ELEMENTS OF DEDUCTIVE LOGIC

Elements of Deductive Logic

Antony Eagle
Exeter College, University of Oxford

Version 3.14; last updated 10th June 2008

The latest version of this book can be found at:
http://users.ox.ac.uk/~sfop0118/

Contents

List of Tables

List of Figures

Preface

About This Book

This book was primarily designed to prepare a student for the paper *Elements of Deductive Logic* offered to mathematically-minded first year philosophy students at Oxford. With a change in regulations, the present book no longer serves that purpose. Yet it should, I hope, provide a useful introduction to logic for interested students, independent of its now defunct role in Oxford. The book was originally written to be read alongside Hodges (2001), and the text remains peppered with references to that book. Yet the book should stand alone; where it disagrees with Hodges (in notation, some technical details, and the discussion of standard classical quantified logic, as well as free logic), I think it is to be preferred. It should provide a good overview of the tableaux system for those who have studied logic using other methods; it should also provide a little bit of extra mathematical sophistication for those logic students who weren't challenged enough by introductory logic, and who are not yet prepared to take on a full formal logic course. Of course, I think that it will provide a good overview and introduction to logic for any interested reader, not just students—the exercises should prove useful to anyone studying this book on their own.

The material in the appendices on mathematical induction and set theory, while not strictly speaking part of the syllabus for an introductory mathematical logic course, are essential for a full understanding of the present text, and obligatory for any further progress in the field.

The book was typeset using LaTeX, an excellent mathematical typesetting system. If you need regularly to write mathematical formulae, or to produce figures and tables and automatic bibliographical references, or you just want footnotes that stay on the correct page, you may want to explore this program: it is easy to learn, fast, and free: http://www.tug.org/begin.html. I use and recommend Lulu for print-on-demand: www.lulu.com.

Conventions I use single quotes ‘, ’ for mentioning sentences or items, rather than using them; I use the Quinean *quasi-quotes* ‘⌜’, ‘⌝’ for their usual purpose, but mostly set them aside in favour of presupposing that common sense will guide use and mention distinctions; see page 38. I refer to sections using the notation ‘§2.3.1’; this refers to chapter 2, section 3, subsection 1, on page 30. I refer to numbered tables, figures, theorems and corollaries by their numbers. A list of figures and a list of tables each appear after the table of contents. ‘■’ indicates the end of a proof (‘QED’). Often, but unfortunately perhaps not always, I will *italicise* a piece of new terminology when it is introduced. Bibliographic references are in the Author (year) format, so that ‘Jeffrey (1991, 55)’ refers to page 55 of Richard Jeffrey’s book *Formal Logic: its scope and limits*. These appear in the Bibliography at the end of the book (page 164). After each work in the bibliography appears a list of pages upon which that work is referred to. Other notation is defined as it is introduced in the course of the text; see the summary in Appendix D.

As this way of approaching logic is no longer on the official syllabus here in Oxford, I am no longer actively revising this text. I may make minor corrections (typographical and other errors); the most recent version will be available electronically from http://users.ox.ac.uk/~sfop0118/.

Acknowledgements

Thanks go to JC Beall for putting me onto the qobitree package for typesetting tableaux; to Barry Taylor, who first taught me tableaux; and to Allen Hazen, who first taught me logic. Particular thanks to my students: Thomas Bloom, Xiao Cai, Tolly Collins, Richard Curtis, Anupam Das, Neil Dewar, Alastair Evans, Ursula Hackett, Tom Hyatt, Maciej Kula, David Lee, Victoria Lee, Zaichen Lu, Phoenix Ma, Quentin Macfarlane, Jack Marley-Payne, Mehmet Noyan, Leo Ringer, Sheena Sodha, and Rebecca Taylor. I am particularly grateful to Claire Coutinho and Michelle Hutchinson, who were the guinea pigs for the very first version (thanks especially to Claire for the whiteboard eraser I’ve used ever since: teaching this material would be much harder without it). This book has benefited greatly from their comments on and corrections of my errors and infelicities. Any remaining errors are, of course, my own.

ARE
Oxford, 10th June 2008

Chapter 1

Introduction: The Role and Point of Logic

1.1 Sentences

Language Language is primarily a tool for communication. There are many uses for language, but we will primarily be concerned with *declarative sentences*. These are sentences that can be true (or can be false), or that aim to describe some possible situation. You may not know whether a given sentence is true or false; it may not even be possible to find out; nevertheless, if the sentence is the kind of thing that can be evaluated as true or false by some omniscient being, it counts as a declarative sentence. (Contrast such sentences to questions or commands: these types of sentence cannot even be true or false.)

How do sentences describe? By having a determinate meaning, which is in part determined by the meanings of the words, and in part by how those words are structured into a sentence. The rules governing correct structure of a natural language sentence are known as a *grammar*. Two specific kinds of grammatical mistake concern us here:

1. *Perturbations* are slightly mis-structured sentences that nevertheless have a clear meaning, stemming perhaps from over-generalisation from known cases (typically heard from children), or from importing grammatical rules from other languages (often heard from non-native speakers); and,

2. *Selection violations* which are apparently grammatical sentences involving word choices so odd as to make their meaningfulness doubtful.

What grammar involves One fairly standard idea of a grammar comprises two parts: firstly, a group of *categories* into which all the words of a language can be classified; and secondly, a group of rules governing how items from different categories can be put together. The grammar that you may remember from school certainly has something of this structure: the classification involves the categories of Nouns, Verbs, Adverbs, *&c*., and the rules will involve claims about when items from these categories can be combined together into sentences, such as for instance the claim that an adjective cannot occur without any noun in a grammatical sentence. If this picture is right, then selection violations should not be counted as ungrammatical: for if the words are of the right categories, and are used in accordance with the rules, they should count as grammatical. The 'weird' reaction can then be explained, not as a phenomenon of grammar, but as a phenomenon of *semantics*, or meaning: we cannot in these cases assign any useful determinate meaning to these sentences, which obviously produces an awkward reaction.

The rules of English grammar are extremely complicated, stemming from a long and twisted historical development, and very few people can claim to know them explicitly in any sense (Hiddleston and Pullum, 2005). Yet all can intuitively judge when a sentence 'sounds wrong' to them, though these intuitions are not infallible.

1.1.1 Problems with grammatical sentences

Even perfectly grammatical sentences can cause problems.

The nuisance of ambiguity (i) *lexical ambiguity*: what if two meanings fit the same word? Example: 'I found some money down at the bank.' (ii) *structural ambiguity*: what if there are two structures that fit the same sentence? Example: 'I heard all about you last week.' A special kind: *cross-reference problems* (also called faulty *anaphora*). Example: 'Ben made Antony teach the logic class; he wasn't that happy about it though.' Clear writing, context, and intonation can avoid most ambiguities.

A problematic example: 'fish' The string of letters 'f⌢i⌢s⌢h' can stand for at least four English words:[1](i) 'fish' can be a noun, denoting a water-dwelling animal; (ii) 'fish' can be an intransitive verb, denoting the activity of trying to catch fish; (iii) 'fish' can be a transitive verb, denoting the activity of trying to catch some particular kind of fish (sense 4.a in the OED under 'fish, *v.*[1]'); (iv) 'fish' can be an adjective, meaning 'fishy'. This multiplicity of meanings gives rise to the following claim: any number of repetitions of the string 'fish' can count as a grammatical sentence of English (note: not a true sentence of English). 'Fish!' is easy: the one word command, meaning something like 'Go fishing!'. 'Fish fish' is also easy; it says something like, 'Fish (can or do) go fishing'. The really problematic cases begin around 5 repetition of 'fish', as in 'Fish fish fish fish fish'. One reading of this sentence that makes sense is 'Fishy (as opposed to fake) fish go fishing for other fishy fish'.[2] This sentence shows that our ability to accurately judge grammaticality has quite definite limits: I doubt that, before I gave the explanation, you instinctively judged this sentence to be grammatical. It is also clear that these sentences involve a tremendous amount of lexical ambiguity; in some cases, like 'Fish fish fish fish', every occurrence of the string 'f⌢i⌢s⌢h' can stand for more than one of the four words (consider the alternative readings 'Fishy fish go fishing for fish' [adj., n., vt., n.] , 'Fish go fishing for fishy fish' [n., vt., adj., n.], 'Fish, who are fished for by other fish, themselves go fishing' [n., n., vt., vi.], etc.).

Context dependence Example: 'Today is Monday.' This sentence is true when uttered on a Monday, but the very same sentence is false if uttered any other day. Another example: 'I am tired.' This may be true if uttered by me and false if uttered by you. Some people have thought that a sentence, when it is uttered or inscribed, takes over from the context the speaker, the time, the place, and whatever else is required to fix a specific and *invariant* meaning on the sentence. This is important, since context-dependence is essential to our language, and not merely a nuisance. (Consider: if I suddenly got amnesia, I could know that the sentence 'Antony Eagle is Antony Eagle' is true, but fail to know that the sentence 'I am Antony Eagle' is true, even though, when they are uttered by me, they express the same proposition.)

[1] '⌢' stands for the relation of *concatenation*, or linking together of particular inscriptions of the characters in question.

[2] I leave the consideration of whether longer chains of repetitions of 'fish' also count as sentences to the reader.

Presupposition What if the context can't supply items for the sentence? For example, if I say 'the present king of France is bald', how can you tell whether my sentence is true or false (since there is no such person)? My utterance *presupposes* the existence of such an entity, but the presupposition fails, and the noun phrase has no reference. Hodges (2001, §6) stipulates that such sentences are false; you might want to think about whether this decision is good.

Another case of presupposition occurs when someone utters a sentence which is true, but he knows something more than he is telling. A general rule of conversation is that people utter the most informative sentence about the subject matter that they are able to (they tell 'the whole truth'), and so it is misleading not to do so. Example: 'Some of the girls kissed me', when all of them did. 'Some' usually presupposes 'not all'; but that is no matter of logic, just convention. We shall adopt the *weak reading* of these sentences, and avoid presuppositions as potentially confusing.

Conversational Implicature Rules governing our interpretation of other speakers, like 'Assume the speaker is uttering the most informative statement relevant in the circumstances', are sometimes known as conversational maxims. Similar rules—for example, 'Be relevant in your answers'—apply to speakers. In both cases, the maxims are conventions adopted by language users to more readily facilitate communication between them. Grice (1989), who first proposed that such rules govern the use of language, also observed that obeying these rules is not part of the meaning of the language, but is rather a matter of choice (rather like the way that traffic light colours determine a course of action—someone who runs a light breaks a law but needn't be ignorant of what the red light means). These rules can be used in various ways. If you say that you are hungry, and I reply 'There is a shop across the road', the injunction to be relevant leads you to interpret my utterance as expressing the claim that that shop has food, that I think it is open, that I think you will like the food. Of course it is compatible with the truth of my utterance that all of these further claims are false, so my utterance does not imply them. Rather, my utterance *implicates* these further claims: my utterance, along with assumptions that you are entitled to make as a speaker of English, leads to these further conclusions.

Implicatures can be misleading too. Take the example of 'I kissed some of the girls'. The injunction to be maximally informative leads one to interpret an utterance of this sentence as saying 'I kissed some *but not all* of the girls'. But as the sentence 'I kissed some of the girls; in fact, I kissed all

of them' is not contradictory, this shows that my interpretative conclusion is something that the original sentence implicates, rather than implies. This, I think, explains the appeal of the strong reading of 'some'.

An example due to Grice (1989, 33) is of a letter of recommendation for a philosophy position that merely commends a candidate's regular attendance. Given the injunction to be maximally informative, the natural interpretation is that nothing else positive can be said about this candidate, and hence that they do not deserve the position. Of course many well qualified philosophers are punctual, which shows that what the letter says does not imply that the person is not qualified. Rather, the letter implicates that the person is not qualified.

Vagueness Again, some words are vague: think of what it is to be a 'heap' of sand. It is vague where the boundary between heaps and non-heaps lies. It is not clear, therefore, whether the sentence 'these 357 grains of sand form a heap' is true; hence arguably even 'true' has borderline cases. Worse, even if we divide all the piles of sand into 'definitely a heap', 'definitely not a heap', and 'borderline', there will still be borderline cases: a pile that is the smallest borderline case is, intuitively, not definitely borderline, as it is only one grain different from a pile that is definitely not a heap. So even the borderline cases have borderline cases! This phenomenon, of so-called 'higher-order vagueness', is highly characteristic of genuinely vague terms.

Bizarreness Sometimes the ordinary meanings of words cannot be easily or obviously extended to new cases. Consider the word 'person': is a baby a person? is a human being in a vegetative state a person? is an alien (or could an alien be) a person? These weird cases require decisions and arguments about the meanings of the words, decisions we have not yet made.

1.1.2 Possible Situations

To get a clear case of a declarative sentence, then, we must avoid all these problems. We shall strive to do so; but you must be careful, since they are everywhere in English. One way to do so is to try and make clear in exactly which *possible situations* a sentence would be true. A possible situation is simply that: one which might have been the case, but needn't be in fact the case. We are not concerned here with what could become the case; or what might have been the case; or even what is compatible with what we

know about the laws of nature. We are simply concerned with what is not impossible, when we suppose only what is *necessary*.

We don't have to believe that such things as other possible situations are things just like the actual situation—nothing in accepting (merely) possible situations involves believing that the contents of such situations are as real as actual entities. Rather, you can think of a possible situation as a *way things might have been*, where that is something like a description according to which things are some way. Obviously the actual way things are is such that, according to it, things are some way; but the actual situation is also more than that, in that the way that things are according to the actual situation is also the way real existing things in fact are.

Propositions We shall take it that a sentence, once all its contextual features are specified, expresses a unique *proposition* (a meaning). Many sentences can express the same proposition; witness the possibility of reliable translation from one language to another. That is, the English sentence 'it is sunny' expresses the same proposition, the same meaning, as the French sentence 'Il est ensoleillé'. Most of the difficulties that arose in the previous subsection were due to sentences that failed to express a unique proposition. Most of the cases of ambiguity involved sentences that could be used to express many different propositions, in different contexts. And ungrammatical sentences mostly express no proposition at all in any context. So if language is to fulfill its communicative intent, there must be a single determinate meaning assigned to each sentence: and that meaning is what the sentence is used on that occasion to communicate.

This use of proposition as the meaning of a disambiguated sentence in a given context is very close to what Hodges calls 'belief', as you can see if you think of that sense of 'belief' in which we say that an English speaker and a French speaker may have the same belief about whether it is sunny, though each would express that belief using different sentences, given their differing native languages.

Content and communication Some people think that a proposition expressed by a sentence just is the set of possible situations which that sentence correctly describes: the proposition expressed by 'Il est ensoleillé' is the set of all possible situations in which it is sunny. Communication, on this view, proceeds as follows: a conversation begins with both conversants having a set of possible situations in mind. These possible situations are, intuitively, the set of possibilities that are compatible with all that the person knows.

An utterance of a sentence *u* 'rules out' all those possibilities in which *u* is not true, so that, over the course of the conversation, both conversants will rule out from their own stock of possibilities all those incompatible with everything that has been said so far. At the end of the conversation, if it has gone successfully, both conversants will have a smaller stock of possibilities that, for all they know, might be actual: that is the sense in which they know more at the end than at the beginning. (There is also a useful treatment of presupposition in this framework: *p* is presupposed if and only if both conversants have ruled out possibilities where *p* is false before the conversation begins.)

Exercises for §1.1

Exercise 1.1.1: What is a 'declarative sentence'? What other kinds of sentences are there? Give at least three examples.

Exercise 1.1.2: Are 'bank' (river shore) and 'bank' (financial institution) the same word with two meanings, or two words with the same spelling? What facts might help us decide this question?

Exercise 1.1.3: Could the sentence 'I am here now' be false (express a false proposition)? Could a speaker who uttered that sentence ever utter a falsehood by it? (Consider an answering machine message.)

Exercise 1.1.4: Give an explanation in terms of conversational implicature of the following examples:

1. Saying 'She produced notes closely corresponding to the tune of *Mamma Mia*' rather than 'She sang *Mamma Mia*'.
2. Replying to 'Is Lara seeing anyone at the moment?' by saying 'She has been spending a lot of time in New York'.
3. Uttering the, apparently trivial, sentence 'Boys will be boys'.

Exercise 1.1.5: Are there such things as 'possible situations'? What are the consequences of this for our theory of meaning?

1.2 Syntax and Semantics

It is now appropriate to draw a fundamental distinction between *syntax* and *semantics*, one which runs fundamentally through this book. To get a hold on this distinction, think of the English word 'happiness'. This is a linguistic entity which when written down has nine letters, in a particular order. It also has two significant parts: 'happi-' and '-ness', each of which occur in other contexts (in 'happy', though spelt differently, and in for example

'ugliness'). These parts are called *morphemes* by linguists: the structural components of English words (Harley, 2006, ch. 5). In this case the word 'happiness' has a *root* morpheme, 'happy', and a *suffix*, '-ness'. There are (quite complicated) rules governing which kinds of roots can go with which suffixes (and prefixes). These rules depend on structural properties of English sentences, and we won't go into them very deeply at all here. If you recall a little bit of school grammar, this example might help: 'happy' is of the grammatical category Adjective; 'happiness' is of the category Noun. In fact, every term ending in the suffix '-ness' is of the category Noun, and almost always '-ness' can only be attached to stems of the category Adjective: '*runningness' or '*happilyness' are not correctly formed in English.[3]

Syntax: the structure of language These facts, about the spelling (orthography) and structure (morphology) of words, are *syntactic* facts: they are facts about the structure and organisation of a linguistic item in terms of the nature and linguistic classification of its parts. Words have a syntactic structure, given by their morphology. Sentences also have a syntactic structure, the study of which is called grammar (we met it earlier in this chapter). Grammar tells us how words of certain linguistic categories are, in some natural language, combined into sentences, but also how they can be combined into structurally significant parts of sentences (clauses, noun phrases, etc.). But this is still syntactic information, because these generalisations about the behaviour of language depend only on the linguistic category of the entities involved, and not on the *meaning*. To extend our example from before, the sentence 'Happiness is important' has a structure Noun ('Happiness') followed by Verb Phrase ('is important'), and the Verb Phrase itself has the structure of a verb ('is') and an adjunct adjective ('important') which complements and is licensed by the verb in question. The simplest kinds of declarative sentences have this basic Noun Phrase-Verb Phrase structure, and it is a basic fact of English grammar that only such linguistic items with this kind of structure (and in that kind of order, since English lacks the complex inflectional structure of a language like German) count as grammatical declarative sentences (Hiddleston and Pullum, 2005, ch. 4).

[3] '-ness' can also go with some nouns in informal contexts: one might use it to indicate the individual essence of some category of things as in a sentence like 'Breaking down in the middle of a crucial job is when your copier really expresses its copierness most fully' (Harley, 2006, 124).

Semantics: the meaning of language The study of meaning, and how it can be attached to individual words and to sentences as a whole, is called *semantics*. Though obviously studying the syntax of English and never studying the semantics (or vice versa) would give you a pretty terrible understanding of English, it is fairly clear that the two disciplines can be studied largely independently of each other (and indeed they are separate academic specialisations within linguistics). To take our examples, the word 'happiness' and the word 'ugliness' have the same morphological structure, and while they are spelled differently this hardly tells us anything about the meanings of these terms. Similarly the sentences 'Happiness is important' and 'Ugliness is overrated' have the same basic Noun phrase-verb phrase grammatical structure, but hardly mean the same because of that. It is only after we associate the word 'happiness' with the concept HAPPINESS that we have given that linguistic item a meaning. In general, semantics has proceeded by giving *truth conditions* for sentences: the fundamental semantic theory for a language will be one that tells you under which circumstances a given sentence is true. This typically goes by the most trivial kind of connection: 'Happinesss is important' is true, after all, just in case happiness is important! But if we think about giving the truth conditions for other languages, that connection is not so trivial: to know that 'Le bonheur est important' is true just in case happiness is important is to come to know something non-trivial about the semantics of French, just as to know that the conditions under which that sentence is true differ from those under which the sentence 'La laideur est surestimée' is true.

Natural and Formal Languages In general, the syntax and semantics of natural language is extremely difficult. To give an account of 'Happiness is important' seems fairly easy: it is true if and only if the item denoted by 'happiness', the quality of being happy, has the property denoted by 'important', that is, if the quality of being happy has relevant value. This does seem to be the case, and we might regard it as a model of this kind of 'subject-predicate' sentence: see §3.3. But the chances of giving an adequate account of the present sentence (namely, this very sentence) seem slim at present, given its complex grammatical structure and the essential occurence of modal and self-referential terms in the sentence. In the present book we sidestep this problem: the languages we study are artifical invented languages, with some relevant similarities to natural language, and some of the same expressive power, but with far simpler syntax and semantics. We shall then use these artificial languages as models for the richer and more

complicated situation with respect to natural language; indeed, the clarity and simplicity of these artificial languages will enable them to perform some natural language tasks more effectively than the natural languages themselves, at least once we have adequately specified translations between the natural and artificial languages. To see what some of these tasks might be, it is now time to look at what the purposes of natural languages might be. For us, the important role of natural language in in the expression and articulation of *belief*, and in the regulation of belief through *argument*.

1.3 Belief

Belief and Believing When someone thinks that the world is a certain way, we normally describe them as believing that the world is that way. So, for example, I think that Oxford is an old university, and it is correspondingly correct to describe me as believing that Oxford is an old university. This mental state that I am in—namely, believing that Oxford is an old university—is a particular kind of ongoing habitual mental event. As such, it occurs in my mind and is of the same category as thinkings, wantings, desirings: namely, mental states. But like thinkings and desirings, believing has an associated mental attitude: a belief (corresponding to a thought or a desire). Beliefs, thoughts and desires are also mental phenomena, though they are mental objects rather than mental states.

Belief as a 'propositional attitude' In addition to the mental object, a belief, we often use the term 'belief' to denote the *content* of someone's mental object. So my belief is a mental entity, with the content 'Oxford is an old university'; we often say that my belief is that Oxford is an old university. When telling someone about my belief, it is perfectly appropriate to tell them about the contents of my beliefs rather than about the mental (let alone brain) states that constitute the mental object which that belief is. A belief in this sense is an attitude to a content, or what we just earlier called a proposition. More specifically, belief is that attitude to a proposition or possible situation of thinking it to be actually the case. Desire can work in a very similar way, but the attitude towards a proposition characteristic of desire is not that of thinking that one's desires are actual, but rather wanting one's desires to be actual. So while I believe that Oxford is an old university, and that belief is (if correct) true, my desire to eat icecream now in no way depends on it being true that I am eating icecream now (and it would be

weird to say I desired to eat icecream now if it fact I was doing it right now!).

Belief and desire together guide our actions. We act so as to get our desires satisfied, but we could hardly do that effectively if we didn't have any internal guide as to what kinds of things will lead to our desires being satisfied. The role of belief is to provide this constraint on effective satisfaction of our desires: we act as if our beliefs were true, so the things we do to get what we want reflect they way we believe the world to be.

Language and belief Of course, it is impossible to directly use beliefs in communication; the propositions which are the content of those beliefs are abstract objects, 'meanings', and one can hardly deal directly with them. But since some sentences describe possible situations, and thus propositions, we can use sentences to express those propositions which are (the content of) our beliefs. This gives the primary reason why sentences are important: The main benefit of declarative sentences and language in this regard is that they are publicly available in a way in which beliefs are not. Thus to articulate our beliefs, to influence other's beliefs and to provide reasons for our actions, we need to make use of articulable declarative sentences.

We can even use language as a rough guide to belief (and presumably this is how we form beliefs about what other people believe): a person believes a proposition exactly if they would assent to any sentence expressing that proposition. So I believe that Australia is a beautiful desert country, and correspondingly I would agree to the statement that 'Australia is a beautiful desert country', if it were put to me. This is rough because there must be many qualifications: they would assent if they were being honest, if they had no compelling reasons not to so assent, if they could understand the sentence, if the sentence wasn't too long, if they weren't annoyed with the questioner.... But, with these qualifications left tacit, we can still see the relevance of belief to assent to and assertion of declarative sentences.

Consistency of belief If our beliefs really do represent possible situations, then there must be a possible situation that describes the sum of our beliefs. That is, our beliefs must be *consistent*: there must be a possible situation that our beliefs collectively represent. Violating this standard of consistency can lead to obvious problems: if there is no possible situation our beliefs describe, then the actual world cannot be accurately represented by those beliefs, and we will be unable to act sensibly or reasonably.

Consistency of language This goes over to our language too: a group of sentences which describe our beliefs, or a possible situation, can also be consistent or inconsistent. Since beliefs are private (i.e. in our heads) it is only by our public use of language that we can detect inconsistency in our representations, so it is very important that we can check the inconsistency of a group of sentences. This is the role of *logic*.

Exercises for §1.3

Exercise 1.3.1: Does language always provide a reliable guide to belief? What about in the case of animals or infants?

Exercise 1.3.2: Why is it important that belief be consistent, especially considering that most likely we all have inconsistent beliefs (because we have so many, from so many sources, and no way to attend to them all at once)?

1.4 Arguments

Change in view One very important use of language is to get people to change what they believe. This might be because they believe inconsistent things, and we want to help them come to have a correct idea of what the world is like; or it might be because they have a consistent but false representation of the world. An *argument* is a special type of linguistic object, designed to persuade or convince someone of some fact. Though arguments are normally used by people, we can consider them as objects in their own right, with various special properties.

Actual versus Abstract Arguments Just as a belief is a particular mental entity with a propositional content which is of primary interest, so too arguments can be regarded as linguistic objects with content which is their primary interest for us. Correspondingly we will tend to regard an argument as an abstract object which can be expressed in any number of linguistic forms; of course it will always be expressed in some particular language, and (particularly if it is essentially used to convince someone of something) it must be publically expressed. The goodness or badness of an argument will depend on its content; but the linguistic structure of a given expression of an argument can be used as a proxy for that content and can, if we are lucky, enable us to determine the goodness of that argument by purely structural techniques.

The fundamental slogan We can, if we wish, express this in slogan form: *logic is about how arguments can be good or bad in virtue of their structure, once that structure is articulated and made clear*. Why we should wish to look at structure rather than content is a good question (see page 65). But because we shall also show that there is a correspondence between the structural techniques for demonstrating the goodness of an argument (via proofs) and content-based techniques for demonstrating the goodness of an argument, we can think of formal logic as giving us another set of tools for the same task, namely, distinguishing good arguments from bad.

One way of making this distinction is to appeal to a distinction used linguists between *function* words and *content* words (Harley, 2006, 117-9). This is not an entirely obvious distinction, but basically it comes down to a distinction between words that mostly convey grammatical information (function words), and those that provide the content which is organised by that grammatical structure (content words). Typical examples of function words include *conjunctions* ('and', 'or'), *determiners* ('some', 'most', 'enough', 'this'), *pronouns* ('I', 'you') and *complementisers* ('that', as in 'I believe *that p*', 'whether') (Harley, 2006, 190–6). Hoping that these examples give you a rough idea of the kind of thing a function word is, we can perhaps rephrase the fundamental slogan: *logic is about how arguments can be good or bad in a way that depends only on the function terms they include, and is relatively independent of the content words used.* In effect, this is the dictum I follow below: in constructing artificial models of English arguments, we will treat conjunctions (ch. 2), determiners (§§3.5 and 3.2.4) and pronouns (§3.2) in a formal way and look at the logical laws specifically applicable to each class of terms. Further extensions of the logic we examine can then go on to look at more complicated pieces of English functional structure, including but by no means restricted to completementisers.

Parts of arguments An argument consists of the following parts: it has a *conclusion*, which is the proposition that the argument is supposed to persuade you of; and it has some *premises*, which are the evidence, or the reasons given, for the conclusion. Usually these are flagged when the argument is expressed in natural language. In English, we do so as follows: **conclusion** *because* **premise 1** and **premise 2**; or **premise 1**; **premise 2** *therefore* **conclusion**.

Powers of persuasion How does an argument persuade someone? By connecting with consistency, discussed above. An argument attempts to

demonstrate that it would be *inconsistent* to believe the premises of the argument and *not* to believe the conclusion. So if someone believes the reasons, you give an argument that shows they must, to avoid the problems of inconsistency, also believe the conclusion.

Validity A truly compelling argument, therefore, will have the following property: *if the premises were to be true, the conclusion **must** also be true*. An argument with this property is called *valid*. If the argument is very lucky, then its premises will also be true; a valid argument with true premises is called *sound*. Sound arguments are obviously very important practically speaking, but in a certain sense soundness is a fragile property of an argument, because even a valid argument is only sound in those rare situations in which all the premises happen to be true.

A valid argument, by contrast, is valid in every situation if it is valid in even one situation. If we think of the premises and conclusions of abstract arguments as propositions, and we think of propositions in the possible situations sense we introduced earlier, validity can be characterised as follows: an argument is valid if the set of possible situations which is the conclusion is included in (is a subset of) the set of possible situations which is the intersection of the premises.[4] This fact, about the relationships between sets of possible situations, isn't a fact about any one of those possible situations, but is rather a fact that transcends each individual situation to look at them all from a global perspective. It is a somewhat surprising fact that there are interesting and important results to be obtained from such an abstract a global view; nevertheless, it is so. The study of which arguments are valid arguments is called *deductive logic*, and (pretty obviously) that is the main focus of this book.

Once we recognise that we have often been wrong about our beliefs, we cannot guarantee that our arguments are sound. *But we can guarantee, if our arguments are valid, that we are being rational and reasonable even if our beliefs turn out to be mistaken.*

Counterexamples It follows from what we have said above that an argument is valid exactly when its premises and the *opposite* (what logicians call the *negation*) of its conclusion are inconsistent. This set of sentences is called the *counterexample set*; if it describes a possible situation, that situation is a *counterexample* to the argument.

[4] If you are unfamiliar with the notion of a set, and the set operations of subset and intersection, you might wish to consult Appendix B.

It follows fairly quickly that deductive logic is the study of arguments to which there are *no counterexamples whatsoever*. This is a very strong property for an argument to have: no matter how vivid their imagination, there is no way for someone who remains committed to the premises to rationally avoid being committed to the conclusion.

Inductive logic There are, of course, arguments of a different nature to this. For instance, the argument 'The sun has risen every day for the past million years; therefore it will rise again tomorrow' is a convincing argument, yet it is not valid. This is because it is possible that the sun is destroyed somehow during the night, and will not rise. Here, the premises give a very good reason for believing the conclusion, but not absolutely conclusive. We are not concerned with such arguments here; they are sometimes called *inductive arguments*, and the study of them is called *inductive logic*. It is a much less advanced and much more complicated field than deductive logic. I mention them only to set them aside. Good deductive and inductive arguments are both examples of what Hodges (2001, 41) calls *rational* arguments: those whose premises provide good reason to believe the conclusion. So maybe there are cases in which one can have rational beliefs even based on deductively invalid arguments. What is certain is that if your rational beliefs are based on deductively valid arguments, you will have rationally managed your beliefs, given your initial beliefs.

Aside: Inference and Implication Arguments are attempts to persuade people who accept the premises to accept the conclusion. Yet it is a perfectly reasonable response to a good argument that the hearer might come to reject the premises, because the conclusion is unacceptable to them. Thus, if someone believes p, and p implies q, they have two options: either come to accept q, or come to reject p. If there are further rules governing which option they should take, those rules would govern human *inference*: how to reason, in the light of arguments. Logic, as we study it, gives no such rules—we are only concerned with *implication* relations between sentences, not what reasonable human beings should believe on the basis of those implication relations (Harman, 2002). So whether you count as rational all things considered will depend on what it is precisely you believe. If you are unlucky enough to start from unreasonable beliefs, and you come to believe the deductive consequences of those beliefs via valid arguments, you will still have unreasonable beliefs. But again, it is certain that regulating belief by deductively valid argument will never move you from reasonable beliefs

to unreasonable ones.

Exercises for §1.4

Exercise 1.4.1: If you believe in the premises of an argument, and you think that the argument is valid, *must* you come to believe the conclusion?

Exercise 1.4.2: When people actually attempt to persuade someone, do they always have to give sound arguments?

Exercise 1.4.3: What does a counterexample set to an argument show about the argument? What if the negation of the conclusion is self-contradictory? What if a premise is self-contradictory?

1.5 Logic and Philosophy

If you are like most people reading this, you will be taking logic as part of a philosophy degree. One question you might be asking yourself is: what does all this have to do with philosophy? The two subjects are certainly closely connected historically, but there are reasons for this historical connection. The remarks immediately above about arguments provide the key to these reasons: since philosophers are mostly concerned with providing and evaluating arguments for the various positions they defend and oppose, the correct rules for giving and evaluating arguments have been of great importance to philosophers. Other disciplines of course also involve argument, but most other disciplines—with the obvious exception of mathematics—are considerably more tolerant of what argumentative moves are permitted and accepted as appropriate. Normally, most philosophers (and most mathematicians) do not give their arguments in anything like the formal languages this book will go on to describe. Nevertheless, the fact remains that without the tools and background knowledge of logic to enable you to evaluate an argument as precisely and carefully as could be required, you will not succeed in philosophy—or at least the philosophy you produce will not be worth the time spent writing or reading it.

Chapter 2

Propositional Logic

2.1 Beginning Formalisation

Clarity of structure Sometimes it can be difficult to see how a sentence relates to other sentences in an argument, perhaps because one cannot see its structure clearly enough, or (more commonly) because other features of the English expression confuse us concerning the structure. (Example: the sentence 'if it snows, then it precipitates' can also be expressed 'it precipitates if it snows' or 'it snows only if it precipitates': these are confusing expressions because they might lead one to incorrect application of tableaux rules—see below.)

Abstraction and substitution It is often important, therefore, to *abstract away* from the other content of the sentence, leaving only the logically important parts explicit. We then *substitute* letters ($p, q, r \ldots$) to stand in for the other parts of the sentence, resulting in a formula. A different letter should stand for each different meaningful part of the sentence.

Example: 'and' For instance, if a sentence has '*and*' as its main connective, we can substitute one letter to stand for the first sentence connected by 'and', and a different letter to stand for the second sentence. It is common too to substitute the word 'and' for a logical symbol; we shall use '$\wedge$'. So the sentence 'It is not snowing and 2+2=4' might be symbolised: '$p \wedge q$', where 'p' stands for 'It is not snowing' and 'q' stands for '2+2=4'.

Sentence Functors Introduced The word 'and' is an example of a *sentence functor*: it connects one or more (two, in the case of 'and') sentence-level parts of a sentence into a larger sentence. Other examples are 'or', 'not', and 'maybe'. We shall use sentence functors as the main markers of how to divide a sentence. We shall see them further, below.

Levels of analysis It is obvious that the sentence '2+2=4' has further meaningful parts: '2' for instance, refers to the number 2. But '2' is not true or false by itself. Later in this course we shall look at more detailed analyses of sentences, but at the moment the lowest *level of analysis* we shall go to is that of individual propositions. That is, we shall analyse a sentence until we have got only those parts that (i) can be true or false; and (ii) have no parts that can be true or false, plus (iii) whatever connects those parts to each other into a complex sentence. These most basic parts we shall call *elementary propositions*. Another way of thinking about it is that, at the moment, we are concerned with only those parts of sentences which are themselves sentences, and have no sentences as parts.

Analysis of sentences Accordingly, we shall proceed as follows. We are given a sentence, with some structure. We divide it up into its immediate parts that are sentences, and the words or phrases which connect those sentences to each other (or, those things which make a new sentence when combined with one or more sentences, i.e. sentence functors). We divide up those further sentences into their constituent parts, keeping track once again of the connectives. We continue until we only have sentences that have no sentences as parts. We assign each of these basic sentences a *sentence letter*, and record a correspondence: for instance, we shall write a dictionary as in Table 2.1.

Table 2.1 A dictionary of elementary propositions.

'p'	=	'It is snowing'
'q'	=	'2 + 2 = 4'

We shall then recombine those sentence letters with the sentence functors we identified, to get a formalised version of our original sentence. This is a formal analogue of the original sentence, which abstracts away from the original content, and highlights the relevant structure, of sentences which have other sentences as meaningful sub-parts.

Sentence Functors Again A sentence functor takes one or more sentences as input, and yields a sentence as output. All the connectives we have considered so far are sentence functors: 'and' for example, takes two sentences and yields a more complex sentence. 'not', or 'maybe' are examples of one-place sentence functors. 'if . . . then . . . else' is a three-place connective. We are currently concerned with how sentence functors structure complicated sentences.

Truth Functors The most important class of sentence functors for our purposes are *truth functors*. These have the following very important property: the truth value of a complex sentence structured by a truth functor depends only on the truth values of its component sentences. 'and' for instance, is a truth functor: the sentence '$p \wedge q$' is true if p is true and q is true, and false otherwise.

Non-Truth Functors Many non-truth-functional operators exist too; their logic is unfortunately beyond us this term. Consider: 'Possibly, there is a talking donkey'. This sentence can be analysed into the operator *possibly* and the sentence 'there is a talking donkey'. This particular sentence is true; the contained subsentence is false. But consider: 'possibly, 2+2=7'. Again the same analysis: the subsentence is false, but now the whole sentence is also false. Since 'possibly' gives a variable outcome on input sentences of the same truth value, it is not a truth functor. Similar things go on when we consider time and tense: 'There was a computer in this room in 1950'; and 'There was a desk in this room in 1950'. Consider also: 'It ought to be the case that (there is universal happiness)' as opposed to 'It ought to be the case that (there is universal despair)'. That is, if a two place sentence functor $\star$ is not a truth functor, then there are cases where the complex sentences $\varphi = \varphi_1 \star \varphi_2$ and $\psi = \psi_1 \star \psi_2$ have different truth values, even though φ_1 has the same truth value as ψ_1, and φ_2 has the same truth value as ψ_2.

We will consider, at this stage, only truth functors, setting other sentence functors aside. One reason for this is that the logic of other sentence functors is not uniform: different logical principles exist to deal with modalities (like possibility), tense and normative statements (like 'ought' and 'should'), and they are not easily dealt with in a simple and uniform manner. The second reason is that they are a natural generalisation of truth-functor logic, so it makes sense to begin with the latter.

Testing Non-Truth Functor-hood Here is a simple way to show a sentence functor is not a truth functor. Begin by listing all the sentences of English, beginning with the two word sentences, then three word sentences, then four, and so on. Since there are only finitely many words in English, there are only finitely many two, three, four, *&c*. word sentences; but there are infinitely many possible sentences of English, as we can always make a longer sentence from conjoining shorter sentences. Take some one-place sentence functor '$\bullet$'; if it is not a truth functor, there will be two sentences φ and ψ with the same truth value but such that '$\bullet\varphi$' has a different truth value from '$\bullet\psi$'. So begin to go through the list of all English sentences; first compare the behaviour of '$\bullet$' on each pair of two-word sentences with the same truth value; then on each pair of two- and three-word sentences with the same truth value; then two- and four-word sentences; then three- and four- word sentences, *&c*. Proceding in this way, for any pair of sentences of any length, at some finite time from the beginning of this long process, they will be compared. If any pair of sentences exists such that '$\bullet$' gives different truth values on those inputs, that pair will turn up after a finite time. Thus, after a finite time, if '$\bullet$' is *not* a truth-functor, we will mechanically generate a pair of sentences which demonstrates this.

However, if '$\bullet$' *is* a truth functor, we shall just keep comparing longer and longer sentences forever. At every stage we will not yet have produced the counterexample to the hypothesis that '$\bullet$' is a truth functor; but that could either be because the first counterexample is yet to come, or because there is no counterexample. If we could perform infinitely many comparisons in a finite time, we could determine whether it is, or is not, a truth functor, with certainty; but we cannot perfom infinitely many comparisons in a finite time. As such, no conclusive demonstration seems to be available that a given English sentence functor is a truth functor: all we can safely and conclusively say is that some English sentence functors have, in all their observed behaviour, conformed to the hypothesis that they are truth functors. But as it is possible that even 'and' turns out, on some odd and lengthy pairs of sentences we've not yet seen, to not have obey the properties of a truth functor, we cannot conclusively say that 'and' is to be modelled by '$\wedge$', which is certainly a truth functor because of the way we define it (Table 2.6).

Truth Functors and Truth Conditions The English word 'and' is plausibly a truth functor: it seems to have the semantic function of producing a complex sentence from two simpler sentences that is true just in case each of the simpler sentences is true. This seems to fit well with the semantic

project of giving truth conditions we discussed earlier (page 9): we specify the meaning of a complex sentence structured by 'and' in terms of the conditions under which it is true, which are just the conditions under which the less complex constituents of that sentence are true. So 'She studied and she failed' is true just in case she studied, and also she failed.

Consider, by contrast, the word 'but'. If this is a truth functor, it seems to be the same as 'and': if a sentence 'φ but ψ' is true, then both φ and ψ must be true ('She studied but she failed' wouldn't be true if she didn't study, or if she passed). Yet 'but' has different properties from 'and': it carries some additional information, namely, that the second sentence is *unexpected* in the light of the first sentence. 'She studied but she failed' seems to indicate that, since she did study, it is true but somewhat surprising that she failed.[1]

Does this pose a problem for truth conditional semantics? No: only if we think that the only conditions under which a sentence can be true must be explicitly part of the content of that sentence. This does hold in the case of 'and': all the ingredients required to evaluate the truth of the sentence are given when the sentence is given. This is not true in the case of 'but': whether a sentence involving 'but' is true also depends on the state of mind of the speaker (whether they expected the second conjunct or not), and the sentence needn't involve any explicit mention of facts about that. (Of course normally when someone uses a sentence involving 'but' they manage to communicate facts about their state of mind.) None of this stops their from being fairly straightforward truth conditions for 'but', however: the sentence 'p but q' is true iff p is true, q is true, and q is unexpected (for the speaker/writer) given p. This is simply to warn you that truth functional connectives and truth conditional semantics are separate projects; if a sentence connective isn't a truth functor that doesn't mean that sentence doesn't have truth conditions.

Truth Functors and Formalisation We can now see the rationale behind our strategy of formalising sentences. Truth functors are such that all that matters for determining the truth value of the whole sentence about the meaning of the elementary propositions is their truth-values. Since we are concerned at this stage with consistency, the only important thing about a sentence is its truth value in some possible situation, for that alone determines whether some set of sentences is consistent or otherwise in that situ-

[1] 'but' and 'and' also have different properties when used, as in this sentence, to conjoin terms of other grammatical categories. Notice how bad it would have been to say '"and" **but** "but" also have different properties…' (Harley, 2006, 191).

ation. So we can abstract away from all the details of meaning of the constituent sentences, except for the truth values of the simplest parts, and how the sentences are truth-functionally structured. We shall see that, despite the enormous amount of detail we discard, this is a very powerful technique.

Independence Here is a possible problem. Let us say we are given the English sentences p = 'This figure has three sides' and q = 'This figure is a triangle'. Intuitively, since being a triangle just means being a three sided plane figure, there is a connection between these two sentences. But the formalisation process loses this connection: p and q have nothing to do with one another, once we abstract away from their meaning. So, it seems, we can consider the possible situation where p is true and q is false, and so on. The problem is, there is no such situation, because of the meaning connection between the two sentences. We shall solve (or dissolve) this problem in two ways. Firstly, we shall try to, whenever possible, decompose our sentences into *independent* sentences, each of which can be true or false independently of the other. Our ultimate goal is to get down to 'logical atoms'; however, this seems a hopeless idealisation in most cases. When we cannot do this, we should introduce a new sentence to capture the meaning relations: in this case, the sentence 'if p then q' does the job for us, taken as a truth of logic, or a truth about possibilities, because this sentence rules out the possibility of a non-three-sided triangle, for example.

Exercises for §2.1

Exercise 2.1.1: What is a sentence functor? What distinguishes truth functors from other sentence functors? For each of the following words, say whether they are a sentence functor or a truth functor, making sure to justify your answers by using the functors in sentences:

1. 'That φ is more likely than that ψ';
2. 'It is reasonable to believe that φ';
3. 'not both φ and ψ';
4. 'φ, regardless of ψ';
5. 'φ if ψ, otherwise χ';
6. 'φ, despite the fact that ψ';
7. 'φ entails ψ';
8. 'Doubtless, φ'.

Exercise 2.1.2: For each of the following sentences, (i) identify the elementary propositions; (ii) identify the sentence functors; (iii) substituting letters ($p, q, r \ldots$) for the elementary propositions, give a semi-formalisation of the sentences (i.e. still using the sentence functors, use the letters to give a formal sentence that has the same meaning as the original sentence).

1. 'Either it's raining, or it's not';
2. 'We'll only go out if it's not raining';
3. 'The beach is normally nice and sunny this time of year';
4. 'Whether or not it's sunny now, we can't risk it raining later';
5. 'Normally, the beach is nice and sunny this time of year';
6. 'Whether or not it's sunny now, it's fairly probable that it will rain later';
7. 'You can have ice cream or chocolate, but not both'
8. 'You can have at most one of chips, hotdogs or a burger';
9. 'You're not allowed to have more than one snack; otherwise, you won't eat your dinner';
10. 'Even if you will eat your dinner, that doesn't mean you can have junk food.'

Exercise 2.1.3: Can you give a convincing defense of the idea that 'and' is a truth functor? What, if anything, is wrong with the argument given against this claim on page 20?

2.2 Truth Tables

Truth values and complex sentences We have said truth functions are the most important ones we will consider. These are sentence functors whose product's truth value depends just on the truth values of its parts. So if we look at all the possible situations, then we should be able to tell what the truth of the complex sentence is in those possible situations.

Possible situations for truth functors For truth functors, describing a possible situation is easy. All we have to do is specify, for each basic sentence letter, whether it is true or false. (This follows from our earlier remark that all that matters to the meaning of a basic sentence is its truth value: if possible situations track meanings, then possible situations for truth functors are individuated by truth values.) To check all the possible situations, therefore, all we need to do is check all the possible combinations of assignments of truth and falsity to the basic sentences.

So if we have just two sentence letters, the easiest way to encode a possible situation is to make a table (where '$\top$' means 'true', and '$\bot$' means 'false'): see Table 2.2.

Table 2.2 Possible situations for two sentence letters.

p	q
$\top$	$\top$
$\top$	$\bot$
$\bot$	$\top$
$\bot$	$\bot$

This, as you can see, is all the combinations for two sentences. There are more combinations, obviously, for more sentence letters.

These possible situations are not fully fledged *possible worlds*: for one thing, they are not maximally specific about every possible proposition (they don't give every proposition a truth value), but only about the elementary propositions we are concerned with. For another thing, we can see they will do their job just as well regardless of the meaning of p, q, r, *&c*. All that matters is truth value; so two sentences with the same truth value in every possible situation will have to be logically equivalent: for instance 'a triangle is a three-sided figure' and 'a triangle is a three-angled figure' must be equivalent, even though it seems there are shades of different meaning between talking about angles and talking about sides.

Substitution Indeed, we can now introduce an important but neglected topic: substitution. Because of what we have just said about truth and possible situations, it is easy to see that the following principle is true. (We use lower-case greek letters φ, ψ as variables for sentence letters, and upper-case greek letters Ξ, Π as variables over sentences.)

Restricted Uniform Substitution Assume that in some possible situation sentence letters φ and ψ have the same truth value. Suppose further that φ appears as a constituent part of some formal sentence Ξ, and ψ does not so appear. Then if we uniformly substitute ψ in every place that φ appears in Ξ, then the resulting sentence Ξ' will have the same truth value in that situation as Ξ does.

We shall return to this principle in what follows; it will establish, for instance, the generality of logic, because (we shall see) it will not matter which sentence letters we use in a particular logical formal sentence, because there

will always be possible situations where other sentence letters have exactly the same truth values. This will become clearer below.

2.2.1 Truth Tables for Standard Truth Functors

We can then summarise the results we had earlier about truth functors, by looking at some important truth functors.

Not Since 'not p' ($\neg p$) is true just when 'p' is false, and vice versa, the truth table is easy: see Table 2.3.

Table 2.3 Truth table for negation ('not').

p	$\neg p$
$\top$	$\bot$
$\bot$	$\top$

Or 'p or q' ($p \vee q$) is true just when 'p' is true, or 'q' is true, or both: see Table 2.4.

Table 2.4 Truth table for disjunction ('or').

p	q	$p \vee q$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\top$
$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\bot$

If…then 'if p then q' ($p \to q$) is *false* when 'p' is true and 'q' is false; we say it is true otherwise, just as in Table 2.5. This connective is often called the *material conditional*.

And 'p and q' ($p \wedge q$) is true when both 'p' and 'q' are true, as in Table 2.6.

Table 2.5 Truth table for the material conditional ('if… then').

p	q	$p \to q$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\top$

Table 2.6 Truth table for conjunction ('and').

p	q	$p \wedge q$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\bot$
$\bot$	$\bot$	$\bot$

2.2.2 Defining truth functors

Numbers of Possible Situations For some given number of sentences, how many possible situations are there? It is clear that two possible situations differ from one another if and only if they differ on the value of at least one proposition; if we therefore have n elementary propositions, there are 2^n possible situations, assuming 2 truth values. That means, a two-place connective will need 2^2 rows of its truth table.

How many truth functors? A truth functor, we have said, is a sentence functor such that the truth value of a complex sentence with that functor as main connective depends only on the identity of the connective and the truth values of the contained sentences. From this, we can say that two truth functors § and ‡ differ if they give a different truth value for a complex sentence on the same row of the truth table (i.e. in the same possible situation they have different truth values). How many different truth functors are there? If we consider two-place truth functors, like 'and' and 'or', how many two place truth functors are there? This is a simple problem in combinatorics: there are 4 possible rows of the truth table, and 2 possibilities for each row: that means 2^4 different assignments of truth values to each row, that is, 16 different operators. There are 2^2 different one-place connectives; and 2^8 different 3-place connectives, *&c*.

Can we get by with less? We may not need all 16 operators. Why? Consider the following: say we need to define an arbitrary truth function, say with a truth table as in Table 2.7.

Table 2.7 Truth table for an arbitrary operator.

p	q	$p \ddagger q$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\bot$

Disjunctive Normal Form We can easily see that this truth functor can be defined in the following way. First, look at the situations in which it is true: that is, when p is true and q is true; or when p is false and q is true. So it is true just when $(p \wedge q) \vee (\neg p \wedge q)$ is true; and false otherwise. This last sentence is in Disjunctive Normal Form (DNF): it is a *disjunction* (sentence involving 'or' as the main connective) of *conjunctions* (sentences involving 'and' as the main connective) of *literals* (sentence letters or negated sentence letters). As can be readily seen, a DNF sentence is a specification of just which (mutually exclusive) possible situations make the sentence true. It is fairly obvious that any possible assignment of truth values to rows can be written as a formula in DNF: simply figure out which rows one wishes the sentence to be true on, and write down a DNF formula that is true in just the specified rows.

So now we have shown that any 2-place truth functional operator can be defined in terms of 'and', 'or' and 'not', since any such operator will have a characteristic truth table, and any such operator can therefore be replicated by a DNF formula involving only disjunction, conjunction and negation.

Degeneracy Of course, this involved one complication: we need the possibility of *degenerate* disjunctions and conjunctions. A degenerate disjunction is a disjunction with only one disjunct; a degenerate conjunction has only one conjunct. So $p \vee q$ is in disjunctive normal form, even though p and q are not conjunctive: they are, however, degenerately conjunctive, and we allow that—see Bostock (1997, 39–40).

Expressive Completeness That is, 'and', 'or' and 'not' are *expressively complete* (or expressively adequate), with respect to this limited context of

truth functions. What this means is, for any truth functional sentence S involving only truth functional connectives, there is another sentence S', such that (i) S' involves only 'and', 'or' and 'not'; and (ii) S' is true in exactly the same situations as the original sentence S. Anything we are able to express in truth functional language, therefore, can be expressed using a sentence involving only 'and', 'or' and 'not': this set of connectives is expressively complete.

Still further reduction Indeed, we can go further. As can be seen from Table 2.8, 'and' and 'or' are interdefinable too:

Table 2.8 Interdefinability of disjunction and conjunction, with the aid of negation.

p	q	$p \wedge q$	$\neg(\neg p \vee \neg q)$	$p \vee q$	$\neg(\neg p \wedge \neg q)$
$\top$	$\top$	$\top$	$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\top$
$\bot$	$\top$	$\bot$	$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$

So we can, in principle, replace any conjunction by negation and disjunction; and any disjunction by negation and conjunction.

Expressively Complete Operators Can we go still further? The answer is yes: there exist two, two place operators which are expressively complete on their own, because they can define both negation and conjunction, for example.

The first is the so-called *Sheffer Stroke*, '|', read 'not-both' (see Table 2.9). As can be seen from the table, the Sheffer stroke can express '$\neg$', and both '$\vee$' and '$\wedge$', and so can be shown to be expressively complete either through the expressively complete set $\{\vee, \neg\}$ or the expressively complete set $\{\wedge, \neg\}$.

Table 2.9 Truth table for the Sheffer Stroke.

p	q	$p\|q$	$\neg p$	$p\|p$	$p \wedge q$	$(p\|q)\|(p\|q)$	$p \vee q$	$(p\|p)\|(q\|q)$
$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$
$\top$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\top$
$\bot$	$\top$	$\top$	$\top$	$\top$	$\bot$	$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\top$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$

The second doesn't have a name or standard symbol, as far as I'm aware, but we shall use '§' to symbolise it. It has the truth table as in Table 2.10—it can be read 'not-either'.

Table 2.10 Truth table for 'not-either'.

p	q	$p \S q$
$\top$	$\top$	$\bot$
$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\bot$
$\bot$	$\bot$	$\top$

It is an easy exercise to show that § is expressively complete. To show that the Sheffer stroke | and our other connective § are the *only* expressively complete single operators for purely truth functional sentences is only a little more difficult. It is easy to see that any connective ‡ which has the same truth value for p as for $p‡p$ cannot define negation; that rules out the eight connectives with $\top$ on the top row, and rules out the further four connectives which have $\bot$ on the bottom row. That leaves four possible operators (thankfully both | and § amongst them). The other remaining operators are $\langle \bot, \top, \bot, \top \rangle$ and $\langle \bot, \bot, \top, \top \rangle$; and it is clear that these are simply equivalent to $\neg q$ and $\neg p$, respectively, and we know that negation alone is not expressively complete. This shows the result we wanted. (See also Hodges (2001, §24, Theorem XI) and Bostock (1997, §§2.7 & 2.9).)

Exercises for §2.2

Exercise 2.2.1: If two elementary propositions have the same truth value in *every* possible situation, do they have the same meaning?

Exercise 2.2.2: Give a truth table for the connective 'φ if ψ, otherwise χ'.

Exercise 2.2.3: Define $\rightarrow$ in terms of disjunction and negation. Is $\{\rightarrow\}$ expressively complete? What about $\{\rightarrow, \neg\}$? Let **F** be the 0-place truth functor that always takes the value $\bot$. Is $\{\mathbf{F}, \leftrightarrow\}$ expressively complete?

Exercise 2.2.4: Express the formula $\neg(\varphi \vee \neg(\psi \rightarrow (\varphi \wedge \chi)))$ in disjunctive normal form.

Exercise 2.2.5: Show that '§', defined in Table 2.10, is expressively complete.

2.3 More on Truth functors

2.3.1 Equivalence

Truth functional equivalence Sentences φ and ψ are *truth-functionally equivalent* just in case they have the same truth value in every row of a truth table. We write '$\varphi \equiv \psi$' to express that φ and ψ are truth-functional equivalents. This is also known as logical equivalence for truth-functional logic. It is completely obvious that every sentence is logically equivalent to itself, that if $\varphi \equiv \psi$ then $\psi \equiv \varphi$ (this is known as *symmetry*), and that if $\varphi \equiv \psi$ and $\psi \equiv \xi$ then $\varphi \equiv \xi$ (*transitivity*) (Hodges, 2001, 109–10).

Any sentence can be put into DNF It is clear from what was said on page 27 that any operator is equivalent to an operator defined by a sentence in DNF. This observation actually establishes something else as well: any sentence is truth-functionally equivalent to a sentence in DNF. This is because any sentence has a truth value at every row of a truth table: so every sentence, in some sense, defines an operator on the basic sentences which make it up. So all we need to do is give a DNF sentence which has the same truth table as the given sentence, and we have found a truth-functionally equivalent sentence.

Double negation For instance, it is clear that $\varphi \equiv \neg\neg\varphi$, since the negation operator simply 'flips' the truth value, and if we apply it twice, that 'flips' the truth value back to the original value. So 'double negation' is equivalent to the original sentence. We can see by inspection of the truth table (2.11) that this judgement is correct.

Table 2.11 Truth functional equivalence of p and $\neg\neg p$

p	$\neg\neg p$
$\top$	$\top$
$\bot$	$\bot$

De Morgan Equivalences We can see also by inspection of truth tables (Table 2.12) that $\neg(\neg p \wedge \neg q) \equiv p \vee q$; also $\neg(\neg p \vee \neg q) \equiv p \wedge q$. These relations between negation, conjunction and disjunction are known as the De Morgan equivalences (after Augustus De Morgan, a nineteenth century logician).

Table 2.12 The De Morgan equivalences

p	q	$p \vee q$	$\neg(\neg p \wedge \neg q)$	$p \wedge q$	$\neg(\neg p \vee \neg q)$
$\top$	$\top$	$\top$	$\top$	$\top$	$\top$
$\top$	$\bot$	$\top$	$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\top$	$\top$	$\bot$	$\bot$
$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$

Conjunctive Normal Form Above we used disjunctive normal form to prove expressive completeness, but we could equally well have used *conjunctive normal form* (CNF). As the name suggests, a sentence is in CNF iff it is a conjunction of disjunctions of literals, again allowing for degeneracy. It is a tedious, but easy, task to show that every formula can be written in CNF (it uses the definition of DNF above, and the De Morgan laws). For more on CNF and DNF, see Bostock (1997, §2.6).

2.3.2 Duality

The De Morgan laws are one example of a curious property of $\wedge$ and $\vee$: they are what is known as *duals* of each other (Bostock, 1997, §2.10). We may characterise the dual of a truth-functor as that truth-functor whose truth table results from that of the given truth-functor by replacing every occurrence of $\top$ by $\bot$ and every occurrence of $\bot$ by $\top$. It is evident that $\vee$ and $\wedge$ fit this:

φ	ψ	$\varphi \wedge \psi$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\bot$
$\bot$	$\bot$	$\bot$

$\Leftrightarrow$

φ	ψ	$\varphi \vee \psi$
$\bot$	$\bot$	$\bot$
$\bot$	$\top$	$\top$
$\top$	$\bot$	$\top$
$\top$	$\top$	$\top$

But other truth functors fit this characterisation as well; for example, $\varphi \leftrightarrow \psi$ is dual to the operator we might symbolise $\not\leftrightarrow$ (such that $\varphi \not\leftrightarrow \psi$ is true just in case φ and ψ have different truth values).

Duality of $\wedge$ and $\vee$ We may prove a more general result about the duality of $\wedge$ and $\vee$. Let φ be a formula, and let $\widetilde{\varphi}$ be the formula that results from writing '$\neg$' directly in front of every sentence letter in φ. So if $\varphi = (p \vee q) \vee \neg(p \wedge \neg r)$, $\widetilde{\varphi} = (\neg p \vee \neg q) \vee \neg(\neg p \wedge \neg\neg r)$. If φ is a formula, let φ^D be the formula that results from exchanging $\vee$ for $\wedge$ throughout φ (and vice versa). So, in our example, $\varphi^D = (p \wedge q) \wedge \neg(p \vee \neg r)$. Now we prove:

Theorem 1 (Duality). *For any formula φ, $\varphi^D \equiv \neg\widetilde{\varphi}$.*

Proof. We show this by *induction on the length of sentence φ*: see Appendix A for more on this technique. The base case is where φ is just a sentence letter: so that $\varphi^D = p$, and $\neg\widetilde{\varphi} = \neg\neg p$, which are equivalent by Double Negation (Table 2.11).

The induction cases are three (we here neglect the other connectives as they may be defined in terms of $\vee, \neg, \wedge$).

1. The theorem holds of ψ, and $\varphi = \neg\psi$. Then $\varphi^D = (\neg\psi)^D = \neg(\psi^D)$. Because the theorem holds of ψ, $\neg(\psi^D) \equiv \neg\neg\widetilde{\psi}$; but $\neg\neg\widetilde{\psi} = \neg\widetilde{\neg\psi} = \neg\widetilde{\varphi}$, which proves the case.

2. $\varphi = \psi \wedge \chi$, and the theorem holds of ψ and χ. $\varphi^D = (\psi \wedge \chi)^D = \psi^D \vee \chi^D$. By the induction hypothesis, $\psi^D \vee \chi^D \equiv \neg\widetilde{\psi} \vee \neg\widetilde{\chi}$. By De Morgan, $\neg\widetilde{\psi} \vee \neg\widetilde{\chi} \equiv \neg(\widetilde{\psi} \wedge \widetilde{\chi})$, which is $\neg(\widetilde{\psi \wedge \chi}) = \neg(\widetilde{\varphi})$.

3. $\varphi = \psi \vee \chi$, and the theorem holds of ψ, χ. $\varphi^D = (\psi^D \wedge \chi^D)$; by the induction, $\psi^D \wedge \chi^D \equiv \neg\widetilde{\psi} \wedge \neg\widetilde{\chi} \equiv \neg(\widetilde{\psi} \vee \widetilde{\chi})$, by De Morgan; by definition, $\neg(\widetilde{\psi} \vee \widetilde{\chi}) = \neg(\widetilde{\psi \vee \chi}) = \neg(\widetilde{\varphi})$.

That completes the induction. ■

This theorem establishes something general about duals: for every law involving '$\wedge$', there will be a corresponding law involving '$\vee$', under the translation scheme given by Theorem 1. (Further consequences of duality are drawn in §2.9.2, p. 60.)

Self-Duality Of course, given our characterisation of duality, every truth functor has a dual. If we denote the dual of any connective c by $c^\star$, we may all a set of truth functors $\mathbb{C} = \{c_1, \ldots, c_n\}$ *self-dual* iff $\mathbb{C} = \{c_1^\star, \ldots, c_n^\star\}$. It is obvious that $\{\neg\}$ is self-dual; the truth table for $\neg$ is obviously the same as that for $\neg^\star$. More interestingly, given the result above, the set $\{\wedge, \vee\}$ is self-dual, as $\vee = \wedge^\star$ and $\wedge = \vee^\star$.

Exercises for §2.3

Exercise 2.3.1: Are $\varphi \to (\psi \to (\chi \to \pi))$ and $\varphi \to \pi$ truth functionally equivalent? What about $((\varphi \to \psi) \to \chi) \to \pi$ and $\varphi \to (\psi \to (\chi \to \pi))$?

Exercise 2.3.2: Is the self-dual set of connectives $\{\to, \to^\star\}$ expressively adequate? Is self-duality either necessary or sufficient for expressive adequacy of a set of connectives?

Exercise 2.3.3: Show that if a two-place truth functor $\oplus$ is self-dual, then the function that expresses it, $f_{\oplus}$, must be such that $f_{\oplus}(\top, \bot) \neq f_{\oplus}(\bot, \top)$ and $f_{\oplus}(\bot, \bot) \neq f_{\oplus}(\top, \top)$. Using this result, establish how many self-dual two-place truth functors there are.

Exercise 2.3.4: Assuming that every sentence is truth functionally equivalent to a sentence in CNF form, show that a necessary and sufficient condition for a propositional formula to be expressible with just the truth functors '$\rightarrow$' and '$\wedge$' is that the formula has the value $\top$ in the structure that assigns the value $\top$ to each sentence letter.

2.4 Peculiarities of the Truth Functors when compared to Natural Language

The standard truth functors We have already meet each of the standard truth functors, classified by their characteristic truth tables: 'and' ($\wedge$), 'or' ($\vee$), 'not' ($\neg$), 'if…then' ($\rightarrow$), and 'iff' ($\leftrightarrow$). These are names for the operators on truth-values which behave in accordance with certain truth tables.

English truth functors The words we use to name the truth functors already exist in English, of course. There is something to this: after all, it is at least part of the meaning of the English word 'and' that a complex sentence with 'and' as its main connective is true just when both sub-sentences that are joined by the 'and' (what are called the *conjuncts* of the conjunctive sentence) are true. So the operator '$\wedge$' mimics in some sense the English sentential connective 'and'. Indeed, for '$\wedge$' and 'and' the match in behaviour is pretty good (even if we can't prove conclusively that the match is perfect: page 20). But does this always occur?

'Or' versus '$\vee$' Consider the sentence 'You can either have ice-cream or cake for dessert,' uttered by one's parent. This sentence, when uttered by normal English speakers, does *not* mean 'You can have ice-cream, cake, or both, for dessert': we frequently use 'or', then, to express that two options are mutually exclusive alternatives. Call this use *exclusive or*. Sometimes, however, we do use 'or' in the inclusive sense (what Hodges (2001, 22) calls the 'weak sense'). Consider the sentence on a company brochure 'You can contact us by post or phone': this sentence does not entail that one must choose an exclusive alternative means of contact. We opt in our discussion for this inclusive sense of 'or' in understanding $\vee$. If we write the exclusive

or as '$\overline{\vee}$', then it is clear that $p \overline{\vee} q \equiv (p \vee q) \wedge \neg(p \wedge q)$. (Prove this yourself using a truth table.)

2.4.1 'If…then' versus '→'

The problems with 'or' are nothing compared to the interpretation of →. Consider the following conditionals:

1. If 2 + 2 = 5, then I'm a monkey's uncle.
2. If 2 + 2 = 5, then James Joyce is the author of *Mansfield Park*.
3. If Joyce wrote *Mansfield Park* and Joyce did not write *Mansfield Park*, then Hodges wrote *Mansfield Park*.
4. If Joyce wrote *Ulysses*, then 2 + 2 = 4.

I suggest that none of these conditional sentences is correctly assertible in ordinary English, except perhaps the first. Yet they are all true, given the truth table for $\varphi \rightarrow \psi$ (which is the same as the truth table for $\neg\varphi \vee \psi$). In the second and third, the *antecedent* (the 'if' part of the conditional) is false in every situation, and so by the truth table for →, any sentence can be put in as the *consequent* (the 'then' part of the conditional) and yield a true compound sentence. It doesn't matter whether the consequent sentence is relevant to the antecedent or not: but most English conditional claims involve some degree of relevance of the consequent to the antecedent. So too in the fourth sentence: since the consequent is always true in every situation, the truth table for → tells us that no matter what sentence is put in as the antecedent, the compound sentence is true. The first sentence is the only slightly assertible sentence; but, I suggest, it really means something like 'It is not the case that 2 + 2 = 5', rather than meaning some conditional claim. The problems get worse if we consider examples of accidentally true conditionals: sentences $p \rightarrow q$ that are true in a situation because p happens to be false or q happens to be true in that situation (consider 'If grass is yellow, then I love logic').

Truth Connections between 'If' and → Many people have found these results about conditional claims disturbing, and most have concluded that → doesn't capture all that can be said about English conditional sentences. But it does capture part of what can be said: for it is true that an English conditional is *false* when the antecedent can be true, but the consequent false,

and that is what $\rightarrow$ says also. So 'If Joyce wrote *Ulysses*, then Joyce was not a very good writer' is false, since in our actual situation, the antecedent is true and the consequent false. So $\rightarrow$ does agree with 'if… then' in some crucial aspects, and at least as far as truth functionality goes, they are in perfect agreement—that is, no truth-functor apart from $\rightarrow$ does a better job of interpreting the non-truth-functional English conditional. We might emphasise this point as follows, following Jackson (1987, 4–6): imagine Detective Black is following McNulty's trail, and he comes to a fork in the path. Black thinks to himself, 'McNulty went either to the left or the right'; he infers from this that if he were to find out that McNulty didn't go right, he would conclude that McNulty went left. So Black is willing to conclude (just on the basis of the disjunctive claim) that 'If McNulty didn't go right, then McNulty went left'. Similarly, from the conditional claim 'If McNulty didn't go right, then he went left', Black is willing to conclude that either McNulty went right, or he went left. But that means that Black is willing to infer 'If $\neg\varphi$ then ψ' from '$\varphi \vee \psi$'; and similarly that he is willing to infer '$\varphi \vee \psi$' from 'If $\neg\varphi$ then ψ'; which seems to indicate that Black regards the conditional claim as equivalent to the disjunction of the negated antecedent and the consequent, just as the material conditional suggests. As far as truth is concerned, English conditionals and the material conditional behave in precisely the same way.

Defending the Material Conditional Account Indeed, some have argued that the fact that $\rightarrow$ and 'if… then' correspond in truth is enough to show that they have the same meaning. Such philosophers must explain away the badness of conditional claims like 'If Joyce wrote *Mansfield Park* then $2+2=4$'. Typically they do this by appealing to *conversational implicature* (recall p. 4). They argue as follows: while these problematic conditionals are in fact true, they sound terrible to our ears because they fail some other conditions which govern the correct use of sentences.

To see the strategy involved, consider this proposal about defective disjunctive utterances. Imagine that someone believes φ, and does not believe ψ. If they utter the sentence '$\varphi \vee \psi$', while they do say something true by their own lights, it is nevertheless apt to mislead their hearers. Why does it mislead? Because we normally expect that speakers will be as informative and as brief as they can be. Given that if a speaker believes one disjunct of a disjunction, it is briefer and more informative to utter just that disjunct (and if they believe both, it is more informative to utter a conjunction). So when we hear a disjunction, we normally assume that the speaker is not commit-

ted yet to either disjunct. If we know on other grounds that the speaker is committed to one of the disjuncts, or rejects another, then such utterances will strike us as defective: not because they are false, but rather because the speaker obviously violates one of the rules that govern normal honest communication.

Since the material conditional is a disjunction, it is not surprising that some defenders of the material conditional account of 'if…then' appeal to this proposal about disjunctions. So Grice argues, in effect, that in the same way that it violates a rule of communication to assert '$\varphi \vee \psi$' just on the grounds that one is certain of φ, similarly one cannot assert '$\varphi \rightarrow \psi$' just on the basis that one is certain that ψ or certain that $\neg\varphi$. In the case of trivial material conditionals (tautologous consequent or contradictory antecedent), everyone can be assured from the start that one is certain of the consequent or certain of the negation of the antecedent; thus uttering such a trivial conditional is forbidden by the rules of honest communication. That the utterance of such conditionals is forbidden, even though they are true, is sufficient to explain why the trivial conditionals seem so bad, thus saving the material conditional account. This same approach can also explain why we baulk at non-trivial but irrelevant conditionals like 'If bananas are yellow then Joyce wrote Ulysses'; such conditionals are equivalent to disjunctions with an obviously false first disjunct (since everyone knows that bananas are yellow), and so violate a rule that demands brevity.

Of course not everyone is satisfied by these arguments, and it is pretty clear already that they can't be sufficient for *all* English conditionals. Consider the obvious differences between these conditional sentences:

- If Oswald didn't shoot Kennedy, somebody else did.

- If Oswald hadn't shot Kennedy, somebody else would have.

These don't have the same actual truth value, so apparently cannot be given a truth-functional analysis.

Moreover, there has been considerable controversy even amongst defenders of the material conditional account about whether the Gricean strategy works; Jackson for example rejects Grice's defense. Given the scope of this book, we shall have to leave this extremely interesting debate here, having sampled some of the reasons for and against the material conditional account. More details can be found in Jackson (1987) and Bennett (2003, ch. 2–3).

Exercises for §2.4

Exercise 2.4.1: (i) Does 'if the cat is on the mat, then the dog is outside' entail 'either the cat is not on the mat, or the dog is outside'? (ii) Does 'either the butler or the gardener did it' entail 'if the butler didn't do it, then the gardener did'?

What significance do your answers to (i) and (ii) have for the thesis that the English conditional 'if φ then ψ' is the truth functor '$\rightarrow$'?

Exercise 2.4.2: John argues that, since φ and ψ together entail ψ, if follows that ψ entails 'if φ, ψ'.

Mary claims that 'if φ, ψ' entails 'if $\neg\psi$, $\neg\varphi$'.

1. Show how it would be possible to use John's conclusion and Mary's claim to argue that '$\varphi \rightarrow \psi$' entails 'if φ, ψ'.
2. Assess John's argument and Mary's claim.

2.5 A Logical Language: $\mathscr{L}$

Logic concerned with abstract formulae What exactly are these formalised analogues of English sentences, connected by operators, but, as we can see from the principle of Restricted Uniform Substitution, somehow more or less independent of the meanings of the original English sentences? We can treat logic as concerned with not formalised abstracted English sentences, but rather with abstract *formulae*, which can be associated with English sentences by virtue of sharing a common form. Formalising English, then, is a not a matter of discarding parts of English sentences, but rather uncovering just which formulae of some logical language are structured similarly to the English sentence.

The formal language $\mathscr{L}$ To that end, we now introduce a formal language, like English in some ways, but very unlike it in others. We shall call it $\mathscr{L}$. For the similarities, the language has items that can be true or false (something like declarative sentences), it has a grammar which determines the correct ways to build up truth-evaluable items from some basic stock of linguistic primitives; and it has sentence functors. The differences are even more striking, however: this language has only a very limited stock of sentence functors (the truth functors), the language has a very different stock of linguistic primitives; and the language has no obvious meanings.

Formality and Schemas This last point deserves some elaboration. Our language will be a formal one, where the allowable rules for manipulating

sentences are purely mathematical and do not distinguish different sentences based on their meanings. English, too, can be treated like that, but it is more difficult: it is hard to see the sentence 'The boy is tired' as a grammatical string of letters rather than a meaningful sentence, though it must be done when considering grammaticality in full generality. Our language is a little more like some mathematical examples you may have seen. For instance the equation '$x = y$': it is a grammatical string of mathematical symbols, and it has some meaning, but you cannot tell that it is true or not, unless you know the values of x and y. But we do know that if '$x = y$' is true, then '$x^2 = y^2$' is also true; that is a purely formal, meaning independent operation on that sentence. Thus we mostly concern ourselves with formula *schemas* like $\ulcorner \varphi \wedge \psi \urcorner$, that are formulae once some formulae are substituted into the variable places $\ulcorner \varphi \urcorner$ and $\ulcorner \psi \urcorner$.

That, then is how we will proceed. First we will describe our formal language, and specify the rules for the correct formation of sentences. Then we will tell how to *interpret* our language: how to specify the meanings for the language (hint: remember those truth tables and possible situations we discussed earlier).

Aside: Quasi-Quotation I used the symbols '$\ulcorner$' and '$\urcorner$' two paragraphs ago. These are called *quasi-quotes*, and their use is as follows. Normally, when we wish to mention a sentence (or linguistic object) rather than to assert its content (or to use the linguistic object), we enclose that sentence or object in quotation marks. Sometimes, however, that approach doesn't work. For instance, I said above that we often use schemas like $\ulcorner \varphi \wedge \psi \urcorner$. But it is not true that we often use '$\varphi \wedge \psi$', because that is not a well-formed formula (see immediately below), mixing as it does parts of $\mathscr{L}$ (notably, the '$\wedge$') with variables over sentence letters. We use quasi-quotes instead, to flag that the item enclosed is part of no language, and says nothing. Hence $\ulcorner \varphi \wedge \psi \urcorner$ cannot be *disquoted* according to the ordinary schema, originally due to Tarski:

Tarski T-schema 'S' is true iff S.

Of course, once we substitute sentence letters of $\mathscr{L}$ for the variable φ, the resulting sentence can be well-formed and assertible, and can be used and mentioned in a perfectly straightforward manner. That is all by way of explanation of the symbols used; the rationale should merely be noted. The notation was introduced in Quine (1951, §6, 33–7).

2.5.1 Well-formed formulae of $\mathscr{L}$

Alphabet We begin by specifying the basic items of our language. We begin with an *alphabet*, just like in English, except our alphabet is a little different. The 'letters' of our alphabet are the following list: 'p', '$'$', '(', ')', '$\neg$', '$\wedge$', '$\vee$', '$\rightarrow$', '$\leftrightarrow$'. (Some of these, like the parentheses and the prime, are punctuation marks.)

Sentence letters As might be expected, we want something to correspond to the elementary propositions that we abstracted away from in the process of formalisation. So the language contains infinitely many *sentence letters*, or *sentential variables*; we used $p, q, r \ldots$, before, one for each possible sentence. In $\mathscr{L}$, the sentences are of the form $p\underbrace{'\ldots'}_{n}$, with n (zero or more) occurrences of '$'$' occurring after 'p'. Since we shall never deal with more than a small number, though the sentence letters are really written p', p', p'', ..., we allow ourselves to use q and r, *&c.* as abbreviations for p' and p'', *&c.*

Complex formulae These are our basic items: they are all counted as formulae of our language, capable of being independently true or false once interpreted. But they are not the only formulae: so now we decide how to build more complex formulae up out of them. We give rules as in Table 2.13.

Table 2.13 Inductive definition of well-formed formulae.

1. 'p' is a basic formula.
2. If 'φ' is a basic formula, then 'φ'' is a basic formula.
3. Every basic formula is a formula.
4. If 'φ' is a formula, '$\neg\varphi$' is a formula.
5. if 'φ' is a formula, and 'ψ' is a formula, then '$(\varphi \wedge \psi)$' is a formula.
6. if 'φ' is a formula, and 'ψ' is a formula, then '$(\varphi \vee \psi)$' is a formula.
7. if 'φ' is a formula, and 'ψ' is a formula, then '$(\varphi \rightarrow \psi)$' is a formula.
8. if 'φ' is a formula, and 'ψ' is a formula, then '$(\varphi \leftrightarrow \psi)$' is a formula.

Nothing else is a formula unless it comes under the scope of these rules.

Inductive Definition What we see in Table 2.13 is a perfect example of an *inductive definition*. We have, in fact, two inductive definitions in one: one defining what it is to be a *basic formula*, and one defining what it is to be a *formula*. Let us quickly look at how this definition works. First, we decide on a *base case*: that p is a basic formula. Then we give a rule that shows how, from any basic formula, to get another basic formula. This is the induction step, and is very powerful. For instance, since p is a basic formula, p' is a basic formula, by rule (2). Then, by rule (2) again, p' is a basic formula. And so on: we don't need to apply the rule to get the infinitely many basic formulae, since we've already said they are all basic formulae simply by showing how to form them from less complex basic formulae. So too with formulae in general: we have a base case (being appropriately enough, a basic formula), and we show then how to get from formulae to more complex formulae. Finally, we close off the definition: nothing that isn't generated compatibly with the rules is a formula. For more details, see §B.2.

Well-formed formulae Any string of basic items of the language drawn from the alphabet, and can be shown to have been constructed in accordance with the rules above, is a *well-formed formula*; nothing else is. So '$((p \wedge p') \vee (\neg p \rightarrow \neg(p \leftrightarrow p'')))$' is well formed; '$(p\neg \vee' p')'(\wedge')$', though formed from the same alphabet, is not well formed. We shall consider only well-formed formulae, or wffs, in what follows.

Scope and ambiguity Why the parentheses? This is simply to distinguish '$\neg p \vee p'$' from '$\neg(p \vee p')$', that is, to show the *scope* of the negation without ambiguity. When there is no risk of ambiguity in what follows, we shall omit the parentheses. From now on, too, I will write q, r and so on as sentence letters; and φ, ψ, *&c*. will be variables over formulae.

2.5.2 Interpreting $\mathscr{L}$

Meaning and Interpretation Now that we have our language, we need to give it some meanings. We do this by specifying an *interpretation*, that is, an assignment of meanings to our sentence letters. So, for instance, on page 18, we gave an interpretation of 'p' as meaning 'It is snowing', and 'q' as meaning '$2 + 2 = 4$'. If we consider starting from $\mathscr{L}$, one way we could proceed is to assign elementary propositions of English to the basic

formulae, such that complex truth-functor sentences of English will be constructed by using the appropriate truth-function symbol of $\mathscr{L}$. But this is misleading, because it makes it look like $\mathscr{L}$ is simply English written in a funny way. The correct way to conceive of the project is as follows: $\mathscr{L}$ is a language, just as English is a language. And just as there is a distinction to be drawn between the grammar of English sentences, and the meanings of those sentences, so there is a similar contrast in $\mathscr{L}$. The specification of the rules governing wffs in the last section told us the syntax, or grammar, of $\mathscr{L}$. We now need to give an account of the meanings of $\mathscr{L}$. The basic items that need to be assigned meaning are the sentence letters. The interpretation of the other parts of the language—the connectives, and parentheses—is fixed by the logic. Because our logic is truth-functional, it is easy to see that, whatever we assign as the meanings of the sentence letters, the only dimension of meaning which really concerns us in *truth-value* of a given sentence. Our truth-functional language is effectively blind to any further distinctions in meaning between sentences that have the same truth value in a given possible situation. So, in fact, we can take a shortcut: an interpretation of $\mathscr{L}$ will simply be an assignment of truth-values to the sentence letters.

Translation That's not to say that $\mathscr{L}$ and English have nothing to do with one another, of course. English and Spanish don't depend on one another for their meanings, yet we can still *translate* from one to the other. It is easy to see that, if our translation is good, any argument that is intuitively valid in English should be valid in Spanish too. That's not particularly useful, since Spanish and English are similarly powerful and complicated languages. But if we can translate from English into $\mathscr{L}$, then the fact that $\mathscr{L}$ has, as we will soon see, a very simple way of establishing whether an argument is valid, will enable us to check whether an English argument is valid when translated. A good translation into $\mathscr{L}$ will translate true claims of English into sentence letters assigned $\top$ by the interpretation, but in such a way as it preserves the logical structure of the English claims (so a true English conditional will be translated into a wff of the form '$\psi \to \varphi$', not just some arbitrary truth). Then a valid argument in English, that is valid due to the truth-functional structure of the premises, will correspond to a similarly valid argument in $\mathscr{L}$.

Limitations of $\mathscr{L}$ Of course this technique is suitable only for interpreting truth-functional sentences of English, so we can see that the expressive capabilities of $\mathscr{L}$ are more limited than those of English. Of course, once

we recognise that $\mathscr{L}$ is a tool with limited applicability, we shall not worry too much when some things are beyond its powers. A hammer does an excellent job of hammering nails; the recognised fact that it does not calculate square roots too shouldn't blind us to its power and usefulness for its intended purpose.[2] We shall, in the later part of the course, design a better formal language that will do more, that is, express more of English.

Exercises for §2.5

Exercise 2.5.1: What is a well-formed formula? What is the difference between basic and other kinds of well-formed formulae, and why do we define wffs in two steps?

Exercise 2.5.2: Say whether each of the following is a wff according to Table 2.13:

1. '$p \wedge q$';
2. 'p';
3. '$p\neg$';
4. '$(p \wedge q)$';
5. '$((p \wedge \neg(q \vee (r \wedge s))) \wedge s))$';
6. '$(r \vee (p \wedge ((p \wedge \neg(q \vee (r \wedge s))) \wedge s)))$'.

2.6 Truth Tables and Consistency

Structures One important thing to note is that as far as $\mathscr{L}$ is concerned, the truth values of the basic sentences are all that matters for the interpretation. We call an assignment of truth-values to basic formulae a *structure*. It is now clear what the truth tables we discussed earlier are: they encode all the possible structures for a given sentence of $\mathscr{L}$. An function that assigns truth values to sentence letters will be called a *valuation function*: it is a function that takes basic formulae as arguments, and yields truth values as values. It is easily observed that every structure is compatible with just one valuation, and vice versa; these two ways of talking are equivalent.

[2] But the fact that English can express claims that $\mathscr{L}$ cannot is not evidence for the widely believed (but so far as I can see, not well supported) claim that some fully developed natural languages have greater expressive power than others.

Truth We say that a wff of $\mathscr{L}$ is true in a structure $\mathscr{S}$ just when the assignment of truth values to basic sentences suffices to fix the truth value of the whole wff as true, in accordance with the truth tables. For example, consider the wff $(\neg(p \vee \neg q) \wedge (p \rightarrow q))$. This wff is true in the structure $\mathscr{S}$ that assigns $\bot$ to p and $\top$ to q; or, as we shall now write, using the truth value function v, $v(p) = \bot$ and $v(q) = \top$. Let us show this.

Checking truth in all structures For a given wff φ of $\mathscr{L}$, and a given structure $\mathscr{S}$, we determine the truth value $v(\varphi)$ in $\mathscr{S}$ as follows. Let us take $\varphi = (\neg(p \vee \neg q) \wedge (p \rightarrow q))$. We begin by writing out the structure as follows:

p	q	$(\neg$	$(p$	$\vee$	$\neg$	$q))$	$\wedge$	$(p$	$\rightarrow$	$q)$
$\top$	$\top$									
$\top$	$\top$									
$\bot$	$\top$									
$\bot$	$\bot$									

Then copy the values for p and q into the columns underneath their places in the wff, as follows:

p	q	$(\neg$	$(p$	$\vee$	$\neg$	$q))$	$\wedge$	$(p$	$\rightarrow$	$q)$
$\top$	$\top$		$\top$			$\top$		$\top$		$\top$
$\top$	$\top$		$\top$			$\bot$		$\top$		$\bot$
$\bot$	$\top$		$\bot$			$\top$		$\bot$		$\top$
$\bot$	$\bot$		$\bot$			$\bot$		$\bot$		$\bot$

Then we can successively fill in the truth values for the subformulae. We can see that the subformula $\neg q$ appears; hence from the truth table for $\neg$, we can fill in another row: so too with the subformula $p \rightarrow q$ and the truth table for $\rightarrow$:

p	q	$(\neg$	$(p$	$\vee$	$\neg$	$q))$	$\wedge$	$(p$	$\rightarrow$	$q)$
$\top$	$\top$		$\top$		$\bot$	$\top$		$\top$	$\top$	$\top$
$\top$	$\top$		$\top$		$\top$	$\bot$		$\top$	$\bot$	$\bot$
$\bot$	$\top$		$\bot$		$\bot$	$\top$		$\bot$	$\top$	$\top$
$\bot$	$\bot$		$\bot$		$\top$	$\bot$		$\bot$	$\top$	$\bot$

Now we can fill in the $\vee$ column because we have both immediate subformulae of $p \vee \neg q$; and then we can fill in the remaining $\neg$ column; and then we can fill in the $\wedge$ column, which is the main connective for this wff, and hence the values under it are the values for the wff in each of the four possible structures (we write them in bold, for easier identification):

p	q	$(\neg$	$(p$	$\vee$	$\neg$	$q))$	$\wedge$	$(p$	$\rightarrow$	$q)$
$\top$	$\top$	$\bot$	$\top$	$\top$	$\bot$	$\top$	$\bot$	$\top$	$\top$	$\top$
$\top$	$\top$	$\bot$	$\top$	$\top$	$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$

We can easily see, now, that this wff is true only in the structure $v(p) = \bot, v(q) = \top$, and false in all other structures.

A more formal approach We can do the same if we look at the behaviour of the valuation function v. For any given structure $\mathscr{S}$, we can give rules on the valuation function $v_{\mathscr{S}}$ for how they apply to complex wffs (it is obvious that a structure is defined by assignments to the basic wffs, so we need no rules for them). We summarise these rules in Table 2.14. It can easily be seen that these rules capture exactly the truth tables for the various truth functor connectives that we defined in §2.2.1; these rules, then, capture the logical meaning of a connective, since the rules are same for every structure, but the valuation on basic sentences differs from structure to structure. The kind of valuation function we've described is sometimes called a *Boolean valuation*, or *two-valued*. There are other kinds of valuation: for instance, we may consider a valuation that assigns *three* truth values to basic sentences: perhaps 'true', 'false' and 'undecided'. The structures for such a valuation are obviously different, and the rules need to be changed also, but it can be done. We shall not in this course consider any other than Boolean or two-valued logics.

Table 2.14 Rules on Boolean valuation function $v_{\mathscr{S}}$ for complex wffs in a structure $\mathscr{S}$.

$v_{\mathscr{S}}(\neg\varphi) = \top$	iff	$v_{\mathscr{S}}(\varphi) = \bot$.
$v_{\mathscr{S}}(\varphi \wedge \psi) = \top$	iff	$v_{\mathscr{S}}(\varphi) = \top$ and $v_{\mathscr{S}}(\psi) = \top$.
$v_{\mathscr{S}}(\varphi \vee \psi) = \top$	iff	$v_{\mathscr{S}}(\varphi) = \top$ or $v_{\mathscr{S}}(\psi) = \top$ (or both).
$v_{\mathscr{S}}(\varphi \rightarrow \psi) = \top$	iff	$v_{\mathscr{S}}(\varphi) = \bot$ or $v_{\mathscr{S}}(\psi) = \top$ (or both).
$v_{\mathscr{S}}(\varphi \leftrightarrow \psi) = \top$	iff	$v_{\mathscr{S}}(\varphi) = v_{\mathscr{S}}(\psi)$.
$v_{\mathscr{S}}(\varphi\|\psi) = \top$	iff	$v_{\mathscr{S}}(\varphi) = \bot$ or $v_{\mathscr{S}}(\psi) = \bot$ (or both).

Substitution Again One quick observation: we can now see in what way restricted substitution was rightly named: for it only allowed substitution of sentence letters for sentence letters. We should, it seems, allow inter-substitution of any formulae with the same truth value for any formula in

a sentence, and that will give the same truth value (in the same structure). That is:

Uniform Substitution Suppose in some structure $\mathscr{S}$, formulae φ and ψ have the same truth value. Suppose further that φ appears as a constituent part of some formula Ξ, and ψ does not so appear. Then if we uniformly substitute ψ in every place that φ appears in Ξ, then the resulting formula Ξ' will have the same truth value in $\mathscr{S}$ as Ξ does.

(Of course, this is a substitution *schema*: there is really an infinity of such substitution principles, one for each φ, ψ and Ξ.) Any formula Ξ' which results from an operation of uniform substitution on Ξ is known as a *substitution instance* of Ξ.

Recall our definition of logical equivalence (§2.3.1). It is clear that, since logical equivalent sentences have the same truth value in every structure, that we can always substitute logically equivalent formulae into a formula and preserve the truth value (Hodges, 2001, thm. X, p. 111). That means we can substitute logically equivalent sub-formulae uniformly into a formula Ξ, and the result is a formula Ξ' which is logically equivalent to the original formula ($\Xi \equiv \Xi'$).

Consistency defined We can turn these observations into a formal and mathematical account of the concept of consistency, one of our original motivations.

Consider a truth table for some well formed formula φ. Recall that a set of sentences was defined as consistent (page 11) just when there was a possible situation that those sentences describe. Accordingly, since truth tables describe each of the possible situations we are considering, a set of sentences would be consistent just if there was a row of a truth table in which each of that set of sentences was true.

Truth tables test consistency Truth tables, therefore, are a test of consistency, in the following fashion. Begin by taking a set S of sentences of $\mathscr{L}$. Write down a truth table that includes each and every basic sentence that appears as part of a sentence in S. Write each member of S out, and evaluate their truth according to the truth table.[3] If there is one row, i.e. one possible situation, in which a $\top$ appears for each member of S, then S is consistent; if there is no such row, S is inconsistent.

[3]If you are unfamiliar with the notion of membership of a set, you might wish to consult Appendix B.

We can use the approach using valuation functions in exactly the same way; we simply check the truth of the wff on all valuation functions. We can make this quicker if we wish. Assume that φ is the wff we wish to test for consistency. If there is a valuation function $v_{\mathscr{S}}$ such that it assigns $\top$ to φ, then φ is obviously consistent. So assume that $v_{\mathscr{S}}(\varphi) = \top$. Then we can apply, in reverse, the rules governing the valuation function: so if φ is a conjunction $\chi \wedge \psi$, we know that $v_{\mathscr{S}}(\chi) = \top = v_{\mathscr{S}}(\chi)$; if φ is a disjunction, we know that at least one conjunct gets assigned $\top$, *&c*. We can then, exploring these options, 'decompose' φ into its constituents, until either we uncover the values $v_{\mathscr{S}}$ assigns to basic wffs such that φ is true (so φ is consistent), or there is no such consistent valuation function, in which case φ is not consistent. The tableau method, which will be introduced below (§2.8), is basically a syntactical reformulation of this semantic method of evaluating consistency.

Logical truth, tautologies and contradictions Consider a set consisting of a single sentence of $\mathscr{L}$, $\{\varphi\}$. If that set is, by itself, inconsistent, then we call φ a *contradiction*. If the set $\{\neg\varphi\}$ is inconsistent, then we call φ a *tautology*. We also call tautologies *logical truths*: this is because they are true in every situation, regardless of the meanings of the constituent sentences, and hence those meanings are irrelevant to the truth of φ, which is, we say, true in virtue of logic alone, and logical form alone. Such formulae are also known as *theorems* of logic. It is easy to see, in our earlier terminology, that a tautology is true in every structure, and hence gets $\top$ on every line of the truth table; equivalently, if for every structure $\mathscr{S}$, $v_{\mathscr{S}}(\tau) = \top$, then τ is a tautology.

Substitution into tautologies Since a tautology is true in every situation, we can see that arbitrary formulae can be substituted into it, regardless of the situation. Since all the truths of logic are tautologies, we can see that all the truths of logic are purely formal, in the sense that substitution of arbitrary sub-formulae doesn't affect their truth value. This is the sense in which logic is formal or topic-neutral—the meanings of the constituent sentences are irrelevant.

Exercises for §2.6

Exercise 2.6.1: What does it mean for a wff to be true in a possible situation, or structure? How does the valuation function v correspond to the truth tables for the truth functors in $\mathscr{L}$.

Exercise 2.6.2: What is the principle of uniform substitution? Why is it important that we have this principle in our logic?

Exercise 2.6.3: How do truth tables test for consistency? Use truth tables to test the following wffs:

1. '$p \wedge \neg p$';
2. '$p \vee \neg p$';
3. '$(p \rightarrow q) \wedge (p \rightarrow \neg q)$';
4. '$p \wedge ((p \rightarrow q) \wedge (p \rightarrow \neg q))$';
5. '$(p \wedge (p \rightarrow q)) \wedge \neg q$';
6. '$\neg(p \rightarrow (p \rightarrow (q \rightarrow p)))$';
7. '$p \vee \neg(p \wedge q)$';
8. '$p \vee (\neg p \wedge \neg q)$'.

Exercise 2.6.4: Knaves always lie, while knights always tell the truth. In Tasmania, where everybody is one or the other (but you can't tell which by looking), you encounter two people, one of whom says "He's a knight or I'm a knave". What are they?

2.7 Arguments

2.7.1 Truth Tables and Arguments

Truth tables test validity Recall that an argument is valid when, if the premises are true, the conclusion must also be true. That means, in every possible situation, if the premises are true, so too must the conclusion be—which entails that the set of sentences consisting of the premises and negation of the conclusion must be inconsistent, for there is no possible situation in which they jointly obtain. So we can use truth tables to test arguments for validity too. Indeed, when an argument is invalid, truth tables will not only tell us, but give us a possible situation, a row on the truth table, which is the counterexample set to that argument: namely, that row where the counterexample set is consistent.

Example Consider the argument '$\neg p; p \vee q; q \leftrightarrow r$; therefore r'. The counterexample set is: $\{\neg p, p \vee q, q \leftrightarrow r, \neg r\}$. The truth table can be seen in Table 2.15. It is easy to see that this argument is valid, since there is no row of the truth table, no possible situation, in which each of these sentences is true. Therefore there is no counterexample to the argument, and hence it is valid.

Table 2.15 The argument from premises $\{\neg p, p \vee q, q \leftrightarrow r\}$ to conclusion r is valid.

p	q	r	$\neg p$	$p \vee q$	$q \leftrightarrow r$	$\neg r$
$\top$	$\top$	$\top$	$\bot$	$\top$	$\top$	$\bot$
$\top$	$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\top$
$\top$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\bot$
$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\top$
$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\bot$
$\bot$	$\top$	$\bot$	$\top$	$\top$	$\bot$	$\top$
$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\bot$
$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\top$

Irrelevant arguments Consider the argument 'p; q; therefore p'. This is intuitively valid; we can certainly, though perhaps not usefully, infer p from p itself. The counterexample set for this argument is $\{p, \neg p, q\}$; it is obvious, even without truth tables, that this set is inconsistent, and hence the original argument is valid. But notice a curious thing. Consider now a closely related counterexample set, $\{p, \neg p, \neg\neg q\}$ (simply replacing 'q' by the equivalent '$\neg\neg q$'). This is the counterexample set for the argument p; $\neg p$ therefore $\neg q$. This seems crazy: from a contradiction, a logical falsehood, anything follows! But we have to accept that the first argument is valid; and from our definition it follows immediately that the second argument is exactly the same, so it must be valid too. Also, we must admit that no practical harm will come of this: if we were to believe a contradiction, then that would be problem enough for us to have to revise our beliefs; it is no further problem that we should thereby be committed to everything by logic. This argument then provides a way of dramatising the problem with believing a contradiction.[4]

Relevance Many people have tried to augment the original definition of validity in terms of necessary truth preservation by adding a requirement of *relevance* (Beall and van Fraassen, 2003, §7.3). Intuitively, in a persuasive argument the premises should be relevant to the conclusion, and these strange arguments from necessary falsehoods (or to necessary truths) do not satisfy this constraint. It turns out to be extremely difficult to make this intuitive thought about relevance into a precise notion that can be apparent

[4]This apparent problem with arguments recalls some of the problems with conditional claims (page 34): we shall soon see that there is a good reason for this!

just from the formal logical structure of an argument. After all, we can use different words to refer to the same things, and the same words to refer to different things, so it will not be obvious when a given sentence is or is not relevant to another. In some sense, we've already given a minimal theory of relevance: the idea that in a valid argument the truth values of the premises and conclusion are connected in some way, and hence must be relevant to each other to a certain degree. As incorporating a more powerful conception of relevance is difficult, and we will never in practice be led astray by our definition of validity, we will set aside these scruples. After all, these irrelevant arguments will be bad for other reasons: they will not be persuasive, and they rely on premises or conclusions that are in various ways either bad or pointless to assert, so there is a good pragmatic explanation of why these arguments should sound terrible to our ears. If we have this pragmatic explanation of their awfulness readily available, must we also claim that these arguments are invalid?

Truth tables are decidable One can tell that the truth-table method for deciding validity is completely automatic: the rules for the connectives determine the truth values of the complex sentences, and we can see that, in principle, we could program a computer to generate all the combinations of truth and falsity for each basic sentence, and then build up the truth values of the more complex sentences. This automatic procedure would then tell us either that the argument was valid, or that it was invalid, and in the latter case it would provide us with a counterexample set. This fact, that truth tables are automatic in this way, is known as the *decidability* of propositional logic: an automatic procedure can decide, for each set of sentences, whether it is consistent or inconsistent, and hence for each argument, whether it is valid or invalid.

What's wrong with truth tables? Truth tables seem wonderful: but that is something of an illusion. Consider a complicated argument, using 30 different basic sentences. The truth table for such an argument has 2^{30} rows; which is more than 1 billion rows. This is impractical, to say the least, and may even be impossible.[5]

[5] *This note should only be read after completing this chapter.* It must be admitted that this justification for looking elsewhere than truth tables is not always sound. Consider a very complex sentence that has only one sentence letter: for instance $p \vee ((p \vee \neg p) \leftrightarrow (p \wedge \neg(p \rightarrow (p \leftrightarrow \neg\neg p))))$. This truth table has only two rows; but all tableaux for it will be quite long, because of the internal complexity of the sentence. A better justification may be sought in the

Exercises for §2.7

Exercise 2.7.1: Use truth tables to test whether the following arguments are valid; if they are not, give a counterexample situation. Formalise the arguments if necessary.[6]

1. "Thin is guilty," observed Watson, "because (i) either Holmes is right and the vile Moriarty is guilty, or he is wrong and the scurrilous Thin did the job; but (ii) those scoundrels are either both guilty or both innocent; and, as usual (iii) Holmes is right."
2. $p \vee q$; $p \to r$; $q \to r$; therefore r.
3. (i) If Min is home, so is Henry; (ii) If Min is home, Henry isn't home; therefore, Min's not home.
4. We know that Min is on board if Henry is home. So she has to be on board if she's home, because Henry's home if she is.
5. $q \to ((p \to (p \to p)) \to (p \to q))$; therefore $((p \to q) \to p) \to p$.

Exercise 2.7.2: What is curious about the following arguments? Are they valid? Are they good arguments?

1. I'll ski tomorrow; therefore, I'll ski tomorrow if I break my leg today.
2. If either Adams or Brown will operate, and Brown won't operate, then if Adams won't, Brown will.
3. 'We'll win, for if they withdraw if we advance, we'll win; and we won't advance!'

Exercise 2.7.3: Is is a problem for logic if a contradiction entails everything; or that a tautology is entailed by everything?

Exercise 2.7.4: Is the following argument valid? Can you tell by truth tables?

$$\begin{array}{rc} 1. & p_1 \to p_2 \\ 2. & p_2 \to p_3 \\ \vdots & \vdots \\ i. & p_i \to p_{i+1} \\ \vdots & \vdots \\ 100. & p_{99} \to p_{100} \\ \hline C. & p_1 \to p_{100} \end{array}$$

fact that truth tables provide a method for semantic validation of a wff, but we desire a purely syntactic proof method.

[6] Apologies to Jeffrey (1991, §1.18)

2.8 Tableaux

Explicit contradictoriness There are, fortunately, more efficient methods for testing consistency and validity than testing the sentence in every possible situation. Consider the sentence

(2.1) $((((p \vee q) \vee r) \vee s) \vee t) \wedge \neg((((p \vee q) \vee r) \vee s) \vee t)$.

To check the consistency of this sentence, we need a truth table with $2^5 = 32$ rows, and it would take a great deal of time to draw up such a table. But it is clear that this sentence is an explicit contradiction: it is of the form $\varphi \wedge \neg\varphi$. It would be great if we could find a method that would enable us to quickly and easily test for explicit contradictoriness of a sentence; or at least to check whether a sentence has as a consequence a contradiction, without testing all the possible situations in which it might be true.

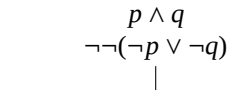
Such a method is provided by *semantic tableaux*. This is a technique for testing the consequences of (sets of) sentences that provides a quick way of seeing if a contradiction exists between them.

Setting up a tableau We begin in the same way as when testing an argument for validity using the truth table method. That is, we test the counterexample set to see if it is consistent. Consider the proof of one of the De Morgan equivalences: say, that $p \wedge q$ entails $\neg(\neg p \vee \neg q)$. We list the counterexample set in a linear fashion, as in Fig. 2.1(a).

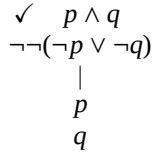
Constructing a valuation We then proceed to draw out consequences of this set, in the following fashion. We try to describe a situation in which the counterexample set is true; that is, we try and narrow down a particular situation which shows the counterexample set to be consistent, and hence that the argument is invalid. We do this by looking at what must be true in order for a sentence we've written down to be true. In this task we use as our guide the rules on valuation functions we discussed earlier (Table 2.14). That is, *we try to construct a valuation function on basic wffs that makes the counterexample set consistent*.

Tableau rules: conjunction So, for instance, take the first wff of the tableau above. It is a conjunction; by the rules for conjunctions, we know that if this wff is to be true ($v(p \wedge q) = \top$), each conjunct needs to be true (both $v(p) = \top$ and $v(q) = \top$). We indicate this by inscribing p and q below

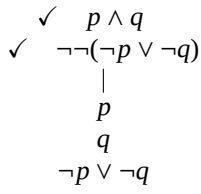
Figure 2.1 Steps in the construction of a tableau.



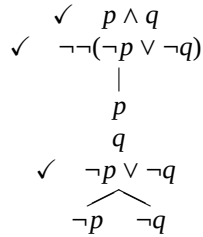
(a) Setting out the counterexample set to begin a tableau.



(b) First steps in the construction of a tableau: conjunction.



(c) Dealing with double negation.



(d) Dealing with disjunction using branching tableaux.

the original pair of wffs, as in Fig. 2.1(b). (We mark a wff we have dealt with by placing a ✓ next to it—this mark is no part of the tableau, but is rather decoration we use to help us keep track of the construction of the tableau.) This action indicates that p and q have to be true in order to make the more complex sentence true; and since p and q are basic sentences, we now know that the basic valuation function if it is to make the counterexample set consistent must evaluate p and q as true.

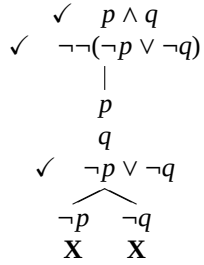
Tableau rules: double negation Next we turn to $\neg\neg(\neg p \vee \neg q)$; we see by the rules for negation, a double negation is true just when the wff without the two negation signs is true. So we inscribe $\neg p \vee \neg q$, and check off the original wff, as in Fig. 2.1(c).

Tableau rules: disjunction Now we come to something of a problem. The rules for disjunction do not specify what must be true in order for the disjunction to be true: they give two options, either of which may be true, and neither of which can be assumed to be true. So we cannot write down just one valuation which captures the content of a valuation function over a

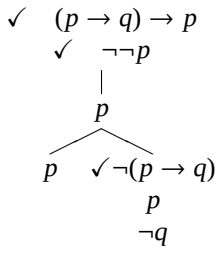
disjunction. In tableaux, we capture this by introducing *branches*: one for each disjunct, as in Fig. 2.1(d).

Leaves, roots and branches Now we can break the wffs in the tableau down no further: we have the smallest parts of the wffs in question. We see, now, if they describe a possible valuation, as follows. Call the wffs inscribed above the line the *root* of a tableau. Call the wffs inscribed at the bottom of a tableau the *leaves*. A *branch* is a sequence of wffs that begins with the root, and terminates with exactly one leaf, and includes every wff above and below every wff within the branch. So, for instance, $\langle p \wedge q, \neg\neg(\neg p \vee \neg q), p, q, \neg p \vee \neg q, \neg p\rangle$ is a branch. A branch is called *closed* if there are two wffs on the branch of the form φ and $\neg\varphi$. So this example branch is closed; indeed, our whole tableau has only closed branches, and we call the whole tableau closed in this case. We mark a branch closed by putting a big bold **X** at its tip, as in Fig. 2.2(a). (Like the ✓, this is decoration; really, a branch is closed iff a wff and its negation both occur, whether we write the **X** or not.)

Figure 2.2 Open and closed tableaux.



(a) A closed tableau.

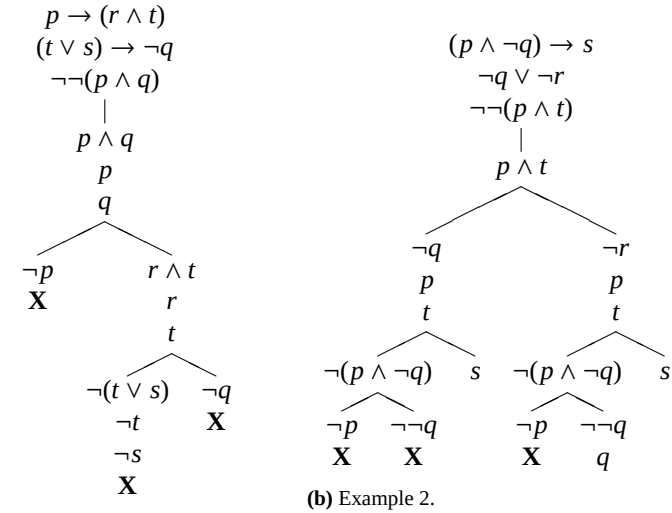


(b) An open tableau.

Consistent valuations If a tableau is not closed, take one of its open branches, call it *B*. This branch *B* contains basic wffs and negated basic wffs. If we construct a valuation function v_B that assigns $\top$ to basic wffs which appear on the branch, and $\bot$ to basic wffs that appear only negated on the branch, we have found a consistent valuation function, and hence have described a possible situation which makes the counterexample set true. Since the tableau in Fig. 2.2(a) has no open branches, there is no valuation function that makes the counterexample set true, and hence no possible situation that is described by the counterexample set: therefore the argument is valid.

Example Consider the tableau in Fig. 2.2(b), with two open branches. Choose the left branch; this defines a valuation function v that makes the counterexample set true: any function such that $v(p) = \top$. We can see by truth tables that this valuation does provide a counterexample to the argument $(p \to q) \to p$ therefore $\neg p$. The right branch, too, provides a counterexample valuation: $v(p) = \top, v(q) = \bot$.

Figure 2.3 Tableaux for examples 1 and 2.



(a) Example 1.

(b) Example 2.

Two more complicated examples

1. Consider the argument $p \to (r \wedge t); (t \vee s) \to \neg q$; therefore $\neg(p \wedge q)$. The tableau is shown in Fig. 2.3(a). This tableau is closed; hence the original argument was valid.

2. Consider the argument $(p \wedge \neg q) \to s; \neg q \vee \neg r$; therefore $\neg(p \wedge t)$. The tableau is pictured in Fig. 2.3(b). As is readily seen, this tableau is open; hence the argument is not valid. The branch terminating $\langle \ldots, \neg r, p, t, s \rangle$ gives the counterexample valuation $v(r) = v(q) = \bot$, $v(p) = v(t) = v(s) = \top$, which describes a possible situation where the argument fails. The tableau for this example shows something that

was implicit in what we said above: that when a rule is applied, *the results of that rule are added to every open branch.*

Complete tableaux rules The full set of rules for propositional (truth-functional) tableaux are set out in Fig. 2.4 (page 57).

Exercises for §2.8

Exercise 2.8.1: Use the tableaux rules to show that the following wffs are theorems, or to give a valuation (structure) which makes the wff false.

1. $(((p \leftrightarrow q) \vee (p \leftrightarrow r)) \vee (q \leftrightarrow r))$.
2. $(\neg p \vee q) \leftrightarrow (p \rightarrow q)$.
3. $\neg p \rightarrow (p \rightarrow q)$.
4. $q \rightarrow (p \rightarrow q)$.
5. $(p \rightarrow r) \rightarrow ((p \wedge q) \rightarrow r)$.

Exercise 2.8.2: Use the tableaux rules to show that the following are valid arguments, or to give a counterexample valuation (structure).

1. $p \wedge \neg q$, $(\neg q \rightarrow \neg p)$ therefore r.
2. q; therefore $(\neg p \vee q) \leftrightarrow (p \rightarrow q)$.
3. $\neg p \vee q$; p therefore q.
4. p; $p \rightarrow (q \vee r)$; therefore $\neg r \rightarrow q$.
5. $p \wedge q$; $q \rightarrow r$; $s \rightarrow \neg t$; $t \rightarrow \neg q$; $r \rightarrow s \vee t$; therefore $\neg t$.

Exercise 2.8.3: Translate the following argument into a sequent of $\mathscr{L}$, providing a suitable interpretation, and commenting on any difficulties.

> If Smith were offered the job, we would have to find a position for her husband. Smith's husband is an electrician; and even if we needed an electrician, we can't afford to employ him. So we cannot have Smith. By contrast, Brown's partner is eminently employable, and indeed would have been a strong candidate in his own right; but Brown himself is not qualified. Smith and Brown are the only alternatives to Jones worth discussing. So Jones should get the job, if she's willing to take it. Thankfully, we've been assured she will.

Decide whether the sequent is correct; if not, provide a counterexample structure.

Exercise 2.8.4: Consider an operator $\odot$ defined by the following tableau rules:

$$\begin{array}{c}(\varphi \odot \psi)\\ \mid \\ (\varphi \odot \varphi)\\ (\psi \odot \psi)\end{array} \qquad\qquad \begin{array}{c}((\varphi \odot \psi) \odot (\varphi \odot \psi))\\ \diagup \quad \diagdown \\ \varphi \qquad \psi\end{array}$$

1. Give an ordinary English reading of '$\varphi \odot \psi$'.
2. If these rules were the only rules for a system of tableau, under what conditions should a branch be counted as closed?

2.9 Semantic Sequents

Some new notation Now we have a formal language, $\mathscr{L}$, and two techniques (tableau and truth-tables) that we can use to establish what we have been calling consistency of a set of formulae, and therefore the validity of arguments that can be phrased in $\mathscr{L}$. We now wish to talk *about* the formal language we have defined, and its properties. We have been using mostly English to do this so far, augmented with some technical terms. We shall now introduce some new notation to talk about the validity of arguments, beginning with the concept of semantic entailment:[7]

Definition 1 (Semantic Entailment). A set of wffs Ξ *semantically entails* a wff φ, written '$\Xi \vDash \varphi$', if every possible structure (row of a truth table, or Boolean valuation function) that makes each member of Ξ jointly true, also makes φ true.

It is obvious from the definition that if $\varphi \in \Xi$, then $\Xi \vDash \varphi$ is correct (Hodges, 2001, §24, Theorem II). Therefore all sequents of the form $\varphi \vDash \varphi$ are correct.

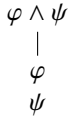
Sequents The sentence schema $\Xi \vDash \varphi$ is known as a *semantic sequent*. The symbol $\vDash$, known as the semantic turnstile, is no part of $\mathscr{L}$: it serves to talk about validity of arguments of $\mathscr{L}$. The relation is obvious: given how we have defined validity, it is clear that an argument '$p_1, \ldots, p_n$; therefore q' is valid just in case the premises semantically entail the conclusion, and we write '$p_1, \ldots, p_n \vDash q$' to express this fact.[8]

Metalanguage and object language We can put this point another way: we call the language we use to talk about arguments and formulae of $\mathscr{L}$ the *metalanguage*, and $\mathscr{L}$ we call the *object language* (it is the object of our discussion). When we use our metalanguage to talk about $\mathscr{L}$ we merely

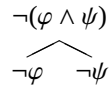
[7] Readers who have not done so already are advised to consult Appendix B at this point; it will be presupposed in what follows.

[8] See Bostock (1997, §1.3) for more on sequents.

Figure 2.4 Propositional Tableau Rules

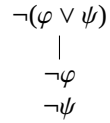


(a) Conjunction



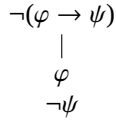
(b) Negated Conjunction

(c) Disjunction

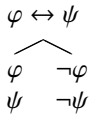


(d) Negated disjunction

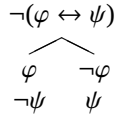
(e) Conditional



(f) Negated Conditional



(g) Biconditional



(h) Negated Biconditional

(i) Double Negation

mention wffs of $\mathscr{L}$, to talk about their semantic or syntactic properties, but not to use them to express what they express.

2.9.1 Manipulating sequents

There are several rules governing what admissible transformations of sequents there are.

Contraction If the sequent $\Xi, \varphi, \varphi, \Theta \vDash \psi$ is correct, then so is the sequent $\Xi, \varphi, \Theta \vDash \psi$ (i.e. duplicates can be removed).

Permutation If the sequent $\Xi, \varphi, \psi, \Theta \vDash \pi$ is correct, then the sequent $\Xi, \psi, \varphi, \Theta \vDash \pi$ is also correct (i.e. order can be rearranged).

Weakening If the sequent $\Xi \vDash \varphi$ is correct, so is the sequent $\Xi, \pi \vDash \varphi$ (i.e. semantic entailment is preserved under addition of new premises) (Hodges, 2001, §24, Theorem I).[9]

Equivalents If $\Xi \vDash \varphi$ is correct, then so is any sequent $\Omega \vDash \psi$ where Ω is obtained from Ξ by replacing a set of subformulae of Ξ by a set of formulae that are true in exactly the same structures, and ψ is a logical equivalent of φ (i.e. sequents are stable under substitution of logical equivalents).

We can easily justify these rules using truth table methods. That is, if two copies of φ are premises already, then removing one φ cannot change any of the structures that the premises are true within, hence contraction. Or we can justify weakening by saying that if ψ is true in every structure where Ξ are all true, then adding a new premise φ only cuts down the numbers of structures, but does not introduce any new structures, hence it is still the case that every one of the new structures also has ψ true. And it is obvious that changing the order of premises does not change the structures that make them true.

The Equivalents rule does the most work for us, however. This rule allows the following manipulation: from the sequent $\varphi, \psi \vDash \chi$ we can legitimately infer $\varphi \wedge \psi \vDash \chi$. That is one example of the fact that all of the rules for logical connectives can be applied to sequents. Moreover, logical equivalents have been semantically defined in terms of structures, so it is easy to see that Equivalents is true, since in considering the correctness of the first sequent, or the second, modified, sequent, we consider exactly the same set

[9] Also called 'thinning', e.g. by Bostock (1997, §2.5.B).

of structures both times, and in the same way. (See also Hodges (2001, §24, Theorem IX).)

2.9.2 Theorems concerning Sequents

Consistency We can apply the above definition of entailment, as expected, to get a semantic link between consistency and entailment.

DEFINITION 2 (SEMANTIC INCONSISTENCY). A set of wffs Ξ of $\mathscr{L}$ is *semantically inconsistent*, written $\Xi \vDash$, if Ξ semantically entails an arbitrary wff. If every structure which makes each member of Ξ true also makes true an arbitrary wff, then every structure which makes every member of Ξ true also makes $p \wedge \neg p$ true (for that is an arbitrary wff); and since no structure makes a contradiction true, there can be no structure which makes every member of Ξ true. A semantically inconsistent set of wffs can't all be true in any structure.

Obviously, we have the following further connection between consistency and entailment: $\Xi \vDash \varphi$ iff $\Xi, \neg\varphi \vDash$ (Hodges, 2001, Equation 23.10).

Theorems and Contradictions This fact alone gives us some interesting properties of sequents. A contradiction is a single wff χ that is inconsistent: $\chi \vDash$. Any contradiction is true in just the same circumstances as any other contradiction (that is, none), so by Equivalents, we can substitute a contradiction into the sequent: $\neg(p \vee \neg p) \vDash$. By the definition of inconsistency, we can swap sides: $\vDash p \vee \neg p$. This sequent expresses the fact that $p \vee \neg p$ is a *tautology*, because it is true in every structure, and hence needs no special premises. We can then use Equivalents again to substitute any tautology for $p \vee \neg p$, and see that the sequent $\vDash \varphi$ says that φ is a theorem.

Logical equivalence The definition of logical equivalence given above (§2.3.1) can also be recast. Since $\varphi \vDash \varphi$ is correct, if ψ is logically equivalent to φ, then $\varphi \vDash \psi$, and $\psi \vDash \varphi$ are both correct sequents too, by Equivalents; and we can define logical equivalence as holding just when two sequents of this form are both true.

Cut and Transitivity Assume that $\Xi, \varphi \vDash \psi$, and $\Xi \vDash \varphi$. It is clear that if every structure in which Ξ is true is one in which φ is true, then it is clear that the set of structures which make Ξ, φ true is just the set of structures which

makes Ξ true. By Equivalents, then, $\Xi \vDash \psi$ Hodges (2001, §24, Theorem III).

This also shows *transitivity*: if $\varphi \vDash \psi$, and $\psi \vDash \pi$, then by Weakening and Permutation $\varphi, \psi \vDash \pi$, and by Cut $\varphi \vDash \pi$.

Substitution If $\Xi \vDash \pi$ is correct, and φ appears as part of Ξ or π (or both), and ψ does not. By Uniform Substitution (page 44), $\Xi' \vDash \pi'$, where Ξ' (and π') are the result of uniformly substituting ψ for φ throughout Ξ (and π) (Hodges, 2001, §24, Theorem V). By extension of our earlier notion, call one sequent a *substitution instance* of another if each formula in the former sequent is the result of applying the same uniform substitution rule to each formula in the latter sequent.

Contraposition $\varphi \vDash \psi$, iff $\neg\psi \vDash \neg\varphi$. We can prove this easily: $\varphi \vDash \psi$ iff $\varphi, \neg\psi \vDash$ iff (by Equivalents) $\neg\neg\varphi, \neg\psi \vDash$ iff (by permutation) $\neg\psi, \neg\neg\varphi \vDash$ iff $\neg\psi \vDash \neg\varphi$.

Duality revisited We are now in a position to prove two further results about duality.

Corollary 1 (Duality II). If $\varphi \vDash \psi$ then $\psi^D \vDash \varphi^D$.

Proof. Assume $\varphi \vDash \psi$. By uniform substitution, $\widetilde{\varphi} \vDash \widetilde{\psi}$. By contraposition, $\neg\widetilde{\psi} \vDash \neg\widetilde{\varphi}$. By Theorem 1, $\psi^D \vDash \varphi^D$. ■

Corollary 2 (Duality III). If $\varphi \equiv \psi$ then $\varphi^D \equiv \psi^D$.

Proof. Immediate from Corollary 1. ■

2.9.3 The Deduction Theorem

We now prove the following important theorem:

Theorem 2 (Deduction). *$\Xi, \varphi \vDash \psi$ is a correct sequent iff $\Xi \vDash \varphi \to \psi$.*

Proof. $\Xi, \varphi \vDash \psi$ is correct iff $\Xi, \varphi, \neg\psi \vDash$ is correct (by consistency); in turn, that is correct iff $\Xi, \neg(\varphi \to \psi) \vDash$ is correct (by Equivalents). That last sequent is correct iff $\Xi \vDash \varphi \to \psi$, by consistency again. Hence we have shown the two sequents are correct in the same cases. ■

Arguments and theorems This theorem has the following consequence: consider some valid argument, $\xi_1, \ldots, \xi_n \vDash \varphi$. By Equivalents, this is correct iff $\xi_1 \wedge \ldots \wedge \xi_n \vDash \varphi$ is correct. And, by the Deduction theorem, this is correct iff $\vDash (\xi_1 \wedge \ldots \wedge \xi_n) \rightarrow \varphi$. That is, an argument is valid iff the conditional with the conjunction of the premises as antecedent, and the conclusion as consequent, is a theorem of logic.

Semantic Sequents and 'Rules of Inference' Some formulations of logic proceed very differently from ours (Bostock, 1997, ch. 5). In those systems ('axiomatic systems'), we begin with some certain truths (the *axioms*, under the intended interpretation) and some *rules of inference* that allow one to move, safely, from those axioms to other derived truths with equal claim to certainty, called *theorems*. While we won't discuss such systems further, it is useful to think about rules of inference in our system. The most common rule in such systems is known as *modus ponens*, the rule that if φ is a theorem, and $\varphi \rightarrow \psi$ is a theorem, then ψ is also a theorem.

THEOREM 3 (MODUS PONENS FOR SEMANTIC SEQUENTS). *If* $\vDash \varphi$ *and* $\vDash \varphi \rightarrow \psi$, *then* $\vDash \psi$.

Proof. Assume that $\vDash \varphi$ and $\vDash \varphi \rightarrow \psi$. By the Deduction theorem, $\varphi \vDash \psi$. By Cut, $\vDash \psi$. ■

Other rules can be given completely similar justifications. Normally, of course, such rules are treated as purely syntactic manipulations on the wffs accepted as axioms; Theorems 6 and 7 will demonstrate that this semantic proof of the acceptability of these rules carries over to their syntactic counterparts.

2.9.4 Interpolation

Definability A wff φ is said to be *defined* in a structure $\mathscr{S}$ if every basic wff that is a constituent of φ is assigned a truth value by $\mathscr{S}$. That is, if there is a row of the truth table that mentions each basic wff, or a valuation function that does so.

Interpolation We now prove the theorem (Hodges, 2001, §24, theorem XII).

Theorem 4 (Interpolation). *If $\varphi \vDash \psi$, and there is a non-empty set B of basic wffs that are constituents of both φ and ψ, then there is a wff χ such that both $\varphi \vDash \chi$ and $\chi \vDash \psi$, and each basic wff in χ is in B.*

Proof. Let $\mathscr{B} = \{\mathscr{B}_1, \ldots, \mathscr{B}_n\}$ name the set of all structures $\mathscr{B}_i$ such that all and only basic wffs in B are assigned a truth value (either $\top$ or $\bot$) in $\mathscr{B}_i$. Call a structure $\mathscr{A}$ an *extension* of a structure $\mathscr{C}$ just in case $\mathscr{A}$ differs from $\mathscr{C}$ only in assigning a truth value to some basic wffs that $\mathscr{C}$ does not assign a value to, and agrees in its valuation of the other basic wffs that $\mathscr{C}$ does assign a value to. If φ is true in a structure, it is true in every extension of that structure.

We now construct a valuation, as follows. For each structure $\mathscr{B}_i$, if that structure has an extension that makes ψ false, assign a $\bot$ to $\mathscr{B}_i$; otherwise assign a $\top$ to $\mathscr{B}_i$. This is an assignment of truth values to structures, and hence is a truth table. We know from §2.3.1 that for any truth table, there is a DNF formula that possesses that truth table and that can be easily constructed from that truth table. Let this formula be defined for our recently constructed truth table: it is our needed interpolant, χ.

χ clearly entails ψ, since by construction no structure that makes χ true can be extended to one that makes ψ false. Moreover, χ is clearly entailed by φ, since if there were a structure $\mathscr{D}$ which made φ true but χ false, $\mathscr{D}$ would be an extension of some $\mathscr{B}_i$ that make ψ true; but by construction χ would be true in $\mathscr{B}_i$ and in every extension of it, including $\mathscr{D}$. But there can therefore be no such $\mathscr{D}$ that both makes χ true and false. ■

Example Consider the correct sequent $\neg(p \vee q) \wedge (p \leftrightarrow r) \vDash (r \rightarrow q) \wedge \neg(s \wedge r)$. The set $B = \{q, r\}$. According to the rules described above, we construct the truth table (Table 2.16). It is clear that the needed interpolant is $(q \wedge \neg r) \vee (\neg q \wedge \neg r)$; indeed, this is logically equivalent to just $\neg r$.

Table 2.16 Example of interpolation truth table.

q	r	?
$\top$	$\top$	$\bot$
$\top$	$\bot$	$\top$
$\bot$	$\top$	$\bot$
$\bot$	$\bot$	$\top$

Exercises for §2.9

Exercise 2.9.1: Prove that $\vDash \varphi \to (\psi \vee \chi)$ iff $\varphi \vDash \psi, \chi$.

Exercise 2.9.2: Show that, if '$\varphi, \psi \vDash \chi$' is an incorrect sequent, it has a substitution instance '$\varphi', \psi' \vDash \chi'$', in which φ' and ψ' are tautologies and χ' is inconsistent.

Exercise 2.9.3: Show, using truth tables as your notion of structure, that the **Weakening** rule is valid for semantic sequents. Is this controversial?

Exercise 2.9.4: All of our sequents have at most one formula on the right side of the turnstile $\vDash$. In this question, we liberalise this, allowing sets of formulae to appear on either side of the turnstile. Now, a sequent $\Xi \vDash \Delta$ is to be read 'whenever *all* the wffs in Ξ are true, *at least one* wff in Δ is true'.

1. Suppose that Ξ and Γ only contain wffs in which the only connectives are from $\{\wedge, \neg, \vee\}$. Define $\Gamma^D = \{\gamma^D : \gamma \in \Gamma\}$, for any Γ (where the ^{D} operation, as in §2.3.2, swaps '$\wedge$' for '$\vee$', and vice versa, wherever they appear). Prove that $\Xi \vDash \Delta$ iff $\Delta^D \vDash \Xi^D$.
2. Give an example to show that $\vDash$ between *sets* of formulae is non-transitive.

Exercise 2.9.5: Show that the following rules of inference are acceptable in our framework:

1. *Disjunctive Syllogism*: If $\vDash \varphi \vee \psi$ and $\vDash \neg\varphi$, then $\vDash \psi$.
2. *Reductio ad Absurdum*: If $\varphi \vDash \psi$ and $\varphi \vDash \neg\psi$ then $\vDash \neg\varphi$.

What is the relationship between these acceptable rules of inference and the semantic sequents which express the correctness of arguments of similar form to the rules (e.g. these sequents: '$\varphi, \varphi \to \psi \vDash \psi$', '$\varphi \vee \psi, \neg\varphi \vDash \psi$', and '$\varphi \to \psi, \varphi \to \neg\psi \vDash \neg\varphi$')?

Exercise 2.9.6: In our system each acceptable rule of inference has a corresponding correct semantic sequent. Consider now this rule of inference in English: if φ is a truth of logic, then 'Necessarily, φ' is true. Is this rule intuitively correct? Is the corresponding sequent '$\varphi \vDash_{\text{English}}$ Necessarily φ' intuitively correct?

Exercise 2.9.7: Find an interpolant for these sequents:

1. $(\zeta \leftrightarrow \psi) \wedge \neg((\rho \to \neg\varsigma) \vee \neg(\varphi \to \psi)) \vDash (\tau \to (\upsilon \to (\neg\varphi \vee \psi)))$;
2. $(\varphi \to (\psi \to (\chi \vee (\xi \wedge \pi))), \xi \vee \rho \vDash \xi \to (\chi \wedge \neg\varphi)$.

2.10 Tableaux and Syntax

The role of valuations We gave rules for the tableaux system above (Fig. 2.4) that were inspired by the constraints on admissible Boolean valuations. But do they presuppose such semantic valuations? It must be admitted that they do not. We can thus propose an entirely formal, or syntactic, view of what a tableau is, just in terms of a set of rules for constructing tableaux that do not involve mention of truth or valuations.

Tableaux syntactically defined A *tableau* is a tree, in the mathematical sense, with the following properties: it has a root node, with one or more branches descending from that root, each branch of which consists of nodes and terminates in a leaf node. Each node of the tree contains one wff of $\mathscr{L}$. A set of wffs $\Xi = \{\xi_1, \ldots, \xi_n, \ldots\}$ *generates* a tableau if the *i*-th node of each branch of the tableau is the formula $\xi_i \in \Xi$, and each further node of the tree is the result of applying one of the patterns of wffs in Fig. 2.4 to a wff that lies earlier on the same branch (i.e. closer to the root). A branch is called *completed* if, for every wff φ on a branch, the resultant wff ψ from applying the appropriate rule to φ is also already on that branch (in the case of branch rules, if both resulting wffs appear on branches that each pass through the node at which φ appears). A tableau for which every branch is completed is itself *completed*. A completed branch is called *closed* if two wffs of the form φ and $\neg\varphi$ appear on that branch; otherwise the branch is *open*. A completed tableau that conforms to these rules is called *closed* if every branch is closed; otherwise it is called *open*. We shall call a tableau that is formed in accordance with these rules a *syntactic tableau*.

Definition 3 (Syntactic Consistency). A set of formulae Ξ is *syntactically consistent* if there exists a completed open tableau generated by Ξ; otherwise it is called syntactically inconsistent.

It is important to note that this definition never mentions truth or structures, and whether it captures the same thing as semantic consistency is yet to be seen, though we shall show soon that it does.

2.10.1 Syntactic Sequents

Syntactic turnstile We now introduce some new notation to talk about the properties of syntactic tableaux, which we shall call tableaux for the remainder of this section. If a set of formulae is syntactically inconsistent, we write:

$$\Xi \vdash . \tag{2.2}$$

Note that we use $\vdash$ to stand for the syntactic turnstile, which differs from the semantic turnstile $\models$: though they look similar for a reason, as we shall see.

Syntactic consequence If there exists a closed complete tableau generated by $\{\Xi, \neg\varphi\}$, we write $\Xi \vdash \varphi$ to express that φ is a syntactic consequence of

Ξ, or is provable using tableaux from Ξ. If Ξ is empty, we write $\vdash \varphi$, and in that case we call φ a syntactic theorem.

Syntactic Deduction theorem Just as we proved the deduction theorem (Theorem 2) for the semantic turnstile, we can prove a syntactic deduction theorem for the syntactic turnstile.

THEOREM 5 (SYNTACTIC DEDUCTION). *$\Xi, \varphi \vdash \psi$ iff $\Xi \vdash \varphi \rightarrow \psi$.*

Proof. If: Assume $\Xi \vdash \varphi \rightarrow \psi$. Then every completed tableau T which is generated by $\{\Xi, \neg(\varphi \rightarrow \psi)\}$ is closed. Take one such tableau, where the first tableau rule applied was the negated conditional rule, Fig. 2.4(f), so that each branch of the tableau begins $\langle \Xi, \neg(\varphi \rightarrow \psi), \varphi, \neg\psi \rangle$. We have already dealt with $\neg(\varphi \rightarrow \psi)$, so the wffs which close every branch on this tableau derive from Ξ or from φ or $\neg\psi$. Hence, we can construct a closed tableau where every branch begins $\langle \Xi, \varphi, \neg\psi \rangle$; hence the set $\{\Xi, \varphi, \neg\psi\}$ generates a closed tableau, hence $\Xi, \varphi \vdash \psi$.

Only if: We assume that $\Xi, \varphi \vdash \psi$. Then there is a closed completed tableaux that is generated by $\{\Xi, \varphi, \neg\psi\}$. If we modify this tableau by inserting $\neg(\varphi \rightarrow \psi)$ above φ on every branch, the modified tableau is closed also. Since φ and $\neg\psi$ can come from $\neg(\varphi \rightarrow \psi)$ (Fig. 2.4(f)), this modified tableau is a complete tableau generated by $\{\Xi, \neg(\varphi \rightarrow \psi)\}$, which means that $\Xi \vdash \varphi \rightarrow \psi$. ■

Why bother? A natural question at this point is why do we bother to define formally syntactic tableaux without reference to truth or structures? The first motive is to continue the process of abstraction from meaning that led us to formal logic in the first place: the continued presence of truth as a basic part of our judgements of validity is potentially confusing. What we want is a completely meaning-independent route to judge the validity of an argument, and truth and meaning are too closely connected. Secondly, some logicians have regarded the very notion of truth as suspect, and regard the activity of proof as the key to mathematical and other reasoning. They, of course, desire a system that formalises what it means to give a proof and makes no reference to the suspect notions, and even if we do not share their suspicions it seems well to separately define notions, if they are truly distinct. Proof of a theorem and the truth of a theorem are quite different activities, that may not necessarily go together, as some past mathematicians with poor methods of proof have shown.

There are some more interesting philosophical reasons as well. Many philosophers have been puzzled about the concepts of meaning and representation, particularly with the deep question of what special features does an object like a wff possess in virtue of which a wff can mean something—unlike, say, a rock, which is also an object but doesn't appear to have the capacity to mean anything. A demonstration that proof doesn't require understanding, and hence doesn't seem to require the meaningfulness of the syntactic items upon which a proof operatres, might seem to hold out the potential of dissolving these philosophical perplexities over meaning and representation. It might also seem to hold out the promise of explaining why objects, like machines, that can perform syntactic manipulations, can seem to exhibit intelligent behaviour; because, as it turns out, those syntactic manipulations that constitute a proof can be independently specified and yet (by the theorems about to be proved) end up corresponding to the semantically significant notions of entailment and validity. So we might have the possibility of artificial intelligence. Indeed, we might be able to explain how it is that a 'meat computer', like the human brain, can perform the intelligent action of which it is so clearly capable, without having to appeal to the essentially mysterious notions of consciousness or a 'soul'.[10]

Connecting syntactic tableaux and structures Having defined what it means to be a theorem, to be consistent, and to be valid, all without any mention of truth or structures, the natural question is: can we show that $\vdash$ and $\models$ hold between exactly the same formulae? That is, does our purely formal system of wff-manipulation correspond to the consequence relations between descriptions of structures? We shall now show that in our case they do: that syntactic tableaux prove everything that is true in our intended system of structures, and they prove only true claims.

Exercises for §2.10

Exercise 2.10.1: Why is it important for the definition of semantic consistency that a closed tableau be complete?

Exercise 2.10.2: What is a syntactic theorem? Prove that for any syntactic theorem τ, for any ψ, $\vdash \psi \rightarrow \tau$ (using purely syntactic methods—do not make reference to structures or truth).

[10] Of course this is not an argument that conscious awareness or the soul do not exist; rather it rests on the fact that, even if they did, they could hardly be expected to explain how the brain can exhibit intelligent behaviour.

Exercise 2.10.3: In what respect does '$\varphi \vdash \psi$' capture the provability of ψ on the basis of φ, while '$\models$' captures the relation of φ making ψ true? Do these relations correspond to separate concepts?

Exercise 2.10.4: In the following, appeal only to syntactic considerations:

1. Prove that, in the propositional calculus, the order in which one applies the eligible rules to the wffs on a branch doesn't affect whether the resulting tableau closes.
2. Prove that $\Gamma \cup \{\varphi \vee \psi\} \vdash$ iff $\Gamma \cup \{\varphi\} \vdash$ and $\Gamma \cup \{\psi\} \vdash$.

2.11 Soundness

We begin by proving soundness of the tableaux system: that is, every syntactic theorem is a semantic theorem. We start with an intermediate lemma. The proof here relies on mathematical induction; see Appendix A (page 147). You might wish to contrast the proof of the same theorem (25.10) in Hodges (2001, 118–9), or that given in Bostock (1997, §4.5).

LEMMA 1. *If there is a valuation function such that $v_{\mathscr{S}}(\neg\varphi) = \top$, there is some branch B on a tableau generated by $\{\neg\varphi\}$ such that for all wffs $b \in B$, $v_{\mathscr{S}}(b) = \top$.*

Proof. We prove the lemma by *induction on the length of tableaux branch B.*
Base: consider the smallest tableaux T generated by $\{\neg\varphi\}$: the single node $\langle\neg\varphi\rangle$. Since this is the only member of the only branch on T, and it is assigned $\top$ by $v_{\mathscr{S}}$, the lemma holds in this case.
Induction: Assume that the lemma holds of a branch B on tableau T_n. Then we show the lemma holds of a branch B^+ on a tableau T_{n+1} obtained from T_n by one additional application of a rule in Fig. 2.4 to a wff in B on T_n. There are three cases.

1. We apply the Double Negation Rule (Fig. 2.4(i)). Then some wff on B is of the form $\neg\neg\chi$, and we add χ to the bottom of the branch to get B^+. Since $v_{\mathscr{S}}$ is a Boolean valuation , $v_{\mathscr{S}}(\chi) = v_{\mathscr{S}}(\neg\neg\chi) = \top$ (see Table 2.14), as required by the lemma.

2. We apply a non-branching rule to some wff on B. Then we add two wffs to the bottom of B. If we applied, for example, the Negated Conditional rule (Fig. 2.4(f)) to $\neg(\pi \rightarrow \tau)$, we added π and $\neg\tau$ to the bottom of B to get B^+. By the rules for Boolean valuation (Table

2.14), $v_{\mathscr{S}}(\neg(\pi \rightarrow \tau)) = \top$ iff $v_{\mathscr{S}}(\pi \rightarrow \tau) = \bot$ iff $v_{\mathscr{S}}(\pi) = \top$ and $v_{\mathscr{S}}(\tau) = \bot$ iff $v_{\mathscr{S}}(\pi) = \top$ and $v_{\mathscr{S}}(\neg\tau) = \top$, as required. Similarly for the Conjunction rule: $v_{\mathscr{S}}(\pi \wedge \tau) = \top$ iff $v_{\mathscr{S}}(\pi) = \top$ and $v_{\mathscr{S}}(\tau) = \top$. The Negated Disjunction rule is left as an exercise.

3. We apply a branching rule to some wff on B. We then get two possible branches, B_1^+ and B_2^+, either of which can satisfy the lemma. For instance, if we apply Disjunction (Fig. 2.4(c)) to $\pi \vee \tau$, $B_1^+ = B \cup \{\pi\}$ and $B_2^+ = B \cup \{\tau\}$. Since by the rules for Boolean valuations (Table 2.14) $v_{\mathscr{S}}(\pi \vee \tau) = \top$ if either $v_{\mathscr{S}}(\pi) = \top$ or $v_{\mathscr{S}}(\tau) = \top$; that is, the lemma holds of at least one of B_1^+ or B_2^+. Similarly for the Negated Conjunction rule: If $\neg(\pi \wedge \tau) \in B$, then $\neg\pi \in B_1^+$ and $\neg\tau \in B_2^+$. $v_{\mathscr{S}}(\neg(\pi \wedge \tau)) = \top$ iff $v_{\mathscr{S}}(\pi \wedge \tau) = \bot$ iff $v_{\mathscr{S}}(\pi) = \bot$ or $v_{\mathscr{S}}(\tau) = \bot$ iff $v_{\mathscr{S}}(\neg\pi) = \top$ or $v_{\mathscr{S}}(\neg\tau) = \top$. The Conditional rule is left as an exercise.

That completes the induction. ■

Now we are in a position to prove soundness.

Theorem 6 (Soundness). *If* $\vdash \varphi$ *then* $\vDash \varphi$.

Proof. Assume that $\vdash \varphi$. Then every complete tableau generated by $\{\neg\varphi\}$ is closed.

If it is not the case that $\vDash \varphi$, then there is a valuation function $v_{\mathscr{S}}$ (defined on structure $\mathscr{S}$) that makes $\neg\varphi$ true. By Lemma 1, there is a tableau T generated by $\neg\varphi$, with a branch B such that for every $b \in B$, $v_{\mathscr{S}}(b) = \top$. Since $v_{\mathscr{S}}$ is a Boolean valuation, if $\psi \in B$, then $\neg\psi \notin B$ (else a wff on B would receive $\bot$ from $v_{\mathscr{S}}$). Hence B cannot be closed; hence T is not closed. But T is generated by $\{\neg\varphi\}$, and must be closed by our initial assumption. So our hypothesis that it was not the case that $\vDash \varphi$ must be wrong; that is, it must be that $\vDash \varphi$. ■

Having proved the theorem, we can easily extend it to the general case of proving that every correct syntactic sequent is a correct semantic sequent. We do this quite easily: take the syntactic sequent, apply the syntactic deduction theorem (Theorem 5), apply the soundness theorem, apply the deduction theorem in reverse (Theorem 2), and we have a correct semantic sequent.

Exercises for §2.11

Exercise 2.11.1: Prove the induction step of the proof for Lemma 1 in the cases where the branch B is extended by (i) the negated disjunction rule; and (ii) the conditional rule.

Exercise 2.11.2: Assuming the soundness theorem, prove that if $\Xi \models \varphi$ is an *incorrect* sequent then $\Xi \vdash \varphi$ is also incorrect.

Exercise 2.11.3: Recall the tableau rules involving the $\odot$ operator from exercise 2.8.4. Are those tableau rules sound?

2.12 Completeness

We now prove the converse of Theorem 6, that is, that every semantic theorem can be proved using tableaux. We begin with a preliminary lemma. The proof here relies on mathematical induction; see Appendix A (page 147). You might wish to contrast the proof of the same theorem (25.11) in Hodges (2001, 119–20), or that given in Bostock (1997, §4.6).

LEMMA 2 (HINTIKKA'S LEMMA). *If B is an open branch on a complete open tableau, then there is a Boolean valuation v_B such that for every wff ψ on B, $v_B(\psi) = \top$.*

Proof. Let v_B be the Boolean valuation (Table 2.14) induced by B, that is, if a basic wff p appears on B, then let $v_B(p) = \top$, and if a negated basic wff $\neg q$ appears on B, let $v_B(q) = \bot$. If r appears as part of a complex formula in B, but does not appear as a literal on B, let $v_B(r) = \top$.

We show by induction on the complexity of ψ that for every wff ψ on B, $v_B(\psi) = \top$.

Base case: ψ contains no binary connectives and at most one connective; then it is either a sentence letter, s, or a negated sentence letter. If s appears on B, $v_B(s) = \top$. If $\neg s$ appears on B, $v_B(\neg s) = \top$, as required.

Induction step: ψ is a complex sentence, and the lemma holds of its less complex parts χ. There are three cases of interest: ψ is a double negation; a branch rule can be applied to ψ; or a list rule can be applied to ψ.

1. $\psi = \neg\neg\chi$. Then χ appears on B, by completeness and openness; and by hypothesis $v_B(\chi) = \top$. But then $v_B(\neg\chi) = \bot$; then $v_B(\neg\neg\chi) = \top$, as required.

2. ψ can have a list rule applied to it. Let us choose Negated Conditional, so $\psi = \neg(\chi \to \pi)$. Then χ and $\neg\pi$ appear on B, and $v_B(\chi) = v_B(\neg\pi) =$

$\top$; therefore by Table 2.14 $v_B(\chi \rightarrow \pi) = \bot$, and $v_B(\neg(\chi \rightarrow \pi)) = \top$, as required. Exactly similar reasoning applies to the other list rules.

3. ψ can have a branch rule applied to it. Let $\psi = \chi \vee \pi$. Then either χ or π appears on B; hence either $v_B(\chi) = \top$ or $v_B(\pi) = \top$, which by Table 2.14 means that $v_B(\chi \vee \pi) = \top$, as required. Exactly analogous reasoning holds for other branch rules.

That suffices to show the lemma. ■

Theorem 7 (Completeness). *If* $\vDash \varphi$ *then* $\vdash \varphi$.

Proof. We prove the contrapositive, namely, that if it is *not* the case that φ is a syntactic theorem (which we write '$\nvdash \varphi$') then it is not the case that φ is a semantic theorem ('$\nvDash \varphi$').

Assume that $\nvdash \varphi$. Then there is a complete open tableau generated by $\neg\varphi$ which has an open branch, B. Let v_B be the valuation induced by B. By Lemma 2, all the wffs on B are evaluated $\top$ by v_B. $\neg\varphi$ appears on B, as the root. Therefore $v_B(\neg\varphi) = \top$, hence $\nvDash \varphi$, since there is at least one possible situation or valuation (namely, v_B) which makes φ false, so it is not a semantic theorem. ■

Again, having proved the theorem, we can easily extend it to the general case of showing that every correct semantic sequent is provable by a correct syntactic sequent. We do this quite easily: take the semantic sequent, apply the deduction theorem (Theorem 2), apply the completeness theorem, apply the syntactic deduction theorem in the right-to-left direction (Theorem 5), and we have a correct semantic sequent.

$\vDash$ is equivalent to $\vdash$ Soundness and completeness together tell us that $\Xi \vDash \varphi$ iff $\Xi \vdash \varphi$, and hence that our system of proof, defined without reference to truth is perfectly the same as the results one gets when thinking about truth in a structure, not proof. It also means that *every acceptable manipulation on semantic sequents holds too of syntactic sequents* (§2.9.1). It also shows us that our mechanical method of proof, using the efficient technique of tableaux, is exactly as reliable as the more laborious detour through valuations or structures (truth tables included).

Exercises for §2.12

Exercise 2.12.1: Prove the induction step for Hintikka's Lemma for (i) the negated conditional rule; and (ii) the negated conjunction rule.

Exercise 2.12.2: Can a system be complete but not sound? If so, give an example. If not, why not?

Exercise 2.12.3: Is it surprising that weakening, permutation, and contraction hold for $\vdash$. Can we show these directly, without using the completeness and soundness theorems?

Exercise 2.12.4: Consider the system we obtain if we add the following rule to the acceptable rules in Fig. 2.4: In any tableau T, if B is not closed, we may add any wff of the form '$(\varphi \vee \neg\varphi)$' at the bottom of B. Assuming that the rules in Fig. 2.4 are sound and complete, prove that the new system is sound and complete.

2.13 Further Metalogical Results: Compactness and Decidability

Having proved the important results of soundness and completeness, we now show some other interesting facts about our logical system of syntactic tableaux and Boolean valuations.

2.13.1 Compactness

We begin by proving a useful intermediate result.

LEMMA 3 (KÖNIG'S LEMMA). *If a tableau has infinitely many wffs appearing on it, then it has an infinite branch.*

Proof. Call a node 'good' if it has infinitely many descendents; obviously if a tableau has a branch which only contains good wffs, that branch is infinitely long. We prove by induction on rank in the tableau T that an infinite tableau has such a good branch. *Basis*: If T has infinitely many wffs on it, then the topmost wff x_1 of T is good. *Induction*: The tableau rules ensure that any node x_i on T has either one or two immediate successors. Assume that x_i is good; we show that at least one immediate successor x_{i+1} of x_i is also good. If we applied a list rule to x_i, and x_i is good, it is obvious that any immediate successor of x_i is good. If we applied a branch rule to x_i, then if both of x_i's immediate successors were not good (with only finitely many descendents), then x_i would not be good; so at least one of x_i's immediate successors x_{i+1} must be good. ■

We may now prove the following useful theorem:

Theorem 8 (Compactness). *For any countably infinite set of formulae* Ξ, $\Xi \models$ *iff there is a finite subset* $\Theta \subset \Xi$ *such that* $\Theta \models$.

Proof. The 'if' direction is obvious, using the structural rules.

The 'only if' direction is more difficult; we'll prove the contrapositive. Assume that every finite subset of Ξ is consistent. Ξ has countably infinitely many wffs. Such sets of wffs may be ordered by the natural numbers, so $\Xi = \{\xi_1, \ldots, \xi_n, \ldots\}$. We now inductively construct a tableau T for Ξ, as follows.

Base: begin by constructing a tableau for ξ_1. Since $\{\xi_i\}$ is a finite subset of Ξ, it is consistent, so the tableau at stage 1 has an open branch.

Induction: if the T has an open branch at stage i, then we add ξ_{i+1} to the *top* of the tableau, and add the consequences of ξ_{i+1} to the bottom of open branches on T; as the set $\{\xi_1, \ldots, \xi_i, \xi_{i+1}\}$ is a finite subset of Ξ, it is consistent, and so T at stage $i + 1$ has an open branch.

At the (infinite) end of this construction, we have a tableau T which is generated by the set $\{\ldots, \xi_2, \xi_i\} = \Xi$, and so has infinitely many nodes; by the construction and König's Lemma (Lemma 3), T has an infinite open branch B such that $\Xi \subset B$. By Hintikka's Lemma (Lemma 2), there is a valuation v_B such that for all $\varphi \in B$, $v_B(\varphi) = \top$; therefore for each ξ_i, $v_B(\xi_i) = \top$; hence Ξ is consistent. ■

Consequences of compactness Since entailment and inconsistency are interdefinable, we also have obviously:

Corollary 3 (Compactness for Entailment). If $\Xi \models \varphi$, then there is some finite subset $\Theta \subset \Xi$ such that $\Theta \models \varphi$.

This also shows that our infinite tableaux are not strictly speaking necessary; for any inconsistent set there is a finite tableau which shows it inconsistent. You may wish to consider the alternative proof-sketch of the theorem presented in Bostock (1997, 173–4), that does not rely on the completeness theorem (note our use of Hintikka's Lemma).

2.13.2 Decidability

Decidability revisited A further consequence of these results is that the tableaux method is decidable. Recall that a logic is decidable if there is an automatic test for whether an argument is valid, that will tell us with certainty either that the argument is valid or that it is invalid (page 49—see also Bostock (1997, 184–7)).

Effective procedures An *effective procedure* for determining a fact is one that terminates in a finite time with a 'yes' or 'no' answer concerning that fact just when the fact is true or false, respectively (Boolos *et al.*, 2003).

THEOREM 9 (DECIDABILITY OF FINITE CONSISTENCY IN $\mathscr{L}$). *For finite* Ξ, $\Xi \vdash$ *can be established by an* effective procedure.

Proof. We begin the proof by observing that all the tableaux rules pictured in Fig. 2.4 break a complex formula into less complex sub-formula, and hence that the successive application of tableaux rules will eventually terminate in basic wffs.

If Ξ is finite, there is a complete tableau generated by Ξ that can be generated by finitely many applications of the tableau rules to the members of Ξ; consideration of the rules of tableaux formation tells us that every application of such a rule yields less complex wffs that the wff the rule operates on; so the tableaux method yields, in a finite number of steps, wffs to which no rules apply. Moreover, this tableau has finitely many nodes, and hence finitely many branches.

If Ξ is syntactically inconsistent ($\Xi \vdash$), then no branch is open; but we need only check, for each of finitely many branches, whether two nodes of contradictory form appear on that finitely long branch; this is clearly performable in finite time. If Ξ is syntactically consistent, then at least one such branch is open, and precisely the same check for inconsistency will determine this too. In either case, we have a finitely performable procedure for determining whether Ξ is consistent or inconsistent. ■

In the case of infinite sets Ξ, things aren't quite so easy.

THEOREM 10 (POSITIVE DECIDABILITY OF CONSISTENCY FOR ARBITRARY WFFS OF $\mathscr{L}$). *If* Ξ *is inconsistent, that is decidable; but there is no effective procedure that will decide if arbitrary* Ξ *is consistent.*

Proof. Let $\Xi = \{\xi_1, \ldots, \xi_n, \ldots\}$, so that each ξ_i is assigned some natural number as an index. If $\Xi \vdash$, then by compactness (Theorem 8) there is a finite subset $\Theta \subseteq \Xi$ such that $\Theta \vdash$. Moreover, it is clear that we can effectively generate finite sets of natural numbers.[11] So each finite subset Θ

[11] *Proof sketch*: Begin by announcing 1. Then proceed inductively: if n is the highest number announced so far, list every set which contains only numbers $\leqslant n$. Since at any point we will only have announced finitely many numbers, there are only finitely many such sets, so this is finitely achievable. Once this is complete, proceed to announce $n + 1$. It's clear that any finite set of natural numbers will be generated at some finite time from the beginning of this series—since each such set has a largest member n, it will be listed before $n + 1$ is announced.

can be effectively generated by generating the indices on the ξ_i, and since (by Theorem 9) finite consistency is decidable, if Θ is inconsistent that will be decided at some finite point. So if Ξ is inconsistent, this test will indicate that some finite subset is inconsistent after some finite time.

However, if Ξ is consistent, no finite subset is inconsistent, and this procedure will continue indefinitely checking consistent sets and showing that they are consistent. It never 'halts' after some finite number of Θ's have been shown consistent, because there are still infinitely many other finite subsets of Ξ that have not yet been checked. So even if Ξ is consistent, this cannot be effectively determined. ■

Exercises for §2.13

Exercise 2.13.1: Prove the 'if' direction of the Compactness Theorem—that is, show that if $\Theta \subset \Xi$ is inconsistent, then Ξ is inconsistent.

Exercise 2.13.2: Can you think of a way of having infinitely many nodes on a tree without having an infinite branch?

Exercise 2.13.3: Show that the decidability of consistency entails the decidability of entailment.

Exercise 2.13.4: Assume, as in Exercise 2.9.4, that $\models$ holds between sets of formulae. Show that if $\Gamma \models \Delta$, then there is a finite conjunction $\Phi = (\varphi_1 \wedge (\varphi_2 \wedge (\ldots \wedge \varphi_n) \ldots)$, such that $\varphi_i \in \Gamma$, and a finite disjunction $\Psi = (\psi_1 \vee (\psi_2 \vee (\ldots \vee \psi_m) \ldots)$, such that $\psi_j \in \Delta$, such that $\Phi \models \Psi$. You may assume compactness.

Chapter 3

Predicate Logic

3.1 Valid Arguments Not Captured by $\mathscr{L}$

3.1.1 Objects and Names

Consider the following argument:

I studied under the finest philosopher of the post-war period;
David Lewis was the finest philosopher of the post-war period;
Therefore, I studied under David Lewis.

$\mathscr{L}$ is unable to capture this argument This argument is obviously valid, but we cannot capture its validity in $\mathscr{L}$. This is because this argument involves premises that have no truth-functional parts, that is, there are no constituents of any of the sentences in this argument that are whole sentences. Therefore, when formalising this argument, the best we could do would be '*p*; *q*; therefore *r*', and that is not a valid argument schema.

We could put the point another way: the possible situations we use in constructing valuations of wffs of $\mathscr{L}$ only assign truth values to basic wffs, independently of the truth values assigned to other basic wffs. All the sentences in this argument are basic wffs, but they are (intuitively) *not* independent. Hence the possible situations we are using are too coarse to capture this argument.

Validity Why, if $\mathscr{L}$ is not able to formalise it, is this argument valid? A natural thought is that the Soundness and Completeness theorems (Theorems 6 and 7) showed that the tableaux system based on $\mathscr{L}$ proved every tautology about arguments; how can there be a valid argument that escapes this net? This is because the soundness and completeness theorems prove that the tableaux system we defined is adequate to capture the possible structures we were concerned with; but these sentences call for, indeed demand, a new kind of structure. The definition of validity, on the other hand, need not be interpreted as referring just to a particular kind of structure. We still say that this argument is valid because the premises cannot be true without the conclusion also being true. If we are to capture this, however, we need different structures for the premises and conclusion to be true within.

Designators and Validity The reason this argument is valid seems to turn on the fact that the first premise is true, and that the name 'David Lewis' refers to the thing that makes the first premise true. The validity of this argument, then, depends on particular properties of particular individuals, or objects, and how we can either (i) name or (ii) describe them. This argument, in particular, turns on our ability to identify the same thing by a name and a description, and realise that we are picking out the same thing by those two means. It turns, that is, on our being able to *designate* the same item by two means, and the argument is valid because the person I studied under has two distinct designators. We shall go into more detail later—see §3.2. We shall see how designators work, so that we can augment our language and provide resources for capturing the validity of the foregoing argument.

3.1.2 Predicates

$$\frac{\text{This rose is scarlet}}{\text{Therefore, this rose is red}}$$

$\mathscr{L}$ is unable to adequately capture the sentences involved in the above argument: at best it takes the form 'p, therefore q'.

Predication The argument is valid because there seems to be some connection in meaning between 'is red' and 'is scarlet': namely, that scarlet (the meaning of 'scarlet') is a kind of red (the meaning of 'red'), and hence 'being scarlet' is part of the meaning of 'being red'. The sentences in question attribute the *predicate* 'is scarlet' to a thing named by 'this rose' (a

demonstrative and a common noun), and in virtue of that must also accept that 'roses' also falls under the predicate 'is red'. That is, then, there is no situation in which something can be scarlet without also being a kind of red, which is what we require for validity.

Subject-predicate sentences Predicates, in this context, are those things that attribute a property to an object (the *subject* of the sentence). Hence the most general form of a subject-predicate sentence is $\ulcorner S$ is $P\urcorner$, where $\ulcorner S\urcorner$ stands for a subject designator, and $\ulcorner$is $P\urcorner$ stands for a predicate, where the 'is' here does not mean 'is identical with', but rather 'is an example of' or 'has the property'. We shall return to predicates below, in §3.3; as it will be seen that they naturally 'go together' with designators to make declarative sentences, which are the fundamental parts of any logic.

Satisfaction The predicate 'is red' is said to be *satisfied* by the designator 'this rose', because the resulting sentence is true. We also sometimes talk of an object 'falling under' a predicate when it satisfies that predicate.

Properties and Predicates Properties, like redness, are the meaning of predicates, like 'is red'. But predicates are parts of language, whereas properties are supposed to be parts of the physical world, or the object which possess the property. This is the same distinction as must be made between names and the objects which they name. Perhaps, even, one might think there are not enough physical properties for all the meaningful predicates: for example, 'is taller than Antony' is a predicate, but is there really a property which only those things taller than me posses, in virtue of which they are taller than I am? Or is it just that they have the property of being h cm tall, where $h > 183$, and their height is what makes that predicate apply to them? We shall not go too much into such questions here.

3.1.3 Quantification

All men are mortal
Some men are unhappy

Therefore, some mortals are unhappy.

The above argument has no correct analysis into $\mathscr{L}$ that captures its validity, though it certainly is valid.

Designators again? Is this argument another instance of the same problem we saw with the argument in §3.1.1? That is, are 'all men' and 'some mortals' designators? It seems not; examine the following argument:

All men are mortal
Steve is a man

Therefore, Steve is mortal.

If 'all men' was a name, then this argument would NOT be valid—since 'all men' occurs only once, the argument would have the form 'S_1 is P_1; S_2 is P_2; therefore S_2 is P_1', and that is easily counterexampled. But, on the contrary, the argument is valid, because the predicate 'is a man' has some connection with the phrase 'all men' in the first premise, which cannot therefore be a pure designator. A similar error goes on if we accept the following flawed argument as valid, as some ancients may have done:

Nothing exists

Therefore, at least one thing (viz., nothing itself) exists.

Sets of objects that satisfy predicates Logicians analyse such phrases as 'all men' and 'some mortals' as talking about sets of entities which satisfy the predicates 'is a man' and 'is a mortal'. Respectively, they say that, everything which is a man has some predicate apply to them; or some, perhaps all, of the things which are mortal have some predicate apply to them. The two arguments we have considered are valid, because in both cases there are relations between the sets of objects which satisfy the predicates. So, in the first argument, the set of unhappy things has an overlap, or intersection, with the set of men, which in turn is completely included in the set of mortal things, hence there is some intersection between the set of unhappy things and the set of mortal things.

In the second argument, we know that the set of mortal things includes the set of men; we are told that 'Steve' names a member of this second set, and hence we know that he is also a member of the set of mortal things.

Quantifiers 'All' and 'some', as well as 'every', 'none', 'half of' and so on are called *quantifiers*: in some sense, they quantify how many, or what proportion, of some set of objects that satisfy one predicate also satisfy another, e.g. 'everything that is a P_1 is also a P_2' or 'Some things that are P_3 are not P_4'; or 'Half of the P_5s are P_6s'. Quantifiers, then, act on predicates to produce quantifier phrases. We shall see more concerning them in §3.5.

Numerical Quantification A special kind of quantification is exemplified in the following valid argument:

There are four things in the cupboard

Therefore, there are more than three things in the cupboard.

Such an argument depends on the fact that $4 > 3$; this can be expressed using quantification and identity as follows: 'The set of items in the cupboard has four non-identical members; therefore, the set of items in the cupboard has at least three non-identical members'. We shall see how to express such sentences using quantification and identity in §3.5.6.

3.1.4 Plan of what is to come

Having now catalogued a number of failures of $\mathscr{L}$, we now attempt to remedy them. We shall use the various kinds of arguments we have looked at to give some kind of taxonomy of the linguistic entities we shall want our augmented logic to possess: in particular, our language should have designators, predicates to combine with designators to make sentences, and some kind of quantificational resources. After looking at how such things appear in natural languages, in particular English, we shall try to construct a formal (purely syntactic) language, a successor to $\mathscr{L}$, which we shall call, somewhat uncreatively, $\mathscr{L}_2$, and we shall give a tableaux system for doing proofs in $\mathscr{L}_2$. Then we shall give a semantics for this language: a description of which things can count as the possible situations that make sentences of $\mathscr{L}_2$ true or false. Then we shall look, again, at the relation between our method of proof, and our account of truth in $\mathscr{L}_2$ structures. The relation, will, unfortunately, not be quite as clear as in the case of $\mathscr{L}$.

3.2 Designators

Designators, as their name suggests, serve to pick out, or designate, things or objects. We will consider four main kinds of designators (Hodges, 2001, 121–4):

1. Names
2. Non-count nouns
3. Singular personal pronouns

4. Definite descriptions.

3.2.1 Names

A name, like ‘David Lewis’, serves to pick out a particular object by labelling it. We attach names by convention or by chance—David Lewis was not called ‘David Lewis’ because it was particularly appropriate for him to be so-called, but just because it was a convention that one chooses, arbitrarily, or because one likes the forename, and the surname was that of his parents.

Direct reference A proper name, then, is a label for an object. A widely, but not universally, held thesis is that names *directly refer*. This means that the meaning that a proper name contributes to a sentence in which it occurs is just the thing to which it refers: the content of a proper name is just the object it designates. This means that names are radically *unlike* descriptions (see §3.2.4), which have varying content from one possible situation to another. (This is because descriptions are more like a guide for finding a referent, and in different circumstances the same guide can lead to different things.) And it means that names do not have ‘meanings’ that help one decide what the name attaches to—whatever meaning they have is exhausted by the item they refer to. This is often connected to the thesis that names are *rigid designators*—that names attach to the same object in all possible situations. So that, for instance, the name ‘Aristotle’ actually attaches to the greatest philosopher of antiquity, but even in a situation where Aristotle was not the greatest philosopher of antiquity, the name ‘Aristotle’ still attaches to him. This issue is discussed at great length in a good and very influential book by Saul Kripke (Kripke, 1980), but we shall do no more than mention it here.

3.2.2 Non-count nouns

According to Hodges (2001, 122), a *non-count noun* is a word that substitutes into the schema ‘I want some φ’ grammatically, but forms an ungrammatical sentence when substituted into ‘I want a(n) φ’. So ‘butter’, ‘politics’, ‘dirt’, ‘perceptiveness’, *&c.*, are all non-count nouns, whereas ‘elephant’, ‘tutor’, *&c.*, are *count nouns*, which we deal with only as part of quantifier phrases (§3.5).

Mass nouns A particularly interesting subclass of non-count nouns are so-called *mass* nouns: like ‘bronze’, or ‘dirt’, where what is denoted is a certain

kind of 'stuff', rather than a certain kind of thing or item. Many logicians and philosophers have been interested in mass terms because they seem to have some intimate connection with how things can be made up out of, or constituted by, stuff: how a field can be constituted by the dirt that fills it, or how a sculpture may be made up out of some bronze.

Abstract nouns Another interesting class of non-count nouns are the *abstract* nouns: 'intelligence', 'perspicuity', and so on. These, again, don't seem to be 'things' in the sense of objects one can separate and identify separately, but we can nevertheless refer to them. They seem to be more like qualities that an object can possess, but that we can nevertheless identify as objects rather than mere properties. They sometimes refer to *abstract objects*, but not all abstract objects are named by an abstract noun: consider the object *the number three (3)*, which is an abstract (i.e. not physical or concrete) object, but it is referred to by the proper name 'Three'.

Cautions We need to be cautious here to avoid being misled. There are a couple of instances of nouns that seem to substitute into Hodges' schema without much difficulty, but which are difficult. (i) Consider first 'Ribena' or 'Coke': it is easy enough to see that these are mass terms, but they are also proper names of the characteristic products for these mass substances. What this shows is that our classification is not mutually exclusive: a term like 'Coke' is both a mass noun *and* a proper noun. Nothing need be difficult about this issue. Perhaps the same occurs, in a more complicated way, with a term like 'philosophy': both 'I want some philosophy' and 'I want a philosophy' seem grammatically acceptable and 'philosophy' appears to be used in much the same way in each. (ii) The other class is more difficult. Take a word like 'tigers'; if we are unspecific as to number, we can (say when ordering supplies for our zoo) correctly say 'I want some tigers'. But we should not thereby think that 'tigers' is a non-count noun, as obviously 'I want seven tigers' is perfectly grammatical. In general, *plurals* are out of the scope of this classification of types of designators, for reasons which can only become clear when we turn to quantification: that will allow us to represent plural nouns in a logically perspicuous fashion that is nevertheless at variance from the surface grammar of English.

3.2.3 Singular personal pronouns

These are words like ‘I’, ‘him’, ‘it’. They serve to designate particular individuals on a particular context of utterance. They are not proper names, since they can name different individuals on different occasions (i.e. in different contexts), and so are not directly referential and certainly aren’t rigid designators. Nor are they descriptions: if, for example, ‘I’ was synonymous with the description ‘the speaker of this sentence’, then I could truly say ‘I might have been different from Antony’ (since that sentence might have been uttered by someone else). Yet we do not think that sentence does possibly express a truth: there is no situation in which I am different from myself, so the description account of ‘I’ is incorrect. These designators instead function more like flexible ‘arrows’ that indicate certain contextually salient individuals when the sentence is uttered: what they happen to be used to refer to.

The word ‘I’, for example, always denotes the speaker of the sentence in which the word occurs (setting aside the use of ‘I’ is quotation of reported speech); it thus has a constant meaning in some sense, though who it refers to on each occasion of use varies. As such, ‘I’ is a context-sensitive expression: ‘I am Antony Eagle’ is true when uttered by me (expresses a true proposition), while ‘I am Antony Eagle’ (the very same sentence) expresses a false proposition when uttered by someone other than me, because in their case ‘I’ refers to them, not me.

Indexicals Indeed, one might wish to broaden this class slightly, and include other words that are used to pick out various features of the context under which a sentence is to be interpreted, either the context of utterance or the context in which we evaluate the sentence. Words like ‘I’, ‘now’, ‘here’, and ‘actually’ pick out various items in the context of the utterance: respectively, the speaker of the utterance (or writer of the sentence), the time of inscription or utterance, the place of inscription, and (according to many) the possible situation that the speaker of the utterance is located within. We shall not have to deal with many of the peculiar complexities of indexical terms.[1]

[1] You might wish to consider, for instance, the *Answering Machine Paradox*: the sentence ‘I am not here right now, please leave a message’ is, when said by the speaker, false. So why, when you hear it over the phone line, does it express a true proposition to you? After all, ‘now’ isn’t normally taken to indicate the time when you are hearing the sentence in which it occurs (Predelli, 1998).

Demonstratives Though Hodges takes them to be like descriptions, it is arguable that the *demonstrative* terms 'this' and 'that', when used referentially on their own (or perhaps when combined with a gesture towards the item in question) may also be subsumed under indexicals. In the case of 'that', for example, the rule might be: 'that' refers to some contextually salient object on the occasion of utterance or inscription. 'Contextually salient' here is not immediately obvious, as it is in the case of 'pure' indexicals like 'I' and 'now' where the content is immediate from the use. Rather, the referent of a demonstrative like 'that' is usually taken to be fixed either by explicit ostension (pointing or other indicating), or by the speaker's intentions for the term to refer to some particular object. This theory, and the theory of indexicals generally, is due to Kaplan (1989).[2]

3.2.4 Descriptions

Descriptions are a most interesting class of designators. A description refers to, or designates, an object by *describing* it. So, for example, 'The finest philosopher of his generation', 'a man without principles', 'the inventor of bifocals'.

Definite and Indefinite Descriptions Descriptions that begin with 'the' are called *definite*, since there is a particular definite object that they intend to pick out. *Indefinite* descriptions, by contrast, attempt to pick out one of a class of entities that has one or more members: 'a person', for example, picks out some arbitrary member of the set of persons, but without any particular person as the referent of the phrase.

Misfiring descriptions It is important to note that descriptions can misfire in a way that other kinds of designators rarely do. For instance, 'the present king of France' misfires, because there is no such person. The definite description 'the student in my logic class' misfires, because there is more than one such person. The important point for our purposes is that when descriptions misfire like this, the sentences containing those descriptions do not become *nonsense*, but seem rather to be false. For instance, 'The present king of France is bald' is a false sentence, not a nonsensical one; just as 'the student in my logic class is named Carol' is false. Some take the existence

[2]For a useful introduction to indexicals and demonstratives, see Braun (2001).

of misfiring descriptions to provide a different lesson: not that sentences involving misfiring descriptions are false, but rather they are neither true nor false, falling into a truth-value 'gap'. The argument is this: the present kind of France is neither bald nor not-bald; but it is implausible that sentences expressing these claims should both be false, as they seem to be negations of one another. Rather, neither can be true. We shall not do more than mention this objection here (Strawson, 1950), but it gets some support from the idea that definite descriptions play a role in English much like that names play, and the thought that sentences involving so-called 'empty' names (say, defunct scientific terms like 'phlogiston', or names of merely fictional characters, like 'Superman') are neither true nor false.

A bad theory of descriptions Some people, realizing that the sentence 'The present king of France is bald' is meaningful though false, thought that the purportedly referring phrase 'the present king of France' must really refer to something: not something *existing*, mind you, but merely (in their terminology) *subsisting*. One can think of a subsisting object as one that is the target of thought and reference, even though no object that really exists is around to be the subject of that thought or reference. Insofar as Pegasus or the round square can be talked about without existing, they are said to subsist. The most prominent advocate of such a theory was philosopher Alexius Meinong, who thought that all putatively referring phrases did in fact successfully refer to something, though perhaps of a non-existent kind (like 'the round square' which refers to something round and square—and of course nonexistent!). This doctrine seems both grossly implausible and also unhelpful—it is no more helpful to be told than some non-existent thing is bald or not bald than it was to be told that the original sentence was nonsense! That said, the view is easily able to deal with one problem case that other theories of descriptions cannot easily: the fact that it strikes us as true that 'the present king of France does not exist'. Other views must reject the straightforward interpretation of this sentence as having subject-predicate form (giving a more complicated logical analysis of the apparent predicate 'exists'), and this is a cost (though perhaps not one worth avoiding by adopting an ontology of subsisting objects).

Russell's Theory of Descriptions The theory that we shall use to analyse the content of a definite description is due to the philosopher and logician Bertrand Russell (Russell, 1956). This theory analyses a sentence involving a description like 'the quickest animal is the peregrine falcon' as meaning

'there is at least one thing which is the quickest animal; and there is at most one thing which is the quickest animal; and that thing is the peregrine falcon.' When applied to our misfiring descriptions, it is straightforward and easy to see that those sentences turn out false, not meaningless, on Russell's view: for if there are less than one, or more than one, item that satisfy the description, then the definite description fails. An indefinite description, for Russell, should be straightforward: 'a person is an animal' means 'there exists at least one thing such that it is a person, and any such thing is an animal'. (This last sentence, then, means something different to 'some person is an animal'.)

One thing to note is that Russell's theory treats definite descriptions quite differently from other designators. Proper nouns, for example, show up in the underlying logical form of an English sentence as individual referring expressions. But while definite descriptions play a syntactic surface role much like proper nouns, Russell's analysis (unlike say Strawson's) makes the underlying logical form of a sentence involving a definite description quite different: it is a quantifier sentence with no individual constants. Of course, so long as the truth conditions are adequate, we can posit any logical form we wish; but is it really plausible that our competence with English is really founded on such a shifty basis, so that there can be a radical mismatch between the grammatical production of English sentences and their truth conditions?

Potential problem: context According to Russell's analysis, when I say 'The blue car is fast', I make an utterance which means 'there is at least one, and at most one, blue car, and it is fast'. Since there is more than one blue car, this utterance is false. But imagine that I make that remark in my friend's garage, where there is one blue sports car, and one red mini. It seems that, in this context, what I say is quite true, because I mean to express a contrast between the red and blue cars which are relevant, and there is indeed exactly one blue car in the relevant context. The problem here, then, is what it means to say 'there is': where does the context suggest I look to find whatever things there are? In this case, context says I look in front of me. In other cases, such as when I say 'the even prime is a small number', I am contextually allowed to look absolutely anywhere, to find the number 2. This is a very general problem, not specific to descriptions, that might be called the problem of *quantifier domain restriction*. Don't worry about what that means now; we will return to it in §3.5.2.

Other problems are not as easy to dismiss. Consider the sentence 'He

is the proud owner of a Rolls Royce'. Analysed according to Russell's proposal, we would imagine that there is at least one, and at most one, proud owner of a Rolls Royce, and he is it. But that is false, presumably; and restricting the domain seems unhelpful—we can see this by assuming a relevant context in which there are two proud Rolls Royce owners, where we may apparently truly say 'Each is the proud owner of a Rolls Royce', which would be (on Russell's theory) to say that each of them is the contextually unique owner of Rolls Royce, and that would be false, not true as intuition delivers. What we really want to say is that *this* use of the definite description is semantically akin to an *indefinite* description, so that the sentence really means 'He is *a* proud owner of a Rolls Royce'. This seems to be what ordinary speakers take the sentence to mean; nevertheless nothing in the surface form of the natural language sentence gives much indication that this is so.

A deeper worry is produced by sentences like this: 'He is tall, handsome, and the love of my life' (Graff, 2001, 10). In this case, the problem is that there seems no straightforward translation of this sentence into a first order language, basically because of the following phenomenon: Russell's theory of descriptions involves making identity claims between the items picked out by a description, and some other designator in the sentence (in this case the pronoun 'he'). So for that to work, the 'is' that occurs in the sentence must be what is called the 'is' of identity: that which we would formalise as '='. But 'tall' and 'handsome' are predicates; and 'he = Tall' isn't even grammatical, let alone true. So for the predication to work, the 'is' in the sentence must be the 'is' of predication. But there is only one 'is' in the sentence—how can it have these two radically different syntactic and semantic roles? No simple solution to this puzzle presents itself, apart from simply accepting that in this case the logical form of the sentence is quite different from its surface structure.[3]

A bad use of Russell's theory Unfortunately, after putting forward this view of descriptions, Russell made a misstep. He said 'Common words, even proper names, are usually really descriptions' (Russell, 1997, 54). This, I think, is a mistake. 'Aristotle' would still have referred to Aristotle even if 'the most famous student of Plato' had described someone else: this seems to drive a wedge between names and descriptions. We shall here keep Russell's theory of descriptions, and indeed go on to give it some for-

[3]Graff's own solution, following Strawson, is the radically non-Russellian claim that descriptions are in fact predicates, and not referring expressions.

mal content once we figure out how to use quantification (§3.5), but we shall not go with Russell in thinking of all designators as disguised descriptions.

3.2.5 Designators in Fiction

Before moving on, a cautionary note. Consider the proper name 'Pegasus' or the description 'the detective who lived at 221B Baker Street'. If these things refer at all, they refer to merely fictional entities: a flying horse and Sherlock Holmes. These things do not really exist. So what contribution can the proper names 'Pegasus' or 'Sherlock Holmes' make to sentences in which they occur? And how can sentences involving the descriptions 'the detective residing at 221B Baker Street' or 'the winged horse ridden by Bellerophon' be anything other than false, as nothing really exists that satisfies those descriptions (assuming straightforwardly that being a fictional object is a way of not being a really existing object)?

Whatever difficulties may result, it seems clear that *syntactically* there is no difference between empty designators and those which successfully refer; insofar as logic is formal, then, it should not treat empty designators in a special way. This is particularly obvious when we consider that we do not need to learn anything new in order to understand the meaning of sentences written in a fiction written in English: our regular semantic competence suffices to render us competent users of fiction-English. For our purposes, then, we shall treat empty designators as logically on a par with regular designators, and set to one side the vexed issues about empty names (Caplan, 2006). Moreover, in our development of the logic of designators, we shall assume for convenience's sake that a designator has a referent: there are no empty names or unsatisfied descriptions. (Hodges, 2001, §28).

3.2.6 Reference

From what we have said so far, it might seem obvious that the purpose of designators is to pick out certain objects, either simply labelling them with a name, giving a recipe for finding them by a description, or playing a more unusual role. That is, designators refer to objects, where an object need not be a physical entity, but is any item that can bear a property.

Complications So designators refer to things. But the situation is complicated by the existence of pairs of sentences like those in Table 3.1.

Table 3.1 Designators used in opaque and non-opaque contexts.

1. Benjamin Franklin lived in the United States, and John Locke lived in England.
2. I believe that John Locke is the inventor of bifocals.
3. Necessarily, Benjamin Franklin is the inventor of bifocals.

If the only purpose of designators is to refer directly to that which they designate, then we shall have some problems. The first one is okay, since Benjamin Franklin, who is designated by the name 'Benjamin Franklin', did indeed live in the United States; and so too for 'John Locke', which refers to the English philosopher Locke. But Benjamin Franklin was also the inventor of bifocals; so the designator 'the inventor of bifocals' should designate him. So the second sentence in the table should mean the same thing as 'I believe that John Locke is Benjamin Franklin', which I can assure you I do not, since I know that one is English and one American. But I also think that it is possible that Locke *may* have invented bifocals: he was a very clever man. So I do not believe that in every possible situation, Benjamin Franklin invented bifocals; and therefore I do not think that 'Benjamin Franklin' necessarily designates the same thing that 'the inventor of bifocals' designates. But I do think that Benjamin Franklin is Benjamin Franklin, and necessarily so: so sentence (3) cannot mean 'Necessarily, Benjamin Franklin is Benjamin Franklin.

Two kinds of reference The preceding observations lead us to the following observation: sometimes designators are used just to refer to what they actually, or in my belief, or here (consider 'next door'), or now (consider 'the present pope') refer to; and sometimes they refer to what they could have referred to in other circumstances. We make a distinction between two types of contexts in sentences in which designators can occur, corresponding to this difference. That is, sometimes the designator has a meaning apart from just its referent, and sometimes we wish to pay attention to that meaning. For example, when considering 'the inventor of bifocals' in an situation where John Locke satisfies the description, we don't want to be confused and attend instead to the actual inventor of bifocals, Benjamin Franklin. This gives rise to the following definition:

Definition 4 (Opacity). An *opaque context* is a part of a sentence which

contains a designator, such that the truth of the whole sentence is not always preserved by substituting designators which refer to the same thing into that part. Designators in such contexts, which refuse such substitution, are not *purely referential*. Contexts which are not opaque are called *transparent*.[4]

Applying the definition This might seem a little confusing, but there is a simple test. Consider some sentence S = '$\ldots a \ldots$', which contains a designator 'a', and where the ellipses fill in the rest of the sentence. Does S feature an opaque context? We use the following test: let $E = \{e_1, \ldots, e_n\}$ be the set of designators which refer to the same thing as 'a' does. If for any $e_i \in E$, S is true, but S_i = '$\ldots e_i \ldots$' is false (i.e. at least one such substitution fails to preserve truth) then 'a' appears in an opaque context in S. So, for instance 'I believe that David Lewis is the finest philosopher of his generation' is true. Let us say that, unknown to me, 'the finest philosopher of his generation' refers to Saul Kripke, who can be referred to by the designator 'Saul Kripke'. Let us, then, substitute these designators: is it true that 'I believe that David Lewis is Saul Kripke'? No! It is not true, so the *belief* context is an opaque one.

Restriction to extensional logic In this course we mention opaque contexts only to note that the logic we shall develop does not include any, and so no English sentence which does can be completely adequately formalised by our logic. That is no complaint: plenty of English sentences do not include such contexts, and it will be enough for us to develop a logic for part of English before we tackle the whole thing. This connects also with an earlier remark we made (on page 19), concerning non-truth functional connectives, like 'necessarily': that their logic is complicated and non-uniform. Such non-truth functional operators often create opaque contexts, and designators occurring within the scope of such an operator will not be purely referential. Our logic, then, will include no such *intensional* operators or contexts: it will be purely *extensional*, which basically means, designators that refer to the same thing are equivalent (and, indeed, predicates which hold of the same things are equivalent, as are propositions true in the same circumstances).

[4]Hodges (2001, §27) makes a similar but more complicated distinction between 'purely referential' uses of a term and other uses.

Exercises for §3.2

Exercise 3.2.1: What is a rigid designator? Why are descriptions not rigid designators? Can you give an argument that suggests proper names should be thought of as rigid designators?

Exercise 3.2.2: Hodges defines the class of non-count nouns as those grammatically substitutable into the schema 'I want some φ'. Give three examples of a non-count noun. Can you think of any arguably count nouns that fit into this schema? Can you think of any non-count nouns that don't fit into this schema?

Exercise 3.2.3: What is an indexical? Should the word 'actually' be thought of as an indexical, or as a rigid designator of the actual situation?

Exercise 3.2.4: What is an indefinite description? What would the logical analysis of an indefinite description be, according to Russell?

Exercise 3.2.5: Are sentences including misfiring definite descriptions false? What about misfiring indefinite descriptions?

Exercise 3.2.6: What is an opaque context? Give two examples of opaque contexts in ordinary English. What is the test for when a designator occurs in an opaque context? What is the relationship between opacity and intensionality?

3.3 Predicates and Relations

Having discussed designators, it is now time to discuss the other items that combine with designators to make up sentences: *predicates*. Indeed, we simply define an (English) predicate as a complex expression that combines with one or more designators to make a grammatical (English) sentence.

We begin by defining an essential concept (see also Hodges (2001, §30)):

Definition 5 (Satisfaction). If an ordered set of designators $\langle a_1, \ldots, a_n \rangle$ is such that they can be attached—purely referentially—to a predicate 'P^n' so as to yield a sentence '$P^n(a_1, \ldots, a_n)$' that is true in some possible situation, we say that the *objects* referred to by the designators *satisfy* the predicate (again, in that possible situation), or fall under it, or that the predicate applies to the designators.[5]

Some further complications to do with the notion of satisfaction are discussed below, §3.8.2, page 132.

[5] Ordered sets, as well as unordered sets, are defined in Appendix B.

3.3.1 Basic Facts about Predicates and Relations

Adicity Since a predicate can attach to one or more designators to make a grammatical sentence, and typically each predicate needs a precise number of designators to make up a sentence, we can classify predicates by the number of designators they attach to, which is sometimes called the *adicity* of a predicate.[6] We have, then, *monadic* (or unary or one-place) predicates, that make a grammatical sentence when combined with just one designator; we also have *dyadic* (or binary or two-place) predicates, triadic (three-place) predicates, and so on.

Some predicates, like 'practices', seem to be *polyadic*, that is, have more than one adicity. For example, 'David practices', and 'David practices the violin' are both grammatical sentences of English, but in the first, 'practices' is monadic, and in the second, it is dyadic. We introduce the policy that we actually have two predicates here, of differing adicities, but similar sound and similar meaning. It is a good idea to avoid such potential confusions, however, and in $\mathscr{L}_2$ we shall.

Designators within predicates Another potential confusion emerges if we consider the predicate 'practices the violin'. It is a string of words such that if a designator, like 'David', is attached to the front of them, we get an English sentence; in fact, just the one we used as the second sentence in the preceding paragraph. So is this sentence 'really' two designators and a dyadic predicate, or is it 'really' a single designator and a monadic predicate? The only sensible approach, of course, is to say that this sentence has *two* grammatical analyses, and which one is appropriate will depend on context. For instance, in an argument like that in Figure 3.1(a), it is obviously okay to take 'practices the violin' as the predicate. But if the argument were, instead as in Fig. 3.1(b), then it is in fact obligatory to take the dyadic predicate reading. Which might lead one to realise that, while excess detail and flexibility, as in taking this predicate to be dyadic, will never fail to render a valid argument, too little detail, i.e. taking the monadic predicate route, might misclassify an argument. So there is at least a pragmatic constraint in favour of the more detailed dyadic reading.

Predicates and properties revisited There is no need to suppose that, just because some designator satisfies a predicate, that the designated object possesses some property that the predicate has as its meaning. For instance,

[6] Sometimes also known as 'arity'.

Figure 3.1 Arguments that can be given a monadic and a dyadic predicate reading.

David practices the violin Gill practices the violin David and Gill practice the violin	David practices the violin Gill practices the piano Two people practice instruments
(a) Formalised with a monadic predicate.	**(b)** Formalised with a dyadic predicate.

consider the predicate 'is grue', defined as 'is green and first observed before 2008, or blue and first observed in or after 2008'. 'Grass is grue' is true; so is 'Grass is green'. Does grass have the property of being grue in addition to the property of being green (if it even has that)? We need not say so, simply in virtue of satisfying the respective predicates. Of course, on *some* views of properties, there is a genuine 'metaphysical' property for every predicate. On other views, there is not. We shall not decide the metaphysical issue here—but we will suppose the *semantical* thesis that for every predicate, there is something which is the *meaning* of that predicate, and we shall call that a property. So since 'grue' has a meaning, there is a semantic property associated with it; but there need not be any metaphysical property associated with it. From now on the word 'property' will simply mean, 'meaning of a monadic predicate'.

Relations Often, *n*-adic predicates, for $n \geqslant 2$, are called *relations*, or *relational predicates*, for the obvious reason that they serve to indicate a relationship between the 2 or more designators found within them. But, again, do not mistake this for the existence of a genuine *metaphysical* relation between the objects designated. For instance, consider the relation '... and — are both objects'. For instance, me and the number 2^{67} both satisfy this relation. But we need not suppose additionally that there is any real or substantive or genuine relation, like causation or spatial/temporal relations, between the two items. They happen to satisfy the same relation, and that is all. Of course, according to some accounts of relations, there is a genuine metaphysical relation for every relational predicate. On other accounts, there need not be. We shall not decide the metaphysical issue here—but we will suppose the *semantical* thesis that for every relational predicate, there is something which is the *meaning* of that relational predicate, and we shall call that a relation. So since 'are both objects' has a meaning, there is a semantic relation associated with it; but there need not be any metaphysi-

cal relation associated with it. From now on the word 'relation' will simply mean, 'meaning of a relational predicate'.

Order and Relations The items that are related by a relation—i.e. the items referred to by the designators flanking a relational predicate—are called the *relata* of the relation. It is an important, if basic, point that order of the relata matters in relations: the mere fact that Darcy loved Lizzie wasn't enough to secure her affections, which shows that 'Darcy loves Lizzie' and 'Lizzie loves Darcy' express different propositions, and hence that order is important. This is why when we defined satisfaction (Definition 5) we used an ordered set of designators to satisfy a predicate.

Hodges on predicates Hodges (2001, 125–7) does not agree with our definition of a predicate; a *Hodges-predicate* is a predicate with *individual variables* x, y, and so on in the 'gaps' where the designators would go. So 'x practices y' is a dyadic Hodges-predicate. Personally, I'd rather use predicates rather than Hodges-predicates, since because there are no individual variables in natural language English, it turns out that there are predicates, but not Hodges-predicates, in English. But formally there is little substantive difference here, except that you may think it slightly more convenient when forming quantifier sentences from Hodges-predicates.

Using Hodges-predicates also clarifies the difference between certain kinds of ambiguous predicates. For instance, the sentences 'Alan loves Steven' and 'Alan loves himself' both contain the predicate 'loves'; but you may think that the second sentence should use a predicate akin to 'loves himself', i.e. 'x loves x', whereas the first should use 'x loves y'. But a predicate like 'loves himself' is not really a two place predicate, if the only option is to fill in the same designator at both apparent places. So I do not think this distinction is of any importance.

Zero-place predicates Sometimes it will be convenient to suppose that declarative sentences are themselves predicates: predicates which need *zero* additional designators added to make them declarative sentences. Do not be alarmed at this: it is a mere convenience. In line with the remarks we have been making above, it most certainly *does not* entail that propositions *are* properties (though that is a view some have held).

3.3.2 Identity

One notable relation is that of *identity*, which we symbolise '=' (Hodges, 2001, §29), and read '— is identical to …'. Quite obviously, a sentence containing two designators which flank the 'is identical to', is true just when those two designators refer to the same thing.

Self-identity or '='? It is obvious, and undeniable, that everything is identical to itself: that is, that every object bears the genuine relation of 'being identical with' to itself. This is a trivial 'discovery' about objects and their properties. The relational predicate of identity, by contrast, holds between designators when they designate the same thing, which is rather more difficult to find out. For instance, Superman is self-identical. But even if 'Superman' and 'Clark Kent' designate the same object, it is much harder for people to figure out whether 'Superman = Clark Kent' is true: witness Lois Lane.

Law of Identity What is equally undeniable is that the same designator, in the same context, refers to the same thing. This gives rise to the *law of identity*: for any designator d, the following is true in every possible situation:

(3.1) $$d = d.$$

From this, it follows that the sentence 'not ($d = d$)' (also written '$d \neq d$') is false in every possible situation, and hence any set of sentences which contains the latter is inconsistent. Our assumption that every designator is presupposed to have a reference (page 87) means that this law isn't falsified by the case where d is empty (doesn't refer), because there is no such case in our logic—though there is in English.

Leibniz' law Another law is simply derived from this first one. If two designators refer to the same thing, that is, if '$c = d$' is true, then obviously they each refer to something which has exactly the same properties as the thing which the other designator refers to, since (of course) it is the same thing. Thus, of course, those two designators will satisfy exactly the same predicates. It was this fact that led us to judge the argument given in §3.1.1 as valid: for it was of the form '$c = d$'; 'c is P'; therefore 'd is P'. *Leibniz' law* says that any argument of this form is valid. Another name for this law is the principle of the *indiscernability of identicals*: that is, there is

no predicate which is satisfied to one designator and not to another if those two designators refer to the same thing. But we must remember our restriction to purely extensional languages (page 89): since if predicates produce opaque contexts, the designators do not necessarily have the same primary reference, even if the things referred to are (actually) identical. Consider again Clark Kent and Superman: even if 'Clark Kent = Superman' is actually true, it does not mean that 'Lois Lane believes that Clark Kent = Superman' follows from the truth that 'Lois Lane believes that Clark Kent = Clark Kent'. This is another reason why we restrict our attention to extensional languages, with extensional predicates that respect Leibniz' law and the intersubstitutability of co-referring designators.

Exercises for §3.3

Exercise 3.3.1: What does it mean for an object to satisfy an 3-adic relational predicate? Give an example of a 3-adic relational predicate, and name some objects which satisfy that predicate, and also some which fail to satisfy that predicate. Why is it important that we define satisfaction in terms of ordered sets of objects?

Exercise 3.3.2: What is the difference between a predicate and a property? What relevance does this distinction have for the following claim: 'identity sentences must be trivial, since self-identity is a property that everything has and nothing lacks, trivially.'

Exercise 3.3.3: What is Leibniz' law? Is the converse of Leibniz' law (the thesis that indiscernibles are identical) plausible?

3.4 More on Relations

Read Appendix B before reading this section.

3.4.1 Meanings and Predicates

We are here discussing relations, not relational predicates. Of course it will turn out that a relation is the meaning of a relational predicate, but since the facts we discuss would be true even if there were no language with relational predicates, these facts must be independent of language.

Sets and Predicates We have stipulated above that we are only to consider extensional predicates. Therefore, every predicate is associated with a set of things which satisfy that predicate, regardless of how we designate those

things. Indeed, every predicate P has, in a given situation, precisely one corresponding set $S_P = \{x : P(x)\}$.

Domains and Individuals Consider the set $S_= = \{x : x = x\}$. In a given situation, the identity predicate applies to everything: nothing is not self-identical, so nothing that is present in that possible situation gets left out. In a situation $\mathscr{S}$, the objects that satisfy the predicate $x = x$ will be called the *domain* of $\mathscr{S}$, which we write $\mathbf{Dom}_{\mathscr{S}}$. The members of the domain we shall call *individuals*.

Predicates and Properties In some situation $\mathscr{S}$, with domain $\mathbf{Dom}_{\mathscr{S}}$, some predicate P will be satisfied by a set of objects S_P. Since the domain includes everything which is in $\mathscr{S}$, it is clear that $S_P \subseteq \mathbf{Dom}_{\mathscr{S}}$. We shall say that a set of individuals in a domain, like S_P, is a *property*, i.e. the meaning of that predicate, in that situation. So we can specify a property X in either of two ways: (i) by listing the members: $X = \{x_1, \ldots, x_n\}$, or (ii) by giving the predicate that *expresses* that property, in that situation: P such that $X = \{x : P(x)\}$. The second approach is frequently simpler and less time consuming. Consider the predicate, 'is a number': much quicker to say, and therefore to delimit the set of all numbers, than to list the set explicitly.

Intension and Extension The first approach is called giving a set *in extension*; the second method is giving a set *in intension*, i.e. by a predicate that gives a rule (satisfaction of the predicate) unifying all the items that are in the set. For some truly heterogenous sets, we may not have a convenient natural language predicate that applies to all and only the members of the set, and in that case, we must resort to giving the set in extension.

Relations and Relational Predicates A relation cannot be completely captured by an unordered set of the items that stand in the relation, as we mentioned above (page 93): it is an ordered sequence of objects which satisfies a relation. But we can capture a relation (i.e. the meaning of a relational predicate R) with an unordered set of ordered n-tuples of instances that satisfy it in a situation $\mathscr{S}$ just as in the case of predicates. In the case of a binary relation, that relation is a set of ordered pairs of relata which satisfy the relational predicate. And while we can't define a relation R in situation $\mathscr{S}$ as a subset of the domain, we can define it as a subset of the Cartesian

product of the domain with itself (§B.3.1):

$$R_{\mathscr{S}} \subseteq \mathbf{Dom}_{\mathscr{S}} \times \mathbf{Dom}_{\mathscr{S}}. \tag{3.2}$$

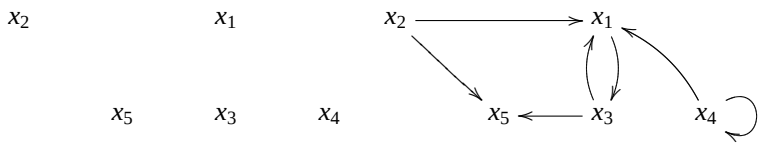
It follows immediately that if $\langle x, y\rangle \in R$ in $\mathscr{S}$, then $\{x, y\} \subseteq \mathbf{Dom}_{\mathscr{S}}$: both relata are members of the domain. Again, we can define a relation, the meaning of a relational predicate, either in extension or in intension.

3.4.2 Binary Relations

We now systematise some features of binary relations.

Graphs and Binary Relations We introduce a useful graphical notation for binary relations defined on a domain $\mathbf{Dom}_{\mathscr{S}} = \{x_1, \ldots, x_n\}$. First we lay out all the members of $\mathbf{Dom}_{\mathscr{S}}$: let us take the domain $\{x_1, x_2, x_3, x_4, x_5\}$. We lay them out as in Fig. 3.2(a).

Figure 3.2 Graphs of Binary Relations



(a) A domain of objects laid out graphically.

(b) The same domain with a relation drawn on.

Consider then some relation R on this domain. It is identical with a set of ordered pairs of members of the domain: let us define

$$R = \{\langle x_1, x_3\rangle, \langle x_2, x_1\rangle, \langle x_2, x_5\rangle, \langle x_3, x_1\rangle, \langle x_3, x_5\rangle, \langle x_4, x_1\rangle, \langle x_4, x_4\rangle\}. \tag{3.3}$$

Then, for any pair $\langle x_i, x_j\rangle \in R$, we draw an arrow from the items x_i to x_j in the figure, as in Fig. 3.2(b). The resulting picture, of members of the domain ('nodes') with arrows between them, is called the *graph* of R on that domain $\mathbf{Dom}_{\mathscr{S}}$. It is obvious that given a large enough sheet of paper, and enough time, we could draw the graph of any relation defined on a finite domain. (Note that we can also graphically represent a monadic property as a region on the graph, drawing a circle around some of the nodes, as it is a subset of the domain.)

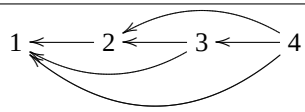
Graphs and Properties of Relations If we have a graph $\mathscr{G}$ of some relation, we can classify various properties of that relation by looking at features of the graph. We begin by defining three important properties of relations: reflexivity, transitivity and symmetry. (Exercise 3.4.1 will ask you to consider these properties of relations defined on the empty domain.)

Reflexivity A relation is *reflexive* if every node on its graph has an arrow to itself; it is *irreflexive* if no node on its graph has an arrow to itself. The graph in Fig. 3.2(b) is neither reflexive (witness x_5) nor irreflexive (witness x_4): it is then *non-reflexive*.

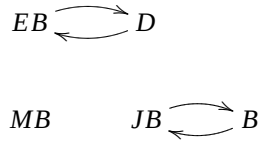
Transitivity A relation is *transitive* if, whenever there is an arrow from x to y, and an arrow from y to z, there is also an arrow from x to z. Fig. 3.3(a) shows 'greater than', a transitive relation on natural numbers. The relation is *intransitive* if there are no instances of transitivity, and *non-transitive* if it is neither transitive nor intransitive.

Symmetry A relation R is *symmetric* if, whenever there is an arrow from x to y on its graph, there is also an arrow from y to x. An arrow from x to x is obviously symmetric. The relation is *asymmetric* if there are no 'returning' arrows at all, and *non-symmetric* if it is neither symmetric nor asymmetric. A relation is *weakly asymmetric* if the only returning arrows are from x to itself: so that xRy and yRx only if $x = y$. Fig. 3.3(b) shows the graph for the relation 'is married to' on the domain {Darcy, Elizabeth Bennett, Bingley, Jane Bennett, Mary Bennett}.

Figure 3.3 Some properties of relations illustrated by graphs.



(a) Graph of the transitive relation $>$ on $\{1, 2, 3, 4\}$.

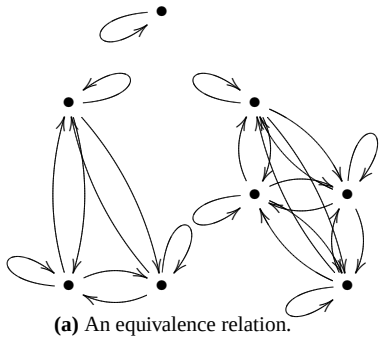


(b) Part of the graph of the symmetric relation 'is married to' in *Pride and Prejudice*.

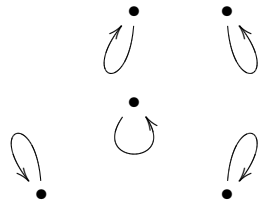
Equivalence Relations If a relation is symmetric, reflexive, and transitive, it is called an *equivalence relation*. One such relation is illustrated

in Fig. 3.4(a)—as can be seen, with a lot of nodes the arrows get rather messy. We can see that an equivalence relation *R* 'divides up' the domain into groups: we call these groups *equivalence classes* under the relation *R*, and we say that *R partitions* the domain into such classes.

Figure 3.4 Equivalence Relations



(a) An equivalence relation.



(b) Identity is an equivalence relation.

If *x* and *y* are in the same equivalence class under relation *R*, we can introduce a new *abstract noun* to capture this, as follows. Let *R* be an equivalence relation, and let *x* and *y* be related by *R*. Then we can say that *x* has the same *R*-ness as *y*. For instance, 'being exactly as tall as' is an equivalence relation; if *x* and *y* are exactly as tall as each other, then we say that *x* and *y* have the same 'tall-ness', or *height*. *R*-ness, height in this case, is an *abstraction* from the equivalence relation *R*. As you might expect, sentences involving two things being the same *P* as each other can be 'un-abstracted' into statements about an equivalence relation. Think also of 'hardness', 'intelligence', 'personality type', *&c*. It is easy to see why this works: an equivalence relation partitions the domain into regions, and on the extensional conception of a property as a subset of the domain, any region defines a property—in the example of height, the property of being (say) 168cm tall.

One famous example from the foundations of mathematics comes from one of the founders of predicate logic, Gottlob Frege. He was looking for a way to define the concept of a number, and proposed that the concept of number be defined by an abstraction from an equivalence relation, as we've been discussing. The equivalence relation he chose, which we can call *equinumerosity*, is defined over the domain of sets, and holds just when there is a one to one mapping between two sets (see page 157). The abstraction is that two sets have the same *number* just when they fall into the same

partition under equinumerosity. For further details, see Frege (1884).

Identity The identity relation = holds only between things and themselves; it is as pictured in Fig. 3.4(b). As we can see, this is an equivalence relation: it is reflexive, transitive (since $x = x$, and $x = x$, therefore $x = x$), and symmetric. Identity partitions the domain into equivalence classes of one element, and we can abstract it into 'thing'; i.e. if $x = y$, then x is the same thing as y. (Perhaps this is explanation as to why the domain of things in a situation is defined as the set of self-identical objects?)

Connectedness One further property of relations should be noted. R is *connected* if, for any two distinct nodes on the graph (any $x, y \in \mathbf{Dom}_{\mathscr{S}}$ such that $x \neq y$), there is an arrow from x to y, or vice versa (i.e. either $R(x, y)$ or $R(y, x)$). The relation 'greater than', from Figure 3.3(a), is connected. This relation of connectedness is sometimes called *trichotomy*, because it is equivalent to the condition that for any x and y, $(x = y \vee xRy \vee yRx)$.[7]

Ordering a Domain An *partial ordering* on a domain is a relation R that is reflexive, transitive and weakly asymmetric. A *strict* partial ordering is a transitive and asymmetric relation. A *total* order is a connected partial order; a strict total order is a connected strict partial order. Some examples:

Partial Order The relation $\subseteq$ on a set of sets (§B.3)—because (i) if $S \subseteq S'$, then S' cannot be a subset of S unless $S' = S$; (ii) subset is clearly transitive; and (iii) there are many pairs of sets such that neither is a subset of the other (see Figure 3.5 for an example).

Strict Partial Order The relation $\subset$ on a set of sets;

Total Order The relation $\leqslant$ on the natural numbers;

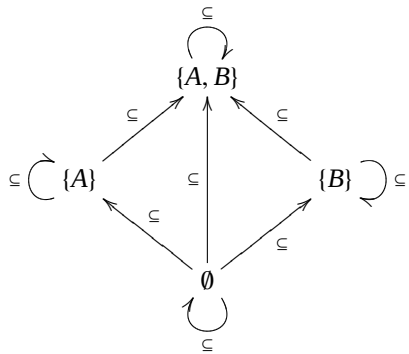
Strict Total Order The relation $<$ on the natural numbers.

[7] There is another property of binary relations, sometimes called *weak connectedness*, which holds iff for any nodes x and y, there is some path that joins x to y moving only along arrows (perhaps in the 'backwards' direction). For instance the relation pictured in Fig. 3.2(b) is weakly connected, while the relation pictured in Fig. 3.4(a) is not.

More formally, let S^* be the relation $(S(x, y) \vee S(y, x))$. A relation R is weakly connected on $\mathbf{Dom}_{\mathscr{S}}$ iff for all nodes x and y in $\mathbf{Dom}_{\mathscr{S}}$, there exists a sequence of nodes $z_1, \ldots, z_n$ such that $R^*(x, z_1)$ and $R^*(z_1, z_2)$ and ... and $R^*(z_n, y)$.

One final ordering relation is worth noting: R is a *well-ordering* of a domain $\mathbf{Dom}_{\mathscr{S}}$ if it is a connected relation such that every non-empty subset d of $\mathbf{Dom}_{\mathscr{S}}$ has a unique least member under R: it is obvious that $<$ is a well-ordering of the natural numbers, because every set of natural numbers has a least member under $<$. But the analogous relation $<$ on the positive *and negative* numbers is not a well-ordering, because there is no least negative number (and, for the same reasons, $>$ does not well-order the natural numbers).

Figure 3.5 Partially ordering the set $\{\{A, B\}, \{A\}, \{B\}, \emptyset\}$ by $\subseteq$.



Some examples from natural language We can find natural language relational predicates that illustrate these points (Hodges, 2001, §32).

- *Sameness*: We have already discussed sameness relational predicates above, but consider some further examples: 'born in the same town as', 'same age as', 'same specificity as'.

- *'At least as much', &c.*: If I have at least as much icecream as you, and you have at least as much as Victoria, then (i) I have at least as much as Victoria; (ii) I have at least as much as myself; (iii) everyone has either at least as much icecream as I do, or I have at least as much as them. That is, 'at least as much as' is reflexive, transitive, and connected. Hodges calls these *reflexive comparative* predicates; if we state them 'both ways around', we get an equivalence relation: 'I have at least as much icecream as you, and you have at least as much as me; therefore, we have exactly the same amount'.

- *More and less*: 'Cooler', 'sweatier' and 'richer' are *irreflexive* comparatives: they are asymmetric and transitive. They are related to the reflexive comparatives, as follows: if $M(x, y)$ is irreflexive comparative, then not $M(y, x)$ is a reflexive comparative. So, for instance, 'x is richer than y' means 'y is not at least as rich as x'.

Exercises for §3.4

Exercise 3.4.1: If R is defined on the empty domain, can it be reflexive? Can it be irreflexive? What about transitivity and symmetry?

Exercise 3.4.2: Can a relation be asymmetric and reflexive? Can a relation be transitive, non-symmetric and irreflexive? Can a relation be connected and irreflexive?

Exercise 3.4.3: Show that connectedness and trichotomy are equivalent.

Exercise 3.4.4: What is a relation, conceived of as a property? What is the difference between specifying a relation intensionally and specifying it extensionally?

Exercise 3.4.5: A relation is *anti-symmetric* iff the only 'two-way' arrows are loops. Give a formal condition that expresses this informal claim, and give a real life example of an anti-symmetric relation.

Exercise 3.4.6: 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.

Exercise 3.4.7: Specify, without using graphs, what conditions a relation must satisfy to be:

1. Non-reflexive;
2. Intransitive;
3. An equivalence relation;
4. A partial ordering of the domain;
5. Connected.

Give examples of each kind of relation.

Exercise 3.4.8: Show that a well-ordering of D is a strict total ordering of D, but not *vice versa*.

Exercise 3.4.9: What is the relation between abstract nouns and equivalence partitions?

Exercise 3.4.10: 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)?

Exercise 3.4.11: We might normally expect 'is similar to' to be a symmetric relation: after all, if there is a respect in which *a* is similar to *b*, then *b* must be similar to *a* in that very same respect. But many people seem to judge that similarity is *not* symmmetric:

> When people are asked to make comparisons between a highly familiar object and a less familiar one, their responses reveal a systematic asymmetry: The unfamiliar object is judged as more similar to the familiar one than vice versa. For example, people who know more about the USA than about Mexico judge Mexico to be more similar to the USA than the USA is to Mexico. (Kunda, 1999, 520)

Can you provide a rationale behind 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?

3.5 Quantification

With the resources currently at our disposal, we can make declarative sentences concerning whether or not some items designated by some designators $d_1, \ldots, d_n$ satisfy some predicate P. But we cannot yet make any remarks about how many things satisfy a given predicate, or the relationships between one predicate and another, or indeed any remarks about the domains of predicates and relational predicates at all.

'All' and 'some' Thus we introduce a new device: *quantification*, designed explicitly to be able to make quantitative and qualitative remarks about predicates, while abstracting away from the detail of exactly which items satisfy those predicates. For instance, to assert that all roses are red we need not name every rose 'r_1',...,'r_n' and assert '**Red**(r_i)', for each i. Rather, we should wish to say "Every item, if it is a rose, then it is red"; or, to deny the claim, we need not specify a named non-red rose, but rather assert "There is some item, such that it is a rose and it is not red". We shall introduce devices to capture these English locutions.

In general we may think of a quantifier as relating two predicates: telling us how many of the things which fall under the first predicate also fall under the second. In our case, we're interested if all (or some) things which fall under the first fall under the second. Having defined the domain as the set of things which fall under the predicate 'is self-identical' (§3.4.1), it is easy

to see even quantification over the whole domain as falling in the scope of this framework.

Quantifiers and Truth Functors It is tempting but dangerous to think of these quantifiers as something like disjunction and conjunction, at least in finite cases: if all roses are red, that means that Rose 1 is red **and** Rose 2 is red **and** ... **and** Rose *n* is red. Similarly, if some rose is red, that means that Rose 1 is red **or** Rose 2 is red **or** ... **or** Rose *n* is red. But in infinite cases, these disjunctions and conjunctions must be infinite, and no human or artificial language of any plausibility involves infinitely long sentences. Moreover, in cases where there are more items than names (as in the case of the real numbers, assuming a name to be a finite linguistic expression), it will not be possible at all to represent a quantified sentence even as an infinite disjunction of simple designator-predicate sentences.

3.5.1 Quantifiers in English

Of course 'all' and 'some' are not the only quantifiers in natural language. Consider the following sentences:

- More than half of the people voted for Bush.
- Most voters seem to be foolish.
- Several issues were considered important.
- A few troublemakers disrupted the polls.
  Thousands of evangelicals turned out to vote.
- The election was a disaster.

Profiles We can classify these quantifications into four types, using the sentence schema '*x* is *P*', and slotting quantifiers to apply to, or *bind* the variable *x* (Hodges, 2001, 160–2):

1. 'At least one thing that is *S* is *P*': this tells us how many *S*s are also *P*s, but nothing about how many non-*S*s are *P*s. Consider also 'No *S* is *P*', 'Thousands of *S* are *P*', *&c*.
2. 'Every *S* is *P*': this tells us how many *S*s are non-*P*s, namely, none. We can also consider 'All but one *S* is *P*', *&c*.

3. 'Half of the *S*s are *P*s': this tells us what proportion of the *S*s are *P*. Other examples include 'Most *S*s are *P*s', *&c*.

4. Definite descriptions: 'The *S* is (is not) *P*'.

Linguistics and Quantifiers Linguists recognise quantifiers are being part of a larger class of words called *determiners* (Harley, 2006, 191–3). Grammatically, all determiners attach to a noun (but *not* a designator in our sense) to form a noun phrase; some (like 'all', 'at least one', 'each') attach to count nouns ('each dog' but '*each dirt'); some (like 'much', 'little') attach to mass nouns ('much dirt', but '*much dog'); some (like 'enough') attach to mass nouns or plural count nouns ('enough doctors' but '*enough doctor'); some (like 'some' itself) seem not really to care ('some doctor', 'some dirt', 'some dogs'). These determiners all tell how much or how little of the class of things designated by the noun are being referred to, hence the name 'quantifier'. The noun here thus functions implicitly to yield a class of items of that kind (which is why a quantifier attached to a designator would be weird: 'some Antony' can only work if we think of selecting a member from the class of people named 'Antony').

Chaos? It may seem difficult to see how to to systematise all these natural language quantificational idioms into a formal language, and we may see that some are quite difficult. But we shall see how, in principle, to do so, with 'all', 'some', and identity.[8]

Definite Descriptions Indeed, we've already seen how to deal with the fourth kind: definite descriptions (§3.2.4). We analysed the definite description 'the *S* is *P*' as 'there exists at least one thing that is *S*, and at most one thing that is *S*, and that thing is *P*'. Note that this is equivalent to: 'there is a thing that is *S*; and there are not two or more distinct things that are both *S*; and all *S* are *P*'. So if we can use 'all', 'some' (i.e. one or more), and whether things are distinct or non-distinct, we can analyse definite descriptions. We shall see how to do this in what follows. Definite descriptions (and

[8]Yet we shouldn't be too cocky; one the most sophisticated treatment of quantifiers in natural language linguistics (so-called 'generalized quantifier theory'), these quantifiers we can deal with in first-order logic turn out to be exceptional and unusual cases. No first-order treatment of the quantifier 'most' can be given, for example, and if we introduce apparatus to deal with it, we end up giving a reanalysis of the quantifiers 'for all' and 'there exists' too. But that is technical material far beyond the scope of this book.

demonstratives, like 'this' and 'that') are also determiners; but they have a quite different linguistic role to play than quantified noun phrases.

3.5.2 Domains of Quantification

Since quantifications operate on predicates, we can adopt the idea of a domain of quantification straight over from the domain of a situation in which a predicate is satisfied or not (page 96). In a given situation, $\mathscr{S}$, the domain $\mathbf{Dom}_{\mathscr{S}}$ is all the individuals that are possible candidates for satisfaction of some predicate; alternatively, those that do satisfy the predicate '$x = x$'. So, for instance 'all golfers like golf' is true in a situation if every member of the domain that is a golfer also likes playing golf (and doesn't just do it to help their business relationships, for example).

Conditions on domains In our golfing sentence, we quantify over a domain that includes some items, and we set down a condition: if any of those items are golfers, then those things must also like golf. But does this decide whether or not some non-golfers are to be included in the domain? For instance, is the number 7 to be included? It is clearly not a counterexample, since it is not a golfer at all. But it is also quite clearly therefore not relevant to assessing the truth or falsity of the quantified claim. We shall make the stipulation that the domain is to include as many things as are consistent with the acceptable assertion of the quantified sentence: numbers, dogs, alien beings, *&c.* are all in to the domain of quantification, for they do no harm as long as we ensure that we get the initial conditions right; i.e. as long as we make sure our claim that 'everything likes golf' is suitably restricted to 'everything that plays golf, likes it'.

Quantifier domain restriction But this gives rise to another problem that we noted earlier: sometimes we make quantified claims that seem to refer to a smaller or more restricted domain than would strictly speaking be correct. For instance, if we assert 'There is no more beer', we may very well know that there is plenty of beer at the bottle shop. So how can we assert that sentence?

Context The answer is that the context makes some restriction of the domain of the quantification acceptable. For instance, if we are having a party and everyone is leaving early, and you ask me why they are all leaving, I may correctly assert in explanation 'There is no more beer', meaning, of

course, that there is no beer left at the party, which is the salient explanation. By contrast, it would be inappropriate for you to reply to my assertion 'But there is plenty of beer—just not here'. For while that might very well be true in the wider context, it is mere pedantry to assert it, and indeed it is false in the relevant context, which is restricted to the domain of things at the party.

A caution So the lesson of quantifier domain restriction is that we need to be cautious in deciding what structures a sentence is meant to apply to. So, while you might think that actual sentences are just meant to apply to the actual world as a whole, that would be incorrect, since some actual sentences assert propositions that are true only in smaller structures than the whole world: structures that include, for example, a strict subset of all the entities in the world. For more on quantifier domain restriction, see Stanley and Szabó (2000).

Exaggeration Sometimes, indeed, it is acceptable to assert a sentence like 'Everyone knows who David Lewis is!' If this sentence is to be true, it must also be restricted in scope to a certain domain: perhaps professional philosophers in the English speaking world, for example. Hodges (2001, 164) seems to think that exaggeration is distinct from quantifier domain restriction; I am inclined to disagree, since both rely at bottom on the ability of contextual clues to limit the domain of quantification to relevant or salient cases. In exaggerations, that restriction is to 'important' or 'core' cases. To refute my assertion, for example, it would suffice to bring to our attention the case of a tenured professor of philosophy who had not heard of David Lewis; but it may not serve to refute my assertion if the only countervailing case is a junior assistant lecturer in a very marginal discipline of philosophy, for example. (You may wish to think for yourself whether or not quantifier domain restriction can deal with all exaggerations.)

Empty Domains Should the domain of quantification have at least one member? Many logics do require that there be at least one item in the domain; indeed, our logic does if the sentences we are trying to decide the truth value of contain designators, which we have assumed to have a primary referent in the domain. But we shall not stipulate that every proper domain must contain at least one member. This will come up again below, when we are deciding how to interpret our quantifiers (page 114).

3.5.3 'All' and 'Some'

The two quantifiers we shall concern ourselves with are 'Every thing is such that . . . ' and 'there exists at least one thing such that . . . ', shortened to 'for all things, . . . ' and 'for some thing(s), . . . '.

Variables To understand how quantifiers work, we now have to introduce a new category of linguistic item: the *variable*. In ordinary English, the variables are words like 'thing': items that are governed by a quantifier, and can also appear connected with a predicate, as in 'Everything is self-identical', which can be read as 'Every thing is such that: that thing is identical to itself'. In formalising natural language we use variables appearing near both the quantifiers and the predicates to play this role—rather than, for example, relying on anaphoric reference between 'thing' and 'itself' to indicate which quantifiers attach to which predicates. That is, we render our quantifiers like so: 'for all x, $\ldots x \ldots$', where $\ldots x \ldots$ is a predicate phrase which involves some predication of x.

Binding places, not predicates We can create quantified sentences involving some predicate without binding every possible way that a designator might attach to that predicate. That is, what is bound is one or more *argument places* of a predicate, not the predicate as a whole. So, for instance, if we have the 2-place predicate 'loves', a designator a, and a universal quantifier, we can make the sentences: 'everything loves itself'; 'everything loves a'; 'a loves everything' and 'a loves a'. The predicate 'a loves' can be bound by the universal quantifier into 'For all things, a loves that thing'. But importantly, the predicate 'loves' can be bound into 'For all things, that thing loves. . . ': which is not a sentence at all, but rather a more complex predicate! This can be bound again, by a different quantifier, but the English phrasing can become rather difficult! We'll see how to deal with this below (§3.5.4).

Free occurrences of a variable Consider the predicate 'is red', and the variable 'thing'. 'thing is red' is obviously not a declarative sentence; but if we prefix a quantifier like so: 'Every thing is such that: that thing is red', then we get a declarative sentence. We say that in the original phrase, the variable occurred *free* or *unbound*: that is, no quantifier applied to the variable even though it was attached to the predicate. If a variable occurs free in a phrase, then it can be bound by the appropriate quantifier. This is

how quantifiers apply to predicates: they describe how many things might satisfy the predicate by attaching to some 'surrogate' object—the variable—that attaches to the predicate. When a variable is bound by a predicate, it no longer functions as a variable: it functions rather simply as part of the larger sentence, simply serving to indicate which predicate the quantifier applies to.

Universal Quantification The quantifier 'for all' is also called the *universal quantifier*. For example, the sentences 'Everything is self-identical'; 'Every man is mortal'; 'All of us want to go to the party'; 'Each client is entitled to a free consultation' are all universal quantifications. Some of these sentences have an 'absolute' form: they say of everything that some predicate applies to it. But some others are conditional: 'every client is entitled to a free consultation' does not say of every thing that it is entitled to a free consultation, but rather says of everything that, if it is a client, then it is entitled to a free consultation. That is, the universal quantifier binds the one-place conditional predicate 'if . . . is a client, then . . . is entitled to a free consultation'. More simply, the variable occurs free twice in this phrase, and is bound by one quantifier. The phrase itself is a conditional: in our old notation, 'If the thing is a client → the thing is entitled to a free consultation'.

A universally quantified sentence is true just in case everything in the domain of the quantifier satisfies the predicate. This is true even in the case of conditional predicates, but those might be understood better as true just when everything in the domain which satisfies the predicate appearing in the antecedent, also satisfies the predicate appearing in the consequent. Note that this has the consequence that, if nothing in the domain satisfies the predicate appearing in the antecedent of the conditional, then the universally quantified sentence can still be true, because of the way that we have stipulated that conditionals of our formal language work (§2.4.1).

Existential Quantification The other kind of quantifier we deal with is the *existential quantifier*, 'for some'. Though 'some' has a natural language meaning of 'a few', for us it will simply mean 'for at least one thing'. The sentences 'There is a man who speaks Spanish', 'For some cats, there are dogs that chase them', 'There are some cold deserts', 'There exist consistent sets of sentences' are all examples of existential quantifications. Each of these sentences has an absolute form: they say that something meeting some set of conditions exists. When we apply multiple predicates to a sin-

gle variable, we do so not in a conditional form, as in the case of universal quantification, but in *conjunctive* form: 'There exists something such that it is a man *and* that thing can speak Spanish'. (Contrast this with 'Everything is such that, if it is a man, then it can speak Spanish'.) If we tried the conditional form, we would get the sentence 'There exists something such that, if it is a man, then it can speak Spanish'. This latter sentence can be true even if there are no men, which is not what we intended the original sentence to say!

An existentially quantified sentence is true just if there is at least one thing in the domain of quantification which satisfies each of the predicates that the quantifier applies to. No existentially quantified sentence can be true in an empty domain.

3.5.4 Multiple Quantifiers and Scope

We talked above of 'universally quantified' and 'existentially quantified' sentences. We mean by this, a sentence which has as its main quantifier, a quantifier of that type. This notion is trivial in sentences which have only one quantifier as the main logical operator, but it becomes interesting when we consider complex sentences which may have two or more quantifiers.

Scope Not every sentence has universal or existential form. The sentence 'Every dog is happy and some cat is happy' has the form '(For all things that are dogs, those things are happy) $\wedge$ [For some thing, it is a cat and it is happy].' This is a conjunction, and has conjunctive form. So what we mean by existential or universal form, is that the operator which has largest *scope* in the sentence is a quantifier of some particular form, just as we mean by 'conjunction' a sentence in which the conjunction takes widest scope. So we decide by seeing what the main operator of a sentence is, just as in the propositional case: only now quantifiers may be among the operators.

Multiple Quantification What if a sentence has two quantifiers appearing in it, like 'every dog is some man's best friend'? How are we to understand this sentence; and how does it differ from the sentence 'every dog is every man's best friend'? We decide by looking at which predicates are bound by which quantifiers. Therefore we must begin by finding a method of linking quantifiers to predicates. We do this by making our variables clear: instead of just 'thing', we can write 'thing_1' and 'thing_2', *&c*. So we shall have 'for every thing_1 such that thing_1 is a dog, there is a thing_2 such that thing_2 is

a man, and thing$_1$ is thing$_2$'s best friend'. That is clearly something of a mouthful, so we introduce some new, more concise, notation.

'$\forall x$' and '$\exists y$' Our new notation is as follows. Instead of 'thing$_1$' and so on, we use quasi-mathematical notation for variables: 'x', 'y', 'z',.... Instead of 'for all things', we shall write '$\forall$'; instead of 'there exists at least one thing', we shall write '$\exists$'. To identify which variable goes with which predicate, we adopt the following convention: we write the variable next to the quantifier; and we use no repeat occurrences of an unbound variable.

Example and scope considerations So, for instance, we write 'every dog is some man's best friend' as '$\forall x$ (x is a dog $\rightarrow \exists y$(y is a man $\wedge$ x is y's best friend))'. There is no hope of confusion here: the universal quantifier clearly binds the xs, and the existential quantifier clearly binds the y. The scope is clear too: the existential quantifier is within the scope of the universal quantifier. We can give a more rigorous test of this, by stating that a sentence is of universal form if the last variable in the predicate to be bound was bound by a universal quantifier; similarly for existential. So in this case we had the predicate 'x is a dog $\rightarrow \exists y$(y is a man $\wedge$ x is y's best friend)', where the 'x' occurs free. (We call a predicate of this form an *open sentence*—it will be closed, and hence a declarative sentence, once the variables are all bound.) The 'x' was bound by $\forall x$, and that therefore closed the sentence and led it to be of universal form.

Quantifier rearrangement If a quantifier, like '$\exists y$', does not bind any variable before it in a sentence, it can be brought to the front. Thus we could have written our example as '$\forall x \exists y$(x is a dog $\rightarrow$ [y is a man $\wedge$ x is y's best friend])'. Then it is simply the order of quantifiers 'out the front' that governs scope, along with the rules for scope that we discussed above (page 40).

'All', 'every', and 'any' Consider the two sentences (1) 'I don't know anything' and (2) 'I don't know everything'. In the second, the correct analysis is '$\neg \forall x$(I know x)'. In the first, the correct analysis is '$\forall x \neg$(I know x)': the scope of the negation alters, even though the 'surface form' of the English sentences is very similar. We may interpret this as evidence that 'any' takes larger scope than 'every' or 'all'. Consider again the sentences (1) 'some girl won every prize' and (2) 'some girl won any prize': here we feel again that in (2), 'any' should take larger scope than 'some', even

though 'some' occurs at the front of the sentence. This suggests that these sentences should be rendered (1a) '$\exists x \forall y$(y is a prize $\rightarrow$ x won y)' and (2a) '$\forall y \exists x$(y is a prize $\rightarrow$ x won y)' (i.e. any prize was won by some girl). But this latter interpretation is by no means indisputable; and one should be wary of any such broad generalisations when examining natural language. Such difficulties abound in natural language; one has to trust one's own judgment as to the logical form of any given quantified sentence of natural language, and as to how it might be rendered accurately using our quantifiers. (See Hodges (2001, §37).) Practice is the only thing that helps one see and interpret the quantificational structure of natural language sentences.

3.5.5 Quantifiers and Arguments

Say that we have some quantified sentence. What follows from it? There is one general principle that governs our understanding of these issues: namely, that any true quantified sentence is made true by the pattern of property instantiation of the individuals in the domain of quantification. So the consequences of quantified sentences are, in the first instance, to be understood in terms of the individuals which make such sentences true.

Universal Instantiation The first rule we shall consider is *universal instantiation*: a sentence '$\forall x(\ldots x \ldots)$', implies that, for any designator a which refers to some element of the domain of quantification, '$(\ldots a \ldots)$'. That is, substituting 'a' for every bound occurrence of some variable x, and 'dropping' the universal quantifier which binds that variable, will give a true sentence if the original quantified sentence was true. Note that we have required that 'a' already be known to refer to some member of the domain; according to our prior assumptions, that can only be the case if 'a' occurs purely referentially in some other accepted sentence. Thus if we only have a universally quantified sentence, we *cannot* infer that it is true of some individual; since the domain may well be empty for all we know to be true already.

So, for instance, from 'Gold is an element' and '$\forall x$(x is an element $\rightarrow$ x is not a compound' we can infer, firstly, that 'If gold is an element, then gold is not a compound'; and then, by propositional logic from the truth 'gold is an element', that 'gold is not a compound'.

Existential Witnessing If a consistent set of sentences contains an existentially quantified sentence, we cannot generate an inconsistency by giv-

ing an arbitrary name to some item in the domain that makes that existential statement true. It follows that an existentially quantified sentence, '$\exists x(\ldots x\ldots)$' implies, for some previously *unused* designator '*b*', the sentence '$(\ldots b\ldots)$'. '*b*' is understood now to refer some arbitrary one of the entities that make the existentially quantified claim true; such an entity is called a *witness* to the truth of the existential claim. There must be such an entity since there can be no true existential sentence in an empty domain.

The proviso that the name be unused is essential. Consider the following sentences 'Gold is an element'; 'There exists a compound substance'; therefore 'Gold is a compound substance'. This argument is clearly flawed, since the third sentence contradicts the first. If we had introduced a new name 'Kljgfd', by dubbing whatever element of the domain makes the second sentence true, 'Kljgfd is a compound substance' would be true, and not contradictory. Of course, with no independent knowledge about 'Kljgfd' apart from the fact that it is a compound element, nothing further follows.

But if we apply the existential witnessing rule *first*, then the universal instantiation rule, we can get genuine information. So, for instance, if we have the sentences 'There is at least one dog' and 'All dogs are smelly', we can infer 'Fido is a dog' and thence 'Fido is smelly'.

Negated Quantifier Rules We now know how to use quantifiers; but we introduce some simple rules for negated quantifiers. Basically, they tell us how to reduce negated quantifier rules to the two preceding rules, as follows:

$\neg\forall$x '$\neg\forall x(\ldots x\ldots)$ is true in the same circumstances as '$\exists x\neg(\ldots x\ldots)$'. This rule is intuitively sound: if not everything satisfies some predicate, then there must be some thing which is a witness to that fact, i.e. something which fails to satisfy the predicate.

$\neg\exists$x '$\neg\exists x(\ldots x\ldots)$' is true in the same circumstances as '$\forall x\neg(\ldots x\ldots)$'. Again, intuitively, if there is no witness that satisfies some predicate, then everything must fail to satisfy the predicate.

Herbrand sentences and consistency Now that we have these two informally phrased rules that give some sense to what our quantifiers mean, we can start to see how arguments that use them might work. Remember our definition of validity: an argument is valid if the set of sentences consisting of the premises and the negation of the conclusion is inconsistent. An inconsistent set of sentences is one that cannot have all of its members true in some situation. In the propositional case, that meant: some sentence and

its negation were consequences of sentences in the set. In our case, an argument that contains some quantified sentences will be valid just in case a certain set of sentences related to that argument is inconsistent. We give two definitions. Let a *quantifier-free* sentence be one that does not contain any quantifiers, only designators and predicates.

Definition 6 (Herbrand Sentences). A *Herbrand sentence* of a set X is any quantifier-free sentence that either (i) is already a member of X; or (ii) follows by some combination of uses of Universal Instantiation and Existential Witnessing (and negated quantifier rules) from a member of X. Let 'H_X' name the set of Herbrand sentences of X.

H_X is a set of sentences that does not contain any quantifiers. If that is so, then we can apply the propositional logic of chapter 2 to H_X. So we can simply extend our propositional logic to the quantified case:

Definition 7 (Inconsistency with Quantifiers). A set of sentences X, possibly containing quantified sentences, is *inconsistent* iff H_X is semantically inconsistent according to propositional logic (see Definition 2).

Hence we can apply all our knowledge of propositional logic to the quantified case! We shall see how to do this formally below (§3.6 onwards).

Empty domains revisited Since we have decided that some universally quantified sentences can be true in an empty domain, but that no existentially quantified sentence can be true in an empty domain, it follows that there can be no completely general rule which tells one that some arbitrary existentially quantified sentence can be a logical consequence of arbitrary universally quantified sentence, if empty domains are possible. So, for instance, the argument from premise '$\forall xFx$' to conclusion '$\exists xFx$' is invalid, since in the empty domain the premise will be true and the conclusion false.

But this argument, at least when translated into English, seems okay: 'everything is F, therefore at least one thing is F'. If we modify our rule of universal instantiation, we can get a logic which vindicates this argument. Let us see how. The new rule, UI$^+$, says: from '$\forall x(\ldots x\ldots)$', we can infer '$\ldots a\ldots$' for any designator '$a$', whether it has been used or not. But if '$\ldots a\ldots$' is true, then 'a' is a witness for $\exists x(\ldots x\ldots)$. So if we adopted UI$^+$, instead of the original form, we will have ruled out empty domains, but we would have allowed that an existentially quantified sentence follows from the universally quantified sentence of the same predicate. Having a logic which validates this intuitively correct inferential structure has seemed to

many people to be a benefit which outweighs the virtues of caution displayed by those who advocate the possibility of empty domains, so most standard logics are not compatible with empty domains. In what follows, we'll keep UI as the primary rule, but will discuss UI^+ whenever it makes a difference to the topics under discussion—see, for example, the discussion at p. 125.[9]

3.5.6 Numerical Quantification

We have shown how to deal with 'all' and 'some', but we have not shown how to give precise numbers as to how many entities satisfy a given predicate. We should now correct this.

Identity and number If there are two entities, and they are distinct, then we can give them names (say, 'a' and 'b') and make the true non-identity claim '$a \neq b$'. This sentence is a witness for '$\exists x \exists y(x \neq y)$': that is, the quantified sentence is true just when at least two things exist. So too for three things: '$\exists x, y, z(x \neq y \wedge y \neq z \wedge x \neq z)$'. One can in principle see how to construct any sentence of this sort. These sentences say 'at least two (three, four, *&c*.) things exist'. (Clearly, to say that 'at most two things exist' is just to say 'It is not the case that at least three things exist', so this type of sentence poses no trouble.)

If we wish to say that *exactly* some number of entities exist, we need to add another conjunct to these sentences. This conjunct will say that everything is identical to one of the three things already known to be non-identical. So the claim that there are precisely three entities is expressed by the sentence

(3.4) $\exists x \exists y \exists z(x \neq y \wedge y \neq z \wedge x \neq z \wedge \forall w(w = x \vee w = y \vee w = z))$.

Obviously, we can also express 'exactly n' by 'there are at least n and there are at most n', conjoining the obvious translations from the previous paragraph.

Number and other predicates We can then combine these sentences saying how many things there are with other predicates. For some arbitrary

[9] UI^+ is called Rule VII in Hodges (2001, 192); see also the discussion in Bostock (1997, §8.4).

predicate 'P', we can say

$$\exists x \exists y (x \neq y \wedge P(x) \wedge P(y)) \tag{3.5}$$

to mean 'there are two P things', and so on.

Special numerical quantifiers We could, if we wished, introduce special quantifiers that abbreviate these longer sentences. So, for instance, we could introduce '$\exists!^n x P(x)$' to mean 'There are exactly n Ps', where that abbreviates the complicated sentence:

$$\begin{aligned} \exists x_1 \ldots x_n \Big(& (P(x_1) \wedge \ldots \wedge P(x_n)) \wedge (x_1 \neq x_2 \wedge \ldots \wedge x_{n-1} \neq x_n) \\ & \wedge \forall y (P(y) \rightarrow y = x_1 \vee \ldots \vee y = x_n) \Big). \end{aligned} \tag{3.6}$$

We shall not do this, even though in principle we could, and some people do.

Definite descriptions revisited We saw above in §3.2.4 that Russell's analysis of a definite description like 'the S is P' is equivalent to 'there is exactly one S and it is P'. We can now render 'There is exactly one S' as '$\exists x (S(x) \wedge \forall y (S(y) \leftrightarrow y = x))$' (that is, there is at least one S, and if anything else satisfies S, then it is identical to the first thing). (We could write this $\exists!^1 x S(x)$.) We can then conjoin this to the claim that 'every S is P', and we therefore analyse Russellian definite description 'the S is P' as

$$\exists x (S(x) \wedge \forall y (S(y) \leftrightarrow x = y)) \wedge \forall z (S(z) \rightarrow P(z)). \tag{3.7}$$

Exercises for §3.5

Exercise 3.5.1: What are the truth conditions for a sentence of the form 'All Fs are Gs'? How do these truth conditions relate to claims like 'All the customers have left'; 'Everyone likes cheesecake'; 'All round squares are round'. What different contribution do 'every' and 'any' make to sentences in which they occur?

Exercise 3.5.2: We formulate 'all Fs are Gs' by saying 'For all x, if x is F then x is G'. Why don't we formulate 'Some Fs are Gs' by 'For some x, if x is F then x is G'?

Exercise 3.5.3: Why does '$\forall x \exists y R x y$' not mean the same as '$\exists y \forall x R x y$'?

Exercise 3.5.4: What is the relation of Herbrand sentences to inconsistency?

Exercise 3.5.5: Using identity and quantification to express the claim that ' There are three *F*s and three (distinct) *G*s'. Can we use identity and quantification to express 'there are exactly as many *F*s as *G*s'?

3.6 A new formal language: $\mathscr{L}_2$

We can now incorporate the linguistic phenomena we have been discussing into a formal structure. This language will *extend* $\mathscr{L}$, in that every wff of $\mathscr{L}$ will also be a wff of $\mathscr{L}_2$, but not vice versa. We have already been using parts of it as notational conveniences above, mixed in with ordinary English, in a bizarre kind of 'logician's creole', but it is well past time for us to put the grammar and form of this language on a firmer and more regular footing.

Alphabet for $\mathscr{L}_2$ We begin by looking at the extensions to the alphabet. $\mathscr{L}$ had the following alphabet: $p \quad ' \quad (\quad) \quad \neg \quad \wedge \quad \vee \quad \rightarrow \quad \leftrightarrow$, and we let $q, r, \&c.$ stand for p', p'' and so on. In what follows, we no longer imagine that p, q and so on are unanalysed sentences, because we shall analyse their predicative structure also. So our new alphabet has no need of 'p' or ''''; it is detailed in Table 3.2. Given this alphabet, the inductive formation rules for $\mathscr{L}_2$ are described in Table 3.3 (page 119).

Table 3.2 The alphabet for $\mathscr{L}_2$

$(,), \neg, \wedge, \vee, \rightarrow, \leftrightarrow$	The Alphabet of $\mathscr{L}$, without atomic sentences
$x, y, z, \ldots$	Variables
$a, b, c, \ldots$	Designators
$P^1, Q^1, R^1, \ldots$ $\vdots$ $P^n, Q^n, R^n, \ldots$ $\vdots$	n-place Predicates and Relational Predicates
$=$	Privileged 2-place Identity Relational Predicate
$\forall$	Universal Quantifier
$\exists$	Existential Quantifier

One important thing to note in the light of our earlier considerations is that, even though we treat definite descriptions as designators, they are not counted as such for the purposes of our alphabet, since we are presupposing

Russell's analysis of definite descriptions, which eliminates them in favour of complicated constructions involving quantifiers and identity (page 116).

Rules (7) and (8) may require some clarification, though the way they function is actually quite simple. If we have some wff '$R^4(a, b, c, d)$', we wish to be able to form the quantified wff '$\forall x(R^4(a, b, c, x)$' but we also wish to form the wff '$\forall x(R^4(a, x, x, x)$'. Rules (7) and (8), because they talk of replacing a sequence of one or more designators, rather than just a single designator, allow for this formation. In fact, we could make do with a simpler formulation of these rules, because that latter wff could have been formed from the wff '$R^4(a, b, b, b)$': see the exercises.

Having set up the elaborate rules, we can now feel free to break them, in particular, by omitting parentheses where no ambiguity will result, by condensing strings of adjacent universal (or existential) quantifiers (i.e. condensing '$\forall x \forall y \forall z$' to '$\forall x, y, z$'), and by writing, when desired, '$a \neq b$' for '$\neg(a = b)$'.

One interesting thing to note about the rules in Table 3.3 is that expressions of the form 'Rxy' are not wffs. Such an expression, with variables in place of designators yet no quantifiers to bind those variables, is known as an *open sentence*. Some systems of logic permit open sentences to be wffs, and then give a corresponding semantics that assigns a semantic value to such expressions (so, for example, Bostock 1997, 78 explicitly allows that open sentences are well-formed). But it is very difficult to see how such sentences can be sensibly evaluated for truth (unless we simply presuppose that there is a hidden quantifier, as we often do in elementary algebra for example), and since they cannot be so evaluated, systems which assign them a semantic value tend to look slightly unusual (for more discussion on this point, see p. 133). Moreover, it is quite difficult to define formation rules that allow for open sentences to be wffs, but don't also allow expressions with quantifiers preceding arbitrary wffs (like '$\forall xP$') to count as wffs (Bostock, 1997, 80). Our system only permits 'sensible' introductions of quantifiers to bind variables occurring in the scope of the quantifier.

Exercises for §3.6

Exercise 3.6.1: Show that Rule (7) in Table 3.3 could be replaced by the following simpler rule, without loss of generative capacity:

> **(7-)** If φ is a wff, and d is a designator that occurs in φ, then if φ^d_v is the result of replacing each occurrence of d by some variable v that does not occur in φ, '$\forall v(\varphi^d_v)$' is a wff.

Table 3.3 Rules of formation for wffs of $\mathscr{L}_2$

0. Any n-place predicate 'P^n', followed by a sequence of n designators '$(\underbrace{a, b, c, \ldots}_{n})$' (possibly involving repetition) contained in parentheses is a *basic formula* of $\mathscr{L}_2$. For example: '$P^2(a, b)$', or '$R^3(c, c, d)$'. For the special case of the identity predicate '=', we allow the exceptional form '$a = b$' rather than '$= (a, b)$'.

The following rules govern the formation of *well-formed formulae of* $\mathscr{L}_2$ ($\mathscr{L}_2$-wffs) from other wffs:

1. Every basic formula is a wff;
2. If 'φ' is a wff, '$\neg\varphi$' is a wff'
3. If 'φ' and 'ψ' are wffs, so is '$(\varphi \wedge \psi)$';
4. If 'φ' and 'ψ' are wffs, so is '$(\varphi \vee \psi)$';
5. If 'φ' and 'ψ' are wffs, so is '$(\varphi \rightarrow \psi)$';
6. If 'φ' and 'ψ' are wffs, so is '$(\varphi \leftrightarrow \psi)$';
7. if 'φ' is a wff, and some set of designators $D = \{d_1, \ldots, d_n\}$ is such that each member occurs in one or more places in φ, then the result of substituting a variable v that does not occur in φ for every occurrence of each member of D, enclosing the whole in parentheses, and prefixing '$\forall v$'—i.e., '$\forall v(\varphi(\ldots v \ldots))$'—is a wff;
8. if 'φ' is a wff, and some set of designators $D = \{d_1, \ldots, d_n\}$ is such that each member occurs in one or more places in φ, then the result of substituting a variable v that does not occur in φ for every occurrence of each member of D, enclosing the whole in parentheses, and prefixing '$\exists v$'—i.e., '$\exists v(\varphi(\ldots v \ldots))$'—is a wff.

Nothing else is a wff unless its being so follows from these rules.

Exercise 3.6.2: Show how to generate the wff '$\forall x \exists y \forall z(Rxy \rightarrow Ryz)$' using the formation rules in Table 3.3.

Exercise 3.6.3: Using the appropriate interpretation, translate the following claims into the language of $\mathscr{L}_2$:

1. Bill is a barber.
2. Bill cuts everyone's hair.
3. Everyone whose hair Bill cuts, cuts someone else's hair too.
4. Some whose hair Bill cuts also have their hair cut by another.
5. Bill cuts the hair of everyone who doesn't cut their own hair.

Exercise 3.6.4: In this exercise, use the following interpretation:

Domain:	Regular solids
Cx:	x is a cube
Tx:	x is a tetrahedron
Lxy:	x is larger than y

Formalize the following sentences into the language of $\mathscr{L}_2$:

1. There are no more than two cubes.
2. Some cube is smaller than all tetrahedra.
3. At least two cubes are different sizes.
4. Exactly one tetrahedron is intermediate in size between two cubes.
5. If there is more than one tetrahedron, then a tetrahedron is the largest object that is either a tetrahedron or a cube.

Exercise 3.6.5: Using the same interpretation as the previous exercise, which of the following, if any, express the claim that there is at most one cube? Justify your answers.

1. $\forall x \forall y((Cx \wedge Cy) \rightarrow x = y)$.
2. $\forall x \forall y(x \neq y \rightarrow (Cx \rightarrow ((Cy \vee Ty) \rightarrow Ty)))$.
3. $\exists x \forall y(Cx \wedge (Cy \rightarrow x = y))$.
4. $\forall y \neg \exists x(Cy \wedge Cx)$.
5. $\forall y \neg \exists x(Cy \rightarrow Cx)$.

Exercise 3.6.6: Give a set of formation rules for wffs of $\mathscr{L}_2$ that allow open sentences to be well-formed. Justify your rules.

3.7 Tableaux for $\mathscr{L}_2$

Having set up our formal language, we now set up a formal calculus for manipulating wffs of $\mathscr{L}_2$. This will extend the tableaux system of chapter 2 to deal with arguments involving quantifiers. The informal remarks contained in §3.5.5 will serve as a guide, but we should refrain from interpreting these rules until such time as we can set down a formal system of interpretation and semantics for $\mathscr{L}_2$(such time will come in §3.8).

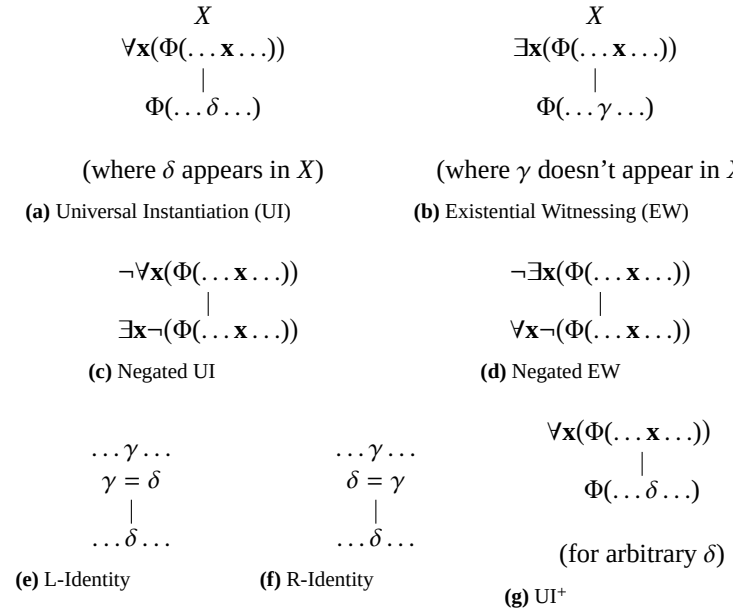
The tableaux system for $\mathscr{L}_2$ includes all of the rules shown in Fig. 2.4, plus the following new rules, shown in Fig. 3.6 (page 122).[10] You must note that I include the rule UI^+ (Fig. 3.6(g)), even though we shall not use that rule in general, preferring the rule pictured in Fig. 3.6(a) (see page 114). We shall however define a system $\mathscr{L}_2^\star$ in which the rule UI^+ replaces the rule UI; we shall mention this system again below. In most systems of first-order logic, UI^+ is the standard basic rule (with UI often not even being mentioned).

3.7.1 $\mathscr{L}_2$-tableaux syntactically defined

We define tableaux mathematically as before, with some slight alterations. A $\mathscr{L}_2$-*tableau* is a tree, in the mathematical sense, with the following properties: it has a root node, with one or more branches descending from that root, each branch of which consists of nodes and which terminates in a leaf node (if it terminates). Each node of the tree contains one wff of $\mathscr{L}_2$. A countable set of wffs $\Xi = \{\xi_1, \ldots, \xi_n, \ldots\}$ *generates* a tableau if the *i*-th node of each branch of the tableau is the formula $\xi_i \in \Xi$, and each further node of the tree is the result of applying one of the patterns of wffs in Fig. 2.4 or (a)–(f) of Fig. 3.6 to a wff that lies earlier on the same branch (i.e. closer to the root). A branch is called *closed* if either (i) two wffs of the form φ and $\neg\varphi$ appear on that branch, or (ii) a wff of the form $\neg(\delta = \delta)$ appears on that branch; otherwise the branch is *open*. A tableau that conforms to these rules is called *closed* if every branch is closed; otherwise it is called *open*. We shall call a tableau that is formed in accordance with these rules a *syntactic* $\mathscr{L}_2$-*tableau*.

[10] Notation: In this section we shall use Φ as a variable over predicates and relational predicates, δ, γ as variables over designators, $\mathbf{x}$ as a variable over variables of $\mathscr{L}_2$. Hence '$\Phi(\ldots\mathbf{x}\ldots)$' means some arbitrary predicate with some variable attaching to it.

Figure 3.6 Additional Rules for Predicate Tableaux.



Definition 8 ($\mathscr{L}_2$-Syntactic Inconsistency). Let Ξ be a set of $\mathscr{L}_2$-wffs. Ξ is *$\mathscr{L}_2$-syntactically inconsistent* if there exists a closed $\mathscr{L}_2$-tableau generated by Ξ.

It is important to note that this definition never mentions truth or structures.

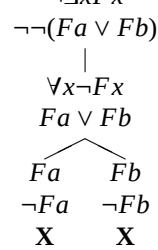
We define $\mathscr{L}_2^\star$-tableau exactly as above, except that we apply the rules seen in (b)–(g) of Fig. 3.6. The expected definition can then be given of $\mathscr{L}_2^\star$-syntactic inconsistency.

You will notice that our rules basically are a method of generating Herbrand sentences (§3.5.5) from quantified sentences in the generating set; and then we simply apply the same standard for closed sentences, with the exception that we take wffs like '$\delta \neq \delta$' as contradictions themselves sufficient to close a branch.

3.7.2 Incomplete Finitely Generated Tableaux

If you recall the discussion in §2.10, you will note a significant change here. We did not talk about completed $\mathscr{L}_2$-tableaux; and we did not talk about using $\mathscr{L}_2$-tableaux to prove consistency of sets of $\mathscr{L}_2$-wffs. The first change is easy to explain. If we have an $\mathscr{L}_2$-wff of the form '$\exists\mathbf{x}(\Phi(\ldots\mathbf{x}\ldots))$' appearing on a tableau, we can apply the EW rule infinitely many times, each time with a new name, and never 'complete' the branch. (And if we have an $\mathscr{L}_2^\star$-wff of the form '$\forall\mathbf{x}(\Phi(\ldots\mathbf{x}\ldots))$', we can do the same thing.) So there is no sense to be made of the notion of applying the rules to formula of ever-decreasing complexity and finally terminating, since some formulae are not 'checked off' after having a rule applied to them.[11] Moreover, if we insist that each formula is to be used only once, we will run into difficulties. Consider, for example, the tableau generated by the set $\{\neg\exists xFx, \neg\neg(Fa \vee Fb)\}$, and shown in Fig. 3.7. The argument corresponding to this tableau is intuitively valid, but if we were only allowed to apply the UI rule to the formula '$\forall x\neg Fx$' once, we would be able to deduce only one of '$\neg Fa$' or '$\neg Fb$', and the tableau would remain open.

Figure 3.7 A tableau requiring two applications of UI to close.



The second problem is much more serious. This is because, with our new rules, it is possible that a tableaux generated by a consistent and finite set of $\mathscr{L}_2$-wffs cannot be completed, and hence we cannot decide whether there is an open branch or not, and hence we cannot decide whether the set is consistent. For example, look at the initial $\mathscr{L}_2$-tableau fragment in Fig. 3.8, generated by the set $\{\forall x\exists y(R(x,y)), \neg R(a,a)\}$ (Jeffrey, 1991, §4.11). We

[11] We can still introduce the idea of a completed $\mathscr{L}_2$-tableau: any $\mathscr{L}_2$-tableau in which, if a rule applies to a wff φ on a branch, there is some wff on some branch descending from φ which is the result of applying that rule. In general, the observations in the text show that completed $\mathscr{L}_2$-tableaux have infinitely long branches.

begin by applying UI to the first node; then we apply EW with a new name, 'b'. Since the branch is not closed, we apply UI to the top node with the new name; then we apply EW with a new name, 'c', and so on, *ad infinitum* (since we'll never run out of new names).

Figure 3.8 An infinitely growing finitely generated tableau.

$$\begin{array}{c} \forall x \exists y (R(x,y)) \\ \neg R(a,a) \\ | \\ \exists y (R(a,y)) \\ R(a,b) \\ \exists y (R(b,y)) \\ R(b,c) \\ \exists y (R(c,y)) \\ R(c,d) \\ \vdots \end{array}$$

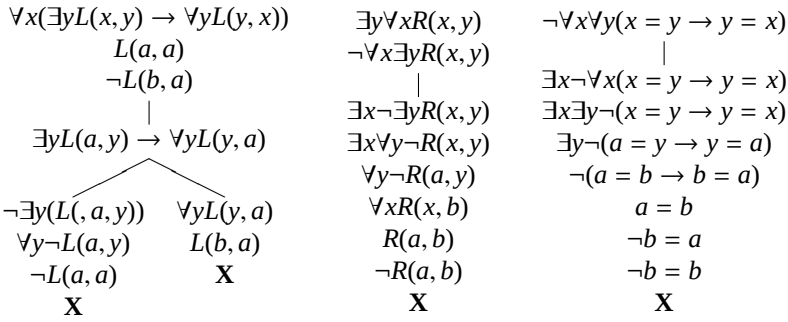
However, as the argument from '$\forall x \exists y (R(x,y))$' to '$R(a,a)$' is clearly invalid, this means that we do not have a test for invalidity any longer—just a test for validity. (To see why it is not valid, think of the interpretation of this argument as the domain of quantification being the natural numbers, and '$R(\mathbf{x},\mathbf{y})$' meaning '**y** is the successor of **x**': no number is its own successor, so the natural numbers provide a counterexample, unfortunately only an infinite counterexample, to this argument.) Of course, in this case we can *see* how the tableaux will continue to unfold, and we can give a story as to why the tableaux won't close in a finite time. But we cannot be assured of being able to see how an arbitrary tableaux will unfold. We must abandon our hopes of being able to use $\mathscr{L}_2$-tableaux as a *decision procedure* for theoremhood of $\mathscr{L}_2$. We shall return below (§3.10) to the question of whether we can use $\mathscr{L}_2$-tableaux as a partial decision procedure, or whether some other method might do better. It is worth noting immediately that, as an earlier remark suggested, $\mathscr{L}_2^\star$-tableaux will do no better than $\mathscr{L}_2$-tableaux here, because the lack of restriction on universal instantiation makes infinite descending chains easier, not harder, to obtain.

3.7.3 Examples of $\mathscr{L}_2$-Tableaux

Regardless of the results of the last section, we can still use $\mathscr{L}_2$-tableaux to prove *validity* of an argument, if it is valid, because $\mathscr{L}_2$-tableaux are a

fine test for inconsistency in the generating set. Again, an argument 'Ξ, therefore φ' is valid just when the set $\{\Xi, \neg\varphi\}$ generates a closed tableau. Let us consider some examples.

Figure 3.9 Some example $\mathscr{L}_2$-tableaux.



(a) Alma's narcissism inflames Barry. **(b)** The all-coveted object. **(c)** Symmetry of Identity.

1. 'All love all lovers, and Alma loves herself. Therefore, Barry loves Alma'. This is valid, as can be seen from the tableau in Fig. 3.9(a), where 'a' names Alma, 'b' names Barry, and 'L^2' is the predicate 'loves' (Jeffrey, 1991, 65).

2. 'Something is coveted by everyone, and therefore everyone covets something'. This is valid: see Fig. 3.9(b). Note the use of the negation rules.

3. 'If $x = y$, then $y = x$'. The tableau in Fig. 3.9(c) does use the special identity rule, and uses the identity-related means of branch closure.

4. The argument 'Everything is P; therefore at least one thing is P'. We should expect this to be invalid given our normal, empty-domain-allowing, rules; but we might expect this to be valid using the rule UI$^+$. This is exactly what appears: the tableau in Fig. 3.10(a) which uses the standard rules is unclosed, while in Fig. 3.10(b), the $\mathscr{L}_2^\star$ tableau using UI$^+$ closes.

5. 'There is some P, and there is at most one P; therefore there is exactly one P'. This is again, valid: see Fig. 3.11. Note that the identity rule has no special role; it is simply used as an ordinary predicate.

Figure 3.10 Empty and Non-empty domain quantifier tableau rules.

$$\begin{array}{c} \forall x P(x) \\ \neg\exists x P(x) \\ | \\ \forall x \neg P(x) \\ \textbf{?} \end{array}$$

(a) An open $\mathscr{L}_2$-tableau using UI.

$$\begin{array}{c} \forall x P(x) \\ \neg\exists x P(x) \\ | \\ \forall x \neg P(x) \\ \neg P(a) \\ P(a) \\ \textbf{X} \end{array}$$

(b) A closed $\mathscr{L}_2^\star$-tableau using UI^+.

3.7.4 Syntactic $\mathscr{L}_2$ Sequents

We now adapt our earlier sequent notation to the case of $\mathscr{L}_2$. If a set of sentences $\{\Xi, \neg\varphi\}$ is $\mathscr{L}_2$-syntactically inconsistent (Def. 8), we write

(3.8) $$\Xi \vdash_{\mathscr{L}_2} \varphi.$$

(From now on, with no risk of ambiguity, we drop the $\mathscr{L}_2$ subscript and just write $\vdash$.) So, for instance, Fig. 3.11 shows that

(3.9)
$$\exists x P(x), \forall x \forall y\, ((P(x) \wedge P(y)) \rightarrow x = y) \vdash \exists x (P(x) \wedge \forall y (P(y) \rightarrow x = y)).$$

As before, if a set of formulae Ξ is inconsistent, we write '$\Xi \vdash$'; if a single formula φ is inconsistent, we write $\vdash \neg\varphi$. Again, we prove

THEOREM 11 (DEDUCTION THEOREM FOR $\vdash_{\mathscr{L}_2}$). *$\Xi, \varphi \vdash_{\mathscr{L}_2} \psi$ iff $\Xi \vdash_{\mathscr{L}_2} \varphi \rightarrow \psi$.*

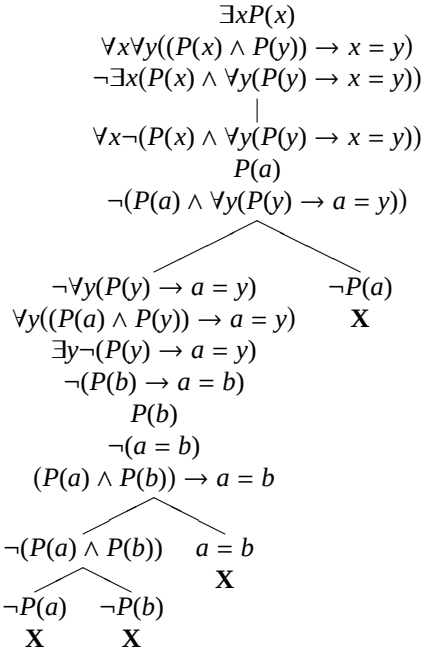
Proof. The earlier proof of Theorem 5 still holds because we have kept all the old tableau rules for $\rightarrow$, which are the only important rules in this proof. ■

$\mathscr{L}_2^\star$ Exactly similar definitions can be given of $\vdash_{\mathscr{L}_2^\star}$, and a deduction theorem also holds.

Exercises for §3.7

Exercise 3.7.1: Using the standard tableau rules (not including UI^+), show that the following arguments are valid:

Figure 3.11 Tableau proof with application to definite descriptions



1. $\exists x \forall y Rxy \vdash \forall x \exists y Ryx$;
2. $\forall x \forall y(Fx \rightarrow (Gy \leftrightarrow Rxy)), \exists x Gx \vdash \exists x Fx \rightarrow (\forall w \forall z((\neg Rwz \wedge Gz) \rightarrow \neg Fw))$;
3. $\forall x \forall y((Tay \rightarrow Txy) \rightarrow Txa) \vdash \exists x Txx$;
4. $\forall x(Tx \rightarrow \neg \exists y(y \neq a \wedge Fxy)) \vdash \forall x(Tx \rightarrow \exists y(Fxy)) \rightarrow \forall x(Tx \rightarrow Fxa)$;
5. $\forall x \forall y(x \neq y \rightarrow (Fx \leftrightarrow \neg Fy)) \vdash \neg \exists x_1 \exists x_2 \exists x_3 \exists x_4(((((x_1 \neq x_2 \wedge x_1 \neq x_3) \wedge x_1 \neq x_4) \wedge x_2 \neq x_3) \wedge x_2 \neq x_4) \wedge x_3 \neq x_4)$;
6. $\forall x \forall y((x = y \wedge Rxy) \rightarrow Sxx) \vdash \neg \exists x \forall y(Ryx \wedge \neg Syx)$;
7. $\forall x \forall y(x \neq y \rightarrow (Fx \vee Fy)) \vdash \forall z(\neg Fz \rightarrow \forall w(\neg Fw \rightarrow w = z))$.

Exercise 3.7.2: Earlier, I said 'if we have an $\mathscr{L}_2$-wff of the form '$\exists \mathbf{x}(\Phi(\ldots \mathbf{x} \ldots))$' appearing on a tableau, we can apply the EW rule infinitely many times, each time with a new name' (p. 123). Do we ever need to apply the EW rule more than once?

Exercise 3.7.3: Refer back to Exercise 3.6.4, and using the sentences in that exercise:

1. Use a tableau to prove that (1), (2), (3) and (4) entail (5). (Warning: this is somewhat involved.)

2. Show that (1), (2) and (3) do not entail (5) by providing a suitable countermodel.

Exercise 3.7.4: Translate the following arguments and check for validity in $\mathscr{L}_2$.

1. Everybody loves exactly one person; so if someone is loved by two people, someone is unloved.
2. Each person is happy only if their society be stable, and a stable society is a just society; so injustice must produce misery.
3. Nobody likes anyone who abuses them, nor (clearly) anyone they abuse; since Jim abuses anyone who doesn't abuse him, everyone dislikes Jim.
4. Every philosopher disagrees with every other philosopher on whether to accept at least one claim; moreover, every claim has been defended by some philosopher or other. And one can only accept or deny a given claim. So if there are only two claims, there are four philosophers.

Exercise 3.7.5: Decide, by appropriate means, whether the following tableaux are correct. If they are correct, prove this using a tableau; if they are incorrect, give a countermodel.

1. $\forall x \exists y Gxy, \neg \exists x \exists y (Gxy \wedge Ax) \vdash \neg \exists x Ax$.
2. $\forall x \exists y \forall z (Rxy \rightarrow (Gyz \wedge Hz)) \vdash \forall x \forall y (Rxy \rightarrow \exists z (Hz \leftrightarrow Gyz))$.
3. $\vdash (\forall x Fx \rightarrow \forall x (Fx \rightarrow \exists y (Fy \vee x \neq y)))$.
4. $\forall x (Fx \rightarrow Gx) \vdash (\exists y (Fy \wedge Gy) \vee \neg \exists z Fz)$.

Exercise 3.7.6: Using the standard rules, is $\forall x (x = x) \vdash a = a$ a correct sequent? Is $a = a \vdash \exists x (x = x)$ a correct sequent? What does this entail about '$\vdash$' in $\mathscr{L}_2$ tableaux?

Exercise 3.7.7: Is it a problem for the completeness of the tableaux technique that the sequent $\forall x \exists y Rxy \vdash Raa$ is incorrect and yet the tableaux for its counterexample set does not close?

Exercise 3.7.8: Is the sentence 'Bill cuts the hair of all and only those who don't cut their own hair' inconsistent? What about the sentence 'Someone cuts the hair of all and only those who don't cut their own hair'? Test these sentences using the standard rules; comment on any differences between them. Does using UI^+ make a difference?

3.8 Structures and Possible Situations for $\mathscr{L}_2$

Having given a system of tableaux based on $\mathscr{L}_2$, it is now time to try and give rules for understanding the meaning of $\mathscr{L}_2$-wffs in a given possible situation. We will of course be guided by our informal remarks on relations, predicates, and designators above, but this section should be regarded as superseding those former sections. (Almost all of the following remarks apply to $\mathscr{L}_2^\star$ also.)

3.8.1 Models and Structures

As we are trying to create some situations in which the sentences of $\mathscr{L}_2$ are assigned meanings, we begin by making precise our informal remarks about meaning from §3.4.1. A domain for our purposes is just a set of objects. Any set of objects, even the empty set $\emptyset$, is a domain.[12] We call the members of the domain *individuals*. We write the domain of a certain situation $\mathscr{S}$ as $\mathbf{Dom}_{\mathscr{S}}$.

Properties as sets of individuals We also remarked in that same section above that an extensional property is regarded as simply the set of individuals in a given situation which satisfy some predicate. This set was regarded as the meaning of the predicate in that situation.

The set of all properties in a given situation is simply the set of all sets of individuals of the domain of that situation. In other words, the set of all properties $\mathbf{Prop}_{\mathscr{S}}$ of a given situation is the *powerset* of the domain: $\wp(\mathbf{Dom}_{\mathscr{S}})$ (page 155).

The set of all *n*-place relations in a given situation is the power set of the set of all ordered *n*-tuples of individuals of the model.[13] This again is easily generated just from the domain $\mathbf{Dom}_{\mathscr{S}}$.

Models A *model* or structure $\mathscr{S}$ is simply a domain with a set of associated properties and relations.[14] Since the set of all properties of every set is simply just the powerset of the domain, and the set of all relations is nearly as easily obtained, we need not specify those separately, if we are content with just the most unrefined conception of a model. But that is not the only kind of model there is.

More refined models One might think that some properties are better than others: some sets of individuals are *natural* sets, that hang together because of a genuine resemblance between all of their members. So, for instance, the set of all red things in a situation is a more coherent set than the set composed of all the red things and all the square things. Then to specify a model

[12] Here lies the difference from $\mathscr{L}_2^\star$: in $\mathscr{L}_2^\star$, we insist that the domain have at least one member. This doesn't matter for soundness (Theorem 14), but it does matter when proving completeness (§3.11.2).

[13] So, for example, the set of all binary relations is the set of sets of 2-tuples, or equivalently, the powerset of the Cartesian product of the domain with itself, i.e. $\wp(\mathbf{Dom}_{\mathscr{S}} \times \mathbf{Dom}_{\mathscr{S}})$.

[14] For more on models see Hodges (1997).

in this more refined sense will involve specifying a domain $\mathbf{Dom}_{\mathscr{S}}$, and specifying a subset of $\wp(\mathbf{Dom}_{\mathscr{S}})$ to count as the natural properties of that model. We then need to specify which relations are to count as natural relations, sets of ordered pairs that hang together because of some resemblance in the relation of their members. We are, in practice, mostly interested in the refined version, since we are interested in structures that make our sentences true because in the model there exist some items that are really related in some genuine manner. But logical truths will be those sentences that are true in all structures, under all interpretations, so we shall have to consider all kinds of models, refined and unrefined alike.

Relations between models If we have two models, $\mathscr{R}$ and $\mathscr{S}$, given a domain and a set of properties and relations for each, we say that $\mathscr{R}$ is *embeddable* into $\mathscr{S}$ just in case (i) There is a function f such that $f : \mathbf{Dom}_{\mathscr{R}} \mapsto \mathbf{Dom}_{\mathscr{S}}$; and (ii) If $\mathbf{P}$ is an n-place relation in $\mathscr{R}$, then $f(\mathbf{P})$ (the image of each member of $\mathbf{P}$ under f) is a n-place relation in $\mathscr{S}$.[15] Such a function f is called an *embedding*. If $\mathscr{S}$ is embeddable into $\mathscr{R}$ also, we say that $\mathscr{R}$ and $\mathscr{S}$ are *isomorphic*. The existence of an isomorphism between two domains of the same size is relatively trivial if one allows *arbitrary* relations; if we restrict our attention to *natural* relations and properties, it is more difficult.

3.8.2 Valuations and Interpretations

As before, we interpret wffs *compositionally*: showing how the truth value of a complex sentence in a model depends on the semantic values of the constituent parts. Only now we have smaller parts that do not have truth values: so there must be some other things to serve as semantic values.

Semantic Values The candidates are obvious. The semantic value of a designator in a model is some individual referent of the designator; the semantic value of a predicate in a model is the property or relation which contains all the things which satisfy that predicate. We have a unified term to describe the semantic value of any item: its *extension*. So when we come to interpret an quantifier-free wff in a model, we need to find (i) individual members of the domain to be the extension of —to be designated by—the designators; and (ii) properties and relations in the situation to be the extensions of—to be the meanings of—the predicates.

[15] i.e. $f(\mathbf{P})$ is defined as the set $\{\langle x_1, \ldots, x_n\rangle : \exists\langle y_1, \ldots, y_n\rangle(\langle y_1, \ldots, y_n\rangle \in \mathbf{P} \wedge \langle x_1, \ldots, x_n\rangle = \langle f(y_1), \ldots, f(y_n)\rangle)\}$.

Interpretations An assignment of designators to individuals and predicates to properties and relations is called an *interpretation*. So a wff of $\mathscr{L}_2$ is not true or false in a model, but *only* under some interpretation of the parts of that wff in that model. That is, we can consider the model which has the domain $\{\mathbf{i}_1, \mathbf{i}_2, \mathbf{i}_3\}$, and one property, $\mathbf{P} = \{\mathbf{i}_1, \mathbf{i}_3\}$. If we have the sentence of $\mathscr{L}_2$, '$P(a) \wedge P(b)$', this sentence is not true or false in this situation, since we have to decide what 'a' designates, and so on. It is true if 'a' designates $\mathbf{i}_1$ and 'b' designates $\mathbf{i}_3$, but false if 'a' designates $\mathbf{i}_2$. A combination of a model and an interpretation will be called a *situation*: for our purposes, sentences are true or false in a model only after we've assigned extensions to all the parts of the sentences, and this combination of model and interpretation is a situation. One consequence of this is that two situations can be different even if all the same objects and properties and relations exist in both situations—for the interpretation may be different.

Example Let us consider the sentence of $\mathscr{L}_2$ '$R(a, b)$'. This sentence is true in a situation if in that situation there is an individual $\mathbf{i}_a$ designated by 'a', an individual $\mathbf{i}_b$ designated by 'b', and a relation **Rel** that is the extension of 'R', such that $\langle \mathbf{i}_a, \mathbf{i}_b \rangle \in \mathbf{Rel}$. A less abstract example might be: The sentence 'Jeff is married to Elyse' can be interpreted as $M(j, e)$, which is true just in case 'j' has as extension the person Jeff, 'e' has as extension the person Elyse, and 'M' has its extension the 'Married' relation where $\langle \text{Jeff}, \text{Elyse} \rangle \in \text{Married}$.

A formal condition A sentence '$P^n(a, b, c, \ldots)$' (a predicate followed by a sequence of n designators) is true in a situation (i.e. in a model, under an interpretation) if the situation assigns $m \leqslant n$ individuals $\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots$ to the designators, and the ordered n-tuple $\langle \mathbf{a}, \mathbf{b}, \mathbf{c} \ldots \rangle$ is a member of the relation **Rel** that is the extension of 'P^n'.

Identity The identity relation '=' is a logical relation, and hence has a constant interpretation: in every situation $\mathscr{S}$, it has as its extension the set $\mathbf{I}_{\mathscr{S}} = \{\langle x, x \rangle : x \in \mathbf{Dom}_{\mathscr{S}}\}$. An identity statement '$a = b$' is true in a situation just in case the extension of 'a' is the extension of 'b'.[16]

[16]The observant reader will have noted that any isomorphism maps the identity relation to the identity relation, which is one reason for thinking it has constant interpretation.

Quantified sentences The interpretation of a quantified sentence seems fairly easy too. There is no item in the domain to be the 'extension' of the quantifier. But the role of Herbrand sentences might appear to give us a clue as to the construction of truth conditions for quantified claims, as follows:

- A sentence of the form '$\forall x\Phi(x)$' is true in a situation $\mathscr{S}$ just in case for every $i \in \mathbf{Dom}_{\mathscr{S}}$, if $\mathbf{P}$ is the extension of 'Φ', then $i \in \mathbf{P}$.
- A sentence of the form '$\exists x\Phi(x)$' is true in a situation $\mathscr{S}$ just in case for some $i \in \mathbf{Dom}_{\mathscr{S}}$, given $\mathbf{P}$ is the extension of 'Φ', then $i \in \mathbf{P}$.

Complications However, these conditions don't work in some cases: for instance, what if the sentence is $\forall x\exists y(L(x,y))$? Our condition says that this sentence is true just if every individual is in the extension of '$\exists y(L(\ldots,y)$': but this itself is *not* a basic predicate that has an extension in $\mathscr{S}$. What we need is that every instance of the universal quantification is true in $\mathscr{S}$, and therefore to show how universally quantified sentences depend on the truth of their less complexly quantified instances. But there is a complication here too, since not every individual in a situation need have a name in the language—what if, for example, there are more individuals than names?[17] Even in this case, there can still be true universally quantified sentences. These complications are resolved as follows: let $\mathscr{S}'$ be a situation which is the same as $\mathscr{S}$ (same model, and mostly the same interpretation) except that some name 'a' of $\mathscr{L}_2$ which may not be assigned an extension in $\mathscr{S}$, is assigned an extension in $\mathscr{S}'$ ($\mathscr{S}'$ is an *extension* of $\mathscr{S}$). A sentence $\forall x\Phi(x)$, where α does not occur in the expression $\Phi(x)$, is true in $\mathscr{S}$ just in case for every individual $i \in \mathbf{Dom}_{\mathscr{S}}$, if '$a$' names i in some extended situation $\mathscr{S}'$ then '$\Phi(a)$' is true in $\mathscr{S}'$. Similarly for the existential quantifier. The extended situation differs from the original at most in what it assigns as the extension of α, so in particular it keeps all other predicates and names with the same interpretation.

Objectual and Substitutional Quantification In technical terms, we insist on the *objectual* reading of the quantifier: what makes a universally quantified sentence '$\forall x\Phi(x)$' true in a given interpretation is that every individual in the domain, when named in some possibly extended situation by some name unused name 'c' (not in 'Φ'), occurs in a true expression of the

[17] As is the case with the real numbers, for example, where a famous result known as *Cantor's theorem* shows that there are more real numbers than possible names for them in any language with only countably infinitely many wffs, like $\mathscr{L}_2$ and indeed all natural languages.

form '$\Phi(c)$' in the extended situation. There is, however, another way of understanding quantification, where instead of thinking of objects that would, when named, make a formula true, we think directly that what makes a quantified claim true is that all of its instances are true. This runs with the intuition that universal quantification is like infinite conjunction; and existential quantification is like infinite disjunction. For the universal quantifer, this interpretation of the quantifers gives the following simple truth conditions: '$\forall xA$' is true in a situation $\mathscr{S}$ iff for all designators d, the if A' is the result of substituting d for all occurrences of x in A, 'A'' is true in S. For obvious reasons, this is called the *substitutional* interpretation of the quantifiers. When there are no unnamed objects, the two theories of the quantifiers coincide. When there are some unnamed objects, they differ radically, and in a way that clearly intuitively seems to favour the objectual reading. But some authors at least have defended the substitutional reading, particularly in the context of nominalist positions about various abstract objects (the idea being, if all one does is substitute designators for variables, then we can have complex designators actually serve proxy for various disreputable objects, and still express quantified truths). But this issue quickly gets technical and is of dubious philosophical motivation in any case; for more, see Kripke (1976).

Satisfaction and Open Sentences Again If we had permitted open sentences to count as wffs (as discussed earlier, p. 118), we would have to have done things differently. Rather than appeal to the simple truth of the sentences in the extended model, we would have had to be more rigorous than we have been about the notion of satisfaction. Up until now I've talked relatively intuitively about some objects satisfying a predicate just in case the predicate is true of them when named. In the case where there are not enough names to go around, or in which the object simply isn't named, we need to talk of an object satisfying an open sentence. In this case we define an additional piece of machinery, a *variable assignment*, which assigns to each variable of $\mathscr{L}_2$ an object in the domain. A situation will now need to be supplemented with a variable assignment, and an open sentence $\Phi(\ldots\mathbf{x}\ldots)$ will be true in $\mathscr{S}$ just in case the variables $\mathbf{x}$ in Φ under variable assignment $\mathscr{A}$ denote objects which fall in the extension of Φ. A closed formula (one which is well-formed and not open) will be true under a variable assignment just in case it is true; so we can now define truth in a situation as truth in that situation under all variable assignments; obviously no open sentence will ever be true. But they will be true under a variable assignment; so now we

could just say of a quantified sentence '$\forall x\Phi$' that it is true in a situation just in case 'Φ' is true under all variable assignments to x in that situation. This isn't the only way to implement this suggestion; for another way, involving an inductive definition of satisfaction on atomic sentences, and the idea of truth as satisfiability on all assignments, see Bostock (1997, 86–9).

Valuations extended to $\mathscr{L}_2$ We can add these new rules to our previous discussion of Boolean valuation functions (Table 2.14), to get *$\mathscr{L}_2$-valuation functions*. We summarise the new rules, and the old (omitting '$\leftrightarrow$' and '|'), in Table 3.4. Remember that we are dealing with situations here, not just models. Note the complex conditions on quantified sentences; note also the special (but entirely orthodox) treatment of the logical predicate '='.[18]

3.8.3 Semantic $\mathscr{L}_2$ Sequents

As before (§2.9), we define semantic sequents for $\mathscr{L}_2$. If a given $\mathscr{L}_2$-wff φ is assigned $\top$ by every valuation function $v_{\mathscr{S}}$—in other words, if it is true in every situation—then we write

(3.10) $$\vDash_{\mathscr{L}_2} \varphi.$$

If φ is true in every situation in which all the members of Ξ are true, we write

(3.11) $$\Xi \vDash_{\mathscr{L}_2} \varphi.$$

Again, we can prove

Theorem 12 (Deduction theorem for $\vDash_{\mathscr{L}_2}$). $\Xi, \varphi \vDash_{\mathscr{L}_2} \psi$ *iff* $\Xi \vDash_{\mathscr{L}_2} \varphi \rightarrow \psi$. (The proof is exactly as for Theorem 2.)

Manipulating Sequents All the rules we introduced above for manipulating sequents still hold of $\mathscr{L}_2$-sequents (§§2.9.1–2.9.2): contraction, permutation, *&c*. The justifications we gave of them will still hold, too, if we simply replace 'formula' with '$\mathscr{L}_2$-wff' and replace 'structure' by '$\mathscr{L}_2$-situation'.

[18] Though our language $\mathscr{L}_2$ does not include any 0-place predicates (see Table 3.3), we could in principle include them. I leave the rules for doing so to an exercise.

Table 3.4 Rules on $\mathscr{L}_2$-valuation function $v_{\mathscr{S}}$ for $\mathscr{L}_2$-wffs in situation $\mathscr{S}$.

Basic $\mathscr{L}_2$-wffs

$v_{\mathscr{S}}(P^n(a_1, \ldots, a_n)) = \top$	iff	**P** is the extension of 'P^n' in $\mathscr{S}$, each designator 'a_j' is assigned an individual $i_j \in \mathbf{Dom}_{\mathscr{S}}$, and $\langle i_1, \ldots, i_n \rangle \in \mathbf{P}$.
$v_{\mathscr{S}}(a_1 = a_2) = \top$	iff	$i_j \in \mathbf{Dom}_{\mathscr{S}}$ is the extension of 'a_j' and $\langle i_1, i_2 \rangle \in \mathbf{I}_{\mathscr{S}}$; i.e. i_1 is i_2.

Complex $\mathscr{L}_2$-wffs

$v_{\mathscr{S}}(\neg\varphi) = \top$	iff	$v_{\mathscr{S}}(\varphi) = \bot$.
$v_{\mathscr{S}}(\varphi \wedge \psi) = \top$	iff	$v_{\mathscr{S}}(\varphi) = \top$ and $v_{\mathscr{S}}(\psi) = \top$.
$v_{\mathscr{S}}(\varphi \vee \psi) = \top$	iff	$v_{\mathscr{S}}(\varphi) = \top$ or $v_{\mathscr{S}}(\psi) = \top$ (or both).
$v_{\mathscr{S}}(\varphi \rightarrow \psi) = \top$	iff	$v_{\mathscr{S}}(\varphi) = \bot$ or $v_{\mathscr{S}}(\psi) = \top$ (or both).

(For the following two conditions, 'a' is a designator which may have no extension in $\mathscr{S}$, '$\Phi(x)$' is a formula with x free and in which 'a' does not appear, and $\mathscr{S}'$ is some extended situation from $\mathscr{S}$: just like $\mathscr{S}$ except possibly in its assignment of an individual i as the extension of 'a'.)

$v_{\mathscr{S}}(\forall x(\Phi(x))) = \top$	iff	For every individual $i \in \mathbf{Dom}_{\mathscr{S}}$, if i is the extension of 'a' in some extended situation $\mathscr{S}'$, then $v_{\mathscr{S}'}(\Phi(a)) = \top$.
$v_{\mathscr{S}}(\exists x(\Phi(x))) = \top$	iff	For at least one individual $i \in \mathbf{Dom}_{\mathscr{S}}$, i is the extension of 'a' in some extended situation $\mathscr{S}'$, and $v_{\mathscr{S}'}(\Phi(a)) = \top$.

Entailment and Open Sentences If a formula '$\forall x\Phi(x)$' is true in $\mathscr{S}$, then if we adopt our variant account of semantics from p. 133, the open sentence $\Phi(x)$ will be true (because, of course, it will be true under every variable assignment). This account, thus, makes it true that an open sentence is equivalent to its *universal closure*, the result of binding all of its free variables with universal quantifiers. But this has some odd views: for instance, if 'Fx' is equivalent to '$\forall xFx$', then $Fx \vDash \forall xFx$ will be correct. If the deduction theorem holds, then $\vDash Fx \rightarrow \forall xFx$ is also a correct sequent. But since, by universal closure, '$Fx \rightarrow \forall xFx$' is equivalent to '$\forall x(Fx \rightarrow \forall xFx)$', then $\vDash \forall x(Fx \rightarrow \forall xFx)$ would be correct—but it certainly should not be (it is false, for example, if not everything is F). This paradoxical result—either something obviously invalid is correct, or the deduction theorem fails—seem to me, and others (Bostock, 1997, 90), to count strongly against the notion that we should extend truth to anything other than closed sentences.

Exercises for §3.8

Exercise 3.8.1: Is the model with domain $\{0, 1, 2, \ldots\}$ and the greater-than relation $>$ as the only non-logical relation isomorphic to the model with domain $\{0, -1, -2, \ldots\}$ and the less-than relation $<$ as the one non-logical relation?

Exercise 3.8.2: Explain the difference between truth in a model and truth in a situation. Are any sentences of $\mathscr{L}_2$ true in some model without being theorems?

Exercise 3.8.3: Explain why this straightforward semantic clause for existentially quantified sentences is inadequate:

$$v_{\mathscr{S}}(\exists x\Phi(x)) = \top \text{ iff } \text{ for some } i \in \mathbf{Dom}_{\mathscr{S}}, \text{ if '}a\text{' denotes } i \text{ in } \mathscr{S}, v_{\mathscr{S}}(\Phi(a)) = \top.$$

Exercise 3.8.4: According to the valuation function given in Table 3.4, can there be any true sentence containing a designator that has no referent—a so-called 'empty name'? Is the sentence 'Superman is the paradigm superhero' true? Does that show that we should give a different clause for truth of basic sentences? Finally, if we allow empty names into our language, is the $\mathscr{L}_2$ tableau rule UI a good rule? If we allow empty names into our language, should a tableau automatically close if $\neg a = a$ appears on it?

Exercise 3.8.5: Though we cannot prove it here, compactness holds for $\mathscr{L}_2$: that is, for every infinite set Γ of formulae of $\mathscr{L}_2$, and for every formula φ of $\mathscr{L}_2$, $\Gamma \vDash \varphi$ iff there is a finite subset $\Delta \subset \Gamma$ such that $\Delta \vDash \varphi$.

1. Specify a formula of $\mathscr{L}_2$ that means 'there are at least n things', where n is an arbitrary positive integer.
2. Specify a set of formulae of $\mathscr{L}_2$ that is true in all and only structures with infinite domains, but is such that each finite subset is true in some structure

with a finite domain.

3. Show that there is no formula of $\mathscr{L}_2$ that is true in a structure $\mathscr{S}$ iff $\mathbf{Dom}_{\mathscr{S}}$ is infinite.

Exercise 3.8.6: How should our semantics be extended to treat zero-place predicates?

3.9 Soundness

Soundness and Completeness Again Having defined two turnstiles, the syntactic turnstile '$\vdash_{\mathscr{L}_2}$' and the semantic turnstile '$\models_{\mathscr{L}_2}$', we again try to see if they can be brought into correspondence, just as we proved for $\mathscr{L}$ in Theorems 6 and 7. Those theorems together show that:

(3.12) $\qquad \vdash_{\mathscr{L}} \varphi \quad \text{iff} \quad \models_{\mathscr{L}} \varphi.$

(The deduction theorems show that proving claim 3.12 is sufficient to prove the total correspondence of the two turnstiles.) We wish to show the corresponding result for $\mathscr{L}_2$. We shall here prove the soundness direction, that if $\vdash_{\mathscr{L}_2} \varphi$ then $\models_{\mathscr{L}_2} \varphi$. We shall make some remarks about the other half of the proof—completeness of $\mathscr{L}_2$—in §3.11.2 below.

In proving the soundness theorem for $\mathscr{L}$, we made use of subsidiary claim to the effect that if every wff on a branch can be assigned $\top$ by some valuation function, then some extension of that branch by the tableaux rules can also have every wff assigned $\top$ (Lemma 1). We start by proving an analogous lemma for $\mathscr{L}_2$.

LEMMA 4. *Let T be a $\mathscr{L}_2$-tableau generated by $\{\neg\varphi\}$, and let there be a situation $\mathscr{S}$ such that $v_{\mathscr{S}}(\neg\varphi) = \top$. Then there exists a branch B on T, and a situation $\mathscr{S}'$, such that for all $\mathscr{L}_2$-wffs $b \in B$, $v_{\mathscr{S}'}(b) = \top$.*

Proof. We prove this just as in the case of Lemma 1, by mathematical induction on length of branches. *Base:* The base case is where B has only one wff on it, namely $\neg\varphi$ itself. Then trivially, we let $\mathscr{S} = \mathscr{S}'$, then $v_{\mathscr{S}'}(\neg\varphi) = \top$.

Induction Step: We assume that the lemma holds for a tableau T and some branch B. We now show that, for some tableaux T' identical to T except that one branch B^+ on T' extends B on T by application of some $\mathscr{L}_2$-tableaux rule, the lemma holds. There are several cases, depending on which rule we apply to B to get B^+:

$\mathscr{L}$ **list and branch rules** Since we have adopted the valuation rules and tableaux rules for $\mathscr{L}$wholesale, the proof of this part of the lemma follows exactly the form of the proof we gave of Lemma 1 (page 67), except that we now deal with $\mathscr{L}_2$-wffs and $\mathscr{L}_2$-valuations, and we take $\mathscr{S}'$ to be $\mathscr{S}$ throughout.

Universal Instantiation B^+ follows from B by an application of the universal instantiation (UI) rule, so B^+ terminates with a wff '$\Phi(a)$'. So we know, by the induction hypothesis, that some wff '$\forall x\Phi(x)$' is on B^+ and true in $\mathscr{S}$, and we also know that some other wff '$\psi(\ldots a\ldots)$' appears on B^+ and is true in $\mathscr{S}$. So (i) we know that 'a' has an extension in $\mathscr{S}$ (hence $\mathscr{S}' = \mathscr{S}$); and (ii) we therefore know by the valuation rules (Table 3.4) that the truth of the universally quantified wff entails that for every individual $i \in \mathbf{Dom}_{\mathscr{S}}$, if i is the extension of 'a' in $\mathscr{S}$, then $v_{\mathscr{S}}(\Phi(a)) = \top$, as desired.

Existential Witnessing B^+ follows from B by an application of the existential witnessing (EW) rule. So B^+ terminates with a wff of the form '$\Phi(b)$', where 'b' does not appear anywhere on B. Then we know that there is an extended situation $\mathscr{S}'$, which does not differ from $\mathscr{S}$ except that in $\mathscr{S}'$ some individual $\mathbf{b}$ is the extension of 'b', and that $v_{\mathscr{S}'}(\Phi(b)) = \top$. Since $\mathscr{S}'$ differs from $\mathscr{S}$ only in assigning a member of $\mathbf{Dom}_{\mathscr{S}}$ as the extension of 'b', and 'b' appears nowhere on B, we know that for all $\varphi \in B$, $v_{\mathscr{S}'}(\varphi) = \top$. Hence this shows the case, as $\mathscr{S}'$ assigns $\top$ to all wffs on B^+.

Negated Existential '$\neg\exists x\Phi(x)$' appears on B, and (by the induction hypothesis) is true in $\mathscr{S}$. Then $v_{\mathscr{S}}(\exists x\Phi(x)) = \bot$, and hence there is not a single $\mathbf{c} \in \mathbf{Dom}_{\mathscr{S}^\dagger}$ such that $\mathbf{c}$ is the extension of any designator 'c' and $v_{\mathscr{S}^\dagger}(\Phi(c)) = \top$, i.e. all such claims are false. Hence choose some arbitrary designator 'a' that appears on B: since for any such designator, $v_{\mathscr{S}^\dagger}(\neg\Phi(a)) = \top$, we know by the valuation rules that $v_{\mathscr{S}}(\forall x\neg\Phi(x)) = \top$, as desired. Let $\mathscr{S}' = \mathscr{S}$ and the case is proved.

Negated Universal '$\neg\forall x\Phi(x)$' appears on B, and (by the induction hypothesis) is true in $\mathscr{S}$. Then $v_{\mathscr{S}}(\forall x\Phi(x)) = \bot$, and hence there exists an $\mathbf{d} \in \mathbf{Dom}_{\mathscr{S}^\ddagger}$ such that $\mathbf{d}$ is the extension of some designator 'd' and $v_{\mathscr{S}^\ddagger}(\Phi(d)) = \bot$. Hence for at least one individual, the extension of 'd', $v_{\mathscr{S}^\ddagger}(\neg\Phi(d)) = \top$, hence by the valuation function $v_{\mathscr{S}}(\exists x\neg\Phi(x)) = \top$. Let $\mathscr{S}' = \mathscr{S}$ and the case is proved.

Identity B^+ extends B by adding a wff '$\Phi(d)$' where '$\Phi(c)$' and '$c = d$' (or its converse) appear on B and are true. Hence the extension **d** in $\mathscr{S}$ of 'd' is also the extension in $\mathscr{S}$ of 'c', and hence if $\mathbf{d} \in P_\varphi$, then $v_{\mathscr{S}}(\Phi(c)) = \top$ iff $v_{\mathscr{S}}(\Phi(d)) = \top$. Obviously the same reasoning applies to both left and right identity rules. Let $\mathscr{S}' = \mathscr{S}$ and the case is proved.

That completes the induction step. ■

Note that this follows pretty quickly:

LEMMA 5. *Let T be a $\mathscr{L}_2^\star$-tableau generated by $\{\neg\varphi\}$, and let there be a situation $\mathscr{S}$ such that $v_{\mathscr{S}}(\neg\varphi) = \top$. Then there exists a branch B on T, and a situation $\mathscr{S}'$, such that for all $\mathscr{L}_2^\star$-wffs $b \in B$, $v_{\mathscr{S}'}(b) = \top$.*

Proof. We need only replace the case of UI in the proof of Lemma 4 with the following modified condition:

Universal Instantiation B^+ follows from B by an application of the universal instantiation (UI) rule, so B^+ terminates with a wff '$\Phi(a)$'. So we know, by the induction hypothesis, that some wff '$\forall x\Phi(x)$' is on B^+ and true in $\mathscr{S}$. So we therefore know by the valuation rules (Table 3.4) that the truth of the universally quantified wff entails that for every individual $\mathbf{a} \in \mathbf{Dom}_{\mathscr{S}'}$, if **a** is the extension of 'a' in $\mathscr{S}'$, then $v_{\mathscr{S}'}(\Phi(a)) = \top$, as desired.

That combines with the other conditions to show the lemma. ■

We are now in a position to prove the soundness theorem for $\mathscr{L}_2$.

THEOREM 13 ($\mathscr{L}_2$ SOUNDNESS). *If $\vdash_{\mathscr{L}_2} \varphi$ then $\models_{\mathscr{L}_2} \varphi$.*

Proof. We prove the contrapositive. Assume that $\not\models_{\mathscr{L}_2} \varphi$. Then there is at least one situation $\mathscr{S}$ where $v_{\mathscr{S}}(\neg\varphi) = \top$. By Lemma 4, there is a $\mathscr{L}_2$-tableau T generated by $\{\neg\varphi\}$ such that for some branch B on T, for all $b \in B$, there is a situation $\mathscr{S}'$ such that $v_{\mathscr{S}'}(b) = \top$. Since $v_{\mathscr{S}'}$ obeys Boolean negation rules (Table 3.4), we know that if 'ψ' $\in B$ then '$\neg\psi$' $\notin B$. Hence B cannot be closed, and T is not closed, so '$\neg\varphi$' is not syntactically inconsistent. But that means $\not\vdash_{\mathscr{L}_2}$, and we have shown the theorem. ■

We can also prove:

THEOREM 14 ($\mathscr{L}_2^\star$ SOUNDNESS). *If $\vdash_{\mathscr{L}_2^\star} \varphi$ then $\models_{\mathscr{L}_2^\star} \varphi$.*

Proof. Left as an exercise for the reader. ■

And also:

Theorem 15. *If* $\Xi \vdash \varphi$ *then* $\Xi \vDash \varphi$.

Proof. Left as an exercise for the reader. ■

Exercises for §3.9

Exercise 3.9.1: Prove Theorem 14 (hint: use Lemma 5).

Exercise 3.9.2: Prove Theorem 15 (you may use earlier results).

3.10 Undecidability

We showed that tableaux did not provide a general decision procedure for predicate logic in §3.7.2. This is because some *consistent* sets of sentences do not yield a complete yet open tableaux: so if we built a machine to mechanically apply the tableaux rules to a given input set of sentences, and halt with an answer of 'Consistent' or 'Inconsistent', we have the problem that for some inputs, the machine will keep going forever and hence will not yield a yes or no answer in a finite time. That means (§2.13.2) there is no effective procedure for consistency in $\mathscr{L}_2$ (and $\mathscr{L}_2^\star$). Indeed, there is no hope for another system of tableaux or logic that is sound and complete for $\mathscr{L}_2$: no such system can be decidable. We cannot prove this result here, because it involves getting a lot clearer about the concept of an effective procedure than we have been—see Jeffrey (1991, ch. 7–8) and, for more mathematical detail, Boolos *et al.* (2003).

Effective Positive Test for Inconsistency Yet things are not completely hopeless. *If* a set of sentences is *inconsistent*, the machine *will* halt in a finite time and tell us that the set was inconsistent. That is, while no test will tell us if a set is consistent, a test will halt in a finite time and tell us that the set was inconsistent (the problem arises when we are waiting for the machine to halt, since we don't know whether the set is consistent and we will be waiting forever, or if the set is inconsistent and the finite time has not yet elapsed).

We can see this informally if we consider infinite descending tableaux (§3.7.2). We could only have generated the infinite chain because no inconsistency appeared in a finite time, hence we were expected to continue to

apply to UI rule to the new names thrown up by the EW rule. However, any inconsistency will appear after some finite number of designators have been introduced, and hence we can be sure that the inconsistency will appear at some stage. It is only those cases where we can continually introduce new designators without any possibility of inconsistency or closure that we have the problem of undecidability.

Weaker results Two weaker systems of logic can be decided however:

1. *Monadic* $\mathscr{L}_2$ If we restrict ourselves to $\mathscr{L}_2$-wffs which feature only monadic (1-place) predicates, the resulting system is decidable. That is, there is an effective procedure for checking consistency or inconsistency of a set of $\mathscr{L}_2$-wffs in which only monadic predicate letters appear. We can see this informally as follows: if we only ever apply the EW rules a finite number of times, then we will only be able to apply the UI rules a finite number of times. Hence after a finite number of times we can only apply the non-quantifier rules, and we know that those rules are decidable in a finite time. And we can show that applying the EW rule only once to each existentially quantified wff is sufficient to establish (in)consistency of the generating set; so we know that there is a decision procedure for (in)consistency of the generating set, since the finite time for the quantifier rules, added to the finite time for the non-quantifier rules, gives a finite time (Boolos *et al.*, 2003, ch. 21).

2. *Single-quantifier* $\mathscr{L}_2$ If we restrict ourselves to wffs which contain only one quantifier, (in)consistency is decidable. Again, we can reasonable only apply the UI or EW a finite number of times, and combined with the decidability of $\mathscr{L}$, that yields the decidability of this system (Jeffrey, 1991, 55).

Exercises for §3.10

Exercise 3.10.1: Does the fact that *we* can recognise that the tableau in Fig. 3.8 will never close show that we have the capability to decide questions that are not answerable by an effective procedure? Could this mean that $\mathscr{L}_2$ really is decidable, by us?

3.11 Further Developments

3.11.1 Terms and Functions

We could, and many logics do, augment their languages with *term-forming operators*. These are phrases that attach to a designator and, instead of forming a sentence as a predicate would, form another designator. So, for instance, 'the father of' attaches to a designator, say 'Albert', and yields another designator, 'the father of Albert'. Such term-forming operators are *functions* in the technical sense: see Appendix §B.4.

Terms and Relations It turns out that nothing is gained by this addition except convenience. Let us consider an arithmetical function: '+' takes two designators and yields a designator which designates the *sum* of the referents of the other designators. Let s be the function which yields the *successor* of a number. Then the following is true of natural numbers:

$$\forall x \forall y (s(x + y) = s(x) + y). \tag{3.13}$$

We express the same sentence using the three-place relation '$\Sigma(\mathbf{x}, \mathbf{y}, \mathbf{z})$', meaning '$\mathbf{x}$ is the sum of $\mathbf{y}$ and $\mathbf{z}$', as follows:

$$\forall x \forall y \exists z \exists w \exists v (\Sigma(z, x, y) \wedge \Sigma(w, z, 1) \wedge \Sigma(v, x, 1) \wedge \Sigma(w, v, y)). \tag{3.14}$$

As can be readily seen, the second sentence is much more cumbersome, but in principle we need no more than relations to express any sentence which involves terms and functions.

Functions and Designators Two further points may be made concerning functions. (i) What about a zero-place function? This yields a designator when supplied with no designators: i.e. is a designator already. So our old designators are a special case of terms and functions. (ii) Definite descriptions are a weird kind of function: they seem to take a predicate and yield a term. That is, 'the F' is a term (it can attach to a predicate, and yield a declarative sentence), but the value of the term seems to be a function of the predicate 'F'. We can, if we choose, liberalise the notion of function to allow this very special term forming operator, noting of course that predications like 'the F is G' stand for complex quantified sentences. Again there is no gain in expressive power of the language, but perhaps a gain in convenience. But if we note that 'any F' also seems to be a term (for it can also

attach to a predicate and yield a declarative sentence), we might be tempted to view all quantification as a species of function... but this would not lead to very convenient results for us.

3.11.2 Completeness

The completeness theorem says

THEOREM 16 ($\mathscr{L}_2$ COMPLETENESS). *If* $\models_{\mathscr{L}_2} \psi$, *then* $\vdash_{\mathscr{L}_2} \psi$.

Unfortunately we do not have the time to prove this theorem here. We can show that our system $\mathscr{L}_2$ is complete for the system of $\mathscr{L}_2$-situations, which include situations with an empty domain.[19] A logic, like $\mathscr{L}_2$, that is compatible with a semantics that includes empty domains is called a *free logic*.

Non-transitive sequents A complication arises, however: since the $\mathscr{L}_2$-tableau generated by $\{\forall x(x = x), \neg a = a\}$ closes immediately, the sequent $\forall x(x = x) \models_{\mathscr{L}_2} a = a$ is correct. Similar considerations show that $a = a \models_{\mathscr{L}_2} \exists x(x = x)$ is correct. However, $\forall x(x = x) \models \exists x(x = x)$ is *not correct*, since it does not hold in the empty domain. Therefore, unlike sequents of $\mathscr{L}$ (page 60), $\mathscr{L}_2$-sequents are not transitive.

The $\mathscr{L}_2^\star$ remedy If, however, we consider only $\mathscr{L}_2^\star$ situations, where the empty domain is ruled out, we can prove completeness for this system. Indeed, the first completeness proof was for a system of classical logic equivalent to $\mathscr{L}_2^\star$. This was established by Kurt Gödel in 1930, and was one of the first of the great results about logic proved in the golden age of the 1930s. A good exposition of this kind of standard completeness theorem, and related theorems like *compactness* (proved for $\mathscr{L}$ earlier in Theorem 8) and the *Löwenheim-Skolem* theorem,[20] can be found in Boolos *et al.* (2003, ch. 12–14); a more elementary presentation is in Jeffrey (1991, §5.8 and §9.8).

[19] A proof of this fact for a system much like $\mathscr{L}_2^\star$ can be found in Bostock (1997, §§8.6–8.7).

[20] This theorem is new to full predicate logic: it basically says that any consistent set of predicate wffs has a model in which the domain is *enumerable*—that is, the domain can be put into one-to-one correspondence with the set of all natural numbers.

3.11.3 Arithmetic

Logic was developed by Frege and Peirce in the 1880s to give a solid foundation for mathematical reasoning. We do not have any distinctively mathematical logical truths, because no mathematical vocabulary has been explicitly introduced. But we can add a set of axioms to logic that can characterise arithmetical truths, for example. That is, we take a set of $\mathscr{L}_2$-wffs which are not logically true, but we assume them to be true: we adopt them as axioms. Therefore we must correspondingly restrict our semantics to those situations in which those sentences are true under the intended interpretation of the meanings of the predicates and the designators of the objects.

The Dedekind-Peano axioms The first set of such axioms to be proposed has gained wide currency. The were proposed by Dedekind (1963), but were popularised by Peano in his work on the foundations of arithmetic, and you will often see them called 'the Peano axioms'. They comprise the postulates found in Table 3.5. These axioms say the following in the intended interpretation: that **0** is a number; the successor of a number is a number; numbers with the same predecessor are identical; every number has a successor; no number precedes **0**; numbers with the same successor are identical; and induction over the natural numbers holds for predicates.

With these axioms we can put arithmetic on a sound footing, in the sense that we can define a system $\mathscr{PA}$ ('Peano arithmetic') over a language enriched with these new predicates 'N' and '$\prec$' as follows: take all the rules of $\mathscr{L}_2^\star$, plus the additional rule that any of the Dedekind-Peano axioms can be added to the bottom of any branch. This system of tableaux is sound for the intended interpretation of arithmetic over the natural numbers, once we introduce abbreviatory function symbols for the relations that we can define over the natural numbers. That is, we can show when the relation Σ holds between three numbers (Equation 3.14), just using the Peano-Dedekind axioms and their new vocabulary:

Definition 9 (Σ). We give an inductive definition (§B.2):

Basis For every x, $\langle x, x, \mathbf{0}\rangle \in \Sigma$;

Induction If $\langle x, y, z\rangle \in \Sigma$ and $x \prec u$, $y \prec v$, and $z \prec w$, then $\langle u, y, w\rangle \in \Sigma$ and $\langle u, v, z\rangle \in \Sigma$.

We may then introduce '+' in the expected way: see Equation 3.13.

Table 3.5 The Dedekind-Peano axioms

In the following, '$N\varphi$' means 'φ is a natural number'; '$\varphi \prec \psi$' means 'φ is the immediate predecessor of ψ' (i.e. '$\varphi + 1 = \psi$'), and '**0**' designates the number 0:

1. $N(\mathbf{0})$;
2. $\forall m \forall n((N(m) \wedge m \prec n) \rightarrow N(n))$;
3. $\forall m, n, n'((N(m) \wedge m \prec n) \wedge m \prec n') \rightarrow n = n')$;
4. $\forall m(N(m) \rightarrow \exists n(m \prec n))$;
5. $\forall n(N(n) \rightarrow \neg(n \prec \mathbf{0}))$;
6. $\forall m, m', n((Nm \wedge Nm' \wedge m \prec n \wedge m' \prec n) \rightarrow m = m')$;
7. $\Big(\Phi(\mathbf{0}) \wedge \forall m \forall n((\Phi(m) \wedge m \prec n) \rightarrow \Phi(n))\Big) \rightarrow \forall n(N(n) \rightarrow \Phi(n))$ (for any predicate 'Φ'—induction schema).

3.11.4 Incompleteness

Unfortunately, $\mathscr{PA}$, though sound, cannot be proved complete: there are true arithmetical claims that cannot be proved within $\mathscr{PA}$. This result is (one consequence of) the celebrated *incompleteness* theorem of Gödel (1992). This is one of the most beautiful and central results of modern logic, but unfortunately not within our grasp at this point. Proving it is one of the key goals of any further course in mathematical logic, and the techniques it uses have been found to be endlessly fruitful in foundational work in logic, set theory, and mathematics more generally.

Exercises for §3.11

Exercise 3.11.1: As on page 142, give a sentence of $\mathscr{L}_2$, using the predicates $\Sigma(x, y, z)$ and $\Pi(x, y, z)$ ('x is the sum of y and z'), that is true in just the same circumstances as the following sentence of arithmetic (where '$\cdot$' is the multiplication function):

(3.15) $$\forall x \forall y(x \cdot s(y) = (x \cdot y) + x).$$

Exercise 3.11.2: Give an inductive definition of the product relation Π.

Exercise 3.11.3: If $\vdash_{\mathscr{L}_2}$ is non-transitive, that also means there are wffs φ, ψ and χ such that $\vdash_{\mathscr{L}_2} \varphi \to \psi$; $\vdash_{\mathscr{L}_2} \psi \to \chi$; and yet $\nvdash_{\mathscr{L}_2} \varphi \to \chi$. Does this failure of transitivity for '$\to$' in $\mathscr{L}_2$ show that '$\to$' is not a real conditional?

3.12 Conclusion

As can be seen from this brief survey of extensions and further directions in logic, the knowledge you now have is only the beginning of what there is to know about this fascinating and rich subject. Though logic is now firmly a part of mathematics, it keeps deep and solid roots in the concepts of valid and invalid arguments, and thence in the ordinary practices of giving and asking for reasons. The study of logic then not only gives you a body of knowledge and facts to absorb, but also a rich grounding in argument and reasoning: essential foundation for any course of study.

Appendix A

Mathematical Induction

Intuition We use, in these notes, a powerful kind of proof known as proof by *mathematical induction*.[1] Mathematical induction relies on the following, intuitive, facts: assume we can show (a) that the first member of some sequence has a property, and (b) that if the preceding member of the sequence possesses the property, so does the succeeding member. (a) and (b) are sufficient, as reflection suggests, to establish that *every* member of the sequence possesses the property.

Natural numbers We usually use induction over a set of items that are ordered by the natural numbers, $\mathbb{N}$. These are the 'counting numbers', zero, one, two, *&c*. That is, $\mathbb{N} = \{0, 1, 2, \ldots\}$. We need also to note that the relation $\leqslant$ ('less than or equal to') is defined on the set $\mathbb{N}$.

A.1 Weak Principle of Induction

The most commonly used principle of induction is the following:

Weak Induction If (i) property P is true of the first member of a set ordered by the natural numbers, and (ii) if P is true of the nth member, then P is true of $n + 1$th member; then those two facts entail that for every member x, P is true of x.

[1] This is to be carefully distinguished from inductive reasoning, in the sense of ampliative non-deductive implication, paradigmatically from observed evidence to conjectures about the unobserved. See page 15.

Using Induction We use induction once we can establish both premises of the weak induction. We do it in two steps: the *base case*, where we establish that P holds of the first member; and the *induction case*, when we *assume* that P holds of the nth member, and on the basis of that assumption we prove that P holds of the $n + 1$th member. We, of course, do not assume that P holds for some particular n: rather, we assume an *arbitrary* n, and show, regardless of the actual value of n, that the induction step holds.

Example of Lemma 1 We saw a number of examples of the use of weak induction above; recall the proof of Lemma 1. In that case, we had a set of all tableaux, ordered by their length of steps, n. The property P was of having the right kind of valuation defined over the branches of the tableau; we showed that if a tableau of length n had the right valuation, then all the tableaux of length $n + 1$ which extended the original tableau also had the right valuation.

Example of Weak Induction Frequently in mathematics we use weak induction on numbers themselves: we want to show that 0 has some property P (written '$P(0)$'), and that if $P(n)$ then $P(n + 1)$, so that P holds of every natural number. For example, say we want to prove that

Theorem 17. *For every natural number n,*

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

Proof. We first prove the *base case*, that the property holds of 0 (the first member of $\mathbb{N}$). That is easy: $0 = \frac{0\cdot(0+1)}{2} = \frac{0}{2} = 0$, which is true. So that part was easy; usually the base case is the easy part of an induction.

Now we wish to prove the *induction step*. That is, we assume the property holds of n, and show it must hold of $n+1$. So we assume: $0+1+\ldots+n = \frac{n(n+1)}{2}$. This assumption allows us to infer:

$$(0 + 1 + \ldots + n) + n + 1 = \left(\frac{n(n+1)}{2}\right) + n + 1 \quad \text{(A.2)}$$

$$= \frac{n(n+1)}{2} + \frac{2(n+1)}{2} \quad \text{(A.3)}$$

$$= \frac{(n+2)(n+1)}{2} \quad \text{(A.4)}$$

$$= \frac{(n+1)((n+1)+1)}{2} \quad \text{(A.5)}$$

Equation A.5 is clearly just an instance of Equation A.1 with '$n+1$' in place of 'n', as required. This proves the induction step. ■

A.2 Other forms of Induction

The weak principle of induction also has equivalent formulations. The first of these is the *strong principle of induction*:

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

The second of these is the *Least number principle*:

LNP If M is a subset of $\mathbb{N}$, and is non-empty, then M has a least member.

Equivalences It can be proved that **weak induction** entails **strong induction**, which in turn entails **LNP**, which in its turn entails **weak induction**. So all three formulations are equivalent. The proofs are relatively easy; we show one of them.

THEOREM 18 (LNP → WEAK). *The Weak Principle of Induction follows from the Least Number Principle.*

Proof. Let us assume that P is a property such that $P(0)$, and for every n, $P(n) \rightarrow P(n+1)$ are both true. We prove that from the LNP and these assumptions, the claim that for every n, $P(n)$ follows (which is the Weak Principle).

Consider $M = \{n : \neg P(n)\}$, i.e. the set of numbers for which P does not hold. By the LNP, M has a least member if it is not empty. Suppose m is the least member of M. $P(0)$ holds, by assumption, so there is some number

$n \geqslant 0$ such that $m = n + 1$. Therefore $n < m$. Since m is the least member of M, it follows that $P(n)$. But since 'if $P(n)$ then $P(n + 1)$' is true, $P(m)$ must be true, but that contradicts our assumption that $m \in M$. So there can be no least member of M (because that leads to contradiction), and a set only has no least member if it has no members at all. Therefore M is empty, and hence there is no number n such that $\neg P(n)$, hence for every number n, $P(n)$. ■

Appendix B

Elementary Set Theory

B.1 Sets

A set is a collection of items, which are called the *members* or *elements* of that set. For instance, there is a set of all kangaroos, the set of all even primes, the set of a all formulae of $\mathscr{L}$. We write down a set by enclosing its members in curly braces: so the set of even numbers less than 10 is written $X = \{2, 4, 6, 8\}$. We shall use capital $X, Y, Z \ldots$ to denote sets. To say of something that it is a member of a particular set, we use '$\in$' to denote the membership relation: so $2 \in \{2, 4, 6, 8\}$, or $2 \in X$. Of course, 3 is not a member of this set; we write $3 \notin X$.

Abstraction For any collection of items, there is a set which contains them—at least if the members can be specified. That is, if we have have a condition P, we can make a set of all and only those things which satisfy P. This is written $\{x : P(x)\}$, and read 'the set of things x such that P applies to x'. For any item y, $y \in \{x : P(x)\}$ iff $P(y)$. We call this an *abstraction* principle: for almost any meaningful predicate we can abstract away from its meaning and leave just the set of things which satisfy it.

Extensionality Our conception of a set is *extensional*: a set is defined by its members. Thus if $x \in X$ iff $x \in Y$, then $X = Y$. But it is possible to have the same set picked out by different conditions: for example $X = \{x : \textbf{Prime}(x) \wedge \textbf{Even}(x)\}$, but also $X = \{x : x = \textbf{Successor}(1)\}$. It might seem

from this that sets of entities which fall under a predicate are not completely adequate to capture the full meaning of predicate.

Russell's paradox; the iterative hierarchy This principle is very powerful, and indeed, quite dangerous: consider the condition '$x \notin x$'. This gives a set $X = \{x : x \notin x\}$. Is $X \in X$? If it is, then $X \notin X$, by the condition, so it must not be. But if it is not a member of itself, then it satisfies the condition, so it it is. Contradiction! This is called Russell's paradox. There can be no such set X, and hence we must have done something illegitimate when applying the abstraction principle. Much ink has been spilled on just how to resolve this problem, and to delimit conditions that give a satisfactory account of when abstraction can and cannot be legitimately used. We shall stipulate the following: no self reference. Or, more concretely, we specify sets in what is called the *iterative hierarchy*. We first allow sets of items which are not themselves sets (such items are often called *ur-elements*). Then, at the next step, we allow sets which contain either ur-elements or sets formed at the first stage. And so on. At the nth stage, we allow sets to contain sets formed at any previous stage, as well as ur-elements. The key is that we never allow sets formed at the same stage to be members of each of other; this is sufficient to rule out Russell's paradox. (This is the same Russell who gave us the theory of descriptions.)

B.2 Inductively defined sets

We can also define sets another way than by giving an explicit condition P: that is, inductively. Recall §2.5.1, where we defined a well formed formula by starting with some things we called basic wffs, and then saying that if something was a wff, then something else generated from it was also a wff. Then we closed the definition, saying that nothing else was a wff unless it met the preceding conditions.

Terminology That is, we gave a *basis*: a set of elements stipulated to satisfy the definition. Then we said that anything related to an element of the set by the *generating relation* was also a member. Finally, nothing except those things is a member of the set: this is the *closure* condition.

Ancestors and ancestrals Another example. We define the set of your ancestors as follows. (Basis) You are your own ancestor. (Generating rela-

tion) Anything which is a parent of one of your ancestors is also one of your ancestors. (Closure) Nothing else is one of your ancestors. Now we have a set of ancestors for each one of us: call that set A_{AE} in my case. Now, everything that is in A_{AE} is a parent of a parent of ... a parent of AE, with for some $n \geqslant 0$ occurrences of 'is a parent of'. That is, if you are ancestor of mine, you are linked to me by some number of occurrences of the parent of relation. The relation 'being an ancestor of' is called the *ancestral* of the 'is a parent of' relation, for obvious reasons. We can consider other ancestrals also: consider the relation 'being the successor of' amongst the natural numbers (as in '2 is the successor of 1'). The ancestral of this relation is the 'greater than or equal to' relation, since if a number is greater than another, it is linked to it by some number (0 or more) of occurrences of the 'is the successor of' relation. Correspondingly, an inductively defined set of a certain generating relation R can be given an explicit abstractive definition from the condition of bearing the ancestral R^* to the members of the basis set. So we could define the set of my ancestors inductively, using R = 'is a parent of' on the basis containing just me, or we could define $A_{AE} = \{x : R^*(x, AE)\}$, where R^* = 'x is an ancestor of AE'. Hence inductively defined sets are a special case of abstractively defined sets, where one abstracts from the ancestral of the generating relation. There are not, therefore, two different kinds of set, but just one; but we do have an additional helpful way of specifying them (since some ancestrals, unlike 'greater than', are very difficult to grasp immediately: consider that one cannot simply identify your ancestors by sight without tracing down the family tree!).

B.3 Relations between, and functions upon, sets

Now we have lots of sets and lots of members. We need some terminology to systematise our discussion.

Sets and Logic There are many relationship between sets that are a matter of logic. Let me illustrate by the following examples.

DEFINITION 10 (SUBSET). A set X is a *subset* of Y, written $X \subseteq Y$ if every member of X is also a member of Y: for every x, if $x \in X$ then $x \in Y$. X is a *strict subset* (or *proper subset*) of Y if $X \subseteq Y$ and $Y \nsubseteq X$: we write this as '$X \subset Y$'.

Note that here, a subset corresponds to an 'if ... then —' statement: X is a subset of Y iff, for any element x, $(x \in X) \rightarrow (x \in Y)$.

There are also functions that combine two sets and yield another set:

Definition 11 (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.

For any two sets, X and Y, is there also always a union? Yes: the set $\{z : (z \in X) \vee (z \in Y)\}$ is the union, which by abstraction always exists. Note the similarity between $\cup$ and $\vee$: the union of two sets has as elements the members of either of them. There is also a generalised notion of union, not just of two sets, but on a set of sets. Let $\mathbf{X} = \{X_1, \ldots, X_n, \ldots\}$. $\bigcup \mathbf{X} = X_1 \cup X_2 \cup \ldots \cup X_n \cup \ldots$ (since these are all unions, there are no scope problems)—or, $\bigcup \mathbf{X} = \{x :$ for some $X_i \in \mathbf{X}, x \in X_i\}$.

Definition 12 (Intersection). If X and Y are sets, the set $X \cap Y$ which contains only members of both is called the *intersection.*

Again, there is a link between $\cap$ and $\wedge$: $z \in X \cap Y$ iff $z \in X \wedge z \in Y$. Again, there is a generalised notion of intersection, not just of two sets, but on a set of sets. Let $\mathbf{X} = \{X_1, \ldots, X_n, \ldots\}$. $\bigcap \mathbf{X} = X_1 \cap X_2 \cap \ldots \cap X_n \cap \ldots$ (since these are all intersections, there are no scope problems)—or, $\bigcap \mathbf{X} = \{x :$ for every $X_i \in \mathbf{X}, x \in X_i\}$.

Definition 13 (Relative Complement). If X and Y are sets, then $X - Y$ is the set of all members of X which are *not* members of Y, called *relative complement* of Y with respect to X.

Again, this is similar to negation: $X - Y = X \cap \{x : \neg(x \in Y)\}$. Using this definition, we propose the following:

Definition 14 (Absolute Complement). If Y is a set, then $-Y$ is the set which contains everything which is *not* a member of Y; i.e. the relative complement of Y with respect to the set of *everything* (see below for a problem with this).

Empty Set We define one special set: the set with no members, called the *empty set* and written '$\emptyset$', defined by the condition $\emptyset = \{x : x \neq x\}$. Since everything is self-identical (§3.3.2), this is obviously an unsatisfiable condition. Should this count as a set? If X is a set, then the relative complement $X - X$ is a set; but this is $\emptyset$. Note that, for every set X, $\emptyset \subseteq X$ (for there could be no counterexample, that is, no member of $\emptyset$ that is not a member of X, since $\emptyset$ has no members).

The Universal set? A problem now arises. Let $\mathbf{U} = -\emptyset$. Then since $\mathbf{U} \notin \emptyset$, which has no members, $\mathbf{U} \in \mathbf{U}$, by definition of complement. But we earlier (in our discussion of Russell's paradox) ruled out self-membered sets! Correspondingly, we must reject $\mathbf{U}$, which involves rejecting either $\emptyset$, or absolute complement. We reject the latter, but keep relative complements with respect to sets we already know exist.

Power set If we have a set X of some items, sometimes we wish to talk not just about those items, but about collections of those items which are subsets of X. The following definition ensures that there is a set of just those subsets:

DEFINITION 15 (POWER SET). Given a set X, the *power set* of X, denoted $\wp(X)$, is defined: $\wp(X) = \{Y : Y \subseteq X\}$.

Notice that, for any X, $\emptyset \in \wp(X)$; and $X \in \wp(X)$.

B.3.1 Ordered Sequences

Since our concept of a set is extensional, it turns out that

$$\{x, y, z\} = \{z, x, y\}. \tag{B.1}$$

That is, no matter what order we write down the members of a set, it still has the same members. But we often wish to describe an ordered collection of items: for instance, the set of all the natural numbers has a natural order '$\geqslant$', and it would be nice to capture this order by putting the natural numbers into a *sequence*. But, on the other hand, we don't want to have two types of collections, so it would be best if we could use regular sets to capture ordered sequences. Thankfully, we can.

Ordered Pairs Let us start with ordered pairs of items. We write an ordered pair of x and y as $\langle x, y\rangle$: of course there is the other ordered pair $\langle y, x\rangle$. We define this in terms of sets as follows:

DEFINITION 16 (ORDERED PAIR). An *ordered pair* $\langle x, y\rangle$ is defined as the unordered set of sets $\{\{x\}, \{x, y\}\}$. (This definition is due to Kuratowski.)

An ordered pair is thus an unordered pair of sets, one of which contains the first member of the pair, and the second contains both members of the pair. It is clear that $\{\{x\}, \{x, y\}\} \neq \{\{y\}, \{x, y\}\}$, and hence that $\langle x, y\rangle \neq \langle y, x\rangle$.

Ordered n-tuples We can extend this definition, and notation, to any number of elements of a sequence, as follows:

$$\langle x_1, \ldots, x_i, \ldots, x_n \rangle = \underbrace{\langle \ldots \langle}_{n} x_1 \rangle, \underbrace{\ldots, x_i \rangle, \ldots,}_{n-2} x_n \rangle. \tag{B.2}$$

This definition is unwieldy, but we shall simply adopt the notation and forget the background complexities.

Cartesian Product The *Cartesian product* of X and Y, written $X \times Y$, is the set of all pairs $\langle x, y \rangle$ such that $x \in X$ and $y \in Y$. This is useful in defining relations; a relation on a domain is just a subset of the product of the domain with itself (p. 129).

B.4 Functions

A *function* f is a designator-forming operator: that is, if 'a' is a designator, and 'f' a function, then '$f(a)$' is also a designator.[1] An example might clarify matters: if 'a' refers to Antony Eagle, and 'f' is the function 'the father of', then '$f(a)$' is a term that refers to the father of Antony Eagle. If 'b' designates 'the largest prime less than 10', and 'g' is the function 'the square of', then '$g(b)$' designates 49. We define these functions as follows: g is the function such that when applied to a number x, the function yields the square of that number, which we write as

$$g : g(x) = x^2. \tag{B.3}$$

Restrictions What does '$g(a)$' designate? What does '$f(b)$' designate? Neither of these questions have sensible answers: the number 7 is not such as to have a father, and I cannot be squared. So functions apply to—are *defined* on—certain objects, and (we say) are *undefined* on other objects.

Arguments and Values In '$f(a) = x$', 'a' is called the *argument* given to f, and 'x' denotes the *value* of the function given that argument. The set of objects that a function applies to (the set of acceptable arguments) is called the *domain* of the function; the set of objects that are values of the function when given as argument a member of the domain is called the *range* of the

[1] For more on functions, see Bostock (1997, §8.2) and Beall and van Fraassen (2003, ch. 2).

function. So the domain of 'the father of' is the set of all people (excepting the first one, if there is such); the range is the set of all male people with children.

Crucially, a function always has exactly *one* value for any argument on which it is defined. But more than one argument can have the same value for a given function: consider the squaring function f: $f(-2) = 4 = f(2)$.

Mappings If D_f is the domain of f, and R_f is the range of f, then we say that f *maps* D_f into R_f, written $f : D_f \mapsto R_f$. If $R_f \subseteq D_f$, then we call the function an *operator* on the domain: consider the binary sentential operator $\vee$, with the domain of wffs and the range is a subset of wffs, the disjunctions. Relatedly, we can say $f : A \mapsto B$ (f maps A *into* B) just in case $D_f = A$ and $R_f \subset B$. If $R_f = B$, we say that f maps A *onto* B. If f maps A both into and onto B, we say that f is a *one to one mapping*: that is, (i) for every $a \in A$, there is one and only one $b \in B$ such that $b = f(a)$; and (ii) for every $b' \in B$, there is one and only one $a' \in A$ such that $b' = f(a')$.

Compound functions If we have a function f, and a function g, if the range of f includes the domain of g, we can consider the compound function $g(f(x))$, which is the result of applying f to x, and then applying g to the result. So, for example, if $f : f(x) = x + 3$ and $g : g(x) = x^2$, then $g(f(x)) = h : h(x) = (x + 3)^2$. Note that for most functions, $f(g(x)) \neq g(f(x))$; e.g. $(x + 3)^2 \neq x^2 + 3$. Sometimes the composition of two functions $f(g(x))$ is written $f \circ g(x)$.

Special compound functions If f is a binary function, and $f(x, y) = f(y, x)$, then we call f *commutative*. '+' is a commutative function on natural numbers. If $f(x, f(y, z)) = f(f(x, y), z)$, then we call f *associative*: '+' is associative also. Note that '$\vee$' is neither associative nor commutative, e.g. since '$\varphi \vee \psi$' is a different wff from '$\psi \vee \varphi$'. But consider the disjunction valuation function $v_\vee$, defined as

$$v_\vee(\varphi, \psi) = \begin{cases} \top & \text{if } v(\varphi) = \top \text{ or } v(\psi) = \top \\ \bot & \text{otherwise.} \end{cases} \tag{B.4}$$

$v_\vee$ is associative and commutative. And so in general we might say that since '$\psi \vee \varphi$' and '$\psi \vee \varphi$' are logically equivalent, '$\vee$' is commutative and associative as far as logic or truth value is concerned.

Inverse and Identity We define the following function: $i : i(x) = x$. This is the *identity* function, which maps every argument onto itself. If we consider two functions f and g such that $g(f(x)) = i(x)$, then we say that g is the *inverse* of f, which we normally write f^{-1}. In some special cases, $f = f^{-1}$: we say that f is its own inverse. Consider for example $f : f(x) = -x$, defined on the natural numbers. Since $--1 = 1$, we can see that this is its own inverse. In classical logic, where $\neg\neg\varphi \equiv \varphi$, negation is its own inverse as far as truth value is concerned.

Functions and relations If R holds between x and y, then there is some function $f_R : f_R(x) = y$. Similarly, if $g : g(x) = y$, then there is some relation R_g such that $R_g(x, y)$ holds.

Appendix C

Further Reading

This book presented a system of *semantic tableaux* for propositional and full first-order logic that follows Hodges (2001). Other textbooks which present similar systems, and may be consulted for further information, include Jeffrey (1991) and Smullyan (1968).

There is more than one way to present this material, however—a quick look at Bostock (1997, part II) should give you some of the flavours. We chose tableaux, but there are lots of different (and different-looking!) systems that are also provable to be sound and complete with respect to $\mathscr{L}_2$-wffs (or $\mathscr{L}_2^\star$-wffs). For instance, there is the *natural deduction* system, presented nicely in Lemmon (1978), or its American variant, the *Fitch-style* natural deduction systems, a good and very accessible account of which appears in Barwise and Etchemendy (2002). Or one could return to the good old days of *axiomatic* systems for logic (where one begins with some claims assumed as true—the axioms—and some rules to derive one truth from another), as presented in (for example) Bostock (1997, ch. 5). With good reason, axiomatic systems have fallen from favour for *teaching* logic, but they are still used in (say) computer implementations of logical reasoning.

A natural next course in logic would take one of two forms. Either it would develop the role of logic in mathematical and arithmetical reasoning, which one of the great stimuli for the initial development of first-order logic by Frege, Peirce, and others. Such a course would look at formalising the notion of computability, that we intuitively treated with when we considered decidability, and develop systems of arithmetic within a formal language, culminating in the proof of Gödel's incompleteness theorems. The latter parts of Jeffrey (1991) develop this material at a beginner's level, but the

best text on this material is Boolos *et al.* (2003). For more on mathematical induction, and the number theory that advanced courses in logic often go on to cover, see Machover (1996, esp. Ch. 0), and also Beall and van Fraassen (2003, §10.3).

The other direction we could go is to extend the logical language so as to capture more of English. This itself can happen in two ways. Firstly, we can revise the logical laws, so we get a better treatment of (say) conditionals or arguments with contradictory premises. Or we can add new operators to our logic, to cope with tense, obligation, or modality. A good starting point for all these matters, and more besides, is Beall and van Fraassen (2003). On the topic of conditionals, in particular, much work has been done to find the logical form of the various English conditional sentences: a good guide to this literature is Bennett (2003).

One trend in recent logic has been to look more closely at the basic machinery of logical entailment, and the mechanics of proofs in particular. Discussion of sequents has been central in this trend; for the introduction of sequents, and the first beautiful results of what is now known as *proof theory*, see Gentzen (1969, ch. 5). This in fact is another way of presenting the same systems for logic, but it is very powerful method for representing all sorts of different logical systems, both stronger and weaker than the ones we consider here. Attention is paid to the structural rules on manipulating sequents (discussed in §2.9) in Restall (2000).

The other direction from proof theory is model theory: the abstract mathematical study of structures and situations in which sentences can be true or false. A good place to start is another book by Hodges (1997).

Now come topics which come strictly outside the scope of logic, but which do concern themselves deeply with logical results and methods. As might be apparent, logic relies closely on languages. There is a general philosophical field called the *philosophy of language* which tries to determine various features of typically linguistic phenomena, like reference, designation, or meaning. One cannot do better in this area than begin with Kripke (1980): a concise and brilliantly influential work on names, necessity, meanings and essentialism. It is a notable and impressive fact that the whole book was transcribed from 3 lectures that Kripke gave entirely without notes! The view which Kripke is concerned to rebut is Russell's (1956) view that names have descriptive content, discussed in §3.2.4 above. If you would like to follow up this issue, the articles in Moore (1993) are a good place to start. The ideas of Grice (1989), which we saw in §1.1.1, have been very influential in so-called *pragmatics*: that part of the philosophy of language which studies the use of particular sentences in a context, and not just

semantics in isolation.

Of course, it is also possible to use these general logical and philosophical results to illuminate natural languages. We constructed our formal languages to model certain important parts of natural languages, while avoiding certain ambiguities and problems of natural language. But once one sees a formal perspective on those fragments of natural language, it can be useful to try and include more of natural language in those models, and to produce formal syntactic and semantic theories of natural language. Here logic connects with linguistics. Formal theories of natural language syntax develop grammars for various natural languages; an accessible account of English grammar is Hiddleston and Pullum (2005), while the classical source of formal mathematical models of natural language syntax is Chomsky (2002). On the semantic side, a useful introduction to modern formal semantics (connecting with Chomskian views about syntax) can be found in Heim and Kratzer (1998).

The introduction to set theory in Appendix B might be usefully supplemented by any number of works. For our purposes, the material in Beall and van Fraassen (2003, ch. 2) is sufficient; they list further references at the end of that chapter. But set theory is not just important for the formulation of meta-logical results, but logic and set theory are crucial for the foundations of *arithmetic*. Frege's original work on the foundations of arithmetic remains highly readable: see Frege (1884). One particularly influential school in the philosophy of mathematics has been *intuitionism*, which gives rise to intuitionistic logic, the characteristic feature of which is the denial of the correctness of sequents of the form '$\neg\neg\varphi \vdash \varphi$'—see Beall and van Fraassen (2003, §6.4) and van Dalen (2001). The philosophy of mathematics more broadly quickly becomes quite technical, but a wonderful informative and entertaining book on mathematics and concepts is Lakatos (1976).

The philosophical foundations of logic have received much attention in recent philosophy, addressing such questions as 'what justifies the logical rules and theorems?'. One nice treatment of such topics is Quine (1970); another is Haack (1996). A good collection of articles on a lot of topics in *philosophical logic* is Goble (2001).

Appendix D

Logical Notation

Alternative notation The notation used in this book is fairly standard, but you may see other notation in the course of your further reading and other study.

Connectives We use '$\wedge$' for the truth-functor 'and'; '&' is commonly used to express this connective. Less commonly, you may see '$\cdot$'. Some authors write conjunctions by simply writing the conjuncts side by side: such authors would write 'PQ' for '$P \wedge Q$'.

Many authors use '$\supset$' for our '$\rightarrow$'; many choose to use '$\rightarrow$' for a generic conditional operator, and reserve '$\supset$' to unambiguously refer to the material conditional. Many authors use '$\equiv$' for our '$\leftrightarrow$'.

You will sometimes see '$\overline{p}$' for '$\neg p$', but more commonly you will see '$\sim$' instead of '$\neg$'. In some non-standard logics, these symbols are used for different 'kinds' of negation. For example, we could distinguish one kind of negation sometimes called 'rejective negation', where by an utterance of '$\neg p$' one *rejects* p, and another kind, such that by an utterance of '$\sim p$' one *asserts* 'not-p'. These amount to the same thing in our system, but, it has been argued, they can come apart.

Quantifiers, Predicates and Identity Sometimes you will see '(x)' instead of '$\forall x$'.

We use '$a \neq b$' to abbreviate what others choose to write as '$\neg a = b$'.

We have written sometimes 'xRy', and at other times '$R(x, y)$'; one also sees 'Rxy'; all three are acceptable as long as you remain consistent.

Punctuation and Metalanguage Hodges (2001) uses brackets '[' and ']' instead of parentheses, but they play precisely the same role. Hodges also uses uppercase letters '$P, Q, R \ldots$' for the sentences letters (you may also see sentence letters drawn from other regions of the alphabet).

We wrote the truth values as '$\top$' and '$\bot$'; many writers render these as '*T*' and '*F*'.

The valuation function $v_A(\varphi)$ is sometimes called an *assignment*, and is sometimes written $|\varphi|_A$.

Bibliography

Barwise, Jon and Etchemendy, John (2002), *Language, Proof and Logic*. Stanford, CA: CSLI Publications. 159

Beall, JC and van Fraassen, Bas C. (2003), *Possibilities and Paradox*. Oxford: Oxford University Press. 48, 156, 160, 161

Bennett, Jonathan (2003), *A Philosophical Guide to Conditionals*. Oxford: Oxford University Press. 36, 160

Boolos, George S., Burgess, John P. and Jeffrey, Richard C. (2003), *Computability and Logic*. Cambridge: Cambridge University Press, 4 ed. 73, 140, 141, 143, 160

Bostock, David (1997), *Intermediate Logic*. Oxford: Oxford University Press. 27, 29, 31, 56, 58, 61, 67, 69, 72, 115, 118, 134, 136, 143, 156, 159

Braun, David (2001), "Indexicals". In Edward N. Zalta (ed.), *The Stanford Encyclopedia of Philosophy*, URL http://plato.stanford.edu/archives/fall2001/entries/indexicals/. 83

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. 87

Chomsky, Noam (2002), *Syntactic Structures*. Berlin: Mouton de Gruyter. 161

Dedekind, Richard (1963), "The Nature and Meaning of Numbers". In *Essays on the Theory of Numbers*, New York: Dover, pp. 29–115. English translation of *Was Sind und Was Sollen die Zahlen*. 144

Frege, Gottlob (1884), *The Foundations of Arithmetic*. Oxford: Blackwell. Translated by J. L. Austin. 100, 161

Gentzen, Gerhard (1969), *Collected Papers*. Amsterdam: North-Holland. Edited by M. E. Szabo. 160

Goble, Lou (ed.) (2001), *Blackwell Guide to Philosophical Logic*. Oxford: Blackwell. 161, 166

Gödel, Kurt (1992), *On Formally Undecidable Propositions of* Principia Mathematica *and Related Systems*. New York: Dover. First published 1962. 145

Graff, Delia (2001), "Descriptions as Predicates". *Philosophical Studies*, vol. 102: pp. 1–42. 86

Grice, Paul (1989), "Logic and Conversation". In *Studies in the Way*

of Words, Cambridge, MA: Harvard University Press. 4, 5, 160

Haack, Susan (1996), *Deviant Logic, Fuzzy Logic*. Chicago: University of Chicago Press. 161

Harley, Heidi (2006), *English Words: A Linguistic Introduction*. Oxford: Blackwell. 8, 13, 21, 105

Harman, Gilbert (2002), "Internal Critique: A Logic is Not a Theory of Reasoning and a Theory of Reasoning is Not a Logic". In Dov Gabbay *et al.* (eds.), *Studies in Logic and Practical Reasoning*, vol. 1, Amsterdam: Elsevier Science. 15

Heim, Irene and Kratzer, Angelika (1998), *Semantics in Generative Grammar*. Oxford: Blackwell. 161

Hiddleston, Rodney D. and Pullum, Geoffrey K. (2005), *A Student's Introduction to English Grammar*. Cambridge: Cambridge University Press. 2, 8, 161

Hodges, Wilfrid (1997), *A Shorter Model Theory*. Cambridge: Cambridge University Press. 129, 160

——— (2001), *Logic*. London: Penguin, 2 ed. viii, 4, 15, 29, 30, 33, 45, 56, 58, 59, 60, 61, 67, 69, 79, 80, 87, 89, 90, 93, 94, 101, 104, 107, 112, 115, 159, 163

Jackson, Frank (1987), *Conditionals*. Oxford: Blackwell. 35, 36

Jeffrey, Richard C. (1991), *Formal Logic: Its Scope and Limits*. New York: McGraw-Hill. ix, 50, 123, 125, 140, 141, 143, 159

Kaplan, David (1989), "Demonstratives". In Joseph Almog, John Perry and Howard Wettstein (eds.), *Themes from Kaplan*, New York: Oxford University Press, pp. 481–563. 83

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. 133

——— (1980), *Naming and Necessity*. Cambridge, MA: Harvard University Press. 80, 160

Kunda, Ziva (1999), *Social Cognition: Making Sense of People*. Cambridge, MA: MIT Press. 103

Lakatos, Imre (1976), *Proofs and Refutations*. Cambridge: Cambridge University Press. 161

Lemmon, E. J. (1978), *Beginning Logic*. Indianapolis: Hackett. 159

Machover, Moshé (1996), *Set Theory, Logic, and Their Limitations*. Cambridge: Cambridge University Press. 160

Moore, Adrian (ed.) (1993), *Meaning and Reference*. Oxford: Oxford University Press. 160

Predelli, Stefano (1998), "I am not here now". *Analysis*, vol. 58: pp. 107–115. 82

Quine, Willard van Orman (1951), *Mathematical Logic*. Cambridge, MA: Harvard University Press. 38

——— (1970), *Philosophy of Logic*. Englewood Cliffs, NJ: Prentice-Hall. 161

Restall, Greg (2000), *An Introduction to Substructural Logics*. London and New York: Routledge. 160

Russell, Bertrand (1956), "On Denoting". In *Logic and Knowledge*, London and New York: Routledge, pp. 41–56. 84, 160

——— (1997), *The Problems of Philosophy*. Oxford: Oxford University Press. 86

Smullyan, Raymond M. (1968), *First-Order Logic*. Berlin: Springer-Verlag. 159

Stanley, Jason and Szabó, Zoltán Gendler (2000), "On Quantifier Domain Restriction". *Mind & Language*, vol. 15: pp. 219–61. 107

Strawson, P. F. (1950), "On Referring". *Mind*, vol. 59: pp. 320–44. 84

van Dalen, Dirk (2001), "Intuitionistic Logic". In Goble (2001). 161