PREFACE

How to Use This Book

This book is intended to be used by you for independent study, with no other reading or lectures etc., much along the lines of standard Open University materials. There are plenty of exercises within the text which we would recommend you to attempt at that stage of your work. Almost all are intended to be reasonably straightforward on the basis of what's come before and many are accompanied by solutions – it's worth reading these solutions as they often contain further teaching, but do try the exercises first without peeking, to help you to engage with the material. Those exercises without solutions might well be very suitable for any tutor to whom you have access to use as the basis for any continuous assessment of this material, to help you check that you are making reasonable progress. But beware! Some of the exercises pose questions for which there is not always a clear-cut answer: these are intended to provoke debate! In addition there are further exercises located at the end of most sections. These vary from further routine practice to rather hard problems: it's well worth reading through these exercises, even if you don't attempt them, as they often give an idea of some important ideas or results not in the earlier text. Again your tutor, if you have one, can guide you through these.

The book is also peppered with notes in the margins, like this! They consist of comments meant to be on the fringe of the main text, rather than the core of the teaching, for instance reminders about ideas from earlier in the book or particularly subjective opinions of the author.

If you would like any further reading in logic textbooks, there are plenty of good books available which use essentially the same system, for instance those by Enderton [12], Hamilton [18], Mendelson [25] and Cori and Lascar [7].

Acknowledgments

I would like to thank all those who have in some way helped me to write this book. My enthusiasm for the subject was fuelled by Robin Gandy, Paul Bacsich, Jane Bridge, Angus Macintyre and Harold Simmons, when I studied at the Universities of Oxford and Aberdeen. Anything worthwhile I have successfully learnt about teaching stems from my colleagues at the Open University and the network of mathematicians throughout the UK who support the Open University by working as Associate Lecturers, external assessors and examiners. They have taught me so much. It has been particularly stimulating writing this book alongside producing the Open University's course on Mathematical Logic (with a very different angle on the subject) with Alan Pears, Alan Slomson, Alex Wilkie, Mary Jones, Roger Lowry, Jeff Paris and Frances Williams. And it is a privilege to be part of a university which puts so much care and effort into its teaching and the support of its students. The practicalities of producing this book owe much to my publishers, Stephanie Harding and Karen Borthwick at Springer; and to my colleagues at the Open University who have done so much to provide me with a robust and attractive LaTeX system: Alison Cadle, David Clover, Jonathan Fine, Bob Margolis and Chris Rowley. And thanks to Springer, I have received much invaluable advice on content from their copy-editor Stuart Gale and their anonymous,

Plainly the blame for any errors and inadequacies of this book lies entirely with me. But perhaps at some deep and subtle level, the fault lies with everyone else!

very collegial, reviewers. I would also like to thank Michael Goldrei for his work on the cover design.

Perhaps the main inspiration for writing the book is the enthusiasm and talent for mathematical logic displayed by my old students at the Open University and at the University of Oxford, especially those of Somerville, St. Hugh's and Mansfield Colleges. In particular I'd like to thank the following for their comments on parts of the book: Dimitris Azanias, David Blower, Duncan Blythe, Rosa Clements, Rhodri Davies, David Elston, Michael Hopley, Gerrard Jones, Eleni Kanellopoulou, Jakob Macke, Zelin Ozturk, Nicholas Thapen, Matt Towers, Chris Wall, Garth Wilkinson, Rufus Willett and especially Margaret Thomas.

This book is dedicated to all those whose arguments win me over, especially Jennie, Michael, Judith and Irena.

2 PROPOSITIONS AND TRUTH ASSIGNMENTS

2.1 Introduction

In this chapter we shall look at statements with a very simple form and arguments about them which rely only on how we use words like 'and', 'or', 'not' and 'implies'. We shall also look at the truth or falsity of the statements and the validity of arguments built up from them. It is best to give one or two examples. For instance, suppose that we are told that

'the temperature outside is at most 20°C or the drains smell'

and believe this statement to be true. Suppose that the weather forecast for tomorrow predicts a temperature of over 20°C. Then we would predict that the drains will smell.

As another example, suppose that we are told about some function $f : \mathbb{R} \longrightarrow \mathbb{R}$ that

'f is not differentiable or f is continuous'

and that we have the further piece of information that f is differentiable. Then from this information we can infer that f is continuous.

These arguments cover entirely different areas of experience, but at a certain level they have a common shape. The statement 'the temperature outside is at most 20°C or the drains smell' in the first argument is built up from two shorter statements

'the temperature outside is at most 20°C'

and

'the drains smell'

connected by the word 'or'. The extra information that the statement 'the temperature outside is over 20°C' is true tells us that the statement 'the temperature outside is at most 20°C' is false, from which we can infer that 'the drains smell' will be true, using our understanding of the word 'or'.

Indeed we call the word 'or' a *connective* as it connects shorter statements to produce a longer one.

Similarly the statement 'f is not differentiable or f is continuous' is built up from the statements

'f is not differentiable'

and

'f is continuous'

by connecting them with 'or'. And given the extra information that 'f is differentiable', so that the statement 'f is not differentiable' is false, our understanding of the word 'or' helps us infer that 'f is continuous'.

The 'not' converts 'f is differentiable' into the longer statement 'f is not differentiable'. We shall also describe 'not' as a connective and consider it in this chapter.

We shall summarize the common feature of these two arguments that will particularly interest us in this chapter as follows. Using the letters p and q to stand for statements like 'the drains smell' and 'f is not differentiable', if we believe the more complicated statement 'p or q' to be true and the statement p to be false, we can infer that q is true.

There are other features of the arguments which are of interest. For instance, one might reasonably question whether the statement 'the temperature outside is at most 20°C or the drains smell' is actually true – it might be true for one person's drains but not for somebody else's – whereas anyone who has studied real analysis would know that for any function $f : \mathbb{R} \longrightarrow \mathbb{R}$, the statement '$f$ is not differentiable or f is continuous' is always true. We shall refine our description of the common features of the argument to account for these factors by saying that

> under any set of circumstances for which the statement 'p or q' is true and p is false, then q is true.

This is then something to do first with how a statement is built up from its component parts, here using 'or', and second how the truth of the statement depends on the truth of these component parts. It is nothing to do with the content of the statements for which p and q stand.

In this chapter we shall discuss a formal language within which we can build up more complicated statements from basic component propositions using symbols like $\vee, \wedge, \rightarrow$ to stand for connecting words, here respectively 'or', 'and', 'implies'. The formal language will have construction rules to ensure that any such complicated expression is capable of being judged to be either true or false, given the truth or falsity of the component parts. For instance, we want to avoid the formal equivalent of expressions like 'or the drains smell': without some statement before the 'or', we would be reluctant to describe this as in a fit state to be pronounced true or false. These construction rules are described as the *syntax* of the language. The framework within which we give some sort of meaning to the formal statements and interpret them as true or false in a given set of circumstances is called the *semantics* of the language. After we have established the basic rules of the language and its interpretation, we shall move on to issues like when one statement is a consequence of others. This will lay the ground for the discussion of a formal proof system for such statements. We shall build on this later in the book when we discuss this idea of consequence for much richer languages involving predicates and quantifiers within which we can express some serious mathematical statements.

'Proposition' is often used to mean a statement about which it is sensible to ask whether it is true or false.

There might appear to be potential for confusion between the formal language we study and the language we use to discuss it. The language we use for this discussion is that of everyday mathematical discourse and is often described as the *metalanguage*. We hope that we won't confuse the two sorts of language – usually the context will make it clear when we are talking about the formal language. However there will be strong links between the two levels of language. For instance, the formal rules for the use of the symbol $\wedge$ intended to represent the word 'and' will inevitably be based on how we use the word 'and' in everyday discourse. Also the desire to represent some part of everyday language in a formal way can force us to tie down how we use everyday language correctly.

What we are about to describe in this chapter, and indeed in the book as a whole, is a mathematical model of a fragment of natural language and arguments using it, not capturing fully their richness and variety. The importance of the model resides in the richness of the resulting theory, its applicability to large tracts of mathematics and, historically, in giving a paradigm for more refined modern analyses of language and argument – indeed, it is *the* model used by virtually all mathematicians and users of logic. Our model will make some hard and fast decisions about how to use terms like 'true', 'false' and 'or' which could legitimately be challenged in terms of how well they model natural language. Your attitude as a reader and student should be to run with our decisions for the purposes of this book, and probably for all the mathematics you will ever do, but to have an open mind to well-reasoned objections to them!

2.2 The construction of propositional formulas

In this section we shall describe the formal language which we shall use to represent statements. The language will consist of some basic symbols and we shall give rules for combining these into more complicated expressions, giving what is called the syntax of the language. We shall describe the way in which we shall give meaning to the formal language, that is, give its semantics, in the following section. However, the syntax and semantics are, perhaps not surprisingly, intertwined, so that considerations of the semantics will influence the specification of the syntax.

We have already indicated that we shall use letters like p and q to stand for basic component propositions, like 'the drains smell' and 'f is continuous', and that we shall use symbols to stand for connectives like 'or' and 'and' to build more complicated propositions from these basic ones. The building process involves stringing together these symbols and letters. We need to be clear what is meant by a string of symbols. A *string* is a finite sequence of symbols. Furthermore, we normally specify the set of symbols which can be used to form a string and we shall do this soon. Just as in everyday language, we have to distinguish which strings of symbols represent anything to which we can usefully give a meaning. In most normal uses a string like)9X7a)) would normally mean nothing and signify that some error has occurred, e.g. a cat has danced on a computer keyboard. For our purposes in this chapter, we will have particular requirements of a statement and this will have a knock-on effect on the strings of symbols in which we shall be interested. For instance, we want to represent statements for which it is meaningful to talk in terms of their truth or falsity. So we would want to exclude from our set of formal statements a string representing

The finiteness of the sequence is of considerable importance in this book. There are other contexts where it makes sense to allow strings to be infinite.

> 'I'll go down to the shops and'

These are examples of how the intended meaning, the semantics, will influence the formal rules of the syntax.

on the grounds that there's something missing after the 'and', preventing us from deciding on its truth on the basis of the truth or falsity of its component parts. Just because we can *utter* the words in this string, we are not necessarily *stating* any idea. More subtly, we want to avoid ambiguity in our formal statements, as for instance with

> 'it is snowing or the bus doesn't come and I'll be late for work'.

The truth of this statement depends on whether its component parts are bracketed together as

'it is snowing or the bus doesn't come' and 'I'll be late for work'

or

'it is snowing' or 'the bus doesn't come and I'll be late for work'.

In the case when 'it is snowing' is true, but both 'the bus doesn't come' and 'I'll be late for work' are false, the first way of bracketing the components gives false while the second gives true. So without some form of bracketing (perhaps done by pausing or emphasis when speaking, or extra punctuation in writing) the original 'statement' is ambiguous, that is, it admits more than one interpretation. For the precision in mathematical argument which the framework of logic helps to achieve, such ambiguity has to be avoided and this is one of the features we shall build into our syntax.

Exercise 2.1

Which, if any, of the following statements is ambiguous?

(a) If it is snowing and the bus doesn't come, then I'll be late for work.

(b) If it is snowing then the bus doesn't come and I'll be late for work.

Solution

We think that (a) is unambiguous, but that (b) is possibly ambiguous. In the context of everyday life, we would normally interpret (b) as saying that 'the bus doesn't come' and 'I'll be late for work' are both consequences of 'it is snowing'. But it is also possible as interpreting it as saying 'if it is snowing then the bus doesn't come' and 'I'll be late for work', so that I'll be late for work regardless of whether it is snowing!

The meaning of (b) could be made clearer by the insertion of some punctuation, like a comma, in an appropriate place.

With considerations like these in mind, we shall define our formal statements as follows. First we shall specify the *formal language*, that is, the symbols from which strings can be formed. We shall always allow brackets – these will be needed to avoid ambiguity. We shall specify a set P of basic statements, called *propositional variables*. From these we can build more complex statements by joining statements together using brackets and symbols in a set S of *connectives*, which are going to represent ways of connecting statements to each other, like $\vee$ for 'or' and other symbols mentioned earlier. We can take any symbols we like for the propositional variables, so long as these symbols don't clash with those used for the brackets and the connectives. To make life easier, we shall adopt the following convention for the symbols we'll use.

Later in this chapter, we shall allow an extra sort of symbol called a *propositional constant*.

> ***Convention for variables***
>
> We shall normally use individual lower case letters like $p, q, r, s, \ldots$ and subscripted letters like $p_0, p_1, p_2, \ldots, p_n, \ldots$ for our propositional variables. Distinct letters or subscripts give us distinct symbols. When we don't specify the set P of propositional variables in a precise way, we shall use p, q, r and so on to represent different members of the set.

Use of symbols like these to represent variable quantities is, of course, very standard in normal mathematics.

Our formal version of statements, which we'll call *formulas*, is given by the following definition.

> ***Definition Formula***
>
> Let P be a set of propositional variables and let S be the set of connectives $\{\wedge, \vee, \neg, \rightarrow, \leftrightarrow\}$. A *formula* is a member of the set $Form(P, S)$ of strings of symbols involving elements of P, S and brackets (and) formed according to the following rules.
>
> (i) Each propositional variable is a formula.
>
> (ii) If θ and ψ are formulas, then so are
>
> $$\neg\theta \qquad (\theta \wedge \psi) \qquad (\theta \vee \psi) \qquad (\theta \rightarrow \psi) \qquad (\theta \leftrightarrow \psi)$$
>
> (iii) All formulas arise from finitely many applications of (i) and (ii).
>
> If we use a different set S of connectives, for instance just $\{\vee, \rightarrow\}$, then clause (ii) is amended accordingly to cover just these symbols.

In many books the phrase *well-formed formula* is used instead of formula. The 'well-formed' emphasizes that the string has to obey special construction rules.

So all formulas are finitely long.

The intended meanings of the connectives are as follows: $\wedge$ will be interpreted by 'and', $\vee$ by 'or', $\neg$ by 'not', $\rightarrow$ by 'implies' and $\leftrightarrow$ by 'if and only if'. With these intended meanings, you can see why clause (ii) of the definition uses $\wedge$, $\vee$, $\rightarrow$ and $\leftrightarrow$ to connect together two formulas, while $\neg$ connects with only one. The brackets used in clause (ii) are a very important part of the definition, playing a crucial part in making it possible to interpret formulas in an unambiguous way.

Taking the set P of propositional variables to be $\{p, q, r\}$, each of the following strings is a formula:

$$q \qquad (p \vee q) \qquad \neg\neg(p \vee q) \qquad (\neg p \wedge (q \rightarrow r)).$$

Let's check that each of these is a formula. The symbol q is a propositional variable, so is a formula by clause (i) of the definition. Each of the symbols p and q are propositional variables and thus formulas by clause (i), so the string $(p \vee q)$ is a formula by clause (ii). A use of the $\neg\theta$ part of clause (ii), taking θ to be $(p \vee q)$, gives that $\neg(p \vee q)$ is a formula; and one more use of the $\neg\theta$ part of clause (ii), this time taking θ to be $\neg(p \vee q)$, gives that $\neg\neg(p \vee q)$ is a formula. Lastly, as p, q, r are formulas, $\neg p$ is a formula by clause (ii), $(q \rightarrow r)$ is a formula by clause (ii), and so $(\neg p \wedge (q \rightarrow r))$ is a formula by clause (ii).

On the other hand, none of the following strings are formulas:

$*$	($*$ is not an allowed symbol)
$\rightarrow$	(the only single symbol formulas consist of a propositional variable)
$q\neg$	(there's nothing following the $\neg$)
$p \vee q$	(any formula using $\vee$ must also have some brackets)
$((\neg r \wedge q))$	(with just one $\wedge$ we can only have one pair of brackets.)

There are usually several reasons why a string fails to be a formula. In each case, we've just given a single one.

For these last two examples of non-formulas it is tempting to say that we know what they are supposed to represent, so let's call them formulas. But they don't conform to our strict rules – for many purposes in this book, you

We can't promise a prize to anyone who spots us forgetting brackets in what we say is a formula. But please let the author know about it.

should regard formulas as capable of being recognized and manipulated by a machine, and we shall keep the instructions for such a machine simple by careful use of brackets. These last two examples can be turned into formulas doubtless expressing correctly what was intended by suitable bracketing as

$$(p \vee q) \quad \text{and} \quad (\neg r \wedge q).$$

The use of brackets is a vital part of how we avoid ambiguity. We discussed earlier the 'statement'

'it is snowing or the bus doesn't come and I'll be late for work'.

A corresponding string is

$$(p \vee q \wedge r)$$

and this is not a formula, for instance because any formula containing one $\vee$ and one $\wedge$ would have to have two pairs of brackets (), rather than the one pair in the string. We discussed the two obvious ways of bracketing the statements 'it is snowing', 'the bus doesn't come','I'll be late for work' together, essentially as one of $((p \vee q) \wedge r)$ and $(p \vee (q \wedge r))$. The moral is that brackets matter a lot.

Exercise 2.2

Explain why each of the following are formulas, taking the set P of propositional variables to include p, q, r, s.

(a) $(r \leftrightarrow \neg s)$

(b) $((r \to q) \wedge (r \vee p))$

(c) $\neg\neg p$

Exercise 2.3

Explain why each of the following is not a formula, taking the set P of propositional variables to include p, q.

(a) $p \leftrightarrow q)$

(b) $(p \,\&\, q)$

(c) $\neg(p)$

(d) $(\neg p)$

(e) $(p \wedge \vee q)$

It's all very well asking you to show that a short string of symbols is a formula – we hope that you had no problem doing this in the last exercise. But for a long string, we really do need something systematic. Likewise it is, we hope, obvious that strings like $(p\neg$ and $(p \wedge q \vee r)$ are *not* formulas. But can we nail down why they are not formulas in a way that will then cope with a long string? We will answer the important question of how one tests whether a string of symbols is a formula first at a fairly informal level and in greater detail later in the section.

The issue of checking whether a string of symbols conforms to given construction rules is of considerable practical importance, for instance in many uses of computers. The details of how to do such checks are normally quite complicated and in this book we don't want them to get in the way of our main objective, which is to investigate the properties of strings which *are* formulas.

Obviously our answer must take account of the definition of a formula.

If a string consists of just a single symbol, then the string is a formula precisely when this symbol is a propositional variable – no string in the sequence can be empty and if clause (ii) has been used, the resulting string consists of more than one symbol.

When a string ϕ contains more than one symbol, it can only be a formula if it is one of the forms $\neg\theta$, $(\theta \wedge \psi)$, $(\theta \vee \psi)$, $(\theta \to \psi)$, $(\theta \leftrightarrow \psi)$, for some shorter strings θ, ψ (which of course also have to be formulas). Whichever of the forms ϕ is, the connective that you can see written down in the list we've just given, rather than any connectives that are hidden within the strings θ and ψ, is given a special name, the *principal connective* of ϕ.

The length of a string will be very important when proving results about those strings which are formulas.

If $\neg$ is the principal connective, this would easily be identified by seeing it at the front (i.e. lefthand end) of the string; and that's the only circumstance under which $\neg$ can be the principal connective. If it is one of the other connectives, one way by which the principal connective can be identified is by looking at brackets. Each appearance of one of $\wedge$, $\vee$,$\to$ and $\leftrightarrow$ brings with it a pair of brackets $(\dots)$. Brackets give vital information about the way a formula has been constructed and constrain which strings can be formulas. For instance, it is pretty obvious that any formula contains an equal number of left brackets (and right brackets), so that any string for which this fails cannot be a formula. A special property of brackets which identifies the principal connective when it is one of $\wedge$, $\vee$,$\to$ and $\leftrightarrow$ can be expressed in terms of the number of brackets to its left. For each occurrence of these connectives, we look at

the number of ('s to its left minus the number of)'s to its left.

For instance, in the formula

$$((p \wedge r) \to (\neg q \vee r)),$$

this 'left minus right bracket count' for the $\wedge$ is 2, for the $\vee$ near the righthand end is also 2, and for the $\to$, which is the principal connective, is 1. What distinguishes the principal connective when it is one of $\wedge$, $\vee$,$\to$ and $\leftrightarrow$ is that its 'left minus right bracket count' equals 1, which accords with our example. As another example, in the formula

Remember that for this purpose we ignore the $\neg$s.

We shall justify these results about brackets later in this section.

$$\neg((p \to \neg(r \vee q)) \to p)$$

the 'left minus right bracket count' of the leftmost occurrence $\to$ is 2, for the $\vee$ it's 3, and for the rightmost occurrence of $\to$, which is the principal connective, it's indeed 1.

Actually, it can be shown that for formulas with more than one symbol whose principal connective isn't $\neg$, there is *exactly one* connective with 'left minus right bracket count' equal to 1, which helps show that strings like

$$p \wedge q \quad \text{and} \quad (p \wedge q \vee r)$$

aren't formulas. For the string $p \wedge q$ there is no connective for which the 'left minus right bracket count' equals 1, while for $(p \wedge q \vee r)$ this equals 1 for more than one connective, both the $\wedge$ and the $\vee$.

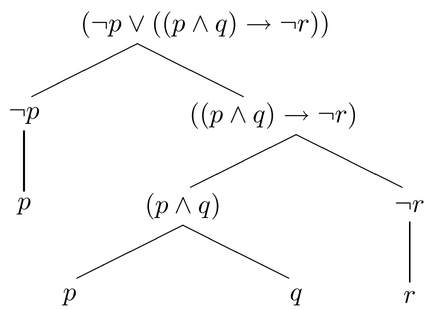
We haven't exhausted here all the ways in which a string can fail to be a formula, but will give an algorithm that detects all of these later in this section.

A very sketchy description of an algorithm for checking a string of symbols to see whether it is a formula is as follows.

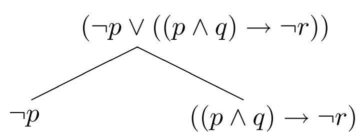
Look to see if it has a principal connective. If so, split it into the appropriate shorter string(s) and repeat the process: look for the principal connective for each of these shorter string(s) and split them up accordingly, and so on. In this way we analyse successively shorter and simpler strings until we reach strings consisting of just a single propositional variable – the shortest legal sort of string. If we don't trip up at any stage of the process, e.g. by failing to find a principal connective, and every analysis gets down to a propositional variable, our initial string was indeed a formula. In any other case, it was not a formula.

If the principal connective is $\neg$, so the string looks like $\neg\theta$, there's only one shorter string, namely θ. All the other connectives join together two substrings, e.g. the $\wedge$ in $(\theta \wedge \psi)$ joins together the substrings θ and ψ with outer brackets added. In the latter case, if the outer brackets are missing, the string can't be a formula.

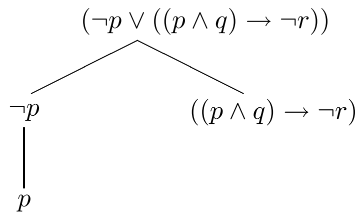
Let's illustrate the process for the string $(\neg p \vee ((p \wedge q) \to \neg r))$, which you can probably see is a formula, by the following diagram.



A diagram like this is called a *tree*, even though you might think that it looks like an upside-down tree! The string we are analysing is placed at the top of the diagram. The branches of the tree go downwards and it is no coincidence that, as our string is actually a formula, each branch ends with a propositional variable. How do we construct the tree? We first write down our original string $(\neg p \vee ((p \wedge q) \to \neg r))$ and attempt to locate its principal connective. If we find a candidate, then what we do next depends on whether it's one of $\{\wedge, \vee, \to, \leftrightarrow\}$ or it's $\neg$. In this case it's $\vee$, one of the first sort, so under the original string we write the two separate substrings which, when joined together with $\vee$ and a pair of outer brackets added, give the original string – here, these are the strings $\neg p$ and $((p \wedge q) \to \neg r)$. This gives the first stage of the diagram:



We join the top string to each of these smaller strings with a line to give a sense of them flowing directly from the top string. We repeat the process for each of these shorter strings. One of them begins with a $\neg$, so there's just the one string, namely p to write underneath it; and this is a propositional variable, so this (upside-down!) branch of the tree has successfully stopped at a propositional variable. The diagram now looks like



We now do the analysis of the other string $((p \wedge q) \to \neg r)$, finding its principal connective to be the $\to$, and we hope that by now you can see how we obtained the full diagram.

Exercise 2.4

Construct a similar sort of tree for each of the following strings.

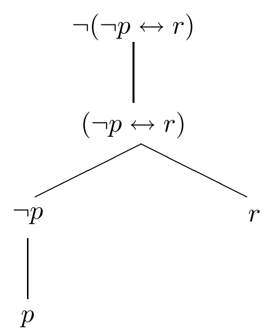
(a) $\neg(\neg p \leftrightarrow r)$

(b) $((p \wedge r) \to (\neg p \leftrightarrow q))$

(c) $\neg((\neg r \vee (r \wedge \neg p)) \leftrightarrow \neg\neg\neg q)$

Solution

(a)



(b) Not given.

(c) Not given.

As we have already observed, in this sort of diagram the constituent parts all are formulas and the branches all end with a propositional variable. It is helpful to have a name for these constituent parts: we call them *subformulas* of the original formula. More formally, we have the following definition.

Definition Subformula

For all formulas ϕ, their *subformulas* are defined as follows, exploiting the construction rules for formulas.

1. If ϕ is atomic, then ϕ is the only subformula of itself.
2. If ϕ is of the form $\neg\psi$, then the subformulas of ϕ are ϕ and all subformulas of ψ.
3. If ϕ is one of the forms $(\theta \wedge \psi)$, $(\theta \vee \psi)$, $(\theta \to \psi)$ and $(\theta \leftrightarrow \psi)$, then the subformulas of ϕ are ϕ, all subformulas of θ and all subformulas of ψ.

So the subformulas of $(\neg p \vee ((p \wedge q) \to \neg r))$ are

$$(\neg p \vee ((p \wedge q) \to \neg r)), \quad \neg p, \quad p, \quad ((p \wedge q) \to \neg r),$$
$$(p \wedge q), \quad q, \quad \neg r, \quad r.$$

We gave the tree diagram for this formula on page 24. Its subformulas are all the formulas involved at some stage of its construction.

Note that p occurs as a subformula in more than one place in the original formula, but we need only list it once as a subformula.

Exercise 2.5

Give the subformulas of each of the following formulas (which appeared in Exercise 2.4).

(a) $\neg(\neg p \leftrightarrow r)$

(b) $((p \wedge r) \to (\neg p \leftrightarrow q))$

(c) $\neg((\neg r \vee (r \wedge \neg p)) \leftrightarrow \neg\neg\neg q)$

We shall now tighten up on some of the details of our sketchy algorithm. First we shall look at an example of how to prove certain sorts of result about formulas, in particular the result that any formula ϕ contains an equal number of left brackets (and right brackets). Although we hope that this is somehow obvious from clause (ii) of the definition of a formula, it is instructive to see how to justify it in a more formal way – we shall need this style of argument several times later in the book to justify much less obvious results about formulas! Our challenge here is to prove something about *all* formulas ϕ, however complex and long they are. The way we shall proceed is by mathematical induction on the *length* of ϕ. There are several sensible measures of the length of a formula ϕ, for instance the total number of symbols in it or the height (that is, length of the longest branch) in its construction tree. Our preferred measure of length is the number of occurrences of connectives in ϕ, so that the length of the formula

We shall say 'the number of connectives' for short.

$$((q \wedge \neg\neg r) \wedge (p \vee (r \to \neg q)))$$

is 7 (consisting of 3 $\neg$s, 2 $\wedge$s, 1 $\vee$ and 1 $\to$). The smallest possible length of a formula is then 0, when ϕ is just a propositional variable p, for some p – the definition of formula allows no other possibility.

The structure of this sort of proof is as follows. Show first that the result holds for all formulas of length 0 – the basis of the induction. Then we do the inductive step: assume that the result holds for all formulas of the given type with length $\leq n$ – this is the induction hypothesis for n – and from this show that it holds for all formulas of length $\leq n+1$. As we are assuming that the hypothesis holds for all formulas of length $\leq n$, this boils down to showing that the hypothesis holds for formulas whose length is exactly $n+1$. By the principle of mathematical induction the hypothesis then holds for formulas of all lengths, i.e. all formulas.

You might like to think why our induction hypothesis isn't simply that the result holds for all formulas of length exactly equal to n, rather than $\leq n$. The reason for this will become clear soon!

Let us now use this method of proof to establish the following theorem about brackets.

Theorem 2.1

Any formula ϕ contains an equal number of left brackets (and right brackets).

Proof

Our induction hypothesis is that all formulas of length $\leq n$ contain an equal number of left and right brackets.

If ϕ is a formula of length 0, it can only be a propositional variable, thus containing an equal number, namely zero, of left and right brackets. Thus the hypothesis holds for $n = 0$.

Now suppose that the result holds for all formulas of length $\leq n$. To show from this that the result holds for all formulas of length $\leq n+1$, all that is needed is to show that it holds for formulas of length $n+1$, as those of shorter length are already covered by the induction hypothesis for n. So let ϕ be such a formula of length $n+1$. As ϕ has at least one connective, it cannot be simply a propositional variable, so must be a formula by an application of clause (ii) in the definition of formula, that is, it must be of one of the five forms

$$\neg\theta, \quad (\theta \wedge \psi), \quad (\theta \vee \psi), \quad (\theta \rightarrow \psi), \quad (\theta \leftrightarrow \psi)$$

where θ and ψ are formulas. We must deal with each of these possible forms. In all five cases, as ϕ has $n+1$ connectives, θ and ψ have at most n connectives, so that the inductive hypothesis will apply to them.

If ϕ is of the form $\neg\theta$, then θ has exactly n connectives. If ϕ is of the form $(\theta \wedge \psi)$, the $\wedge$ accounts for one of the $n+1$ connectives in ϕ, leaving the remaining n connectives to be distributed somehow between θ and ψ.

Case: ϕ is of the form $\neg\theta$

As θ has length n, by the induction hypothesis θ contains an equal number, say k, of left and right brackets. The formation of the string $\neg\theta$ from θ doesn't add further brackets, so that this form of ϕ also contains an equal number, namely k, of left and right brackets.

Case: ϕ is of the form $(\theta \wedge \psi)$

As both θ and ψ have length $\leq n$, by the induction hypothesis θ contains an equal number, say k, of left and right brackets, while ψ contains an equal number, say j, of left and right brackets. The formation of the string $(\theta \wedge \psi)$ from θ and ψ adds an extra left bracket and an extra right bracket to those in θ and ψ, so that this form of ϕ contains an equal number, namely $k+j+1$, of left and right brackets.

It can be shown that the result holds for ϕ of each of the three remaining forms, completing the inductive step. It follows by mathematical induction that the result holds for all $n \geq 0$, that is for all formulas. ■

Exercise 2.6

Fill in the gaps in inductive step of the proof above for the cases when ϕ is of one of the forms $(\theta \vee \psi)$, $(\theta \rightarrow \psi)$, $(\theta \leftrightarrow \psi)$.

Solution

We hope that this is seen as essentially trivial, simply replacing the $\wedge$ in the argument for the case when ϕ is of the form $(\theta \wedge \psi)$ by, respectively, $\vee$, $\rightarrow$ and $\leftrightarrow$. In all but the most fastidious circles, one would merely complete the proof given above of the theorem by saying that the other cases are similar to that of $(\theta \wedge \psi)$!

Exercise 2.7

(a) Use mathematical induction on the length of a formula to show that the number of occurrences of the symbol $\wedge$ in a formula ϕ is less than or equal to the number of left brackets (in the formula.

(b) Does the result of part (a) hold for the symbol $\neg$? Justify your answer.

Recall that our preferred measure of length of a formula is the number of occurrences of connectives in it.

We shall now describe a more comprehensive algorithm for checking whether a string of symbols is a formula. For simplicity in most of the rest of the section, we shall suppose that the language used involves only two propositional variables p and q, the connective $\wedge$ and brackets. It is very straightforward to extend our algorithm to cope with richer languages. There are several possible algorithms and we shall go for one which treats the brackets as the crucial component. Consider the following string, which is a formula:

$$(((p \wedge q) \wedge p) \wedge (q \wedge p)).$$

An undercurrent in the development of the subject is whether there is an algorithm for generating true statements of mathematics. For this to be remotely feasible, we need an algorithm for checking whether a string of symbols is a statement, hence our interest in this algorithm for a very simple language.

By Theorem 2.1, the number of left brackets in a formula equals the number of right brackets. We can check this for the formula above by moving along the string from left to right keeping a count of the difference between the numbers of left and right brackets in the following way. Start at the lefthand end of the string with the count at 0. Whenever we meet a left bracket, we add 1 to the count. When we meet a right bracket, we subtract 1 from the count. In this way we associate a number with each bracket in the string, as follows:

$$\underset{1}{(}\underset{2}{(}\underset{3}{(}p \wedge q\underset{2}{)} \wedge p\underset{1}{)} \wedge \underset{2}{(}q \wedge p\underset{1}{)}\underset{0}{)}$$

We shall call the number associated with each bracket in this way its *bracket count*. The final bracket has bracket count 0, which confirms that the number of left brackets equals the number of right brackets, as we would expect for a formula. Furthermore, the bracket count is greater than zero for any bracket before this final one, where it is 0. These properties apply to all formulas, not just this one, and can be proved using mathematical induction on the length of formulas. Of course, these properties by themselves don't ensure that the string is a formula.

E.g. consider the string

$$\underset{1}{(}p\underset{0}{)}$$

Another important feature of the bracket count is that it helps identify which occurrence of the connective $\wedge$ is the principal connective of the formula. In our example, observe that the principal connective happens to follow a bracket with a count of 1. This is not a coincidence. In any formula in our restricted language containing at least one occurrence of $\wedge$, there will be *exactly one* occurrence which follows a bracket with a count of 1 and this will be the principal connective. This principle works for the further example

Earlier we described this occurrence as having a 'left minus right bracket count' of 1.

$$\underset{1}{(}\,q\wedge\underset{2}{(}\underset{3}{(}\,p\wedge p\,\underset{2}{)}\wedge\, q\,\underset{1}{)}\underset{0}{)}$$

where the relevant bracket is a left bracket (rather than a right bracket) as in our first example. Let's try to explain informally why the principle holds in general.

A formula containing an $\wedge$ will be of the form $(\phi \wedge \psi)$, where the $\wedge$ which is the principal connective is the one that we can see between the ϕ and the ψ – these subformulas might, of course, contain other occurrences of $\wedge$. If the first subformula ϕ contains brackets, then its bracket count is greater than zero for any bracket before its final one, where the count is 0. Thus in the formula $(\phi \wedge \psi)$, which has an extra left bracket at its lefthand end, all the bracket counts for the subformula ϕ increase by 1. That means that the bracket count for $(\phi \wedge \psi)$ looks something like this:

$$\underset{1}{(}\overbrace{\underset{2}{(}\ldots(\ldots\underset{1}{)}}^{\phi}\wedge\overbrace{\underset{2}{(}\ldots(\ldots\underset{1}{)}}^{\psi}\underset{0}{)}$$

It starts by going straight from 1 to 2 as ϕ is entered. It then first goes back to 1 at the final bracket of ϕ, just before the principal connective, as required. If the subformula ψ contains brackets, the bracket counts increase – all the counts for ψ on its own also increase by 1, so the count for $(\phi \wedge \psi)$ only gets back to 1 at the final bracket of ψ, which is then followed by the final bracket of $(\phi \wedge \psi)$.

We leave it to you to think about the cases when one or both of ϕ and ψ contain no brackets, meaning that it is simply a propositional variable.

The algorithm will then be as follows, starting with the string which one wishes to test to see whether it is a formula. At any stage when the algorithm declares that the string is a formula, the process halts. Similarly it halts when the algorithm declares that the string is not a formula.

(i) Test the lefthand symbol of the string.

- (a) If it is one of p and q, check if this is the only character in the string. If this is so, the string is a formula; if not, then it isn't a formula.
- (b) If it is $\wedge$ or a right bracket), then the string isn't a formula.
- (c) If it is a left bracket (, then proceed to step (ii).

(ii) Check whether the string ends in a right bracket). If not, the string is not a formula. Otherwise proceed to step (iii).

(iii) Compute the bracket count. Moving along the string from left to right, locate the first occurrence of $\wedge$ following a bracket with count 1. If there is no such occurrence of $\wedge$, the string is not a formula. Otherwise, proceed to step (iv).

(iv) Use this occurrence of $\wedge$ to split the string into two substrings: one consisting of all the symbols to the left of the $\wedge$ except for the initial left bracket (; and the other consisting of all the symbols to the right of the $\wedge$ except for the final right bracket).

(v) Now apply the algorithm starting with (i) to both of these substrings. If both substrings are formulas, then the string is a formula.

Note that at stage (v), the substrings are shorter than the original (finite!) string, so that the algorithm will stop with a result after a finite number of steps.

Exercise 2.8

How does our algorithm detect the case when there are two occurrences of $\wedge$ which follow a bracket count of 1?

Of course, in such a case the string is not a formula.

Exercise 2.9

Adapt our algorithm for strings built up using the propositional variables p, q and the connective $\wedge$ so that it tests strings which might also include the connective $\neg$.

Now that we have a definition of formula, we can look at how to interpret formulas and discuss their truth or falsity in an interpretation. This is the subject of the next section.

Further exercises

Exercise 2.10

Show that in all formulas θ built up using the propositional variables p, q and the connective $\wedge$, the bracket count is greater than zero for any bracket of θ before its final one where the count is 0.

Exercise 2.11

Suppose that formula ϕ is built up using only $\wedge$ and $\vee$ and has connective length n. What can you say about the number of subformulas of ϕ? What can be said if ϕ might include the connective $\neg$ as well as $\wedge, \vee$?

Exercise 2.12

Show that in all formulas θ built up using the propositional variables p and q using the connective $\wedge$ and containing at least one occurrence of $\wedge$, there is exactly one occurrence of $\wedge$ which follows a bracket with a count of 1.

2.3 The interpretation of propositional formulas

We shall now describe how to give meaning to the formal language, giving what is called its semantics. Recall that we introduced the simplest (shortest!) sort of formula, a propositional variable, by saying that it was intended to stand for a basic component proposition, like 'f is a continuous function'. In normal mathematics, the truth of this will depend on whether the f we are given is indeed a continuous function and, for that matter, what we mean by a continuous function. But for propositional calculus, this level of detail of how a propositional variable is interpreted is much greater than we shall need in this chapter. For purposes like deciding whether one statement or formula is a consequence of others within the propositional calculus, all we shall need to know about each propositional variable is whether, in a particular set of circumstances, it is true or false. Once we have specified how to interpret the connectives, we can then say how the truth or falsity of more complicated formulas depends on that of the propositional variables, which are the basic building blocks. This in turn will allow us to say whether one formula is a consequence of others.

We shall be much more interested in what each basic proposition expresses when we look at the predicate calculus.

Hidden in the preamble above and implicit in earlier discussions in the book is an important decision. Under a given set of circumstances, a statement is either true or false – one or the other, and no sort of half-truth in between. We hope that this seems perfectly reasonable. In everyday mathematics, a statement like 'f is a continuous function' is just one of true or false, depending on the f we are given. However, there are circumstances where it might make sense to describe the truth of a statement in a less black and white way, for instance giving a probability that the statement is true; and one of the ways in which you could extend your knowledge beyond this book is by learning about other ways of analysing what is meant by truth. *For the rest of this book*, our standard measure of the truth of a statement will be in terms of the two distinct values 'true' and 'false'. We shall describe each of these as a *truth value* and abbreviate them by T for 'true' and F for 'false'. So the set of truth values is the two element set $\{T, F\}$.

We have talked informally about knowing whether, in a particular set of circumstances, each propositional variable is true or false. More formally and elegantly, this set of circumstances is a function $v\colon P \longrightarrow \{T, F\}$, where P is the set of propositional variables in our language. The function v gives a truth value to each propositional variable in P, thus describing the set of circumstances. We will explain how to extend such a function v so that it assigns a truth value to each formula built up from P using connectives in a set S – the function so obtained will be called a *truth assignment*. A key step is to specify how to interpret the connectives. For each of these, we shall explain how the truth of a formula ϕ with it as principal connective depends on the truth of the subformulas it connects to form ϕ. We shall look at each of the connectives introduced so far in the book, namely $\neg$, $\wedge$, $\vee$, $\rightarrow$ and $\leftrightarrow$, the intended meanings of which we have already said are, respectively, 'not', 'and', 'or', 'implies' and 'if and only if'.

The issue of other reasonable connectives used in everyday discourse is delayed until Section 2.5.

$\neg$ (negation)

A formula of the form $\neg\theta$ for some formula θ with principal connective $\neg$ is called the *negation* of θ. We shall specify how its truth value is to be related to the truth value of θ. As our intended way of interpreting $\neg$ is as 'not', we want $\neg\theta$ to have the value F (false) when θ has the value T (true) and the value T when θ has the value F, i.e. $\neg\theta$ will have the opposite value to that assigned to θ. We can summarize this by the following table.

We may also sometimes refer to a formula of the form $\neg\theta$ as a *negation.*

θ	$\neg\theta$
T	F
F	T

$\wedge$ (conjunction)

A formula of the form $(\theta \wedge \psi)$ for some formulas θ, ψ with principal connective $\wedge$ is called the *conjunction* of θ and ψ. Each of the formulas θ and ψ is called a *conjunct* of $(\theta \wedge \psi)$. Our intended way of interpreting $\wedge$ is as 'and', so we shall assign $(\theta \wedge \psi)$ the value 'true' exactly when both θ and ψ are assigned the value 'true'. We can summarize this by the following table.

We may also sometimes refer to a formula of the form $(\theta \wedge \psi)$ as a *conjunction.*

So if one or both of θ and ψ are false, then so is $(\theta \wedge \psi)$.

θ	ψ	$(\theta \wedge \psi)$
T	T	T
T	F	F
F	T	F
F	F	F

This sort of table, giving the truth values of a formula constructed from some of its subformulas for all possible combinations of truth values of these subformulas, is called a *truth table.* Here the formula $(\theta \wedge \psi)$ is given in terms of the subformulas θ and ψ. There are four combinations of truth values for θ and ψ, so this truth table has 4 rows. Our earlier table for negation gave the truth values of $\neg\theta$ in terms of the values of the subformula θ, so this truth table only required 2 rows.

$\vee$ (disjunction)

A formula of the form $(\theta \vee \psi)$ for some formulas θ, ψ with principal connective $\vee$ is called the *disjunction* of θ and ψ. Each of the formulas θ and ψ is called a *disjunct* of $(\theta \vee \psi)$. Our intended way of interpreting $\vee$ is as 'or', but unlike 'not' and 'and' earlier, we run into the problem that there is more than one way of using 'or' in English. One way, called the *exclusive* 'or', makes $(\theta \vee \psi)$ true when exactly one of θ and ψ is true – the truth of one of them excludes the truth of the other. For instance, many restaurants offer a fixed price menu with a choice of dishes for each course. The choice for each course is to be read as a disjunction with the exclusive use of 'or' – you can have any one of the soup, terrine and prawn cocktail, but only one. Another use of 'or' is in what is called an *inclusive* way, where $(\theta \vee \psi)$ is true when one or both of θ and ψ are true. For instance, a common sort of argument in maths is along the lines of 'if x or y are even integers, then xy is even', where 'x or y are even' includes the case that both x and y are even. Because this way of using

We may also sometimes refer to a formula of the form $(\theta \vee \psi)$ as a *disjunction.*

'or' is pretty well standard in mathematics, we shall choose to interpret $\vee$ in the *inclusive* way, as given by the following truth table.

θ	ψ	$(\theta \vee \psi)$
T	T	T
T	F	T
F	T	T
F	F	F

$\rightarrow$ (implication)

A formula of the form $(\theta \rightarrow \psi)$ for some formulas θ, ψ with principal connective $\rightarrow$ is called an *implication*. Our intended way of interpreting $\rightarrow$ is as 'implies', or 'if ... then', which suggests some of the rows of the truth table in the following rather backhanded way. In normal use of 'if ... then' in English, from being told that 'if θ then ψ' is true and that θ is true, we would expect ψ to be true. Likewise if we are told that θ is true and ψ is false, then 'if θ then ψ' would have to be false. This then settles two of the rows of the truth table, as follows:

The formula θ is called the *antecedent* and ψ the *consequent* of the implication.

θ	ψ	$(\theta \rightarrow \psi)$
T	T	T
T	F	F
F	T	?
F	F	?

It may not be immediately obvious from normal English how to fill in the remaining two rows of the table, covering the cases where θ is false. It is a constraint of the process of making a simple model of this fragment of natural language and argument that we have to make some sort of decision about the truth value of $(\theta \rightarrow \psi)$ when θ is false, and our decision is to make $(\theta \rightarrow \psi)$ true on these rows, giving the following truth table.

θ	ψ	$(\theta \rightarrow \psi)$
T	T	T
T	F	F
F	T	T
F	F	T

This table might be made more memorable by thinking of it as saying that $(\theta \rightarrow \psi)$ is false only when θ is true and ψ is false – surely circumstances when 'θ implies ψ' has to be false.

The decision we have taken about the bottom two rows in this table is consistent with the way we handle implication in everyday mathematics. We frequently state theorems in the form 'if ... then', for instance the following theorem:

'for all $x \in \mathbb{R}$, if $x > 2$, then $x^2 > 4$'.

We hope that this result strikes you not only as correct (which it is!) but as a familiar way of expressing a host of mathematical results using 'if ... then'. Given that we regard this result as true, we must surely also regard

'if $x > 2$, then $x^2 > 4$'

as being true for each particular x in $\mathbb{R}$, and we want the truth table for 'if ... then' to reflect this. Taking some particular values for x, for instance 3, 1

Giving the value 'true' to the statement 'if $x > 2$, then $x^2 > 4$' even when the particular value of x makes $x > 2$ false is a fair reflection that there is a correct proof of 'if $x > 2$, then $x^2 > 4$'.

and -5, this means we want our truth table for 'if ... then' to give the value T in each of the following circumstances:

when $x = 3$, both $x > 2$ and $x^2 > 4$ have the value T;

when $x = 1$, both $x > 2$ and $x^2 > 4$ have the value F;

when $x = -5$, $x > 2$ has the value F but $x^2 > 4$ has the value T.

These cases correspond to all the rows of our truth table for $\rightarrow$ which result in the value T.

The connective $\rightarrow$ with its intended meaning of 'implies' is perhaps the most important of the connectives we have introduced. This is because a major use of our formal language is to represent and analyze mathematical arguments and theorems, and a salient feature of these is the use of implication both to state and prove results.

Note that the truth of the statement $(\theta \rightarrow \psi)$ does not necessarily entail any special relationship between θ and ψ, for instance that in some sense θ causes ψ. To test your understanding of the truth table of $\rightarrow$, try the following exercise.

Exercise 2.13

The British psychologist Peter Wason (1924–2003) devised a famous experiment involving people's understanding of 'if ... then ...' as follows. The experimenter lays down four cards, bearing on their uppermost faces the symbols A, B, 2 and 3 respectively. The participants are told that each card has a letter on one side and a number on the other side. Their task is to select just those cards that they need to turn over to find out whether the following assertion is true or false: 'If a card has an A on one side, then it has a 2 on the other side.' Which cards should be turned over?

Solution

In the original experiment and in subsequent trials, most people selected the A card and, perhaps, the 2 card. Surprisingly they failed to select the 3 card. According to Wason's obituary in the Guardian (25th April 2003), the experiment 'has launched more investigations than any other cognitive puzzle. To this day – and Wason's delight – its explanation remains controversial.'

The 2 card doesn't need to be turned over to test the assertion!

$\leftrightarrow$ (bi-implication)

A formula of the form $(\theta \leftrightarrow \psi)$ for some formulas θ, ψ with principal connective $\leftrightarrow$ is called a *bi-implication* of θ and ψ. Our intended way of interpreting $\leftrightarrow$ is as 'if and only if', so we shall assign $(\theta \leftrightarrow \psi)$ the value 'true' exactly when the truth value of θ matches that of ψ. We can summarize this by the following truth table.

Why do you think that $\leftrightarrow$ is formally described as 'bi-implication'?

θ	ψ	$(\theta \leftrightarrow \psi)$
T	T	T
T	F	F
F	T	F
F	F	T

Exercise 2.14

We interpret the formula $(\theta \leftrightarrow \psi)$ as 'θ if and only if ψ'. Write down two formulas involving the connective $\rightarrow$, one which represents 'θ if ψ' and the other representing 'θ only if ψ'. Often in everyday maths we interpret $(\theta \leftrightarrow \psi)$ as 'θ is a necessary and sufficient condition for ψ'. Which of your formulas represents 'θ is a necessary condition for ψ' and which represents 'θ is a sufficient condition for ψ'?

And say which is which!

An important point to note about the truth tables for the connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$ and $\leftrightarrow$ is that these are not simply conventions for the purposes of this book when working out the truth values of formulas under an interpretation of the formal language. They also reflect how in normal mathematics we determine the truth of statements made involving their standard interpretations as, respectively, 'not', 'and', 'or', 'implies' and 'if and only if'. In particular when we discuss our formalization of statements and arguments in this book, in 'normal' language (what we called earlier the metalanguage), we shall use these standard interpretations.

Now that we have said how to interpret each of the connectives, we can turn to the interpretation of formulas in general. Our aim is to define a *truth assignment*, that is a special sort of function from the set of all formulas to the set $\{T, F\}$ of truth values, which turns particular truth values given to the propositional variables into a truth value for any formula built up from them, exploiting the truth tables of the connectives. We shall approach this by making more precise what we mean by exploiting these truth tables, starting with the truth table for $\wedge$.

Let $Form(P, S)$ be the set of all formulas built up from propositional variables in a set P using connectives in a set S which includes $\wedge$. We shall say that a function $v\colon\ Form(P, S) \longrightarrow \{T, F\}$ *respects the truth table of* $\wedge$ if

$$v((\theta \wedge \psi)) = \begin{cases} T, & \text{if } v(\theta) = v(\psi) = T, \\ F, & \text{otherwise,} \end{cases}$$

for all formulas $\theta, \psi \in Form(P, S)$. That is, the value of $v((\theta \wedge \psi))$ is related to those of $v(\theta)$ and $v(\psi)$ by the truth table for $\wedge$:

$v(\theta)$	$v(\psi)$	$v((\theta \wedge \psi))$
T	T	T
T	F	F
F	T	F
F	F	F

You can probably guess how we are going to exploit this definition. If we have $v(p) = v(q) = T$ and $v(r) = F$, where p, q, r are propositional variables, and v respects the truth table for $\wedge$, then $v((p \wedge q)) = T$ and $v(((p \wedge q) \wedge r)) = F$.

Likewise we say that v respects the truth tables of $\neg$ and respectively $\vee$ if

$$v(\neg\theta) = \begin{cases} F, & \text{if } v(\theta) = T, \\ T, & \text{if } v(\theta) = F, \end{cases}$$

and

$$v((\theta \vee \psi)) = \begin{cases} F, & \text{if } v(\theta) = v(\psi) = F, \\ T, & \text{otherwise,} \end{cases}$$

for all formulas $\theta, \psi \in Form(P, S)$.

In a similar way we can define that the function v respects the truth tables of other connectives.

In the next section we shall discuss other connectives besides $\neg$, $\wedge$, $\vee$, $\rightarrow$ and $\leftrightarrow$.

Exercise 2.15

Suggest definitions for v respects the truth tables of $\rightarrow$ and $\leftrightarrow$.

Solution

$$v((\theta \rightarrow \psi)) = \begin{cases} F, & \text{if } v(\theta) = T \text{ and } v(\psi) = F, \\ T, & \text{otherwise,} \end{cases}$$

and

$$v((\theta \leftrightarrow \psi)) = \begin{cases} T, & \text{if } v(\theta) = v(\psi), \\ F, & \text{otherwise,} \end{cases}$$

for all formulas $\theta, \psi \in Form(P, S)$.

We can now give the key definition giving the truth value of a formula under a given interpretation of the propositional variables it contains.

Definitions   Truth assignment

Let P be a set of propositional variables and S a set of connectives. A function $v\colon\ Form(P, S) \longrightarrow \{T, F\}$ is said to be a *truth assignment* if v respects the truth tables of all the connectives in S. We shall sometimes call $v(\phi)$ the *truth value of* ϕ *under* v.

We shall sometimes describe this v as a *truth assignment on* P.

If $v(\phi) = T$, we shall often say that 'v *makes* ϕ *true*' or that 'v *satisfies* ϕ'; and we adapt this terminology appropriately when $v(\phi) = F$.

Often we shall not be very specific about the sets P and S in this definition, and will rely on the context making it obvious what these sets are being taken to be.

It might appear to be very cumbersome, if not downright impossible, to give an example of a truth assignment v as we would have to give the value of $v(\phi)$ for every formula ϕ, however long and complicated. Fortunately a truth assignment v can essentially be described simply by giving the values of $v(p)$ for all propositional variables p. As v respects all relevant connectives, it can be shown that $v(\phi)$ is completely determined by these values of $v(p)$. Furthermore, for any choice of truth values for the propositional variables, there is a truth assignment v taking these given values on the propositional variables. This is the import of the following vital result.

Theorem 2.2

Let P be a set of propositional variables, let S be the set of connectives $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$ and let $v\colon P \longrightarrow \{T, F\}$ be a function. Then there is a unique truth assignment $\overline{v}\colon\ Form(P, S) \longrightarrow \{T, F\}$ such that $\overline{v}(p) = v(p)$ for all $p \in P$.

Proof

Existence can be demonstrated by defining $\overline{v}$ as follows, using what is called *recursion* on the length of a formula, exploiting the construction of a formula from subformulas of shorter lengths, with propositional variables as the basic building blocks with length 0.

If you are not familiar with recursion and the recursion principle, then you can simply take this theorem on trust or you can look at the details in e.g. Enderton [12].

For any formula of length 0, such a formula can only be a propositional variable p in P, in which case define $\overline{v}(p)$ to be $v(p)$.

Now suppose that $\overline{v}(\phi)$ has been defined for all formulas of length $\leq n$. Let ϕ be a formula of length $n+1$. As $n+1 > 0$, ϕ has a principal connective (which is unique), so is one of the forms $\neg\theta$, $(\theta \wedge \psi)$, $(\theta \vee \psi)$, $(\theta \rightarrow \psi)$ and $(\theta \leftrightarrow \psi)$, where θ and ψ are of length $\leq n$, so that both $\overline{v}(\theta)$ and $\overline{v}(\psi)$ have already been defined. Now define $\overline{v}(\phi)$ by using the appropriate row of the truth table for its principal connective with these values of $\overline{v}(\theta)$ and $\overline{v}(\psi)$.

This process, exploiting what's called the *recursion principle*, defines $\overline{v}(\phi)$ for all formulas ϕ and thus defines a function $\overline{v}\colon\ Form(P, S) \longrightarrow \{T, F\}$. Plainly the construction guarantees that $\overline{v}$ respects the truth tables of all the connectives in S, that is, $\overline{v}$ is a truth assignment.

We now need to prove that the function $\overline{v}$ is unique. We suppose that $v'\colon\ Form(P, S) \longrightarrow \{T, F\}$ is another truth assignment with $v'(p) = v(p)$ for all $p \in P$. We shall use mathematical induction to show that for all formulas ϕ of length $\leq n$, $\overline{v}(\phi) = v'(\phi)$, where $n \geq 0$. As every formula has a finite length, this will show that $\overline{v}(\phi) = v'(\phi)$ for all formulas ϕ, so that the functions $\overline{v}$ and v' are equal.

In general, two functions f, g are equal if they have the same domain A and the same effect on each element of the domain, i.e. $f(a) = g(a)$ for all $a \in A$.

Any formula of length 0 has to be a propositional variable p in P, in which case both $\overline{v}(p)$ and $v'(p)$ are, by definition, $v(p)$, and are thus equal.

Now suppose that for all formulas ϕ of length $\leq n$, $\overline{v}(\phi) = v'(\phi)$, and that ϕ is a formula of length $n+1$. Then ϕ is one of the forms $\neg\theta$, $(\theta \wedge \psi)$, $(\theta \vee \psi)$, $(\theta \rightarrow \psi)$ and $(\theta \leftrightarrow \psi)$, where θ and ψ are of length $\leq n$, so that

$$\overline{v}(\theta) = v'(\theta) \text{ and } \overline{v}(\psi) = v'(\psi).$$

In all cases, as $\overline{v}$ and v' are truth assignments, and so respect the truth tables of all the connectives in S, we then have $\overline{v}(\phi) = v'(\phi)$. By mathematical induction, we have $\overline{v}(\phi) = v'(\phi)$ for formulas of all lengths $n \geq 0$, i.e. for all formulas ϕ. ∎

One way of phrasing this result is that any assignment of truth values to the propositional variables of a language can be extended to a unique truth assignment. An important consequence of it is that the effect of a truth assignment v is completely determined by the values it gives to the propositional variables, as we claimed earlier. A full explanation of this is as follows.

Given a truth assignment v, look at the restriction w of the function v to the set P of propositional variables (which is a subset of the domain $Form(P, S)$ of v as each propositional variable is a formula). This means that w is the function from P to $\{T, F\}$ defined by $w(p) = v(p)$ for all $p \in P$. Our task is to show that the effect of v on *all* formulas is determined by the effect of this w just on propositional variables. By the result above, w can be extended to a unique truth assignment $\overline{w}$. This means that (a) $\overline{w}$ is a truth assignment, (b) $\overline{w}(p) = w(p) = v(p)$ for all $p \in P$ and (c) $\overline{w}$ is the only truth assignment with property (b). But the v we started with is a truth assignment and satisfies (b). So by (c) $\overline{w}$ is v.

We have used the letter w rather than the standard notation $v|_P$ for this restriction function in the hope that it will make this passage easier to read!

Now we know that a truth assignment v is determined by the values it gives to the propositional variables, we can look at some examples. In practice we are given the value of $v(p)$ for each propositional variable p. How do we then compute the truth value $v(\phi)$ for a formula ϕ? We exploit the fact that a truth assignment respects truth tables and the construction of ϕ from its subformulas. For a simple formula, the process is easy. For instance, suppose that v is a truth assignment with $v(p) = F, v(q) = T$ and that we want the value of $v((p \wedge (q \to \neg p)))$. As v respects truth tables, we have

$$v(\neg p) = T$$
$$v((q \to \neg p)) = T$$
$$v((p \wedge (q \to \neg p))) = F.$$

Exercise 2.16

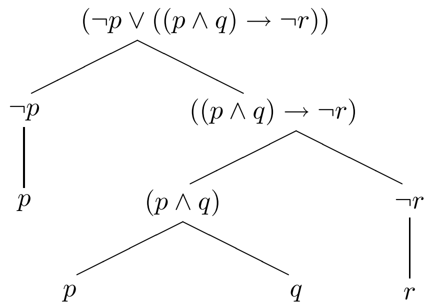
Suppose that v is a truth assignment with $v(p) = T, v(q) = F$. What is the value of $v((p \wedge (q \to \neg p)))$?

Solution

As v respects truth tables, we have

$$v(\neg p) = F$$
$$v((q \to \neg p)) = T$$
$$v((p \wedge (q \to \neg p))) = T.$$

What we have done for these simple formulas is essentially to build up to the truth value of the whole formula by working out the truth values of its subformulas. This method works just as well for more complicated formulas. Think of a formula in terms of its construction tree. Take for instance the formula $(\neg p \vee ((p \wedge q) \to \neg r))$ and the truth assignment v with $v(p) = T, v(q) = T, v(r) = F$. We constructed the tree for this formula on page 24 and reproduce it below for convenience.



For the given v, we work out $v(\psi)$ for all subformulas ψ of ϕ, starting from the simplest subformulas, namely the propositional variables appearing in ϕ, and work our way up the tree till we get up to $v(\phi)$. Here this gives

$$
\begin{aligned}
&v(p) = T\\
&v(q) = T\\
&v(r) = F\\
&v(\neg p) = F\\
&v((p \wedge q)) = T\\
&v(\neg r) = T\\
&v(((p \wedge q) \to \neg r)) = T\\
&v((\neg p \vee ((p \wedge q) \to \neg r))) = T.
\end{aligned}
$$

In practice, when working out $v(\phi)$ for a given v and ϕ, all one uses are the subformulas ψ of ϕ, rather than the whole tree, which saves some paper! But the principle is the same: starting with the values of $v(p)$ for the propositional variables that appear in ϕ, work out the values of $v(\psi)$ for progressively more complicated subformulas of ϕ until one obtains the value of $v(\phi)$ itself.

Of course, once one is familiar with the game, one can often take some shortcuts. In this example, once one has worked out that $v(\neg r) = T$, the truth table for $\to$ gives that $v(((p \wedge q) \to \neg r))$ has to equal T, irrespective of the value of $v((p \wedge q))$. Then, thanks to the truth table for $\vee$, $v((\neg p \vee ((p \wedge q) \to \neg r)))$ has to equal T, irrespective of the value of $v(\neg p)$. But be warned that such shortcuts cannot always be taken.

Exercise 2.17

Let v be the truth assignment defined on the set of propositional variables $\{p, q, r\}$ by $v(p) = T$, $v(q) = F$, $v(r) = F$. Find the truth value under v of each of the following formulas.

(a) $\neg q$

(b) $(\neg p \vee r)$

(c) $(p \leftrightarrow (\neg r \to s))$

(d) $((q \wedge (r \to \neg r)) \vee ((p \vee r) \leftrightarrow \neg q))$

(e) $(p \to ((\neg r \to p) \to (\neg q \vee r)))$

Solution

(a) As $v(q) = F$, we have $v(\neg q) = T$.

(b) The subformulas here are p, r, $\neg p$, and finally the formula $(\neg p \vee r)$ itself. The values of these subformulas under v are

$$\begin{aligned} &v(p) = T \\ &v(r) = F \\ &v(\neg p) = F \\ &v((\neg p \vee r)) = F. \end{aligned}$$

(c) Not a misprint for a change, but a trick question! As we have not specified the value of $v(s)$, we cannot work out the value of $v((p \leftrightarrow (\neg r \to s)))$.

(d) Working out the values of subformulas of

$$((q \wedge (r \to \neg r)) \vee ((p \vee r) \leftrightarrow \neg q)),$$

we have

$$\begin{aligned} &v(p) = T \\ &v(q) = F \\ &v(r) = F \\ &v(\neg r) = T \\ &v((r \to \neg r)) = T \\ &v((q \wedge (r \to \neg r))) = F \\ &v((p \vee r)) = T \\ &v(\neg q) = T \\ &v(((p \vee r) \leftrightarrow \neg q)) = T \\ &v(((q \wedge (r \to \neg r)) \vee ((p \vee r) \leftrightarrow \neg q))) = T. \end{aligned}$$

Perhaps easier to say after the event than to spot beforehand, once one has spotted that $v(((p \vee r) \leftrightarrow \neg q)) = T$, then thanks to the truth table for $\vee$, we must have

$$v(((q \wedge (r \to \neg r)) \vee ((p \vee r) \leftrightarrow \neg q))) = T,$$

irrespective of the value of $v((q \wedge (r \to \neg r)))$. This would have been a bit of a shortcut, but we suspect that sometimes looking for a shortcut might take up time that could have been used working out the truth value the slow way!

(e) Not given.

From now on, we will often not show the intermediate steps in working out $v(\phi)$ and as your confidence increases you might find yourself doing the same.

One fact which we hope seems obvious is that the truth value of a formula ϕ under a truth assignment v does not depend on the values of $v(p)$ for propositional variables p which do not appear in ϕ. Despite being obvious, it's worth seeing how to prove the result, which can be regarded as a consequence of the following exercise. This exercise provides useful practice in proving a result for all formulas of a certain sort by mathematical induction on the length of a formula.

Exercise 2.18

Let v and v' be truth assignments which take the same values for all propositional variables except p, i.e. $v(p) \neq v'(p)$ and $v(q) = v'(q)$ for all other propositional variables q. Show that $v(\phi) = v'(\phi)$ for all formulas ϕ built up using the connectives $\neg, \wedge, \vee$ in which the propositional variable p does not appear.

Solution

We shall prove this by mathematical induction on the length of such formulas ϕ. The induction hypothesis is that $v(\phi) = v'(\phi)$ for all formulas ϕ in which the propositional variable p does not appear, where ϕ has length $\leq n$.

Recall that our preferred measure of the length of a formula is the number of occurrences of connectives in it, here $\neg, \wedge, \vee$.

If ϕ is a formula of the given form, namely one in which p does not appear, with length 0, then it has to be of the form q, where q is a propositional variable other than p. Then we have $v(q) = v'(q)$, that is, $v(\phi) = v'(\phi)$ for this ϕ.

Now suppose that the result holds for all ϕ of the given form with length $\leq n$. To prove from this the induction hypothesis for $n+1$, it is enough to show that the result holds for any formula of the given form with length $n+1$. Let ϕ be such a formula. As ϕ has length $n+1 \geq 1$, ϕ contains at least one connective, so is of one of the forms $\neg\theta$, $(\theta \wedge \psi)$, $(\theta \vee \psi)$, for subformulas θ, ψ. The sum of the lengths of θ and ψ is n, so that both θ and ψ have length $\leq n$. Also as p does not appear in ϕ, it cannot appear in θ or ψ, so that the hypothesis can be used for both these subformulas. We must deal with each of the possible forms.

Case: ϕ is of the form $\neg\theta$

By the induction hypothesis $v(\theta) = v'(\theta)$, so that as v, v' are truth assignments

$$v(\neg\theta) = v'(\neg\theta),$$

that is, $v(\phi) = v'(\phi)$ for this form of ϕ.

Case: ϕ is of the form $(\theta \wedge \psi)$

By the induction hypothesis $v(\theta) = v'(\theta)$ and $v(\psi) = v'(\psi)$, so that as v, v' are truth assignments

$$v((\theta \wedge \psi)) = v'((\theta \wedge \psi)),$$

that is, $v(\phi) = v'(\phi)$ for this form of ϕ.

The case when ϕ is of the form $(\theta \vee \psi)$ is of course similar, completing the inductive step. The result for all $n \geq 0$, that is, for all formulas ϕ in which p does not appear, follows by mathematical induction.

The result of this last exercise confirms our intuition that the truth value of a formula ϕ under a truth assignment v depends only on the values v takes for the propositional variables in ϕ. This means that we can summarize the values ϕ can take under all possible truth assignments by looking only at the different truth assignments on the finitely many variables in it. How many of these latter assignments are there? This is the subject of the next exercise.

Exercise 2.19

(a) How many different truth assignments are there on the set of propositional variables $\{p, q, r\}$?

(b) How many different truth assignments are there on the set of propositional variables $\{p_1, p_2, \dots, p_n\}$, where n is a positive integer?

Solution

(a) For a truth assignment v, there are two choices for the value of $v(p)$. For each such choice there are then two choices for $v(q)$. Also for each choice of $v(p)$ and $v(q)$, there are also two choices for $v(r)$, giving a total of $2 \times 2 \times 2 = 2^3 = 8$ different truth assignments.

(b) Extending the reasoning above, by an easy use of mathematical induction, we can show that there are 2^n different truth assignments.

We can now summarize the truth value of a formula ϕ under each of the different truth assignments on the variables in it by what is called the *truth table* of ϕ. This extends the terminology we used for the truth tables describing how to compute truth values for each of the connectives. If the propositional variables in ϕ are amongst $p_1, p_2, \dots, p_n$, the table would have the form

Normally we would only list variables that are used in ϕ. But there are occasions when it is useful to give the table using more variables than actually appear in ϕ, hence our use of the word 'amongst'.

p_1	p_2	$\dots$	p_n	ϕ
T	T	$\dots$	T	?
T	T	$\dots$	F	?
$\vdots$	$\vdots$		$\vdots$	$\vdots$
F	F	$\dots$	F	?

where each row represents a truth assignment v giving particular truth values to each of $p_1, p_2, \dots, p_n$ and then gives the corresponding value of $v(\phi)$. As there are 2^n different truth assignments on the n propositional variables, the table would have 2^n rows.

Our first example is the formula $(\neg(p \vee q) \rightarrow (p \wedge q))$ using the propositional variables p, q, rather than p_1, p_2.

p	q	$(\neg(p \vee q) \rightarrow (p \wedge q))$
T	T	T
T	F	T
F	T	T
F	F	F

For instance, the second row of the truth table says that when p is given the value T and q the value F, the formula $(\neg(p \vee q) \rightarrow (p \wedge q))$ has the value T.

Equivalently, the second row says that if v is the truth assignment such that $v(p) = T$ and $v(q) = F$, then $v((\neg(p \vee q) \rightarrow (p \wedge q))) = T$.

It often helps one to record the truth values of the subformulas of ϕ in the table to enable one to compute the final values of ϕ itself. In the example above, this could have been recorded as follows.

p	q	$(p \vee q)$	$\neg(p \vee q)$	$(p \wedge q)$	$(\neg(p \vee q) \to (p \wedge q))$
T	T	T	F	T	T
T	F	T	F	F	T
F	T	T	F	F	T
F	F	F	T	F	F

This analysis of the subformulas could also be expressed in a more succinct form as follows.

$(\neg$	$(p$	$\vee$	$q)$	$\to$	$(p$	$\wedge$	$q))$
F	T	T	T	T	T	T	T
F	T	T	F	T	T	F	F
F	F	T	T	T	F	F	T
T	F	F	F	F	F	F	F
3	1	2	1	4	1	2	1

The circled numbers indicate the order in which the columns were filled in. They are not part of the truth table and they could be left out. The values in column 4 tell us the truth value of the entire formula corresponding to the truth values of the propositional variables p, q in the columns labelled 1. Note that column 4 is that in which the principal connective, here $\to$, of the entire formula occurs.

Strictly speaking, the truth table is the simpler one just giving the final value of ϕ for each truth assignment, although this conceals many complicated and tedious computations.

Exercise 2.20

Give the truth table of the formula $((p \to (q \wedge r)) \leftrightarrow \neg(p \vee r))$ using the propositional variables p, q, r.

Solution

One way of presenting the table is as follows.

p	q	r	$((p \to (q \wedge r)) \leftrightarrow \neg(p \vee r))$
T	T	T	F
T	T	F	T
T	F	T	T
T	F	F	T
F	T	T	F
F	T	F	T
F	F	T	F
F	F	F	T

Our rough work involved producing the following table giving truth values of subformulas.

$((p$	$\to$	$(q$	$\wedge$	$r))$	$\leftrightarrow$	$\neg$	$(p$	$\vee$	$r))$
T	T	T	T	T	F	F	T	T	T
T	F	T	F	F	T	F	T	T	F
T	F	F	F	T	T	F	T	T	T
T	F	F	F	F	T	F	T	T	F
F	T	T	T	T	F	F	F	T	T
F	T	T	F	F	T	T	F	F	F
F	T	F	F	T	F	F	F	T	T
F	T	F	F	F	T	T	F	F	F
1	3	1	2	1	4	3	1	2	1

There is no firm rule about the order in which one arranges the different truth assignments into rows, but it pays to be systematic – imagine having to write down the truth table of a formula using 10 variables and wanting to be sure that one has correctly listed all 2^{10} truth assignments somewhere in it! Our preferred method is to organize the rows so that all those assignments under which p_1 takes the value T appear in the top half of the table and all those in which it takes the value F appear in the bottom half. Then, having settled on a particular value of p_1, list all assignments in which p_2 takes the value T above those in which it takes the value F. Having settled on particular values of p_1 and p_2, repeat the process for p_3, and so on for further variables.

Exercise 2.21

Use the outline procedure above to list the rows of a table for a formula ϕ involving the variables p_1, p_2, p_3, p_4.

Solution

p_1	p_2	p_3	p_4	ϕ
T	T	T	T	?
T	T	T	F	?
T	T	F	T	?
T	T	F	F	?
T	F	T	T	?
T	F	T	F	?
T	F	F	T	?
T	F	F	F	?
F	T	T	T	?
F	T	T	F	?
F	T	F	T	?
F	T	F	F	?
F	F	T	T	?
F	F	T	F	?
F	F	F	T	?
F	F	F	F	?

Exercise 2.22

For each of the following formulas, give its truth table (where $p, q, r, p_1, p_2, p_3, p_4$ are propositional variables).

(a) $(p \wedge \neg p)$

(b) $(p \to \neg(q \leftrightarrow \neg p))$

(c) $((r \vee (q \wedge p)) \vee (\neg(q \leftrightarrow \neg r) \to p))$

(d) $(p_1 \to (p_3 \to (\neg p_4 \to p_2)))$

Solution

No solutions are given, but we hope that you found that there was only one row in the table for (d) in which the truth value of the formula was F!

We often describe a formula in terms of other formulas, not all of which are propositional variables, for instance $(\theta \to (\psi \vee \neg\theta))$. Thinking of the construction tree for this formula, we can regard its basic building blocks as being the subformulas θ and ψ. In such a case, we extend the idea of a truth table by giving the value of the whole formula for all possible combinations of truth values of these building blocks, in this case as follows:

θ	ψ	$(\theta \to (\psi \vee \neg\theta))$
T	T	T
T	F	F
F	T	T
F	F	T

Of course, if we knew more about the formulas θ and ψ, it could be that some of the rows above could never arise. For instance, if θ was the formula $(p \wedge \neg p)$, which always takes the value F, the top two rows of the table would be irrelevant. However, potentially θ and ψ could also be distinct propositional variables, so that the whole table is potentially relevant.

Exercise 2.23

For each of the following formulas, give its truth table.

(a) $(\phi \to (\psi \to \phi))$

(b) $\neg(\neg\phi \vee \phi)$

(c) $((\theta \vee (\phi \leftrightarrow \theta)) \to \neg(\psi \wedge \neg\phi))$

We trust that your solution to Exercise 2.23(a) showed that the formula $(\phi \to (\psi \to \phi))$ is true for all possible combinations of truth values of the subformulas ϕ and ψ, so that it is true under all truth assignments. Such a formula is called a *tautology*. Likewise your solution to Exercise 2.23(b) should have shown that the formula $\neg(\neg\phi \vee \phi)$ is false whatever the truth value of the subformula ϕ, so that it is false under all truth assignments. Such a formula is called a *contradiction*. Tautologies and contradictions will prove to be of special importance in much of the rest of the course.

Strictly speaking, we should say that a tautology is true under all truth assignments which are defined on a set of propositional variables including those appearing in the formula. However, here and elsewhere we shall simply talk about 'all truth assignments' as a shorthand for this fuller description.

Simple examples of tautologies are

$$(\phi \vee \neg\phi), \quad (\phi \to \phi) \quad \text{and} \quad (\neg\neg\phi \leftrightarrow \phi),$$

where ϕ is any formula. It is clear that they are tautologies by phrasing them using the intended interpretations of the connectives, 'ϕ or not ϕ' and so on, and verifying that they are tautologies by constructing their truth tables is very straightforward. Likewise $(\phi \wedge \neg\phi)$ is a pretty memorable contradiction – it corresponds well to the way we use the word 'contradiction' in everyday language.

It's well worth actively remembering these particular tautologies and contradictions.

Exercise 2.24

Show that each of the formulas $(\phi \vee \neg\phi)$, $(\phi \to \phi)$ and $(\neg\neg\phi \leftrightarrow \phi)$ is a tautology and that $(\phi \wedge \neg\phi)$ is a contradiction.

As an aside, the philosopher Ludwig Wittgenstein (1889–1951) described a tautology as a statement which conveys no information. Indeed, taking ϕ to be the statement 'it will rain', asserting ϕ gives useful information, for instance influencing one to take an umbrella when going outdoors. But asserting the tautology $(\phi \vee \neg\phi)$ is totally unhelpful in this regard!

Exercise 2.25

Which, if any, of the following formulas is a tautology or a contradiction?

(a) $(p \to (p \to p))$

(b) $((p \to p) \to p)$

(c) $((p \to \neg p) \leftrightarrow (\neg p \to p))$

The last exercise provides a reminder that there are formulas which are neither a tautology nor a contradiction. Tautologies and contradictions are of particular interest in the rest of the book, but don't forget that in general formulas don't have to fall into one of these categories.

Exercise 2.26

Let ϕ be a formula. Show that ϕ is a tautology if and only if $\neg\phi$ is a contradiction.

Solution

In conversation in a class, we would probably accept an informal (but convincing!) argument based on the observation that if the value of ϕ on each row of its truth table is T, then the value of $\neg\phi$ on the corresponding row of its truth table must be F, and vice versa. However, as other problems of this sort might not yield as easily to this sort of analysis, we shall record a more formal way of presenting this solution in case such an approach is needed elsewhere.

We shall show that if ϕ is a tautology then $\neg\phi$ is a contradiction and that if $\neg\phi$ is a contradiction then ϕ is a tautology.

First let us suppose that ϕ is a tautology. We need to show that $\neg\phi$ is false under all truth assignments. So let v be any truth assignment. As ϕ is a tautology we have $v(\phi) = T$. Thus $v(\neg\phi) = F$ (as a truth assignment respects the truth table of $\neg$, but by this stage of your study, you no longer need to say this). Hence $v(\neg\phi) = F$ for all truth assignments v, so that $\neg\phi$ is a contradiction, as required.

Conversely, suppose that $\neg\phi$ is a contradiction. Then for any truth assignment v we have $v(\neg\phi) = F$, so that, as v is a truth assignment, we can only have $v(\phi) = T$. Thus $v(\phi) = T$ for all truth assignments v, so that ϕ is a tautology.

Exercise 2.27

Let ϕ, ψ be formulas.

(a) Show that if ϕ and $(\phi \to \psi)$ are tautologies, then ψ is a tautology.

(b) Is it the case that if ϕ and ψ are tautologies, then $(\phi \to \psi)$ is a tautology?

(c) Is it the case that if $(\phi \to \psi)$ and ψ are tautologies, then ϕ is a tautology?

Now that we have the basic ideas of how to interpret the formal language, we can start to investigate relationships between one formula and others, to work towards our goal of representing mathematical arguments in a formal way. We shall begin with the idea of *logically equivalent* formulas in the next section.

Further exercises

Exercise 2.28

For a formula ϕ built up using the connectives $\neg, \wedge, \vee$, let ϕ^* be constructed by replacing each propositional variable in ϕ by its negation.

So if ϕ is the formula $((q \vee p) \wedge \neg p)$, ϕ^* is $((\neg q \vee \neg p) \wedge \neg\neg p)$.

(a) For any truth assignment v, let v^* be the truth assignment which gives each propositional variable the opposite value to that given by v, i.e.

$$v^*(p) = \begin{cases} T, & \text{if } v(p) = F, \\ F, & \text{if } v(p) = T, \end{cases}$$

for all propositional variables p. Show that $v(\phi) = v^*(\phi^*)$. [*Hint*: This is really a statement about all formulas ϕ of a certain sort, so what is the likely method of proof?]

(b) (i) Use the result of part (a) to show that ϕ is a tautology if and only if ϕ^* is a tautology.

(ii) Is it true that ϕ is a contradiction if and only if ϕ^* is a contradiction? Explain your answer.

Exercise 2.29

Suppose that we are given a set S of truth assignments with an odd number of elements. Let $\mathbf{D}$ be the set of formulas of the language which a majority of the truth assignments in S makes true. Which of the following statements is always true? Give reasons in each case.

(a) For any well-formed formula ϕ, either ϕ or $\neg\phi$ belongs to $\mathbf{D}$.

(b) If ϕ belongs to $\mathbf{D}$ and $(\phi \to \theta)$ is a tautology, then θ belongs to $\mathbf{D}$.

(c) If ϕ and $(\phi \to \theta)$ belong to $\mathbf{D}$, then θ belongs to $\mathbf{D}$.

Exercise 2.30

Prove that any formula built up from $\neg$ and $\to$ in which no propositional variable occurs more than once cannot be a tautology.

Exercise 2.31

Let ϕ, ψ be formulas.

(a) If ψ is a contradiction, under what circumstances, if any, is $(\phi \to \psi)$ a contradiction?

(b) If ϕ is a contradiction, under what circumstances, if any, is $(\phi \to \psi)$ a contradiction?

(c) If $(\phi \to \psi)$ is a contradiction, must either of ϕ, ψ be contradictions?

2.4 Logical equivalence

Formulas can be of great complexity and it is often very valuable to see whether the statements they represent could have be rephrased in a simpler but equivalent way. For instance, in normal language one would normally simplify the statement ‘it isn’t the case that it’s not raining’ into ‘it is raining’, which conveys the same information. The corresponding formal concept is that of logical equivalence, which we define below, and one of the recurring themes in the book is whether, given a formula, one can find a ‘simpler’ formula logically equivalent to it.

Definition Logically equivalent

Formulas ϕ and ψ are said to be *logically equivalent*, which we write as $\phi \equiv \psi$, if for all truth assignments v, $v(\phi) = v(\psi)$.

As ever, we should strictly speaking talk about all truth assignments v on a set of propositional variables including all of those appearing in ϕ or ψ.

For example, we have $(\theta \leftrightarrow \chi)$ is logically equivalent to $(\neg\theta \leftrightarrow \neg\chi)$ for any formulas θ, χ, as can be seen by comparing their truth tables:

θ	χ	$(\theta \leftrightarrow \chi)$
T	T	T
T	F	F
F	T	F
F	F	T

θ	χ	$(\neg\theta \leftrightarrow \neg\chi)$
T	T	T
T	F	F
F	T	F
F	F	T

By writing the tables so that the rows giving the different combinations of truth values for θ and χ are in the same order, it is easy to see that the tables match, so that for all truth assignments v, $v((\theta \leftrightarrow \chi)) = v((\neg\theta \leftrightarrow \neg\chi))$, showing that

$$(\theta \leftrightarrow \chi) \equiv (\neg\theta \leftrightarrow \neg\chi).$$

The definition of logical equivalence could be expressed in these terms, saying that the truth tables of ϕ and ψ match. However, when not all the basic building blocks of one of the formulas appears in the other, as is the case with the logically equivalent formulas θ and $((\theta \wedge \chi) \vee (\theta \wedge \neg\chi))$, the truth tables would have to be constructed in terms of truth values of the same set of subformulas, in this case θ and χ. The definition in terms of truth assignments is more elegant!

Exercise 2.32

Show that $\theta \equiv ((\theta \wedge \chi) \vee (\theta \wedge \neg\chi))$, for any formulas θ, χ.

Solution

Let v be any truth assignment.

If $v(\theta) = F$, then

$$v((\theta \wedge \chi)) = v((\theta \wedge \neg\chi)) = F,$$

regardless of whether $v(\chi)$ is true or false, so that

$$v(((\theta \wedge \chi) \vee (\theta \wedge \neg\chi))) = F = v(\theta).$$

If $v(\theta) = T$, then regardless of whether $v(\chi) = T$ or $v(\chi) = F$ (in which case $v(\neg\chi) = T$), exactly one of $v((\theta \wedge \chi))$ and $v((\theta \wedge \neg\chi))$ equals T, so that

$$v(((\theta \wedge \chi) \vee (\theta \wedge \neg\chi))) = T = v(\theta).$$

Thus for all truth assignments v, $v(\theta) = v(((\theta \wedge \chi) \vee (\theta \wedge \neg\chi)))$, so that

$$\theta \equiv ((\theta \wedge \chi) \vee (\theta \wedge \neg\chi)).$$

Alternatively, we could show that the truth tables match. The truth table of $((\theta \wedge \chi) \vee (\theta \wedge \neg\chi))$ is

θ	χ	$((\theta \wedge \chi) \vee (\theta \wedge \neg\chi))$
T	T	T
T	F	T
F	T	F
F	F	F

To help compare the truth tables, we regard θ as constructed from the subformulas θ, χ, giving the table

θ	χ	θ
T	T	T
T	F	T
F	T	F
F	F	F

which matches the first.

We list a number of very useful simple logical equivalences involving $\neg, \wedge, \vee$ in the following theorem. Some will seem very obvious. Some describe ways in which the connectives interact with each other. All are worth remembering and you will often find it helpful to exploit them.

Theorem 2.3

The following are all logical equivalences.

(a) $(\phi \wedge \psi) \equiv (\psi \wedge \phi)$ (commutativity of $\wedge$)

(b) $(\phi \vee \psi) \equiv (\psi \vee \phi)$ (commutativity of $\vee$)

(c) $(\phi \wedge \phi) \equiv \phi$ (idempotence of $\wedge$)

(d) $(\phi \vee \phi) \equiv \phi$ (idempotence of $\vee$)

(e) $(\phi \wedge (\psi \wedge \theta)) \equiv ((\phi \wedge \psi) \wedge \theta)$ (associativity of $\wedge$)

(f) $(\phi \vee (\psi \vee \theta)) \equiv ((\phi \vee \psi) \vee \theta)$ (associativity of $\vee$)

(g) $\neg\neg\phi \equiv \phi$ (law of double negation)

(h) $\neg(\phi \wedge \psi) \equiv (\neg\phi \vee \neg\psi)$ (De Morgan's Law)

(i) $\neg(\phi \vee \psi) \equiv (\neg\phi \wedge \neg\psi)$ (De Morgan's Law)

(j) $(\phi \wedge (\psi \vee \theta)) \equiv ((\phi \wedge \psi) \vee (\phi \wedge \theta))$ (distributivity of $\wedge$ over $\vee$)

(k) $(\phi \vee (\psi \wedge \theta)) \equiv ((\phi \vee \psi) \wedge (\phi \vee \theta))$ (distributivity of $\vee$ over $\wedge$))

(l) $(\phi \wedge (\psi \vee \phi)) \equiv \phi$ (absorption law for $\wedge$)

(m) $(\phi \vee (\psi \wedge \phi)) \equiv \phi$ (absorption law for $\vee$)

A more accurate description of part (a) would be that it demonstrates the commutativity of $\wedge$ *under logical equivalence*. The normal use of commutativity is with a binary operation $*$ on a set S which has the property that $a * b$ *equals*, rather than is *equivalent* to, $b * a$ for all $a, b \in S$. Hence the phrase 'under logical equivalence' is, strictly speaking, needed for this and other parts of this theorem.

Proof

We shall give an argument for part (f) and leave the rest to you as a straightforward exercise.

There are several acceptable ways to show that $(\phi \vee (\psi \vee \theta)) \equiv ((\phi \vee \psi) \vee \theta)$. One easy way is to write down the truth tables and show that these match. A perhaps more elegant method is to argue using truth assignments as follows.

Let v be any truth assignment.

If $v((\phi \vee (\psi \vee \theta))) = F$, then $v(\phi) = v((\psi \vee \theta)) = v(\psi) = v(\theta) = F$, so that $v((\phi \vee \psi)) = F$, giving

$$v(((\phi \vee \psi) \vee \theta)) = F.$$

Similarly, if $v(((\phi \vee \psi) \vee \theta)) = F$, then $v(\phi) = v(\psi) = v(\theta) = F$, so that

$$v((\phi \vee (\psi \vee \theta))) = F.$$

Thus for all truth assignments v,

$$v((\phi \vee (\psi \vee \theta))) = F \text{ if and only if } v(((\phi \vee \psi) \vee \theta)) = F,$$

so that for all truth assignments v,

$$v((\phi \vee (\psi \vee \theta))) = v(((\phi \vee \psi) \vee \theta)),$$

giving that $(\phi \vee (\psi \vee \theta)) \equiv ((\phi \vee \psi) \vee \theta)$. ■

It would also have been acceptable to show that $v((\phi \vee (\psi \vee \theta))) = T$ if and only if $v(((\phi \vee \psi) \vee \theta)) = T$, for all truth assignments v, but for these formulas involving $\vee$, this would have involved more work.

You may well have noticed similarities between many of the equivalences in this theorem involving $\neg, \wedge, \vee$ and set identities involving set complement (written as $\setminus$), intersection ($\cup$) and union ($\cap$). For instance, the De Morgan Law

$$\neg(\phi \vee \psi) \equiv (\neg\phi \wedge \neg\psi)$$

is very similar to the set identity, for sets A, B regarded as subsets of a set X,

$$X \setminus (A \cup B) = (X \setminus A) \cap (X \setminus B).$$

This is one of the laws introduced by the English mathematician Augustus De Morgan (1806–1871) who made many important contributions to the growth of modern logic.

This is no coincidence, and a moment's thought about how to express what it means to be a member of the sets on each side of the set identity will convince you of the strong connection between the set operators and the corresponding logical connectives.

The sort of connection to which we refer is that $x \in C \cap D$ if and only if $x \in C$ *and* $x \in D$.

Exercise 2.33

Prove the remaining parts of Theorem 2.3.

There are very tempting connections between some pairs of the logical equivalences in Theorem 2.3. For instance, the logical equivalence

$$(\phi \vee (\psi \wedge \theta)) \equiv ((\phi \vee \psi) \wedge (\phi \vee \theta))$$

of part (k) corresponds to interchanging the occurrences of $\wedge$ and $\vee$ in the equivalence

$$(\phi \wedge (\psi \vee \theta)) \equiv ((\phi \wedge \psi) \vee (\phi \wedge \theta))$$

of part (j). These connections are made precise in what is called the Principle of Duality, which can be found in Exercise 2.44 at the end of this section.

We hope that it is pretty obvious that $\phi \equiv \psi$ if and only if $(\phi \leftrightarrow \psi)$ is a tautology. This means that each of the logical equivalences in the theorem above corresponds to a tautology involving $\leftrightarrow$. We think that the logical equivalences are more memorable than the corresponding tautologies!

Exercise 2.34

Show that $\phi \equiv \psi$ if and only if $(\phi \leftrightarrow \psi)$ is a tautology, for all formulas ϕ, ψ.

Much of what we expect the word 'equivalent' to convey about formulas is given by the results in the following exercise.

Exercise 2.35

Show each of the following, for all formulas ϕ, ψ, θ.

(a) $\phi \equiv \phi$

(b) If $\phi \equiv \psi$ then $\psi \equiv \phi$.

(c) If $\phi \equiv \psi$ and $\psi \equiv \theta$, then $\phi \equiv \theta$.

Exercise 2.35 shows that logical equivalence is what is called an *equivalence relation* on the set of all formulas of the underlying language. Logically equivalent formulas, while usually looking very different from each other, are the 'same' in terms of their truth under different interpretations. The set of all formulas logically equivalent to a given formula ϕ is called the *equivalence class* of ϕ under this relation and a natural question is whether each such class contains a formula which is in some way nice. We shall look at an example of one way of answering such a question later in the section.

We shall look at the theory of equivalence relations in Chapter 4.

In Theorem 2.5.

Also of great use are the following logical equivalences involving $\to$ and $\leftrightarrow$, especially those connecting $\to$ with $\neg, \wedge, \vee$.

Theorem 2.4

The following are logical equivalences.

(a) $(\phi \to \psi) \equiv (\neg\phi \vee \psi) \equiv \neg(\phi \wedge \neg\psi) \equiv (\neg\psi \to \neg\phi)$

(b) $(\phi \leftrightarrow \psi) \equiv ((\phi \to \psi) \wedge (\psi \to \phi))$

The formula $(\neg\psi \to \neg\phi)$ is called the *contrapositive* of $(\phi \to \psi)$.

Note that we have exploited the results of Exercise 2.35 to write the result of Theorem 2.4(a) on one line saying that four formulas are logically equivalent, rather than writing down several equivalences showing that each pair of formulas is logically equivalent. These logical equivalences give an indication of how some connectives can be expressed in terms of others – an idea which we shall take further in the next section when we look at the idea of an *adequate* set of connectives, that is, a set of connectives from which we can generate all conceivable connectives, not just $\to$ and $\leftrightarrow$. However, $\to$ and $\leftrightarrow$ are such important connectives in terms of expressing normal mathematical statements that you should not get the impression that their use is somehow to be avoided by the use of Theorem 2.4!

Exercise 2.36

Prove Theorem 2.4.

Further nice and very useful results about logical equivalence are given in the following exercise.

Exercise 2.37

Suppose that $\phi \equiv \phi'$ and $\psi \equiv \psi'$. Show each of the following.

(a) $\neg\phi \equiv \neg\phi'$

(b) $(\phi \wedge \psi) \equiv (\phi' \wedge \psi')$

(c) $(\phi \vee \psi) \equiv (\phi' \vee \psi')$

(d) $(\phi \to \psi) \equiv (\phi' \to \psi')$

(e) $(\phi \leftrightarrow \psi) \equiv (\phi' \leftrightarrow \psi')$

Solution

We give the solution to part (b) and leave the rest to you.

Let v be any truth assignment.

Suppose that $v((\phi \wedge \psi)) = T$, so that $v(\phi) = v(\psi) = T$. We then have

$$\begin{aligned} v(\phi') &= v(\phi) \quad (\text{as } \phi \equiv \phi') \\ &= T \end{aligned}$$

and

$$\begin{aligned} v(\psi') &= v(\psi) \quad (\text{as } \psi \equiv \psi') \\ &= T, \end{aligned}$$

so that

$$v((\phi' \wedge \psi')) = T.$$

Similarly we can show that if $v((\phi' \wedge \psi')) = T$ then

$$v((\phi \wedge \psi)) = T.$$

Thus for all truth assignments v,

$$v((\phi \wedge \psi)) = T \text{ if and only if } v((\phi' \wedge \psi')) = T,$$

so that $(\phi \wedge \psi) \equiv (\phi' \wedge \psi')$.

The results of this exercise can be generalised to show that if θ is a formula containing occurrences of ϕ as a subformula and all these occurrences are replaced by a formula ϕ' where $\phi \equiv \phi'$ to turn θ into the formula θ', then $\theta \equiv \theta'$.

The proof of such a result would involve induction on the length of the formula θ and we would have to be more specific about the connectives being used. We leave an example of such a proof for you as Exercise 2.46.

The set of all propositions in a language using the set of connectives $\{\neg, \wedge, \vee\}$ with logical equivalence taking the place of $=$ is an example of what is called a *Boolean algebra*.

We shall look at Boolean algebras in Section 4.4 of Chapter 4.

The essentially algebraic results of Theorems 2.3 and 2.4 and Exercises 2.35 and 2.37 provide an alternative way of showing formulas are logically equivalent to that of working directly from the definition of logical equivalence. For instance, take the logical equivalence

$$(\phi \wedge (\phi \vee \neg\phi)) \equiv \phi.$$

Using first principles, we could argue as follows. Let v be any truth assignment.

If $v(\phi) = T$, then $v((\phi \vee \neg\phi)) = T$, so that $v((\phi \wedge (\phi \vee \neg\phi))) = T = v(\phi)$.

If $v(\phi) = F$, then $v((\phi \wedge \psi)) = F$ for any formula ψ, so that in particular $v((\phi \wedge (\phi \vee \neg\phi))) = F = v(\phi)$.

Thus $(\phi \wedge (\phi \vee \neg\phi)) \equiv \phi$.

Alternatively, using the algebraic results, as $(\phi \vee \neg\phi) \equiv (\neg\phi \vee \phi)$ (by Theorem 2.3(b)), we have

$$(\phi \wedge (\phi \vee \neg\phi)) \equiv (\phi \wedge (\neg\phi \vee \phi)) \quad \text{(by Exercise 2.37(c))}$$

and by Theorem 2.3(l) we have

$$(\phi \wedge (\neg\phi \vee \phi)) \equiv \phi,$$

so that by Exercise 2.35(c) we have

$$(\phi \wedge (\phi \vee \neg\phi)) \equiv \phi.$$

Exercise 2.38

Establish each of the following equivalences, where ϕ, ψ, θ and all the θ_i are formulas. You are welcome to do this from first principles or by exploiting the results of Theorems 2.3 and 2.4 and Exercises 2.35 and 2.37.

(a) $((\phi \wedge \psi) \vee \neg\theta) \equiv ((\phi \vee \neg\theta) \wedge (\psi \vee \neg\theta))$

(b) $(\phi \rightarrow \neg\psi) \equiv (\psi \rightarrow \neg\phi)$

(c) $(\theta \rightarrow (\phi \vee \psi)) \equiv (\phi \vee (\psi \vee \neg\theta))$

(d) $((\theta_1 \wedge \theta_2) \wedge (\theta_3 \wedge \theta_4)) \equiv (\theta_1 \wedge ((\theta_2 \wedge \theta_3) \wedge \theta_4))$

Solution

We shall give a solution to (a) and leave the rest to you.

$$\begin{aligned} ((\phi \wedge \psi) \vee \neg\theta) &\equiv (\neg\theta \vee (\phi \wedge \psi)) \quad \text{(by Theorem 2.3(b))} \\ &\equiv ((\neg\theta \vee \phi) \wedge (\neg\theta \vee \psi)) \quad \text{(by Theorem 2.3(k)} \\ &\qquad\qquad\qquad\qquad\qquad\qquad \text{and Exercise 2.35(c)).} \end{aligned}$$

But

$$(\neg\theta \vee \phi) \equiv (\phi \vee \neg\theta) \text{ and } (\neg\theta \vee \psi) \equiv (\psi \vee \neg\theta)$$

by Theorem 2.3(b), so by Exercise 2.37(b)

$$((\neg\theta \vee \phi) \wedge (\neg\theta \vee \psi)) \equiv ((\phi \vee \neg\theta) \wedge (\psi \vee \neg\theta)).$$

Then by Exercise 2.35(c),

$$((\phi \wedge \psi) \vee \neg\theta) \equiv ((\phi \vee \neg\theta) \wedge (\psi \vee \neg\theta)).$$

We have shown in Theorem 2.3(e) that $(\phi \wedge (\psi \wedge \theta)) \equiv ((\phi \wedge \psi) \wedge \theta)$ and in Exercise 2.38(d) you were asked to show that

$$((\theta_1 \wedge \theta_2) \wedge (\theta_3 \wedge \theta_4)) \equiv (\theta_1 \wedge ((\theta_2 \wedge \theta_3) \wedge \theta_4)).$$

These equivalences are most easily established by noticing that each of the relevant formulas are true precisely for those truth assignments which make each of the subformulas θ_i true. A generalization of these logical equivalences is given in the following exercise.

Exercise 2.39

Suppose that the formula ϕ is constructed by taking subformulas $\theta_1, \theta_2, \ldots, \theta_n$ in that order and joining them together only using the connective $\wedge$, with brackets inserted in such a way as to make ϕ a formula. Show that ϕ is true precisely for those truth assignments which make each of the subformulas $\theta_1, \theta_2, \ldots, \theta_n$ true. Deduce that any two such formulas (using the same $\theta_1, \theta_2, \ldots, \theta_n$ in that order) are logically equivalent. [*Hint*: Use the version of mathematical induction with the hypothesis that the result holds for all $k \leq n$ where $n \geq 1$.]

Examples for $n = 4$ include $((\theta_1 \wedge \theta_2) \wedge (\theta_3 \wedge \theta_4))$ and $(\theta_1 \wedge ((\theta_2 \wedge \theta_3) \wedge \theta_4))$. There is a general result for associative binary operations similar to the ultimate conclusion of this exercise which we invite you to look at in Exercise 2.45.

Exercise 2.40

State and prove (by any preferred method) a result similar to that in Exercise 2.39 for formulas built up from n subformulas using $\vee$ rather than $\wedge$.

Given the results of these exercises, it will be convenient for us to introduce shorthand notations for a formula which is a successive conjunction of more than two subformulas and for a formula which is a successive disjunction of more than two subformulas.

Notation

If a formula is constructed by conjunction of $\theta_1, \theta_2, \ldots, \theta_n$ so that they appear in that order joined by $\wedge$s and suitably placed brackets, we shall write it as

$$(\theta_1 \wedge \theta_2 \wedge \ldots \wedge \theta_n),$$

that is, we ignore all the brackets around the θ_is except the outermost pair. A further shorthand is to write this as

$$\bigwedge_{i=1}^{n} \theta_i,$$

further ignoring these outermost brackets and using $\bigwedge$ to represent lots of $\wedge$s. Similarly if the formula is constructed by disjunction, i.e. using $\vee$ rather than $\wedge$, we shall write it as

$$(\theta_1 \vee \theta_2 \vee \ldots \vee \theta_n)$$

and use the shorthand

$$\bigvee_{i=1}^{n} \theta_i.$$

Note that several different formulas get represented by the same shorthand. For instance, $((p \wedge q) \wedge (p \wedge r))$ and $(p \wedge ((q \wedge p) \wedge r))$ both get represented as $(p \wedge q \wedge p \wedge r)$. In the contexts where we shall use these shorthands, this won't matter – all that will matter is that the formulas so represented are all logically equivalent.

You might like to think about how many different formulas $(p \wedge q \wedge p \wedge r)$ represents in this way and, more generally, how many are represented by $\bigwedge_{i=1}^{n} \theta_i$. The answer will be one of what are called *Catalan* numbers.

Many of the useful basic logical equivalences can be extended to cover conjunctions and disjunctions of more than two subformulas, as you will see in the following exercise.

Exercise 2.41

Establish the following logical equivalences.

(a) $\neg \bigwedge_{i=1}^{n} \theta_i \equiv \bigvee_{i=1}^{n} \neg \theta_i$

(b) $\neg \bigvee_{i=1}^{n} \theta_i \equiv \bigwedge_{i=1}^{n} \neg \theta_i$

(c) $\left(\bigwedge_{i=1}^{n} \theta_i\right) \vee \left(\bigwedge_{j=1}^{m} \psi_j\right) \equiv \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{m} (\theta_i \vee \psi_j)$

(d) $\left(\bigvee_{i=1}^{n} \theta_i\right) \wedge \left(\bigvee_{j=1}^{m} \psi_j\right) \equiv \bigvee_{i=1}^{n} \bigvee_{j=1}^{m} (\theta_i \wedge \psi_j)$

We are being somewhat casual about brackets in these formulas, in the cause of comprehensibility we hope! The same philosophy will pervade our solutions.

Solution

(a) We shall use mathematical induction on $n \geq 1$. For $n = 1$ the result is trivially true. For the inductive step, we suppose that the result holds for $n \geq 1$ and must show that it holds for $n + 1$. We have

$$\begin{aligned}
\neg \bigwedge_{i=1}^{n+1} \theta_i &\equiv \neg \left(\left(\bigwedge_{i=1}^{n} \theta_i \right) \wedge \theta_{n+1} \right) \\
&\equiv \left(\neg \left(\bigwedge_{i=1}^{n} \theta_i \right) \vee \neg \theta_{n+1} \right) \qquad (\text{as } \neg(\phi \wedge \psi) \equiv (\neg\phi \vee \neg\psi)).
\end{aligned}$$

By the induction hypothesis, $\neg \bigwedge_{i=1}^{n} \theta_i \equiv \bigvee_{i=1}^{n} \neg \theta_i$, so by Exercise 2.37(c) we have

$$\begin{aligned}
\left(\neg \left(\bigwedge_{i=1}^{n} \theta_i \right) \vee \neg \theta_{n+1} \right) &\equiv \left(\left(\bigvee_{i=1}^{n} \neg \theta_i \right) \vee \neg \theta_{n+1} \right) \\
&\equiv \bigvee_{i=1}^{n+1} \neg \theta_i,
\end{aligned}$$

This is a typical use of Exercise 2.37 to replace one subformula by an equivalent subformula.

so that

$$\neg \bigwedge_{i=1}^{n+1} \theta_i \equiv \bigvee_{i=1}^{n+1} \neg \theta_i,$$

as required. The result follows by mathematical induction.

(b) Not given.

(c) This one could be done by mathematical induction on both $n \geq 1$ and $m \geq 1$: first show that the result holds for $n = 1$ and all $m \geq 1$ using induction on m, and then assume that the result holds for some $n \geq 1$ and all m and show that it holds for $n + 1$ and all m. However, there is a shortcut which we will take.

First we fix $m = 1$ and show the result then holds for all $n \geq 1$. We shall write ϕ rather than ψ_1 for a reason which will be revealed later! The result holds trivially for $n = 1$. If the result holds for some $n \geq 1$, we then have

As forecast, we shall be a bit casual about brackets!

$$\begin{aligned}
\left(\bigwedge_{i=1}^{n+1} \theta_i\right) \vee \phi &\equiv \left(\left(\bigwedge_{i=1}^{n} \theta_i\right) \wedge \theta_{n+1}\right) \vee \phi \\
&\equiv \left(\left(\bigwedge_{i=1}^{n} \theta_i\right) \vee \phi\right) \wedge (\theta_{n+1} \vee \phi) \\
&\qquad \text{(as } ((\psi \wedge \theta) \vee \phi) \equiv ((\psi \vee \phi) \wedge (\theta \vee \phi))\text{)} \\
&\equiv \left(\bigwedge_{i=1}^{n} (\theta_i \vee \phi)\right) \wedge (\theta_{n+1} \vee \phi) \\
&\qquad \text{(using the induction hypothesis and Exercise 2.37(b))} \\
&\equiv \bigwedge_{i=1}^{n+1} (\theta_i \vee \phi),
\end{aligned}$$

as required. So by mathematical induction, we have

$$\left(\bigwedge_{i=1}^{n} \theta_i\right) \vee \phi \equiv \bigwedge_{i=1}^{n} (\theta_i \vee \phi),$$

for all $n \geq 1$.

Now to prove the required result, we replace ϕ in the result above by $\bigwedge_{j=1}^{m} \psi_j$ to obtain

$$\left(\bigwedge_{i=1}^{n} \theta_i\right) \vee \left(\bigwedge_{j=1}^{m} \psi_j\right) \equiv \bigwedge_{i=1}^{n} \left(\theta_i \vee \bigwedge_{j=1}^{m} \psi_j\right). \qquad (*)$$

As $(\theta \vee \phi) \equiv (\phi \vee \theta)$, the subsidiary result for $m = 1$ gives

$$\phi \vee \bigwedge_{i=1}^{n} \theta_i \equiv \bigwedge_{i=1}^{n} (\phi \vee \theta_i),$$

which, by replacing n by m, the θ_is for $i = 1, \ldots, n$ by ψ_j for $j = 1, \ldots, m$, and ϕ by θ_i gives

$$\theta_i \vee \bigwedge_{j=1}^{m} \psi_j \equiv \bigwedge_{j=1}^{m} (\theta_i \vee \psi_j).$$

Substituting this in $(*)$ gives the required result, namely

$$\left(\bigwedge_{i=1}^{n} \theta_i\right) \vee \left(\bigwedge_{j=1}^{m} \psi_j\right) \equiv \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{m} (\theta_i \vee \psi_j),$$

for all $n, m \geq 1$.

(d) Not given.

We are about to state a result involving a quite complicated description of a particular sort of formula, as follows:

> ψ is a disjunction of formulas which are conjunctions of propositional variables and/or negated propositional variables.

What does this mean? An example of what we mean is

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

Recall our shorthand for ignoring brackets in long conjunctions and long disjunctions.

Here each of $(p \wedge \neg q \wedge p)$, q, $(\neg r \wedge s)$ and $\neg s$ are conjunctions of propositional variables and/or negated propositional variables – OK, you might not like it, but each of the q and $\neg s$ is a conjunction of just one thing! Also these conjunctions are joined together by $\vee$s to form the disjunction. Such a formula is said to be in *disjunctive form*. A more general version of such a form is

$$\bigvee_{i=1}^{n} \left(\bigwedge_{j=1}^{k_i} q_{i,j}\right),$$

When $n = 1$, no $\vee$ actually appears, and an example of the sort of formula you get is $(p \wedge q \wedge \neg r)$.

where each $q_{i,j}$ is a propositional variable or its negation.

The result will also involve a corresponding form where the roles of $\wedge$ and $\vee$ are interchanged. This is called a *conjunctive form*, which is a conjunction of formulas which are disjunctions of propositional variables and/or negated propositional variables, i.e. of the form

$$\bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{k_i} q_{i,j}\right),$$

where each $q_{i,j}$ is a propositional variable or its negation. An example of this is

$$(p \wedge (\neg q \vee r) \wedge (r \vee \neg p \vee q \vee p)).$$

As with disjunctive form, there are some fairly trivial formulas which are in conjunctive form, like each of q, $\neg p$, $(p \wedge q)$ (for which the k_is in the general form above all equal 1) and $(q \vee r \vee \neg p)$ (for which the n in the general form equals 1). Actually all these trivial formulas are simultaneously in both conjunctive and disjunctive form. The result we are leading towards says that for any given formula ϕ using connectives in the set $\{\wedge, \vee, \neg, \rightarrow, \leftrightarrow\}$, there are logically equivalent formulas, one in disjunctive form and one in conjunctive form, but usually these latter formulas are not the same. For instance, $((p \to q) \to r)$ is logically equivalent to

$$((p \wedge \neg q) \vee r)$$

which is in disjunctive form and

$$((p \vee r) \wedge (\neg q \vee r))$$

which is in conjunctive form. Now for the theorem! This result tells us that any formula, however complicated a jumble of variables and connectives it appears to be, is logically equivalent to a formula with a nice, orderly shape.

Theorem 2.5

Let ϕ be a formula using connectives in the set $\{\wedge, \vee, \neg, \to, \leftrightarrow\}$. Then ϕ is logically equivalent to a formula $\phi^{\vee}$ in disjunctive form and a formula $\phi^{\wedge}$ in conjunctive form.

Proof

First we remove all occurrences of $\leftrightarrow$ in ϕ by replacing all subformulas of the form $(\theta \leftrightarrow \psi)$ by $((\theta \to \psi) \wedge (\psi \to \theta))$. In the resulting formula, we then remove all occurrences of $\to$ by replacing all subformulas of the form $(\theta \to \psi)$ by $(\neg\theta \vee \psi)$. In this way we have produced a formula logically equivalent to ϕ which only uses the connectives $\wedge, \vee, \neg$. If we can prove the result for all formulas of this type, then the result holds for the original ϕ.

These are logical equivalences in Theorem 2.4.

So let's now suppose that the connectives in ϕ are in the set $\{\wedge, \vee, \neg\}$. We have to prove the result for all formulas ϕ of this type and to cope with formulas of arbitrary complexity we shall use induction on the length of ϕ, where, as before, our preferred measure of length is the number of connectives in ϕ.

For a formula ϕ of this type of length 0, ϕ can only be a propositional variable p, which is already in both disjunctive and conjunctive form.

So $p^{\vee} = p^{\wedge} = p$.

Now suppose that the result holds for all formulas using connectives in the set $\{\wedge, \vee, \neg\}$ of length $\leq n$, where $n \geq 0$, and that ϕ is a formula of this type with $n+1$ connectives. As ϕ has at least one connective, it must have one of the following three forms:

$$(\theta \wedge \psi), \quad (\theta \vee \psi), \quad \neg\theta,$$

where crucially the subformulas θ and ψ have length $\leq n$, so that the inductive hypothesis applies to them. We must consider each of these three forms separately and will leave some of the details to you. We'll make heavy use of some of the results of Exercise 2.41 and Exercise 2.37 to replace subformulas by logically equivalent formulas.

Case: ϕ is of the form $(\theta \wedge \psi)$

By the inductive hypothesis θ and ψ are logically equivalent to $\theta^\wedge$ and $\psi^\wedge$ respectively, both in conjunctive form. Then

$$\phi \equiv (\theta \wedge \psi) \equiv (\theta^\wedge \wedge \psi^\wedge).$$

But $(\theta^\wedge \wedge \psi^\wedge)$ is in conjunctive form as both $\theta^\wedge$ and $\psi^\wedge$ are in this form. So we can take $\phi^\wedge$ to be $(\theta^\wedge \wedge \psi^\wedge)$.

That was rather easy, but what about $\phi^\vee$? For this, first note that by the inductive hypothesis θ and ψ are logically equivalent to $\theta^\vee$ and $\psi^\vee$ respectively, both in disjunctive form. So

$$\phi \equiv (\theta \wedge \psi) \equiv (\theta^\vee \wedge \psi^\vee).$$

The $\wedge$ as principal connective of $(\theta^\vee \wedge \psi^\vee)$ means that it is not usually in disjunctive form, so some more work is needed. The formulas $\theta^\vee$ and $\psi^\vee$ are of the forms $\displaystyle\bigvee_{i=1}^{n} \theta_i$ and $\displaystyle\bigvee_{j=1}^{m} \psi_j$ respectively where each θ_i and ψ_j is a conjunction of propositional variables and/or negated propositional variables. So

$$\phi \equiv \left(\bigvee_{i=1}^{n} \theta_i\right) \wedge \left(\bigvee_{j=1}^{m} \psi_j\right)$$

and by Exercise 2.41 part (d), the formula on the right is logically equivalent to

$$\bigvee_{i=1}^{n} \bigvee_{j=1}^{m} (\theta_i \wedge \psi_j).$$

As each θ_i and ψ_j is a conjunction of propositional variables and/or negated propositional variables, so is each $(\theta_i \wedge \psi_j)$. That means that $\displaystyle\bigvee_{i=1}^{n} \bigvee_{j=1}^{m} (\theta_i \wedge \psi_j)$ is in disjunctive form, so that we can take this formula as $\phi^\vee$.

Case: ϕ is of the form $(\theta \vee \psi)$

This is left as an exercise for you.

Case: ϕ is of the form $\neg\theta$

By the inductive hypothesis, θ is logically equivalent to $\theta^\wedge$ in conjunctive form, which we can write as $\displaystyle\bigwedge_{i=1}^{n} \theta_i$, where each θ_i is a disjunction of propositional variables and/or their negations. Using Exercise 2.41 part (a), we have

$$\begin{aligned}\phi \equiv \neg\,\theta \equiv \neg \bigwedge_{i=1}^{n} \theta_i \\ \equiv \bigvee_{i=1}^{n} \neg\,\theta_i.\end{aligned}$$

Each θ_i is a disjunction of propositional variables and/or their negations, so of the form $\bigvee_{j=1}^{n_i} q_{i,j}$, where each $q_{i,j}$ is a propositional variable or its negation. By Exercise 2.41 part (b), $\neg\,\theta_i$ is logically equivalent to $\bigwedge_{i=1}^{n} \neg q_{i,j}$. Each $q_{i,j}$ is of the form p or $\neg p$, where p is a propositional variable. So $\neg q_{i,j}$ is of the form $\neg p$ or $\neg\neg p$. In the latter case $\neg q_{i,j}$ is logically equivalent to p. For each i, j put

$$r_{i,j} = \begin{cases} \neg p, & \text{if } q_{i,j} \text{ is a propositional variable } p, \\ p, & \text{if } q_{i,j} \text{ is } \neg p \text{ where } p \text{ is a propositional variable,} \end{cases}$$

so that $r_{i,j}$ is always a propositional variable or its negation and is logically equivalent to $\neg q_{i,j}$. At last we have

$$\begin{aligned} \phi &\equiv \bigvee_{i=1}^{n} \neg\,\theta_i \\ &\equiv \bigvee_{i=1}^{n} \neg \bigvee_{j=1}^{n_i} q_{i,j} \\ &\equiv \bigvee_{i=1}^{n} \bigwedge_{j=1}^{n_i} \neg q_{i,j} \\ &\equiv \bigvee_{i=1}^{n} \bigwedge_{j=1}^{n_i} r_{i,j}, \end{aligned}$$

which is in disjunctive form and can thus be taken as $\phi^{\vee}$.

We shall leave it as an exercise for you to find a suitable $\phi^{\wedge}$ logically equivalent to ϕ in this case. ■

Exercise 2.42

(a) Explain how to construct a suitable $\phi^{\wedge}$ and $\phi^{\vee}$ in the case when ϕ is of the form $(\theta \vee \psi)$ in the proof of Theorem 2.5.

(b) Explain how to construct a suitable $\phi^{\wedge}$ in the case when ϕ is of the form $\neg\,\theta$ in the proof of Theorem 2.5.

Exercise 2.43

For each of the following formulas, follow the method used in the proof of Theorem 2.5 to find a disjunctive form and a conjunctive form equivalent to it.

(a) $((p_1 \vee (p_2 \leftrightarrow \neg p_1)) \to (\neg p_1 \wedge p_3))$

(b) $(p \to (q \to (r \vee \neg p)))$

(c) $(p \leftrightarrow \neg p)$

A given formula will in general be logically equivalent to several different formulas in disjunctive form. For instance,

$$(p \vee (\neg p \wedge q)) \equiv (p \vee q),$$

where both are in disjunctive form. We shall see in the next section that amongst these different disjunctive forms there are some that fit a standard format, and so can be said to be in a *normal* form. The same thing goes for conjunctive forms.

In this section we have looked at propositional formulas built up from the connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$ and $\leftrightarrow$. But these are not the only connectives we might have introduced based on normal language. For instance, we might have introduced connectives for the likes of 'unless' and 'neither . . . nor'. In the next section we shall look at further possible connectives in a very general way. Then we shall discover a remarkable fact about how even very complicated connectives can always be built up from some of the very simple ones we have met in this section.

Further exercises

Exercise 2.44

Let ϕ be a formula built up using the connectives $\neg, \wedge, \vee$. The *dual* ϕ' of ϕ is the formula obtained from ϕ by replacing all occurrences of $\wedge$ by $\vee$, of $\vee$ by $\wedge$, and all propositional variables by their negations.

For instance, if ϕ is the formula
$(\neg p \wedge ((p \wedge q) \vee r))$,
then ϕ' is
$(\neg\neg p \vee ((\neg p \vee \neg q) \wedge \neg r))$.

(a) Show that ϕ' is logically equivalent to $\neg\phi$. (This is called the *Principle of Duality*.) [*Hint*: Use induction on the length of ϕ.]

(b) Hence, using Theorem 2.4, show that if ϕ, ψ are formulas built up using the connectives $\neg, \wedge, \vee$, then

$$(\phi \rightarrow \psi) \equiv (\psi' \rightarrow \phi')$$

and

$$(\phi \leftrightarrow \psi) \equiv (\phi' \leftrightarrow \psi').$$

(c) Use the following method to show that Theorem 2.3(k) follows from Theorem 2.3(j). Theorem 2.3(j) states that

$$(\phi \wedge (\psi \vee \theta)) \equiv ((\phi \wedge \psi) \vee (\phi \wedge \theta)),$$

so that by the result of Exercise 2.34,

$$((\phi \wedge (\psi \vee \theta)) \leftrightarrow ((\phi \wedge \psi) \vee (\phi \wedge \theta)))$$

is a tautology. Use the result of part (b) above and the result of Exercise 2.28(b)(i) in Section 2.3 to show that

$$((\phi \vee (\psi \wedge \theta)) \leftrightarrow ((\phi \vee \psi) \wedge (\phi \vee \theta)))$$

is a tautology. Hence, by Exercise 2.34,

$$(\phi \vee (\psi \wedge \theta)) \equiv ((\phi \vee \psi) \wedge (\phi \vee \theta)),$$

which is Theorem 2.3(k).

(d) Identify other logical equivalences in Theorem 2.3 which are related in the same way using the method of part (c).

Exercise 2.45

Let X be a non-empty set and suppose that $*\colon X^2 \longrightarrow X$ is a function with the associative property, that is,

$$(x * (y * z)) = ((x * y) * z), \quad \text{for all } x, y, z \in X,$$

where we write the image under the function of the pair (a, b) in X^2 as $(a * b)$. Let $x_1, x_2, \ldots, x_n$ be elements of X and suppose that brackets and $*$s are inserted into the string of symbols $x_1 x_2 \ldots x_n$ to give an expression which can be computed using the function $*$ to give an element of X. Show that the computations of all such expressions (for the same $x_1 x_2 \ldots x_n$ in that order) will result in the same element of X.

For instance, for $n = 4$, each of the expressions

$$(x_1 * (x_2 * (x_3 * x_4)))$$
$$(x_1 * ((x_2 * x_3) * x_4))$$
$$((x_1 * x_2) * (x_3 * x_4))$$
$$(((x_1 * x_2) * x_3) * x_4)$$
$$((x_1 * (x_2 * x_3)) * x_4)$$

give the same element of X.

[*Hints*: Use induction on n with hypothesis that each such expression equals both of

$$(x_1 * (x_2 * (\ldots * (x_{n-1} * x_n) \ldots)))$$

and

$$(((\ldots (x_1 * x_2) * \ldots) * x_{n-1}) * x_n).]$$

Exercise 2.46

Let θ be a formula built up using the connectives $\neg, \wedge$ and let ϕ be one of its subformulas. We shall write $\theta[\phi'/\phi]$ for the formula obtained by replacing all occurrences of the subformula ϕ in θ by the formula ϕ'. Show that if $\phi \equiv \phi'$ then $\theta \equiv \theta[\phi'/\phi]$. [*Hints*: Fix the formulas ϕ and ϕ' and do an induction on the length of θ. But first be much more specific about the meaning of $\theta[\phi'/\phi]$ by defining it as follows:

$$\theta[\phi'/\phi] = \begin{cases} \theta, & \text{if } \phi \text{ does not occur as a subformula of } \theta, \\ \phi', & \text{if } \phi \text{ is the subformula } \theta \text{ of } \theta, \\ \neg\psi[\phi'/\phi], & \text{if } \phi \text{ occurs as a subformula of } \theta \text{ (with } \phi \neq \theta) \text{ and } \theta \text{ is of the form } \neg\psi, \\ (\psi[\phi'/\phi] \wedge \chi[\phi'/\phi]), & \text{if } \phi \text{ occurs as a subformula of } \theta \text{ (with } \phi \neq \theta) \text{ and } \theta \text{ is of the form } (\psi \wedge \chi). \end{cases}$$

Making the description of $\theta[\phi'/\phi]$ makes the problem much easier to solve!]

For instance if θ is the formula

$$((p \wedge p) \wedge (q \wedge \neg(p \wedge p))),$$

ϕ is the formula $(p \wedge p)$ and ϕ' is the formula $\neg\neg p$, then $\theta[\phi'/\phi]$ is the formula

$$(\neg\neg p \wedge (q \wedge \neg\neg\neg p)).$$

2.5 The expressive power of connectives

So far we have looked at formulas built up using the connectives $\wedge$, $\vee$, $\neg$, $\rightarrow$ and $\leftrightarrow$, which have intended interpretations corresponding to uses in everyday language, conveyed by their standard truth tables. Surely there are other connectives which might arise from everyday language, in which case we might ask the following questions.

(i) How many different connectives are there?

(ii) Can some connectives be expressed in terms of others?

(iii) Is there any 'best' set of connectives?

The correspondence between the symbols and everyday language has already required some firm, and even tough, decisions, like 'or' being taken as inclusive rather than exclusive and 'implies' carrying with it the convention that $\phi \rightarrow \psi$ is true when ϕ is false.

It's not too hard to think of some further everyday connectives, although it might be harder to settle on reasonable truth tables for them. For instance, there are the exclusive 'or' (meaning '... or ... but not both'), 'is implied by' and 'unless'.

Exercise 2.47

Suggest truth tables for each of 'ϕ or ψ' with the exclusive 'or', 'ϕ is implied by ψ' and 'ϕ unless ψ'.

These extra connectives are all binary – they connect two propositions. How about connectives requiring more than two propositions? They are perhaps less everyday than the likes of 'and' and 'implies', but they do exist. Take, for instance, 'at least two of the following statements are true: ...'. This doesn't fit in well with the sort of construction rules we've had for well-formed formulas if we leave open how many statements do follow. But we can nail things down by specifying a natural number n (with $n > 2$ to make things interesting!) and modifying this as 'at least two of the following n statements are true: ...'.

Exercise 2.48

(a) (i) How many rows would you need for the truth table of the proposition 'at least two of the following 3 statements are true: ϕ, ψ, θ'?

(ii) Write down the truth table for this proposition.

(b) (i) How many rows would you need for the truth table of the proposition 'at least two of the following n statements are true: $\phi_1, \phi_2, \ldots, \phi_n$', where $n \geq 2$?

(ii) On how many of these rows would you expect the proposition to be true?

Solution

(a) (i) Each of the three propositions ϕ, ψ and θ could take the values T or F independent of the values taken by the other two. Thus there are $2 \times 2 \times 2 = 2^3 = 8$ different combinations of truth values to be taken into account in the table, so that the latter needs 8 rows.

(ii)

We've set out the rows in what we hope is a transparently systematic basis!

ϕ	ψ	θ	at least two of ϕ, ψ, θ are true
T	T	T	T
T	T	F	T
T	F	T	T
T	F	F	F
F	T	T	T
F	T	F	F
F	F	T	F
F	F	F	F

(b) (i) As each of the propositions $\phi_1, \phi_2, \ldots, \phi_n$ could potentially be true or false independent of the values taken by the others, the table would normally require

$$\underbrace{2 \times 2 \times \ldots \times 2}_{n} = 2^n$$

rows.

(ii) Not given.

Rather than producing even more baroque examples of vaguely natural connectives, we'll simply state that we've opened the floodgates of connectives, laying open the possibility that there are lots of connectives of n arguments, for arbitrarily large natural numbers n. We shall leave behind the everyday language descriptions of connectives and concentrate on what characterizes them within 2-valued logic, namely their truth tables. Each truth table in essence describes the values of a function, as in the following definition.

A given truth table might have several descriptions: e.g. 'if ... then' and 'implies' have the same truth table.

Definition Truth function

A *truth function of n arguments* is any function $f\colon \{T, F\}^n \longrightarrow \{T, F\}$.

$\{T, F\}^n$ means $\underbrace{\{T, F\} \times \{T, F\} \times \ldots \times \{T, F\}}_{n}$.

A truth function can often be described nicely by its rule, for instance the function of 3 arguments we saw in Exercise 2.48(a) which takes the value T when at least two of its arguments take the value T. But as often as not, the only way of describing a truth function is by giving its full table of values, in the form:

x_1	x_2	$\ldots$	x_n	$f(x_1, x_2, \ldots, x_n)$
T	T	$\ldots$	T	?
T	T	$\ldots$	F	?
$\vdots$	$\vdots$		$\vdots$	$\vdots$
F	F	$\ldots$	F	?

Exercise 2.49

(a) If f is a truth function of n arguments, how many rows are there in its table of values?

(b) Explain why the number of truth functions of n arguments is 2^{2^n}.

Clearly there are infinitely many truth functions, allowing for all possible values of n. If we want to represent all of these within our formal language, do we need a special connective symbol for each one? Or can we represent some of them in terms of others? We have already seen how some of the more basic connectives are interrelated, in terms of logical equivalence. For instance, we have $(\phi \to \psi) \equiv (\neg\phi \vee \psi)$, so that any use of the connective $\to$ in a formula could be replaced by a construction involving $\neg$ and $\vee$. Likewise we can talk about a formula in the formal language representing a particular truth function, as in the following definition.

More precisely there are countably infinitely many truth functions.

Definition Representing a formula

Let f be a truth function of n arguments $x_1, x_2, \ldots, x_n$ and let ϕ be a formula of the formal language involving propositional variables out of the set $\{p_1, p_2, \ldots, p_n\}$. We shall say that ϕ *represents* f if the table of values of f

x_1	x_2	$\ldots$	x_n	$f(x_1, x_2, \ldots, x_n)$
T	T	$\ldots$	T	?
T	T	$\ldots$	F	?
$\vdots$	$\vdots$		$\vdots$	$\vdots$
F	F	$\ldots$	F	?

matches the truth table of ϕ in an obvious way:

p_1	p_2	$\ldots$	p_n	ϕ
T	T	$\ldots$	T	$f(T, T, \ldots, T)$
T	T	$\ldots$	F	$f(T, T, \ldots, F)$
$\vdots$	$\vdots$		$\vdots$	$\vdots$
F	F	$\ldots$	F	$f(F, F, \ldots, F)$

or, more formally, for any truth assignment v, if $v(p_i) = x_i$ for $i = 1, 2, \ldots, n$, then $v(\phi) = f(x_1, x_2, \ldots, x_n)$.

For instance, the truth function $f_\to$ of 2 arguments given by $f_\to(x_1, x_2) = F$ if and only if $x_1 = T$ and $x_2 = F$ is represented by the formula $(p_1 \to p_2)$. Also the truth function of 3 arguments corresponding to the connective introduced in Exercise 2.48(a) is represented by the formula

$f_\to$ is just the truth function corresponding to 'implies'.

$$((p_1 \wedge p_2) \vee (p_2 \wedge p_3) \vee (p_3 \wedge p_1)).$$

If we were interested in whether both these truth functions could be represented in a language using a limited set of connectives, say consisting of just $\neg$ and $\wedge$, then it so happens it can be done, by exploiting logical equivalences, first $(\phi \to \psi) \equiv (\neg\phi \vee \psi)$ (for appropriate ϕ, ψ) to eliminate the uses of $\to$ and then $(\phi \vee \psi) \equiv \neg(\neg\phi \wedge \neg\psi)$ to eliminate the uses of $\vee$. But perhaps these are just nicely behaved truth functions. With how uncomplicated a set of connectives can we represent all truth functions?

Of course we could have used the logical equivalence $(\phi \to \psi) \equiv \neg(\phi \wedge \neg\psi)$ to eliminate the $\to$ directly, without going via the use of $\vee$.

Pretty remarkably, there are very uncomplicated and small sets of connectives with which one can represent all truth functions. This property merits a definition.

Definition Adequate set of connectives

A set S of connectives is *adequate* if all truth functions can be represented by formulas using connectives from this set.

In many ways the nicest (to the author!) such adequate set is $\{\neg, \wedge, \vee\}$, as is shown in the following theorem.

Theorem 2.6

The set of connectives $\{\neg, \wedge, \vee\}$ is adequate.

Proof

Let $f\colon \{T, F\}^n \longrightarrow \{T, F\}$ be a truth function of n arguments. We shall use the table of values of f to construct a formula ϕ using $\neg, \wedge, \vee$ representing f.

This proof has the bonus that it gives an explicit construction of a formula ϕ using $\neg, \wedge, \vee$ representing a given truth function f and that this formula has a helpful standard shape.

$$\begin{array}{cccc|c} x_1 & x_2 & \ldots & x_n & f(x_1, x_2, \ldots, x_n) \\ \hline T & T & \ldots & T & ? \\ T & T & \ldots & F & ? \\ \vdots & \vdots & & \vdots & \vdots \\ F & F & \ldots & F & ? \\ \hline \end{array}$$

First we deal with the case when $f(x_1, x_2, \ldots, x_n) = F$ on all 2^n rows of the table. Simply take ϕ to be the formula $(p_1 \wedge \neg p_1)$ – the truth table of this formula, regarded as involving variables out of the set $\{p_1, p_2, \ldots, p_n\}$, will give the value F on all lines.

If you really want a formula which involves all of the variables $p_1, p_2, \ldots, p_n$, then take the formula $(p_1 \wedge \neg p_1) \wedge \ldots \wedge (p_n \wedge \neg p_n)$.

Much more interesting, and needing some real effort, is the case when the table of values of f has some rows for which $f(x_1, x_2, \ldots, x_n) = T$. For each such row, coded by a particular n-tuple $\langle x_1, x_2, \ldots, x_n \rangle$ in $\{T, F\}^n$, construct the formula $\theta_{\langle x_1, x_2, \ldots, x_n \rangle}$ as follows:

$$(q_1 \wedge q_2 \wedge \ldots \wedge q_n) \quad \text{where } q_i = \begin{cases} p_i, & \text{if } x_i = T, \\ \neg p_i & \text{if } x_i = F, \end{cases} \text{ for } i = 1, 2, \ldots, n.$$

The key property of this formula is that the only truth assignment v that makes it true is the one corresponding to the row coded by $\langle x_1, x_2, \ldots, x_n \rangle$ in $\{T, F\}^n$, i.e. defined by $v(p_i) = x_i$, for each $i = 1, 2, \ldots, n$.

Check this!

Now let ϕ be the disjunction of all the $\theta_{\langle x_1, x_2, \ldots, x_n \rangle}$ which arise for the truth function f. We claim that ϕ represents f.

Take any truth assignment v, where $v(p_i) = x_i$ for each $i = 1, 2, \ldots, n$. We must show that $v(\phi) = f(x_1, x_2, \ldots, x_n)$ and to do this it's enough to show that $v(\phi) = T$ if and only if $f(x_1, x_2, \ldots, x_n) = T$. If $v(\phi) = T$, then as ϕ is a disjunction of formulas $\theta_{\langle y_1, y_2, \ldots, y_n \rangle}$, one of the latter is true under v. But then one of these θs must be $\theta_{\langle x_1, x_2, \ldots, x_n \rangle}$. By the construction of ϕ, this can only be included when $f(x_1, x_2, \ldots, x_n) = T$, which is what we require. Conversely, if $f(x_1, x_2, \ldots, x_n) = T$, then $\theta_{\langle x_1, x_2, \ldots, x_n \rangle}$ is one of the disjuncts of ϕ; and as the truth assignment v makes $\theta_{\langle x_1, x_2, \ldots, x_n \rangle}$ true, it must then make ϕ true. ■

Let's look at the ϕ which the proof constructs for the truth function f of 3 variables given by the following table:

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
T	T	T	T
T	T	F	T
T	F	T	F
T	F	F	F
F	T	T	F
F	T	F	T
F	F	T	F
F	F	F	F

There are three rows on which $f(x_1, x_2, x_3) = T$, corresponding to the triples $\langle T, T, T \rangle$, $\langle T, T, F \rangle$ and $\langle F, T, F \rangle$.The corresponding θs are

$$\begin{aligned} \theta_{\langle T,T,T \rangle} &: \quad (p_1 \wedge p_2 \wedge p_3), \\ \theta_{\langle T,T,F \rangle} &: \quad (p_1 \wedge p_2 \wedge \neg p_3), \\ \theta_{\langle F,T,F \rangle} &: \quad (\neg p_1 \wedge p_2 \wedge \neg p_3). \end{aligned}$$

Thus f is represented by the formula ϕ obtained by taking the disjunction of these three formulas:

$$((p_1 \wedge p_2 \wedge p_3) \vee (p_1 \wedge p_2 \wedge \neg p_3) \vee (\neg p_1 \wedge p_2 \wedge \neg p_3)).$$

Note that this formula is in what we previously called a disjunctive form, namely a disjunction of formulas each of which is a conjunction of variables, possibly negated. Furthermore these latter conjunctions and the similar conjunctions obtained for almost all other truth functions f all involve the same variables, $p_1, p_2, \ldots, p_n$, which merits saying that the formula is in a 'normal form', to which we give a special name, as in the following definition.

We met this in Theorem 2.5 in Section 2.4.

The exception being the case when $f(x_1, x_2, \ldots, x_n) = F$ for all $x_1, x_2, \ldots, x_n$; but even then we could represent f by the formula $(p_1 \wedge \neg p_1 \wedge \ldots \wedge p_n \wedge \neg p_n)$, which uses all of $p_1, p_2, \ldots, p_n$.

Definition Disjunctive normal form

A formula is said to be in *disjunctive normal form*, often abbreviated as *dnf*, if for some $n \geq 1$ it is a disjunction of formulas of the form

$$(q_1 \wedge q_2 \wedge \ldots \wedge q_n),$$

where for each $i = 1, 2, \ldots, n$, q_i is one of p_i and $\neg p_i$. As an exceptional case, we shall also say that the formula $(p_1 \wedge \neg p_1 \wedge \ldots \wedge p_n \wedge \neg p_n)$ is in disjunctive normal form.

To avoid redundancy in a formula in dnf, we shall also ban any of these conjunctions from appearing more than once in the formula.

Thus a benefit of this proof of Theorem 2.6 is that we know not only that each truth function can be represented by a formula using $\wedge$, $\vee$ and $\neg$, but how to construct such a formula with a nice shape, namely one in disjunctive normal form.

A further benefit of Theorem 2.6 is that it gives a more general result than Theorem 2.5 in Section 2.4. The latter theorem told us that any formula built up using connectives in the set $\{\wedge, \vee, \neg, \rightarrow, \leftrightarrow\}$ was logically equivalent to a formula in disjunctive form. But now we can say that any formula ψ built up using *any* connectives, not just these familiar ones, is logically equivalent to a formula in dnf. All we do is take the truth function f represented by ψ and then apply Theorem 2.6 to get a formula ϕ which represents f, and hence has to be logically equivalent to ψ. We hope that after having seen the proof of Theorem 2.6, the construction of the formulas in the dnf for ψ now seems very natural – we just represent each truth assignment making ψ true by a corresponding conjunction of propositional variables and/or their negations, and then join together the relevant conjunctions by $\vee$s.

Exercise 2.50

Write down formulas in dnf representing each of the following:

(a) the truth function of 2 arguments represented by the formula $(p_1 \leftrightarrow p_2)$;

(b) the truth function of 3 arguments represented by the formula $\neg(p_1 \vee p_3)$, regarded as built up from variables out of the set $\{p_1, p_2, p_3\}$;

(c) the truth function f of 3 arguments where $f(x_1, x_2, x_3)$ is true if at least two of x_1, x_2, x_3 is false.

Solution

We shall give a solution to (b) and leave the others to you.

The formula $\neg(p_1 \vee p_3)$, regarded as built up from variables out of the set $\{p_1, p_2, p_3\}$, represents a truth function f whose table of values matches the truth table of $\neg(p_1 \vee p_3)$ relative to these 3 variables, thus with 8 rows. The rows on which $f(x_1, x_2, x_3) = T$ correspond to the truth assignments making both p_1 and p_3 false, with p_2 given any truth value. So the relevant triples for the dnf are $\langle F, T, F\rangle$ and $\langle F, F, F\rangle$, giving the dnf

$$((\neg p_1 \wedge p_2 \wedge \neg p_3) \vee (\neg p_1 \wedge \neg p_2 \wedge \neg p_3)).$$

There is a similar result to Theorem 2.6 in terms of conjunctive forms, which we also met in Theorem 2.5 of Section 2.4. A formula θ is said to be in *conjunctive normal form*, abbreviated as *cnf*, if, for some n, it is a conjunction of disjunctions of propositional variables and their negations, where each variable in the set $\{p_1, p_2, \ldots, p_n\}$ appears exactly once in each disjunction, for example

$$((p_1 \vee \neg p_2 \vee p_3) \wedge (p_1 \vee \neg p_2 \vee \neg p_3) \wedge (\neg p_1 \vee p_2 \vee \neg p_3)).$$

In the next exercise, we invite you to prove in two different ways that every truth function f can be represented by a formula in cnf.

Exercise 2.51

(a) Prove that any truth function f can be represented by a formula ϕ in cnf by negating a formula in dnf representing a suitably chosen truth function related to f and manipulating the resulting formula.

(b) Prove that any truth function f can be represented by a formula ϕ in cnf by an adaptation of the proof of Theorem 2.6, using the table of values of the function f. [*Hints*: Which rows should you look at? Also for each row you look at, try to construct in a systematic way a formula of the form $(q_1 \vee q_2 \vee \ldots \vee q_n)$, where each q_i is one of p_i and $\neg p_i$.]

(c) Write down formulas in cnf representing each of the truth functions in Exercise 2.50.

As a check, our cnf for $(p_1 \leftrightarrow p_2)$ is $((\neg p_1 \vee p_2) \wedge (p_1 \vee \neg p_2))$.

Once we have one adequate set of connectives, it's easy to show that some other set S is adequate – all one has to do is show that each of $\neg, \wedge, \vee$ can be represented by a formula using connectives in S. For instance, as we have $(\phi \vee \psi) \equiv \neg(\neg\phi \wedge \neg\psi)$, so that $\vee$ can be represented by a formula using $\neg$ and $\wedge$, the set $\{\neg, \wedge\}$ is adequate.

Exercise 2.52

Show that each of the following sets of connectives is adequate.

(a) $\{\neg, \vee\}$

(b) $\{\neg, \rightarrow\}$

(c) $\{|\}$ where $|$ is a two-place connective with truth table:

p	q	$(p \vert q)$
T	T	F
T	F	T
F	T	T
F	F	T

The symbol $|$ is often called 'nand' or 'not (... and ...)' – you have to say it so that the bracketing is unambiguous! – or the *Sheffer stroke*, and $(\phi|\psi)$ is logically equivalent to $\neg(\phi \wedge \psi)$.

The following definition introduces a couple of new symbols which we shall use occasionally later in the book.

Definitions Propositional constants

The symbol $\perp$ is called the *propositional constant for falsity*. The symbol $\top$ is called the *propositional constant for truth*. These propositional constants are used in the construction of formulas as though they are propositional variables, but for every truth assignment v,

$$v(\perp) = F \text{ and } v(\top) = T.$$

What distinguishes a propositional constant from a variable is that its truth value is the same for every truth assignment v.

Thus

$$(\perp \rightarrow (p \rightarrow \perp)),\ \perp,\ \neg\top,\ (q \rightarrow \top)$$

are formulas, while $\perp p$ and $\top\perp$ are not. Also for all truth assignments v,

$$v((\perp \wedge p)) = F, \quad v((\top \wedge p)) = v(p), \quad v((\top \vee \perp)) = T.$$

Exercise 2.53

Show that the set $\{\to, \bot\}$ is adequate.

One of the benefits of knowing that a set of connectives is adequate comes when devising a formal proof system for propositional calculus. A reasonable aim for such a system is that one should be able to prove within it all tautologies, while at the same time not involving too many connectives for which the formal system would have to give rules or axiom schemes. An adequate set of connectives will do nicely!

How might one show that a set S of connectives is *not* adequate? This turns out to be less straightforward and requires a variety of tricks, depending on the connectives involved. For instance, the set $\{\wedge, \vee\}$ is not adequate. Obviously, if we could find a formula involving these connectives that represented $\neg$, the set would be adequate. A spot of experimenting with shortish formulas using $\wedge$ and $\vee$ will not come up with such a formula, but how do we know that there isn't some very long formula that does the job? One general principle to answer this is to look for some special property possessed by *all* formulas of a particular sort built up using connectives in S which thereby rules out that all truth functions can be represented. In the case of $S = \{\wedge, \vee\}$, one such property is the following.

There are other such properties that would do, for instance that for any ϕ built up from $p_1, p_2, \ldots, p_n$ with $\wedge$ and $\vee$, if v is the truth assignment such that $v(p_i) = T$ for all $i = 1, 2, \ldots, n$, then $v(\phi) = T$. That would mean that no truth function f of n arguments such that $f(T, T, \ldots, T) = F$ can be represented.

> Any formula built up using a single variable p (as many times as you like!) with $\wedge$ and $\vee$ always takes the value T under the truth assignment v such that $v(p) = T$.

This means that such a formula could not represent one of the (two) truth functions $f\colon \{T, F\} \longrightarrow \{T, F\}$ such that $f(T) = F$, so that the set S is not adequate.

It is pretty easy to see that this property holds, but it is a useful discipline to prove it properly. A key feature of the proof is that it has to encompass all formulas ϕ of the sort described, and one of the standard ways of doing this is by induction on some measure of the length of ϕ. Have a go at the proof in the next exercise. In our solution we shall as ever use the number of connectives in ϕ as the measure of its length.

The essence of why the property holds is that the truth tables of both $(\theta \wedge \psi)$ and $(\theta \vee \psi)$ give the value T when ϕ and ψ take the value T.

Exercise 2.54

Prove that if ϕ is built up using the variable p with $\wedge$ and $\vee$, and v is the truth assignment such that $v(p) = T$, then $v(\phi) = T$.

Solution

We use induction on the number n of connectives in ϕ.

The base case is $n = 0$. The only formula of the given type with no connectives is p and the required property, namely that if $v(p) = T$ then $v(\phi) = T$, holds trivially.

For the inductive step assume that the property holds for all the formulas of this special form with $\leq n$ connectives, and that ϕ is a formula of this type with $n+1$ connectives. Then ϕ has to be one of the forms $(\theta \wedge \psi)$ and $(\theta \vee \psi)$, where, as θ and ψ have the same form and at most n connectives, both have the required property, i.e. if $v(p) = T$ then $v(\theta) = v(\psi) = T$. Then whichever form ϕ has, the truth tables of $\wedge$ and $\vee$ ensure that $v(\phi) = T$, as required.

The result follows by mathematical induction.

As usual with such arguments, it is almost always essential to frame the property of formulas ϕ of the special sort so that it includes the case when ϕ is a propositional variable, which contains no connectives – hence $n = 0$. For the inductive step, the $(n+1)$th connective, in this case, joins two formulas whose combined connective length is n – we don't know how many connectives are in each, but we do know that in each case it's n or less.

Use your ingenuity to resolve the following problems and provide suitably convincing arguments!

Exercise 2.55

Show that none of the following sets of connectives is adequate.

(a) $\{\neg\}$

(b) $\{\rightarrow\}$

(c) $\{\vee, \perp\}$, where $\perp$ is the propositional constant for falsity.

Exercise 2.56

Let $S = \{\neg, \leftrightarrow\}$.

(a) Show that every truth function of one argument can be represented by a formula using connectives in $\{\neg, \leftrightarrow\}$.

(b) By finding a property possessed by all formulas built up from two propositional variables p and q using $\neg$ and $\leftrightarrow$ (and verifying that this property does indeed hold), show that the set $\{\neg, \leftrightarrow\}$ is not adequate.

Exercise 2.57

There are 16 different possible truth tables for a two-place connective $*$. For which of these is $\{*\}$ an adequate set of connectives? In each case explain why it gives or does not give (as appropriate) an adequate set.

An alternative approach to showing that a set S of connectives is not adequate is to investigate, for each non-negative integer n, how many truth functions of n arguments can be represented using S. If for some n this number is less than 2^{2^n}, then S is not adequate. Meanwhile, knowing how many, and which, truth functions can be represented by S is of interest in its own right.

Exercise 2.58

How many truth functions of n arguments can be represented using the set $\{\wedge\}$?

Solution

Our method is to investigate whether there is some sort of 'normal' form for formulas ϕ built up using variables in the set $\{p_1, p_2, \ldots, p_n\}$ and connectives out of the set S, in this case just the connective $\wedge$. It would be nice if each ϕ was logically equivalent to just one formula in this normal form, so that counting the number of formulas in normal form gives the number of different truth functions which can be represented. In this case, there are some very useful logical equivalences involving $\wedge$ which lead to such a normal form. They are as follows:

Logically equivalent formulas, regarded as using variables out of the set $\{p_1, p_2, \ldots, p_n\}$, represent the same truth function. So to count the different truth functions representable, we just want one formula out of each class of logically equivalent formulas.

$$(\phi \wedge (\psi \wedge \theta)) \equiv ((\phi \wedge \psi) \wedge \theta) \quad \text{(associativity)},$$
$$(\phi \wedge \psi) \equiv (\psi \wedge \phi) \quad \text{(commutativity)},$$
$$(\phi \wedge \phi) \equiv \phi \quad \text{(idempotency)}.$$

Recall that the correct terminology here is that $\wedge$ is associative, commutative and idempotent under logical equivalence. The associativity and commutativity are particularly helpful, because using these we can show that given any formula ϕ built up just using $\wedge$, any rearrangement of the variables in ϕ gives a formula logically equivalent to ϕ. In particular we can rearrange the variables in ϕ so that all the p_i, for a given i, are in the same subformula θ_i, and obtain a formula logically equivalent to ϕ of the form

$$(\theta_{i_1} \wedge (\theta_{i_2} \wedge (\ldots \wedge \theta_{i_k})\ldots))),$$

where the propositional variables appearing in ϕ are $p_{i_1}, p_{i_2}, \ldots, p_{i_k}$, with $i_1 < i_2 < \ldots < i_k$. For example, the formula

Not all of the n propositional variables in $\{p_1, p_2, \ldots, p_n\}$ might appear in a given ϕ.

$$((p_3 \wedge p_1) \wedge (p_3 \wedge ((p_4 \wedge p_1) \wedge p_3)))$$

is logically equivalent to the formula

$$((p_1 \wedge p_1) \wedge ((p_3 \wedge (p_3 \wedge p_3)) \wedge p_4)).$$

The idempotency of $\wedge$ under logical equivalence then gives us that each component θ_i is logically equivalent to just a single p_i. Thus a given ϕ is logically equivalent to a normal form which is a simple conjunction of just the p_is appearing in ϕ, that is,

$$(p_{i_1} \wedge p_{i_2} \wedge \ldots \wedge p_{i_k}),$$

Back to being casual about brackets, thanks to associativity!

where the propositional variables appearing in ϕ are $p_{i_1}, p_{i_2}, \ldots, p_{i_k}$ with $i_1 < i_2 < \ldots < i_k$. In the example above, this would be the formula

$$(p_1 \wedge p_3 \wedge p_4).$$

The number of distinct formulas using variables in the set $\{p_1, p_2, \ldots, p_n\}$ which are in this normal form equals the number of non-empty subsets of this set, namely $2^n - 1$.

The set $\{p_1, p_2, \ldots, p_n\}$ has n elements, so has 2^n subsets, including the empty set.

The same sort of method pays dividends in many parts of the following exercises.

Exercise 2.59

How many truth functions of n variables can be represented by using each of the following sets of connectives?

(a) $\{\vee\}$

(b) $\{\neg\}$

(c) $\{\neg, \wedge, \vee\}$

(d) $\{\leftrightarrow\}$

(e) $\{\neg, \leftrightarrow\}$

The result of this gives a nice alternative way of showing that the set $\{\neg, \leftrightarrow\}$ is not adequate.

Exercise 2.60

(a) Let f be a truth function of n arguments such that $f(T, T, \ldots, T) = T$. Show that f can be represented by a formula using connectives in the set $\{\wedge, \vee, \rightarrow\}$. [*Hints*: f can be represented by a formula ϕ using $\{\neg, \wedge, \vee\}$ in dnf (or cnf, whichever you prefer). The fact that $f(T, T, \ldots, T) = T$ gives just enough information about ϕ to enable all the occurrences of $\neg$ to be eliminated using $\wedge$, $\vee$ and $\rightarrow$, with the aid of logical equivalences such as $(\neg\theta \vee \psi) \equiv (\theta \rightarrow \psi)$.]

(b) Hence show that the number of truth functions of n arguments representable using $\{\wedge, \vee, \rightarrow\}$ is $2^{2^n - 1}$ and also find the number when using $\{\wedge, \rightarrow\}$.

You may be surprised to know that there is no known nice formula for the number of truth functions of n arguments representable using such a simple set as $\{\wedge, \vee\}$, which has been sought for a variety of applications, but for which only not very good upper and lower bounds have been found.

The problem for $\{\wedge, \vee\}$ is equivalent to ones in the contexts of Boolean algebra and sets, e.g. how many different sets can be created from up to n sets by taking unions and intersections.

Further exercises

Exercise 2.61

Let $*$ be a ternary connective with the following truth table.

p	q	r	$*(p, q, r)$
T	T	T	F
T	T	F	T
T	F	T	T
T	F	F	T
F	T	T	T
F	T	F	T
F	F	T	T
F	F	F	F

(a) Prove that $\{*\}$ is not adequate.

(b) Is $\{*, \rightarrow\}$ adequate? Prove your answer.

Exercise 2.62

Suppose that ϕ is a formula in disjunctive normal form (but not necessarily involving all the propositional variables in each disjunct). Prove that ϕ is a contradiction if and only if for each disjunct ψ of ϕ there is some propositional variable p such that both p and $\neg p$ appear in ψ.

Exercise 2.63

The result that $\{\wedge, \vee, \neg\}$ is an adequate set of connectives can be restated by saying that all truth functions $f\colon \{T, F\}^n \longrightarrow \{T, F\}$, for all $n \geq 0$, can be obtained by (repeated) composition of the three truth functions $f_\wedge$, $f_\vee$ and $f_\neg$ (of respectively 2, 2 and 1 arguments) corresponding to the connectives $\wedge$, $\vee$ and $\neg$. Ignoring the particular interpretation of T and F as truth values, we can conclude from this that the set of all functions from X^n to X, where X is a two element set and $n \geq 0$, can be obtained by composition of a finite number (three!) of relatively simple functions. Show that the same applies for any finite set X, that is, that there is a finite subset of the set of all functions from X^n to X for all $n \geq 0$ from which any function in the set can be obtained by composition.

2.6 Logical consequence

In this section we shall start looking at arguments involving propositional formulas, which will be our first step towards modelling mathematical proof. One's expectation of a (correct!) mathematical argument is that it should involve statements which follow from previous statements in some sort of convincing way, right up to the desired concluding result. It is a tall order trying to nail down all the ways in which mathematicians are convinced by an argument, so in this section we shall concentrate on just one requirement of a convincing argument, as follows. One measure of a statement ϕ following from statements in a set Γ is that whenever all the statements in Γ are true then ϕ must also be true. So, for instance, in everyday maths it follows from the statement that the function f from $\mathbb{R}$ to $\mathbb{R}$ is differentiable, along with all sorts of other tacit assumptions about the arithmetic of $\mathbb{R}$, that f is continuous. It is not the case that every function f is differentiable, but whenever one does have an f that is differentiable, then it must also be continuous. We shall capture this general idea by the following definition.

Think of the statements in Γ as the assumptions underlying the argument.

Definitions Logical consequence

Let Γ be a set of formulas and ϕ a formula involving propositional variables in a set P. Then ϕ is a *logical consequence* of Γ, or equivalently Γ *logically implies* ϕ, when for all truth assignments v on P, if $v(\gamma) = T$ for all $\gamma \in \Gamma$, then $v(\phi) = T$. We write this as $\Gamma \vDash \phi$.

In the case where Γ is the empty set, we write $\vDash \phi$ to say that for all truth assignments v, $v(\phi) = T$, i.e. ϕ is a tautology.

When ϕ is not a logical consequence of Γ, we write $\Gamma \nvDash \phi$. Similarly when ϕ is not a tautology, we write $\nvDash \phi$.

Another description of $\Gamma \vDash \phi$ is that every truth assignment satisfying Γ also satisfies ϕ. Informally, ϕ is true whenever Γ is true.

So we have

$$\models ((p \wedge q) \rightarrow p),$$

as the formula is a tautology. We have

$$\{q, (r \rightarrow \neg p)\} \models (q \vee r)$$

as for each of the truth assignments v satisfying both q and $(r \rightarrow \neg p)$, v also satisfies $(q \vee r)$. Thus we have

Three of the eight different truth assignments on $\{p, q, r\}$ satisfy both q and $(r \rightarrow \neg p)$.

$$\{q, (r \rightarrow \neg p)\} \not\models (q \rightarrow r)$$

as there is a truth assignment v satisfying both q and $(r \rightarrow \neg p)$ which does not satisfy $(q \rightarrow r)$, for instance v defined by

$$v(p) = v(q) = T, \ v(r) = F.$$

Exercise 2.64

Decide which of the following logical implications hold.

(a) $\{p, \neg r\} \models (q \rightarrow (r \rightarrow \neg p))$

(b) $\{p, (q \leftrightarrow r)\} \models (q \rightarrow (r \rightarrow \neg p))$

(c) $\models ((p \rightarrow q) \rightarrow p)$

(d) $\{(p \vee q)\} \models ((p \rightarrow q) \rightarrow q)$

(e) $\{p_{2i} : i \in \mathbb{N}\} \models ((p_{17} \rightarrow p_{14}) \rightarrow p_{87})$

(f) $\{(p_{2i} \rightarrow p_i) : i \in \mathbb{N}\} \models ((p_{34} \vee p_{17}) \rightarrow p_{17})$

Solution

(a) In any truth assignment v satisfying all the formulas in the set $\{p, \neg r\}$, we must have $v(p) = T$ and $v(r) = F$. Then $v((r \rightarrow \neg p)) = T$, so regardless of the value of $v(q)$, we have $v((q \rightarrow (r \rightarrow \neg p))) = T$. Thus it is the case that $\{p, \neg r\} \models (q \rightarrow (r \rightarrow \neg p))$.

(b) The truth assignment v defined by $v(p) = v(q) = v(r) = T$ satisfies all the formulas in $\{p, (q \leftrightarrow r)\}$, but $v((q \rightarrow (r \rightarrow \neg p))) = F$. So it is not the case that $\{p, (q \leftrightarrow r)\} \models (q \rightarrow (r \rightarrow \neg p))$.

Equivalently, we have shown that $\{p, (q \leftrightarrow r)\} \not\models (q \rightarrow (r \rightarrow \neg p))$.

(c) Not given.

(d) Suppose that the truth assignment v satisfies $(p \vee q)$. If $v(q) = T$, then $v(((p \rightarrow q) \rightarrow q)) = T$. If $v(q) = F$, so that $v(p) = T$, we have $v((p \rightarrow q)) = F$, so that $v(((p \rightarrow q) \rightarrow q)) = T$.

Thus in all cases where $v((p \vee q)) = T$, we have $v(((p \rightarrow q) \rightarrow q)) = T$, so that $\{(p \vee q)\} \models ((p \rightarrow q) \rightarrow q)$.

(e) Not given.

(f) Not given.

Exercise 2.65

Is $\Gamma \not\models \phi$ equivalent to saying $\Gamma \models \neg\phi$?

When we extend the definition of logical consequence to the more complicated, and mathematically more useful, predicate languages in Chapter 4, you will see that the idea does capture something of great importance. To give you a foretaste, the set Γ might give axioms for an interesting theory, for instance group theory, and $\Gamma \vDash \phi$ will then mean that in every structure which makes all of Γ true, i.e. in every group, the formula ϕ is also true. Then the property of groups that statement ϕ represents holds for *all* groups.

In, for instance, our solution to Exercise 2.64(d), we establish a logical consequence $\Gamma \vDash \phi$ by direct appeal to the definition, by looking at all truth assignments which satisfy Γ. But this is quite far from how we usually infer statements from others within a mathematical proof. For instance, while it is the case that a function f being continuous is a logical consequence of f being differentiable, this is normally established by a sequence of several non-trivial steps. In general, we tend to use quite small steps in proofs and in the exercise below we give logical consequences corresponding to some very small such steps. From now on in this section we shall concentrate on inferences involving propositional formulas. Later in the book we shall look at inferences involving a richer language, closer to one usable for everyday mathematics.

Notation

We shall sometimes cheat on set notation for the Γ in $\Gamma \vDash \phi$ by dropping some of the set brackets $\{\ \}$, writing e.g.

$$\theta, \psi \vDash \phi \text{ instead of } \{\theta, \psi\} \vDash \phi,$$
$$\Gamma, \theta \vDash \phi \text{ instead of } \Gamma \cup \{\theta\} \vDash \phi$$
$$\text{and } \Gamma, \Delta \vDash \phi \text{ instead of } \Gamma \cup \Delta \vDash \phi.$$

We hope that the context will make it clear what is meant.

None of the parts of the next exercise should be very challenging, but even if you don't attempt them all, do read all the parts of the exercise as they are potentially more important than their simplicity suggests.

Exercise 2.66

Let ϕ, ψ, θ be formulas. Show each of the following.

(a) $(\phi \wedge \psi) \vDash \phi$

(b) $(\phi \wedge \psi) \vDash \psi$

(c) $\phi, \psi \vDash (\phi \wedge \psi)$

(d) $\phi \vDash (\phi \vee \psi)$

(e) $\phi \vDash (\psi \vee \phi)$

(f) If $\phi \vDash \theta$ and $\psi \vDash \theta$, then $(\phi \vee \psi) \vDash \theta$.

With this and later exercises, turn the symbols into natural language when you think about the problem. For example, you might think of part (c) as saying that any assignment making both the formulas ϕ and ψ true must make the formula $(\phi \wedge \psi)$ true.

Solution

We shall give a solution only to part (a), to give you an idea of how simple a convincing explanation can be!

If a truth assignment v satisfies $(\phi \wedge \psi)$, then from the truth table of $\wedge$ we must have $v(\phi) = T$ $(= v(\psi))$.

These simple logical consequences are important because they illustrate how we infer using the connectives 'and' and 'or'. Here are some more simple logical consequences of similar importance in representing small steps of inference.

Exercise 2.67

Let ϕ, ψ, θ, χ be formulas. Show each of the following.

(a) $\phi, (\phi \to \psi) \vDash \psi$

(b) If $\phi \vDash \psi$ then $\neg\psi \vDash \neg\phi$.

(c) If $\phi \vDash \psi$ and $\theta \vDash \chi$, then $(\phi \wedge \theta) \vDash (\psi \wedge \chi)$.

(d) If $\phi \vDash \psi$ and $\theta \vDash \chi$, then $(\phi \vee \theta) \vDash (\psi \vee \chi)$.

The result of the first part of this exercise, inferring ψ from ϕ and $(\phi \to \psi)$, has been regarded as so important that it has been given a special name, *Modus Ponens*. It is plainly a crucial feature of how to infer with $\to$. When we come to our formal proof system in the next chapter, we shall adopt a formal rule of inference corresponding to Modus Ponens. One yardstick of a formal system will be whether it can mirror other simple logical consequences – if it cannot, what hope for deriving those that are more complicated!

We hope that it comes as no surprise that there are strong connections between logical consequence and the connective $\to$ which represents implication. One such connection is given by the following theorem.

Theorem 2.7

Let Γ be a set of formulas and ϕ, ψ be formulas. Show that

$$\Gamma, \phi \vDash \psi \quad \text{if and only if} \quad \Gamma \vDash (\phi \to \psi).$$

We shall be very keen to match this result within our formal system.

Exercise 2.68

Prove Theorem 2.7.

Exercise 2.69

Let $\gamma_1, \gamma_2, \ldots, \gamma_n$ be finitely many formulas and ϕ a formula. Show that

$$\gamma_1, \gamma_2, \ldots, \gamma_n \vDash \phi \quad \text{if and only if} \quad \vDash ((\gamma_1 \wedge \gamma_2 \wedge \ldots \wedge \gamma_n) \to \phi).$$

There are useful connections between logical consequence and logical equivalence, as you can show, we hope straightforwardly, in the next exercise.

Exercise 2.70

Let ϕ, ψ, θ be formulas.

(a) Show that $\phi \equiv \psi$ if and only if $\phi \vDash \psi$ and $\psi \vDash \phi$.

(b) Suppose that $\phi \equiv \psi$. Show that

(i) if $\phi \vDash \theta$, then $\psi \vDash \theta$;

(ii) if $\theta \vDash \phi$, then $\theta \vDash \psi$.

The next exercise involves a very straightforward and useful result about cascading logical consequences. Again, this illustrates further connections between $\vDash$ and $\to$.

Exercise 2.71

(a) Let ϕ, ψ, θ be formulas. Show that if $\phi \vDash \psi$ and $\psi \vDash \theta$, then $\phi \vDash \theta$.

(b) Let Γ be a set of formulas and $\phi_1, \phi_2, \ldots, \phi_n$ finitely many formulas such that

$$\Gamma \vDash \phi_1, \quad \phi_1 \vDash \phi_2, \quad \phi_2 \vDash \phi_3, \quad \ldots, \quad \phi_{n-1} \vDash \phi_n.$$

Show that $\Gamma \vDash \phi_n$.

One way of using the result of the last exercise is to combine simple logical consequences to establish a more complicated logical consequence. But not every complicated logical consequence would necessarily get broken down in such a linear fashion. There might be other routes, for instance as in the following exercise.

Exercise 2.72

Let Γ, Δ be sets of formulas and ϕ, ψ, θ formulas. Suppose that $\Gamma \vDash (\phi \vee \psi)$, $\Delta, \phi \vDash \theta$ and $\psi \vDash \theta$. Show that $\Gamma, \Delta \vDash \theta$.

We have previously introduced the symbols $\perp$ and $\top$ as the propositional constants for falsity and truth, respectively. Here's an easy exercise involving them.

Recall from Section 2.5 that these propositional constants are used in formulas just like propositional variables, but that for every truth assignment v, $v(\perp) = F$ and $v(\top) = T$.

Exercise 2.73

Let ϕ, ψ be any formulas. Show the following.

(a) $(\phi \vee \neg\phi) \equiv \top$

(b) $(\phi \wedge \neg\phi) \equiv \perp$

(c) $\psi \vDash \top$

(d) $\perp \vDash \psi$

(e) $\top \vDash \psi$ if and only if ψ is a tautology.

(f) $\psi \vDash \perp$ if and only if ψ is a contradiction.

So $\top$ is a tautology and $\perp$ is a contradiction.

Parts (b) and (d) are of particular interest. If we have a set of formulas Γ and a formula ϕ for which both ϕ and $\neg\phi$ are logical consequences of Γ, it is easy to show that Γ logically implies the contradiction $(\phi \wedge \neg\phi)$, so that using (b) and (d) (and essentially the result of Exercise 2.71(b)), we can show that all formulas ψ are logical consequences of Γ. Such sets Γ most certainly exist – for instance, just take Γ to be the set $\{p, \neg p\}$ for a propositional variable p – but they are somehow rather undiscriminating when it comes to investigating their logical consequences! In fact, for such a set Γ, there are no truth assignments making all of the formulas of Γ true. We shall ask you to show this, phrasing the result using the following new terminology.

Definition Satisfiable

The set Γ of formulas is *satisfiable* if there is some truth assignment v which satisfies Γ, i.e. $v(\gamma) = T$ for all $\gamma \in \Gamma$.

Plainly the issue of whether there are circumstances in which all statements in a given set can be made simultaneously true, i.e. are satisfiable, will be of interest, for instance when the statements attempt to axiomatize a mathematical theory.

Exercise 2.74

Let Γ be a set of formulas. Show that $\Gamma \vDash \bot$ if and only if Γ is not satisfiable.

Exercise 2.75

Let Γ be a set of formulas and ϕ a formula such that both $\Gamma \vDash \phi$ and $\Gamma \vDash \neg\phi$. Show that $\Gamma \vDash \psi$, for all formulas ψ.

The converse result holds trivially!

Exercise 2.76

Let Γ be a set of formulas and ϕ a formula. Show that $\Gamma \cup \{\neg\phi\}$ is satisfiable if and only if $\Gamma \nvDash \phi$.

Solution

We shall show that if $\Gamma \nvDash \phi$, then $\Gamma \cup \{\neg\phi\}$ is satisfiable, and we leave the rest of the proof to you!

Suppose that $\Gamma \nvDash \phi$. So it is not the case that all truth assignments v which satisfy Γ also satisfy ϕ. So there is some truth assignment v which satisfies Γ and doesn't satisfy ϕ. As $v(\phi) \neq T$, this means $v(\phi) = F$, so that $v(\neg\phi) = T$. Therefore this v satisfies $\Gamma \cup \{\neg\phi\}$, which is thus satisfiable.

Exercise 2.77

Let Γ be a set of formulas. Show that Γ is satisfiable if and only $\Gamma \nvDash \phi$ for some formula ϕ.

It might be tempting to think that a set of formulas Γ for which $\Gamma \vDash \perp$ is in some sense silly. But such a set is often of great value within a mathematical proof, in the context of what is called *proof by contradiction*. You should have met this before and, if you are anything like the author, have been so excited by this method of proof that you spent a long period trying to use it in every mathematical argument! A classical argument is that found in Euclid (at around 300 BC) to show that there are infinitely many primes. A modern version of this proof goes as follows.

This has the grander Latin name of *reductio ad absurdum*.

Suppose that there are only finitely many primes, listed as $p_1, p_2, \ldots, p_n$. Consider the number $N = p_1 p_2 \ldots p_n + 1$. As division by any p_i leaves remainder 1, none of $p_1, p_2, \ldots, p_n$ divides N. But N can be factorized as a product of primes, so there is another prime dividing N not equal to one of $p_1, p_2, \ldots, p_n$. This contradicts that $p_1, p_2, \ldots, p_n$ lists all the primes. Thus there are infinitely many primes.

For the purposes of this section, the underlying structure of this proof by contradiction is as follows: to prove that ϕ follows from the set of formulas Δ, we assume the negation of ϕ, i.e. $\neg\phi$, and from this and Δ derive a contradiction. Hey presto! This means that from Δ we can infer the hoped for ϕ. Formally we have the following theorem.

In the proof above, ϕ says that there are infinitely many primes while Δ in some way gives more fundamental properties of the integers.

Theorem 2.8 Proof by contradiction

Let Δ be a set of formulas and ϕ a formula. If $\Delta, \neg\phi \vDash \perp$ then $\Delta \vDash \phi$.

Proof

Suppose that $\Delta \cup \{\neg\phi\} \vDash \perp$. Then by the result of Exercise 2.74, there are no truth assignments making all of the formulas in $\Delta \cup \{\neg\phi\}$ true. This means that if the truth assignment v makes all the formulas of Δ true, $v(\neg\phi)$ must be false, so that $v(\phi)$ must be true; that is, every truth assignment satisfying Δ also satisfies ϕ. Thus $\Delta \vDash \phi$. ■

Exercise 2.78

Show that the converse of the theorem above holds, that is, if Δ is a set of formulas and ϕ a formula such that $\Delta \vDash \phi$, then $\Delta \cup \{\neg\phi\} \vDash \perp$.

Exercise 2.79

Let Δ be a set of formulas and ϕ a formula. Show that if $\Delta, \phi \vDash \perp$ then $\Delta \vDash \neg\phi$.

Exercise 2.80

Let ϕ, θ be formulas such that $\phi, \theta \vDash \perp$. Show that $\phi \vDash \neg\theta$ and $\theta \vDash \neg\phi$. Is it necessarily the case that $\vDash (\neg\phi \wedge \neg\theta)$?

When we extend our formalisation of language to predicate languages and write down some set of formulas Γ axiomatizing something of real mathematical significance, like the theory of groups or of linear orders, we shall not only be interested in whether a particular formula ϕ is a consequence of Γ but in finding all formulas which are consequences of Γ. One tempting possibility is that for a given a set of formulas Γ and any formula ϕ, we have that $\Gamma \models \phi$ or $\Gamma \models \neg\phi$.

We might even have that both hold, so that by the result of Exercise 2.75, $\Gamma \models \psi$ for all formulas ψ. But then by Exercise 2.74, there are no truth assignments making Γ true.

Exercise 2.81

Let Γ be the set $\{(p \vee q)\}$, where p, q are propositional variables. Is it the case that for all formulas ϕ in the language using just these two propositional variables that $\Gamma \models \phi$ or $\Gamma \models \neg\phi$?

Solution

No. Consider the formula ϕ given by p. Then taking the truth assignment v defined by $v(p) = F,\ v(q) = T$, we have $v((p \vee q)) = T$ and $v(p) = F$, so that $\Gamma \not\models \phi$. Taking the truth assignment u defined by $u(p) = T,\ u(q) = T$, we have $u((p \vee q)) = T$ and $u(\neg p) = F$ (here $\neg\phi$ is $\neg p$), so that $\Gamma \not\models \neg\phi$.

So in general, given a set Γ, we should not expect that for all formulas ϕ, $\Gamma \models \phi$ or $\Gamma \models \neg\phi$. There are some sets Γ for which this holds, besides those for which there is no truth assignment making all formulas of Γ true, and they are described as *complete*. We shall discuss these some more in the next chapter.

For an example giving many such sets Γ, see Exercise 2.87 later.

Some comments on decidability

The question of *how* we can tell whether ϕ is a logical consequence of the set Γ is rather interesting. One of the hopes of those who developed the ideas of describing interesting parts of mathematics using axioms was that the logical consequences of these axioms would be *decidable*, meaning that there is an algorithmic procedure which would, after a finite number of steps, say whether or not a given formula ϕ is a logical consequence of a set of formulas Γ.

See Davis [10] for some of the history, leading up to precise definitions of 'decidable' and, indeed, of 'algorithmic procedure'. For the exciting theory stemming from these ideas, see Cutland [9], Enderton [12], Epstein and Carnielli [14] or Kleene [22].

We shall have to wait till Chapter 5 for the predicate languages which we might use to axiomatize some interesting mathematics. For the moment we shall just discuss the decidability of $\Gamma \models \phi$ for propositional languages.

If Γ is the empty set, there is a very straightforward algorithmic procedure for deciding whether $\models \phi$, i.e. whether ϕ is a tautology – just construct its truth table and check whether it takes the value T on all the finitely many rows of this table.

We hope that you are willing to agree that one can give a finite set of simple instructions for producing the truth table of a formula ϕ that can be undertaken in finitely many steps.

What can be done when Γ is non-empty? If the set Γ is finite, we can create the formula $(\bigwedge_{\gamma \in \Gamma} \gamma \to \phi)$, look at its truth table and exploit the result of Exercise 2.69. The formula is a tautology exactly when $\Gamma \models \phi$ and checking its truth table involves just finitely many steps. For the mean-minded there is then the further question of whether the method involving the construction of a truth table is practicable. Checking whether a formula is a tautology involves a finitely long process which is good news for many purposes. But if there is a large number n of propositional variables involved, the number of

As for what is practicable, you may have observed the author's reluctance to produce truth tables of long formulas or ones involving more than 8 rows!

rows in the truth table, 2^n, will be so large as to make the process impracticable. So there is an interest in thinking about other ways of checking whether $\Gamma \vDash \phi$.

When Γ is an infinite set, it is highly unlikely that $\Gamma \vDash \phi$ will be decidable. For instance, suppose that the language has propositional variables in the set $\{p_i : i \in \mathbb{N}\}$. An attempt by brute force of checking all truth assignments v to see first whether v satisfies Γ and, if so, whether v also satisfies ϕ, is likely to be far from an algorithmic procedure, as there are uncountably infinitely many different truth assignments on the variables $\{p_i : i \in \mathbb{N}\}$, so the full check couldn't happen in finitely many steps. But sometimes an infinite set Γ has a simple enough structure to make checking whether $\Gamma \vDash \phi$ practicable. For instance, if Γ is the set

$$\{p_0\} \cup \{(p_i \to p_{i+1}) : i \in \mathbb{N}\},$$

there is actually only the one truth assignment making all the formulas of Γ true, namely v where $v(p_i) = T$ for all $i \in \mathbb{N}$. So checking whether $\Gamma \vDash \phi$ for a given formula ϕ is simply a matter of working out the truth value $v(\phi)$ for this v and seeing whether this is T! There are more challenging examples of this positive behaviour – see for example Exercise 2.86 below. But normally with an infinite set Γ, we are doomed. For instance, it can be shown that there are subsets I of the set $\mathbb{N}$ for which there is no algorithmic procedure for deciding whether or not a given natural number is in I. Take such a set I and let Γ be the set $\{p_i : i \in I\}$. Then we can't even decide whether $\Gamma \vDash p_n$ for each $n \in \mathbb{N}$ – this is equivalent to deciding whether $n \in I$ – let alone whether $\Gamma \vDash \phi$ for more complicated ϕ.

There's a bit of a reminder about the set-theoretic background and the theory of infinite sets in Section 6.4 of Chapter 6. The ideas of checking whether a truth assignment satisfies infinitely many formulas in Γ and then whether the possibly infinitely many truth assignments satisfying Γ also satisfy ϕ are meaningful to most modern mathematicians. But the practicalities of doing this checking are another matter!

A rather clever question which one can ask when Γ is infinite and $\Gamma \vDash \phi$ is whether all of the infinite amount of information coded in Γ is needed to logically imply ϕ. Perhaps there is some finite subset Δ of Γ for which $\Delta \vDash \phi$.

We shall address this question for infinite Γ using the soundness and completeness theorems in Section 3.3 of Chapter 3.

Exercise 2.82

Let Γ and Δ be sets of formulas with $\Delta \subseteq \Gamma$, and let ϕ be a formula.

(a) Show that if $\Delta \vDash \phi$ then $\Gamma \vDash \phi$.

(b) Give a counterexample to show that the converse of (a) is false.

In general the given subset Δ won't be suitable. But this doesn't mean that if $\Gamma \vDash \phi$ there might not be *some* non-trivial subset Δ of Γ for which $\Delta \vDash \phi$.

Within the predicate calculus, the logical consequences ϕ of a set Γ will be of much greater mathematical interest than those we have been looking at in this chapter. For instance, if Γ axiomatizes group theory, its logical consequences will be all statements that must be true for all groups. We shall be able to axiomatize group theory with finitely many axioms Γ. But checking whether $\Gamma \vDash \phi$ for this finite set Γ cannot be done by the same finite process as we gave above for dealing with formulas. For a finite set of propositional formulas Γ and a given ϕ, there are essentially only finitely many different truth assignments needed to check to see whether $\Gamma \vDash \phi$. But for the axioms of group theory, there are infinitely many groups making these axioms true for which we would have to check whether ϕ is also true. This cannot give a finite process for checking whether $\Gamma \vDash \phi$. For predicate calculus we thus have to investigate an alternative way of establishing logical consequence. Our chosen route is to look more closely at how we actually prove results within mathematics and it is both wise and very revealing to look first at how we might formally handle proofs using the relatively simple propositional formulas we have to hand right now. This is what we shall do in the next chapter. Bear in mind that our main aim will be to produce a formal proof system within which derivations correspond to logical consequences. That such proof systems can be found represents a considerable achievement!

Further exercises

Exercise 2.83

Which of the following sets of formulas are satisfiable? (p, q, r and the p_i, $i \in \mathbb{N}$, are propositional variables.)

(a) $\{(p \rightarrow q), (q \rightarrow r), (r \rightarrow p)\}$

(b) $\{(p \vee (q \leftrightarrow \neg p)), \neg(p \vee q)\}$

(c) $\{(p_i \leftrightarrow \neg p_j) : i < j,\ i, j \in \mathbb{N}\}$

Exercise 2.84

Three individuals, Green, Rose and Scarlet, are suspected of a crime. They testify under oath as follows.

This is based on an exercise in Kleene [22] attributed to H. Jerome Keisler.

GREEN: Rose is guilty and Scarlet is innocent.
ROSE: If Green is guilty, then so is Scarlet.
SCARLET: I am innocent, but at least one of the others is guilty.

(a) Could all the suspects be telling the truth?

(b) The testimony of one of the suspects follows from that of another. Identify which!

(c) Assuming that all three are innocent of the crime, who has committed perjury?

(d) Assuming that everyone's testimony is true, who is innocent and who is guilty?

(e) Assuming that those who are innocent told the truth and those who are guilty told lies, who is innocent and who is guilty?

Exercise 2.85

Suppose that L is a propositional language which, besides the usual connectives, also includes constants $\top$ (for true) and $\bot$ (for false). For any formula ϕ of L and any propositional variable p, write for short:

(ϕ/p) for $(\phi[\top/p] \vee \phi[\bot/p])$,

where $\phi[\psi/p]$ is the result of substituting the formula ψ for the variable p throughout ψ. Prove the following.

(a) $\phi \vDash (\phi/p)$.

(b) If $\phi \vDash \theta$ and p does not occur in the formula θ, then $(\phi/p) \vDash \theta$.

(c) If $\phi \vDash \theta$, then there is a formula ψ involving at most the propositional variables in common to ϕ and θ such that $\phi \vDash \psi$ and $\psi \vDash \theta$.

(d) If $\phi \vDash \theta$ and ϕ and θ have no propositional variables in common, then either $\neg\phi$ or θ is a tautology.

The desirable result of part (c), that the logical implication of θ from ϕ should somehow depend only on propositional variables in common to the formulas, is called *Craig's interpolation lemma*, after the American logician Bill Craig. The formula ψ is called the *interpolant* between ϕ and θ.

Exercise 2.86

Consider a propositional language with variables p_i for each $i \in \mathbb{N}$. Let Γ be the set of formulas

$$\{(p_n \to p_m) : n, m \in \mathbb{N}, n < m\}.$$

(a) Which of the following sets of formulas is satisfiable? In each case, justify your answer.

 (i) $\Gamma \cup \{p_0\}$

 (ii) $\Gamma \cup \{\neg p_0, p_1\}$

 (iii) $\Gamma \cup \{\neg p_2, p_1\}$

 (iv) $\Gamma \cup \{\neg p_1, p_2\}$

(b) Describe all the truth assignments which satisfy Γ, explaining why they satisfy Γ and how you know that you have found all possible assignments.

(c) For which pairs (m, n) is the formula $(p_n \to p_m)$ a logical consequence of Γ and for which pairs is this not the case?

(d) Find and describe an algorithmic procedure which, for any formula ϕ, decides whether $\Gamma \vDash \phi$. (For a given ϕ, such a procedure would have to decide within a finite number of steps whether or not $\Gamma \vDash \phi$.)

Exercise 2.87

Suppose that the language has the set $\{p_i : i \in \mathbb{N}\}$ of propositional variables and let Γ be the set $\{q_i : i \in \mathbb{N}\}$, where q_i is either p_i or $\neg p_i$ for each $i \in \mathbb{N}$. Show that for any formula ϕ in this language, exactly one of ϕ and $\neg\phi$ is a logical consequence of Γ.

Exercise 2.88

Suppose that Γ is an infinite set of formulas in the language with the finite set of propositional variables $\{p_1, p_2, \ldots, p_n\}$. Is the following argument correct?

> Every formula in Γ is logically equivalent to one of the 2^{2^n} formulas in dnf using these variables, so that there is a finite set Σ of these latter formulas such that, for all formulas ϕ in the language
>
> $$\Gamma \vDash \phi \quad \text{if and only if} \quad \Sigma \vDash \phi.$$
>
> As Σ is finite, there is an algorithmic procedure for deciding whether, for any formula ϕ, $\Sigma \vDash \phi$. Therefore there is an algorithmic procedure which, for any formula ϕ, decides whether $\Gamma \vDash \phi$.