Chapter 2
Propositional Logic

One of the basic notions in mathematics, whether we are talking of geometry, arithmetic or any other area, is that of truth and falsity. Reasoning is what mathematics is all about, and even if we are discussing a topic such as geometry, we need a mathematical infrastructure to reason about the way the truth of a statement follows from other known truths. One area of mathematics dealing with these concepts is *propositional logic*.

2.1 What Is a Proposition?

A proposition is a statement which is either true or false. We may not know whether or not it is true, but it must have a definite value.

Example 'Mice eat snakes', 'The world is round', 'Elephants are pink' and 'Whenever it rains I always carry an umbrella' are all examples of propositions. On the other hand, 'Where am I?', 'Go to school', 'The dark side of the moon' are examples of non-propositions. The first is a question, and the second is an imperative, and are thus not statements. The third is a phrase and not a whole sentence. You would be baffled if I were to say: 'Tell me whether the following is true ...' and then proceed to tell you 'Go to school'. With a proposition you may not know the answer, but you do know that it has to be either true or false. Examples of such propositions are 'Zejtun Red Stars play in the Maltese Premier league' and 'Jim Morrison died at the age of 28.'[1] Although you may not be able to answer 'true' or 'false' if asked about the truth of these propositions, you would be able to say 'I do not know, but it is either true or false'. ◇

[1] Assuming, of course, that you are not an ardent fan of The Doors.

G.J. Pace, *Mathematics of Discrete Structures for Computer Science*,
DOI 10.1007/978-3-642-29840-0_2, © Springer-Verlag Berlin Heidelberg 2012

Exercises

2.1 Which of the following are propositions:

(a) If I am wrong, then I am an idiot.
(b) Choose a number between 5 and 10.
(c) Have I already arrived at the town of True?
(d) 7 is the largest number.
(e) Numbers are odd.

2.2 The Language of Discourse

Consider the propositions 'I am carrying an umbrella', 'It is raining' and 'I am drenched'. From these propositions we can construct more complex propositions such as 'I am carrying an umbrella and it is raining', 'if it is raining and I am carrying an umbrella, then I am drenched' and 'I am carrying an umbrella unless it is raining.' We will now define a number of operators which will allow us to combine propositions together. Why should we bother doing this? Consider the compound proposition 'It is raining and I am drenched'. Whatever the state of the weather and my clothes, I might as well have said 'I am drenched and it is raining'. Similarly, had I said 'I am carrying an umbrella and it is raining', I might as well have said 'it is raining and I am carrying an umbrella'. Obviously, I can generalise: for any propositions P and Q, we would like to show mathematically that saying 'P and Q' is the same as saying 'Q and P'. To reason mathematically, we should have a clear notion of what operators we will be using and their exact mathematical meaning.

Definition 2.1 *Given two propositions P and Q, the* conjunction of P and Q *is written as $P \wedge Q$. We read this as 'P and Q', and it is considered to be true whenever both P and Q are true.* ■

Example We can write 'it is raining and I am carrying an umbrella' as 'it is raining' $\wedge$ 'I am carrying an umbrella'. If it is raining, and I am carrying a closed umbrella, this compound proposition is true, whereas if it is not raining, but I am carrying an open umbrella, it would be false. ◇

Definition 2.2 *Given two propositions P and Q, the* disjunction of P and Q *is written as $P \vee Q$. We read this as 'P or Q', and it is considered to be true whenever either P is true, or Q is true, or both.* ■

Example We can write 'I am carrying an umbrella or I am drenched' as 'I am carrying an umbrella' $\vee$ 'I am drenched'. If I am carrying a closed umbrella in the rain, this is true. If I am under the shower, carrying an open umbrella, this is still true. It would be false if I am dry and carrying no umbrella. ◇

Definition 2.3 *Given two propositions P and Q, we write $P \Rightarrow Q$ to mean 'If P is true, then so is Q', or 'P implies Q'. $P \Rightarrow Q$ is considered to be true unless P is true and Q is false.* ■

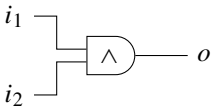
Example We can write 'If it is raining then I am drenched' as 'It is raining' $\Rightarrow$ 'I am drenched'. If it is raining, but I am inside and thus not wet, this proposition is false. If it is raining and I am having a shower, then it is true. If it is not raining, then it is true whether or not I am wet.

As another example, consider a politician who makes the statement 'If the economy improves, we will build a new hospital'. This can be written as 'Economy improves' $\Rightarrow$ 'A new hospital is built'. Does it correspond to our everyday use of the natural language statement? Let us consider the possible cases separately. If the economy improves and a new hospital is built, the politician has made a true statement. On the other hand, if the economy improves but no hospital is built, the politician clearly lied. Now let us look at the truth of the politician's statement if the economy does not improve. If the hospital is not built, everyone would agree that the politician did not make a false statement. What if a hospital is built anyway? One can argue that the politician made no commitment for the possible eventuality of the economy not improving. Therefore, even in this case, the statement was not a lie. If we compare this analysis to the meaning of implication, we see that the use of implication is justified. ◇

Definition 2.4 *Given two propositions P and Q, we take their bi-implication, written $P \Leftrightarrow Q$, to mean 'P if and only if Q'. This is considered to be true whenever P and Q are equivalent: either P and Q are both true, or P and Q are both false.* ■

Example 'You will pass the exam if and only if you study and you work out the exercises' can be written formally as 'You will pass the exam' $\Leftrightarrow$ ('you study' $\wedge$ 'you work the exercises'). If this is a true proposition, not only does it mean that if you do not study or if you do not work out the exercises you will not pass the exams, but also that if you do study and work the exercises, you will pass the exam.

Another use of bi-implication can be seen when describing a logic gate. Consider the following conjunction gate which takes two input wires i_1 and i_2 and outputs their conjunction on wire o:



If we use a proposition for each wire (i_1, i_2 and o) which is true if that particular wire carries a high signal, we can describe the behaviour of the circuit as: the output wire is high if and only if both input wires are carrying a high signal. This can be written as: $o \Leftrightarrow (i_1 \wedge i_2)$. The behaviour of the other logic gates can be similarly described. ◇

Definition 2.5 *Given a proposition P, the* negation of P *is written as $\neg P$. We read this as 'not P', and it is considered to be true whenever P is not true (P is false).* ■

Example 'If it is raining and I am not carrying an umbrella, then I will get wet or I will run quickly home' can be written as:

$$(\text{'it is raining'} \wedge \neg \text{'I am carrying an umbrella'}) \Rightarrow (\text{'I will get wet'} \vee \text{'I will run quickly home'}) \quad \diamond$$

We call a sentence made up of these operators a propositional formula. It is called *a well-formed formula* if is syntactically correct. The propositional formulae given in the examples above are all well-formed while, for example, ($\wedge$ 'it is raining') is not. We will describe well-formed formulae using the following table:

wff	::=	basic proposition
	|	(wff)
	|	*true*
	|	*false*
	|	$\neg$ wff
	|	wff $\wedge$ wff
	|	wff $\vee$ wff
	|	wff $\Rightarrow$ wff
	|	wff $\Leftrightarrow$ wff

It means that a well-formed formula (wff) can take any of the forms separated by |. This describes the *syntax* of the language—it identifies the class of formulae which we will be able to reason about. Note that we have added two entries we have not mentioned: '*true*' and '*false*', which are the propositions which are always true and always false, respectively.

Example We will now translate English sentences into formal propositional logic formulae. Remember that natural languages tend to be informal and imprecise, so sometimes more than one translation exists, depending on how they are interpreted.

Fixing the car: I am not sure what is wrong with my car. It is either the carburetor or the distributor. My mechanic said, whenever there is a problem with the carburetor, the sparking plugs need changing. Another mechanic told me that the distributor and the sparking plugs are closely connected. If one finds problems in one, there are invariably problems with the other.
Let C be the proposition *'there is a problem with the carburetor';* D be the proposition *'there is a problem with the distributor';* and S be *'the sparking plugs need changing.'*
The statements about the way the problems are related can be expressed as:

- It is either the carburetor or the distributor: $C \vee D$.
- Whenever there is a problem with the carburetor, the sparking plugs need changing: $C \Rightarrow S$.

- The distributor and the sparking plugs are closely connected. If one finds problems in one, there are invariably problems with the other: $D \Leftrightarrow S$.

What do you think is wrong with my car?

Eating habits: Jack and Jill have just moved in together. They have discovered that their eating habits are quite different. For Jack to be happy, he insists that, if it is cold but he is not very hungry, he would want to eat soup. Jill, on the other hand, never eats soup when it is cold. Assuming that at their place on any one day only one dish is cooked, express these conditions formally.
Let S be the proposition for '*today, Jack and Jill will eat soup*'; H for '*Jack is hungry*'; and C for '*it is cold*'.

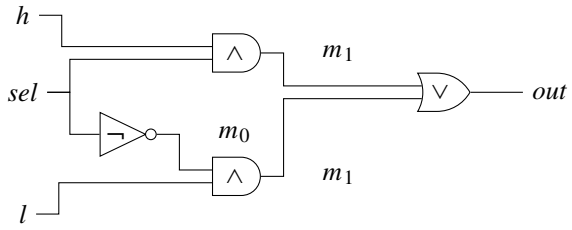
- Jack's constraint for happiness: $(C \wedge \neg H) \Rightarrow S$.
- Jill's constraint for happiness: $S \Rightarrow \neg C$.

There are different ways of writing Jill's constraint. For instance, another valid way of writing it is $C \Rightarrow \neg S$, or even $\neg(C \wedge S)$. Which to choose is largely a matter of interpretation of the English statement. Luckily, we will later develop mathematical tools to show that these three ways of writing her statement are equivalent.
Based on these known facts, do you think that they can live happily together?

Circuits: An engineer is trying to implement a multiplexer circuit which takes three input wires *sel*, l and h, and one output wire *out*. The behaviour of a multiplexer is very simple—when the selector wire *sel* is high, then the output wire *out* should be equal to the input h, while if *sel* is low, the output should be equal to the input l.
We will use a propositional variable for each wire in the circuit—so *sel* will be a propositional variable which is true if the input *sel* carries a high signal. Similarly, we will have variables *out*, l and h.
Before designing a circuit to implement the multiplexer, the engineer lists a number of properties it should satisfy:

- If *sel* is high, then *out* should be equal to h: $sel \Rightarrow (out \Leftrightarrow h)$.
- If *sel* is low, then *out* should be equal to l: $\neg sel \Rightarrow (out \Leftrightarrow l)$.
- If l and h have the same value, then the output should be the same as either of them: $(l \Leftrightarrow h) \Rightarrow (out \Leftrightarrow h)$.

The engineer then proceeds to implement the multiplexer circuit using negation, conjunction and disjunction gates as shown below:



Note that we have named the intermediate wires between gates to be able to talk about their value. As we have already seen, the behaviour of logic gates can be de-

scribed using the propositional logic operators. For instance, if a conjunction gate takes inputs i_1 and i_2 and has output o, its behaviour is described as $o \Leftrightarrow i_1 \wedge i_2$. The behaviour of the circuit designed by the engineer is thus equivalent to the conjunction of all the formulae describing the logic gates:

$$\begin{aligned} & (out \Leftrightarrow m_1 \vee m_2) \\ \wedge\ & (m_0 \Leftrightarrow \neg sel) \\ \wedge\ & (m_1 \Leftrightarrow sel \wedge h) \\ \wedge\ & (m_2 \Leftrightarrow m_0 \wedge l) \end{aligned}$$

Do you think that the engineer's circuit really satisfies the properties identified earlier? ◇

In arithmetic, to avoid overusing brackets, we give the operators a relative precedence. Thus, $3 \times 2 + 4$ is unambiguously interpreted as $(3 \times 2) + 4$. Mathematically, we say that multiplication has a higher precedence than addition—in an unbracketed expression, the multiplication is to be applied before addition. The operators with higher precedence can be seen as if surrounded by brackets. The following table shows the precedence levels of the propositional logic operators:

$$\underset{\longrightarrow}{\neg \quad \wedge \quad \vee \quad \Rightarrow \quad \Leftrightarrow}$$
$$\textit{decreasing precedence}$$

Therefore, $\neg A \Rightarrow B \wedge C \Leftrightarrow D$ effectively means $((\neg A) \Rightarrow (B \wedge C)) \Leftrightarrow D$. Although we can reduce brackets to a minimum, we usually use brackets to distinguish between $\wedge$ and $\vee$, and between $\Rightarrow$ and $\Leftrightarrow$. Therefore, we would usually write $A \vee (B \wedge C)$ even if $A \vee B \wedge C$ would do. Similarly, we write $A \Leftrightarrow (B \Rightarrow C)$ when $A \Leftrightarrow B \Rightarrow C$ would do.

Another issue is how to interpret an unbracketed expression with more than one use of the same operator, such as $A \wedge B \wedge C$. Should this be interpreted as $(A \wedge B) \wedge C$ or as $A \wedge (B \wedge C)$? Later on, we will prove that the two expressions are logically equivalent. However, syntactically, the expressions are different, and we thus have to decide which expression we mean when we leave out the brackets. Furthermore, the equivalence does not hold for all operators. Consider, for instance, subtraction. $(7 - 1) - 3$ is not equal to $7 - (1 - 3)$. Usually, when we write $7 - 1 - 3$, we mean $(7 - 1) - 3$. This is called *left associativity.* An operator $\oplus$ is said to be *right associative* when $x \oplus y \oplus z$ is to be interpreted as $x \oplus (y \oplus z)$. Conjunction and disjunction are left associative, while implication and bi-implication are right associative. Therefore, using precedence and associativity, we could write $A \wedge B \wedge C \Rightarrow D \Rightarrow E$ instead of $((A \wedge B) \wedge C) \Rightarrow (D \Rightarrow E)$.

Exercises

2.2 Write in mathematical notation:

(i) My loves include Sally or Jane. If I love Sally, I also love Jane.
(ii) Being a coward is the same as being a hero. If I were a hero, then either I can assess dangers well, or I am very foolish. In reply to the question 'Are you foolish?', I (truthfully) reply 'If that were so, then I am a coward.'

(iii) Swimming is fun unless it is not hot. If it is hot, it may be fun swimming, but playing tennis is certainly not.
(iv) In a circuit consisting of two negation gates connected in sequence, the output of the circuit is equal to its input.

2.3 Explain in English:

(i) B_1 and B_2 stand, respectively, for bulbs 1 and 2 being lit. S_1 and S_2 represent whether or not switches 1 and 2 are on or off. An engineer observes that $S_1 \wedge S_2 \Rightarrow B_1$, and $S_1 \Rightarrow B_2$. He also noticed that $B_2 \Rightarrow \neg S_2$ and that $B_1 \vee B_2$.
(ii) $T =$ I think, $A =$ I am. It has been said that $T \Rightarrow A$ and $A \vee \neg A$.
(iii) $A =$ I am thinking of an armadillo; $T =$ the animal I am thinking of has a tail; $F =$ the animal I am thinking of has feathers. I say: $T \vee \neg F \Rightarrow A$, $\neg(F \wedge T)$.
(iv) With the propositions $O =$ God is omnipotent and $C =$ God can create a rock She cannot lift, it is then true that $(O \Rightarrow C)$ and $(C \Rightarrow \neg O)$.

2.4 Add brackets to the following expressions:

(i) $A \wedge B \wedge C \Leftrightarrow A \wedge (B \wedge C)$.
(ii) $\neg A \vee B \Leftrightarrow A \Rightarrow B$.
(iii) $A \Rightarrow B \Rightarrow C \Leftrightarrow A \wedge B \Rightarrow C$.

2.5 Which of the formulae given in the previous exercise do you think are true no matter the values of A and B?

2.3 Model Theory

We have just defined what a formulae in propositional calculus should look like. We have also seen how such formulae correspond to English sentences, which can be either true or false. The next step is to build a series of mathematical tools to reason about these formulae. For instance, some statement can be shown to be true no matter what—if the weather report were to predict "Tomorrow, either it will rain or it will not," we can immediately conclude that the forecast is correct. We may also want to use the mathematical tools we will develop to show that two sentences are equivalent ("It is cold and it is raining" is intuitively equivalent to "It is raining and it is cold") or that one sentence follows from another ("A lion escaped from the zoo" follows from "No tiger escaped, and either a lion or a tiger, or both, escaped from the zoo"). Although these examples are short and easy to understand, in practice, when modelling complex systems or statements, one typically ends up with huge formulae to reason about. For example, if we were to translate a small electronic chip into a propositional formula, the result would have thousands of basic propositions and thousands of pages of formulae describing the behaviour of the chip.

2.3.1 Truth Tables: Propositional Operators

Since basic propositions can only be either true or false, we can list all possible values and use rules to generate the value of a compound formula. This approach would not work in another field of mathematics, such as numbers, where the number of values a basic variable can take ranges over an infinite collection.

2.3.1.1 Negation

Consider negation. We would like to draw up a table which precisely describes the value of $\neg P$ for any possible value of proposition P. The following table should satisfy our needs:

P	$\neg P$
true	*false*
false	*true*

The double vertical line indicates that what comes to the right is the answer. Before the double line, we list all possible values of the propositional variables. In this case, since we have one variable, we have two possibilities. Truth tables used to define the meaning of the propositional operators are *basic truth tables.* They are to be considered different from the truth tables of compound formulae which are *derived* from basic truth tables. We call such tables *derived truth tables*.

Example Using the basic truth table for negation, we can now draw up the derived truth table for the compound formula $\neg(\neg P)$:

P	$\neg P$	$\neg(\neg P)$
true	*false*	*true*
false	*true*	*false*

First of all, note that we have now used two double lines to separate the basic values from intermediate values we need to calculate the final result. To build the table we would start off with the left column filled (or the columns before the first double line when we have more than one variable), then proceed to use the basic truth tables to fill in the blanks, column by column. For example, to fill in the first entry of $\neg P$ column, we note that P is *true* (from the first column). Looking that up in the basic truth table, we find that the entry under $\neg P$ should be *false*. When we reach the last entry of the $\neg(\neg P)$, we note that $\neg P$ has the value *true* (from the second column), and thus the basic truth table decrees that the entry under $\neg(\neg P)$ will be *false*. $\diamond$

Exercises

2.6 Construct the truth table for $\neg(\neg(\neg P))$.

2.7 We noted that, since we have one variable, we will have two entries in the table. How many entries would we have with two variables? What about three? Can you generalise for n variables?

2.3.1.2 Conjunction

Let us now turn our attention to conjunction. Constructing the basic truth table for $P \wedge Q$ is straightforward:

P	Q	$P \wedge Q$
true	*true*	*true*
true	*false*	*false*
false	*true*	*false*
false	*false*	*false*

We can now draw a number of more interesting derived truth tables.

Example Let us start by drawing the truth table for $\neg Q \wedge P$. First we observe that, using the precedence rules, we can add brackets to get $(\neg Q) \wedge P$. We will have two columns before the first double line (one for P, one for Q). The last column will obviously have $\neg Q \wedge P$. Finally, we need to add a column $\neg Q$ (the only subformula we need).

P	Q	$\neg Q$	$\neg Q \wedge P$
true	*true*	*false*	*false*
true	*false*	*true*	*true*
false	*true*	*false*	*false*
false	*false*	*true*	*false*

Consider filling one of the entries. The last entry in the rightmost column is to be filled by the value of $\neg Q \wedge P$ when we know (from the previous columns) that the value of $\neg Q$ is true and that of P is false. Consulting the second row of the basic truth table for conjunction we know that this last entry has to take the value *false*. ◇

Example Let us try another formula $P \wedge \neg(Q \wedge \neg R)$. Since we now have three variables, P, Q and R, we will have eight rows in the truth table. What about columns for intermediate values? Well, we need $\neg(Q \wedge \neg R)$ to calculate the value of the whole expression. To calculate this, we need to calculate the value of $Q \wedge \neg R$, for which we need the value of $\neg R$.

P	Q	R	$\neg R$	$Q \wedge \neg R$	$\neg(Q \wedge \neg R)$	$P \wedge \neg(Q \wedge \neg R)$
true	*true*	*true*	*false*	*false*	*true*	*true*
true	*true*	*false*	*true*	*true*	*false*	*false*
true	*false*	*true*	*false*	*false*	*true*	*true*
true	*false*	*false*	*true*	*false*	*true*	*true*
false	*true*	*true*	*false*	*false*	*true*	*false*
false	*true*	*false*	*true*	*true*	*false*	*false*
false	*false*	*true*	*false*	*false*	*true*	*false*
false	*false*	*false*	*true*	*false*	*true*	*false*

◇

Exercises

2.8 Draw the truth tables for the following formulae:

(i) $\neg(\neg P \wedge \neg Q)$
(ii) $P \wedge Q \wedge R$ (Note: use associativity)
(iii) $Q \wedge P$

2.3.1.3 Disjunction

By now the pattern of how to draw up truth tables should be quite clear. It is sufficient to be given the basic truth table of a new operator, and we can derive truth tables of formulae using that operator. Let us look at the basic truth table of disjunction.

P	Q	$P \vee Q$
true	*true*	*true*
true	*false*	*true*
false	*true*	*true*
false	*false*	*false*

Note that, as we have already discussed, the or operator yields true, even if both its operands are true. You may argue that it would make more sense to have an exclusive-or meaning attached to the operator (where exactly one of the operands is true for the result to be true), but this is just a matter of choice of notation. Later on in this chapter, one of the exercises is to define such an operator and derive its properties.

Example Let us draw the truth table for $P \vee (Q \vee R)$:

P	Q	R	$Q \vee R$	$P \vee (Q \vee R)$
true	*true*	*true*	*true*	*true*
true	*true*	*false*	*true*	*true*
true	*false*	*true*	*true*	*true*
true	*false*	*false*	*false*	*true*
false	*true*	*true*	*true*	*true*
false	*true*	*false*	*true*	*true*
false	*false*	*true*	*true*	*true*
false	*false*	*false*	*false*	*false*

◇

Exercises

2.9 Draw the truth tables of the following formulae:

(i) $\neg(\neg P \vee \neg Q)$
(ii) $(P \vee Q) \vee R$

(iii) $P \vee P$
(iv) $\neg P \vee Q$

2.3.1.4 Bi-implication

Recall that the bi-implication, or if-and-only-if, operator results in true if the two operands have the same value.

P	Q	$P \Leftrightarrow Q$
true	*true*	*true*
true	*false*	*false*
false	*true*	*false*
false	*false*	*true*

Example Let us draw the truth table of $P \wedge Q \Leftrightarrow Q \wedge P$:

P	Q	$P \wedge Q$	$Q \wedge P$	$P \wedge Q \Leftrightarrow Q \wedge P$
true	*true*	*true*	*true*	*true*
true	*false*	*false*	*false*	*true*
false	*true*	*false*	*false*	*true*
false	*false*	*false*	*false*	*true*

◇

Example Here is another example: $P \Leftrightarrow P \vee Q$

P	Q	$P \vee Q$	$P \Leftrightarrow P \vee Q$
true	*true*	*true*	*true*
true	*false*	*true*	*true*
false	*true*	*true*	*false*
false	*false*	*false*	*true*

◇

Exercises
2.10 Draw the truth tables of the following formulae:

(i) $P \Leftrightarrow (Q \Leftrightarrow R)$
(ii) $(P \vee Q) \wedge (P \Leftrightarrow Q)$
(iii) $P \Leftrightarrow P$

2.3.1.5 Implication

Finally, let us turn our attention to implication. Recall that, if P is true, then Q must be true for $P \Rightarrow Q$ to be true. What about when P is false? We chose to take the view that when a politician says "If the economy improves, then we will build a new

hospital," the statement is always true if the economy fails to improve. Therefore, if P is false, $P \Rightarrow Q$ is true, no matter the value of Q.[2]

P	Q	$P \Rightarrow Q$
true	*true*	*true*
true	*false*	*false*
false	*true*	*true*
false	*false*	*true*

Example Let us look at one final example to illustrate implication: $(P \Rightarrow Q) \Rightarrow R$:

P	Q	R	$P \Rightarrow Q$	$(P \Rightarrow Q) \Rightarrow R$
true	*true*	*true*	*true*	*true*
true	*true*	*false*	*true*	*false*
true	*false*	*true*	*false*	*true*
true	*false*	*false*	*false*	*true*
false	*true*	*true*	*true*	*true*
false	*true*	*false*	*true*	*false*
false	*false*	*true*	*true*	*true*
false	*false*	*false*	*true*	*false*

◇

Exercises

2.11 Draw the truth tables of the following formulae:

(i) $(P \Leftrightarrow Q) \Rightarrow (P \Rightarrow Q)$
(ii) $P \vee Q \Rightarrow P \wedge Q$
(iii) $P \Rightarrow P$
(iv) $P \Rightarrow (Q \Rightarrow R)$
(v) $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$

2.12 Four cards lie on a table, each of which, we are told, has a number written on one side and a letter on the other. The visible card faces show A, B, 4 and 8. We are asked to check whether the implication 'if a card has A on one side, then it has a 4 on the other' is true. Which cards would you have to turn over to check the truth of the statement?

2.3.1.6 Applications

Let us look at the examples we gave before at the end of Sect. 2.2, and draw truth tables for the formulae we came up with.

[2]Of course, if you are not happy with this interpretation, as we said in the case of disjunction, you are free to define another operator with your preferred semantics. However, if you choose the meaning of the new operator to be such that the politician's statement would be interpreted as a lie when a hospital is built despite the fact that the economy has not improved, you will discover that your operator is nothing other than bi-implication. Try it out to check.

Fixing the car: Recall that we set C to be the proposition '*there is a problem with the carburetor*', D the proposition '*there is a problem with the distributor*', and S '*the sparking plugs need changing*.'
The statements about the way the problems are related were expressed as follows:

- It is either the carburetor or the distributor: $C \vee D$.
- Whenever there is a problem with the carburetor, the sparking plugs need changing: $C \Rightarrow S$.
- The distributor and the sparking plugs are closely connected. If one finds problems in one, there are invariably problems with the other: $D \Leftrightarrow S$.

Now the whole statement was that all three expressions were true: $(C \vee D) \wedge (C \Rightarrow S) \wedge (D \Leftrightarrow S)$. Let us draw the truth table:

C	D	S	$C \vee D$	$C \Rightarrow S$	$D \Leftrightarrow S$	$(C \vee D) \wedge (C \Rightarrow D)$	$(C \vee D) \wedge (C \Rightarrow S) \wedge (D \Leftrightarrow S)$
true	*true*	*true*	*true*	*true*	*true*	*true*	*true*
true	*true*	*false*	*true*	*false*	*false*	*false*	*false*
true	*false*	*true*	*true*	*true*	*false*	*true*	*false*
true	*false*	*false*	*true*	*false*	*true*	*false*	*false*
false	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*true*	*true*	*false*	*true*	*false*
false	*false*	*true*	*false*	*true*	*false*	*false*	*false*
false	*false*	*false*	*false*	*true*	*true*	*false*	*false*

Now, we know that the statements are true, so the situation must refer to the first or fifth line, where we note in both cases, D and S are true. Therefore, there is a problem with the distributor, and the sparking plugs need changing.

Eating habits: We set S to be the proposition for '*today, Jack and Jill will eat soup*', H to be '*Jack is hungry*', and C to be '*it is cold*'.

- Jack's constraint: $(C \wedge \neg H) \Rightarrow S$.
- Jill's constraint: $S \Rightarrow \neg C$.

Again, let us draw the truth table of $((C \wedge \neg H) \Rightarrow S) \wedge (S \Rightarrow \neg C)$.

C	H	S	$(C \wedge \neg H) \Rightarrow S$	$S \Rightarrow \neg C$	$((C \wedge \neg H) \Rightarrow S) \wedge (S \Rightarrow \neg C)$
true	*true*	*true*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*
true	*false*	*true*	*true*	*false*	*false*
true	*false*	*false*	*false*	*true*	*false*
false	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*true*	*true*	*true*
false	*false*	*true*	*true*	*true*	*true*
false	*false*	*false*	*true*	*true*	*true*

It is worth noticing that, if it is cold and Jack is not very hungry, then there is no way to satisfy both constraints (rows 3 and 4). Unless Jack and Jill live in a country with a tropical climate or Jack never feels very hungry when it is cold, they will have to cook separate dishes.

Using a similar approach, we could draw the truth table for the circuit designed by the engineer, and the properties which the engineer expected it to satisfy. The only problem is that the circuit had seven propositional variables—three inputs, one output and three additional wires used to connect the logic gates. Since with every variable the number of entries in a truth table would double, the number of entries to describe all possible behaviours of the circuit would require $2^7 = 128$ entries.

2.3.2 Properties of Propositional Sentences

What we have done up to now is to be able to define the meaning of a sentence in terms of a table which we can use to check the value of a particular expression. We have made use of the truth tables informally to reason about situations expressed as propositional logic sentences. Now we need to define a number of concepts, using which we will be able to formulate better our conclusions.

2.3.2.1 Tautologies

Definition 2.6 *We say that a well-formed formula is a* tautology *if all the entries of the rightmost column in its truth table are true.* ■

Informally, a tautology is something which is true no matter what the value of its constituent subformulae.

Example $P \vee \neg P$ is called the law of the excluded middle: there is no middle way; either something is true, or it is false.[3] We can show that this law holds by showing that it is a tautology.

P	$\neg P$	$P \vee \neg P$
true	*false*	*true*
false	*true*	*true*

Since the entries in the last column are all true, the statement is a tautology. ◇

[3] Charles Dodgson, writing under the pseudonym Lewis Carroll, mentions this in *Alice Through the Looking Glass*:

> *'Everybody that hears me sing it–either it brings the tears into their eyes, or else–'*
> *'Or else what?' said Alice, for the Knight had made a sudden pause.*
> *'Or else it doesn't, you know.'*

Example We can now also formalise what we were saying in the previous section. Consider the example with the car fixing problem. What we informally concluded from the given statements was that the distributor and the sparking plugs need changing, or to write it formally $D \wedge S$. Recall that the given information was expressed as: $(C \vee D) \wedge (C \Rightarrow D) \wedge (D \Leftrightarrow S)$. Our claim is that, if the given data is correct, then the distributor and the sparking plugs need changing:

$$((C \vee D) \wedge (C \Rightarrow D) \wedge (D \Leftrightarrow S)) \Rightarrow (D \wedge S).$$

It suffices to show that the above formula is a tautology, which we do by drawing up its truth table (we leave out some intermediate columns already given elsewhere):

C	D	S	$(C \vee D) \wedge (C \Rightarrow D) \wedge (D \Leftrightarrow S)$	$D \wedge S$	$((C \vee D) \wedge (C \Rightarrow D) \wedge (D \Leftrightarrow S)) \Rightarrow (D \wedge S)$
true	*true*	*true*	*true*	*true*	*true*
true	*true*	*false*	*false*	*false*	*true*
true	*false*	*true*	*false*	*false*	*true*
true	*false*	*false*	*false*	*false*	*true*
false	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*
false	*false*	*true*	*false*	*false*	*true*
false	*false*	*false*	*false*	*false*	*true*

So the statement that the given information implies that we need to fix the distributor and change the plugs is true. ◇

Exercises

2.13 Show that $(P \Rightarrow Q) \Leftrightarrow (\neg P \vee Q)$ is a tautology.

2.14 In Exercise 2.2, I gave some details about Sally and Jane. I love at least one of them, and if I love Sally, then I also love Jane. Show (using a tautology) that these constraints guarantee that I love Jane.

2.15 Using a tautology, show that two negation gates connected in sequence output the same value given as input.

2.16 Jack and Jill's dilemma can be expressed as 'If it is cold and Jack is not hungry, then at least one of them will be unhappy.' Express this formally, and show it to be a tautology.

2.3.2.2 Contradictions

A contradiction is the opposite of a tautology—an expression which is not true for any choice of values of the variables.

Definition 2.7 *We say that a well-formed formula E is a* contradiction *if all the entries in the rightmost column of its truth table are false*. ■

Example Intuitively, we know that P and $\neg P$ are opposite of each other. We can express this by showing that $P \wedge \neg P$ is a contradiction.

P	$\neg P$	$P \wedge \neg P$
true	*false*	*false*
false	*true*	*false*

Since the entries in the last column are all false, $P \wedge \neg P$ is a contradiction. ◇

Example Using a tautology, we already saw that, when Jack is hungry and it is cold, Jack and Jill will be arguing about what to eat. We can also reach this conclusion by showing that, if $H \wedge C$ (Jack is hungry and it is cold) and both Jack and Jill's constraints hold, we end up with a contradiction.

C	H	S	$\neg H$	$\neg H \wedge C$	$((C \wedge \neg H) \Rightarrow S) \wedge (S \Rightarrow \neg C)$	$(H \wedge C) \wedge ((C \wedge \neg H) \Rightarrow S) \wedge (S \Rightarrow \neg C)$
true	*true*	*true*	*false*	*false*	*false*	*false*
true	*true*	*false*	*false*	*false*	*true*	*false*
true	*false*	*true*	*true*	*true*	*false*	*false*
true	*false*	*false*	*true*	*true*	*false*	*false*
false	*true*	*true*	*false*	*false*	*true*	*false*
false	*true*	*false*	*false*	*false*	*true*	*false*
false	*false*	*true*	*false*	*true*	*true*	*false*
false	*false*	*false*	*false*	*true*	*true*	*false*

This means that there is no way to satisfy their constraints under these circumstances. ◇

Exercises

2.17 Show that $(P \Rightarrow P) \Rightarrow \neg(P \Rightarrow P)$ is a contradiction.

2.18 Show that $(P \Rightarrow \neg P) \Leftrightarrow P$ is a contradiction.

2.19 Jill lies 'If Jack eats fish then he eats meat.' Jack retorts that he doesn't eat meat. Can Jack be telling the truth?

2.20 If a formula E is a contradiction, can you construct a formula in terms of E which is a tautology? What about another contradiction?

2.3.2.3 Equivalence

Tautologies and contradictions tell us a lot about formulae. However, most interesting formulae are neither one nor the other. We can relate different formulae even if they are neither tautologies nor contradictions.

Definition 2.8 *Two propositional formulae E and F are said to be semantically equivalent if the values in their truth tables match. We write this as $E \models\!\models F$.*

If they do not match on at least one row, we say that E and F are not semantically equivalent, and we write $E \not\models\!\models F$. ■

It is easy to confuse semantic equivalence with bi-implication. It is true that saying that $E =\!\models F$ is the same as saying that $E \Leftrightarrow F$ is a tautology. However, note that the symbol $=\!\models$ is not a symbol which can appear in a well-formed propositional formula. When we talk about tautologies, semantic equivalences, etc., we are talking *about* propositions, and we are not writing other propositions.

This may take some time to sink in, but it is an important mathematical concept. For the moment, remember to look for symbols in the formula to decide whether it is a propositional formula or a statement about a propositional formula.

Example Semantic equivalence, in a certain sense, is equality between expressions, once we evaluate their symbols. For example, $P \Leftrightarrow Q$ is semantically equivalent to $P \Rightarrow Q$ and $Q \Rightarrow P$ (even the symbol used suggests this). If this is so, we should be able to show that $P \Leftrightarrow Q$ is semantically equivalent to $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$:

P	Q	$P \Rightarrow Q$	$Q \Rightarrow P$	$P \Leftrightarrow Q$	$(P \Rightarrow Q) \wedge (Q \Rightarrow P)$
true	*true*	*true*	*true*	*true*	*true*
true	*false*	*false*	*true*	*false*	*false*
false	*true*	*true*	*false*	*false*	*false*
false	*false*	*true*	*true*	*true*	*true*

Since all the entries of the two expressions match, we can conclude that they are semantically equivalent: $P \Leftrightarrow Q =\!\models (P \Rightarrow Q) \wedge (Q \Rightarrow P)$. $\diamond$

Such equivalencies allow us to replace one expression with another (semantically equivalent one) within an expression to simplify it. We will talk more about this later, but for the moment it suffices to note that in this way we can reason about formulae by replacing semantically equivalent ones.

Example We will now show a number of properties of conjunction by constructing truth tables:

1. $P \wedge P =\!\models P$

P	$P \wedge P$	P
true	*true*	*true*
false	*false*	*false*

2. $P \wedge Q =\!\models Q \wedge P$

P	Q	$P \wedge Q$	$Q \wedge P$
true	*true*	*true*	*true*
true	*false*	*false*	*false*
false	*true*	*false*	*false*
false	*false*	*false*	*false*

3. $P \wedge (Q \wedge R) \models\!\models (P \wedge Q) \wedge R$

P	Q	R	$P \wedge Q$	$Q \wedge R$	$P \wedge (Q \wedge R)$	$(P \wedge Q) \wedge R$
true	*true*	*true*	*true*	*true*	*true*	*true*
true	*true*	*false*	*true*	*false*	*false*	*false*
true	*false*	*true*	*false*	*false*	*false*	*false*
true	*false*	*false*	*false*	*false*	*false*	*false*
false	*true*	*true*	*false*	*true*	*false*	*false*
false	*true*	*false*	*false*	*false*	*false*	*false*
false	*false*	*true*	*false*	*false*	*false*	*false*
false	*false*	*false*	*false*	*false*	*false*	*false*

In all cases, we can deduce the semantic equivalence by inspecting the truth tables. ◇

This last example is an important one. When we define a new operator we would like to know whether these properties hold.

Definition 2.9 *A binary operator* $\oplus$ *is said to be* idempotent *if* $x \oplus x = x$, commutative *or* symmetric *if* $x \oplus y = y \oplus x$, *and* associative *if* $x \oplus (y \oplus z) = (x \oplus y) \oplus z$. ■

Over propositional formulae, we can use semantic equivalence as our notion of equality.[4] In the previous example, we have thus shown that conjunction is idempotent, commutative and associative. Therefore, an unbracketed conjunction such as $P \wedge Q \wedge R$ gives the same resulting truth table, whether we take conjunction to be left or right associative.

Exercises

2.21 Show that $\mathit{false} \wedge P \models\!\models \mathit{false}$, while $\mathit{true} \wedge P \models\!\models P$. Can you find similar equivalences for $\vee$?
2.22 Show that $\neg P \vee Q \models\!\models P \Rightarrow Q$.
2.23 Show that $\neg(\neg P \wedge \neg Q) \models\!\models P \vee Q$.
2.24 Show that $\neg(\neg P \vee \neg Q) \models\!\models P \wedge Q$.
2.25 Show that $\Rightarrow$ is not commutative. Is it associative?
2.26 Show that $\vee$ is idempotent, commutative and associative.
2.27 $P \not\models\!\not\models Q$ is not equivalent to $P \models\!\models \neg Q$. Give an example of two propositions P and Q to show that these two statements are not the same.

[4] Equalities should obey certain laws, which will be mentioned later on. For the moment, we can assume that semantic entailment satisfies these rules.

2.3.2.4 Semantic Entailment

While it is useful to have a notion of equivalence between propositional statements, in a logical argument the statements should follow from one another but not necessarily be equivalent. For instance, if at a certain point in a logical argument we know that, whenever at least one of two switches is on, then the bulb will be lit, and we also know that the first switch is on, we conclude that the bulb is on. Note that the conclusion is not equivalent to the knowledge we had before. However, for the argument to be valid, if the first statement is true, then so should the second statement. We call this notion *semantic entailment*.

Definition 2.10 *A well-formed propositional formula E is said to semantically entail formula F if every entry in the truth table of E which is true is also true in the truth table of F. We write this as $E \models F$. If E does not semantically entail F, we write $E \not\models F$.* ■

It is easy to see that $E \models\!\models F$ is the same as $E \models F$ and $F \models E$, hence the symbol. As noted before with semantic equivalence, it is easy to confuse semantic entailment with implication. In fact, saying that $E \models F$ is the same as saying that $E \Rightarrow F$ is a tautology.

Example We are told that a particular file is not owned by either Pam or Quentin. From this, it should follow that the file is not owned by Pam. The general rule from which this follows is: $\neg(P \vee Q) \models \neg P$. To check whether or not it holds in general, we draw the truth table for the two expressions.

P	Q	$P \vee Q$	$\neg(P \vee Q)$	$\neg P$
true	*true*	*true*	*false*	*false*
true	*false*	*true*	*false*	*false*
false	*true*	*true*	*false*	*true*
false	*false*	*false*	*true*	*true*

Since in every instance where $\neg(P \vee Q)$ is true, so is $\neg P$, it follows that $\neg(P \vee Q) \models \neg P$. ◇

Example Semantic entailment gives us another way of expressing our conclusions about the car example we gave earlier. To say that the car conditions we were given guarantee that the sparking plugs and distributor are to blame, we can simply write:

$$(C \vee D) \wedge (C \Rightarrow D) \wedge (D \Leftrightarrow S) \models D \wedge S$$

What about the carburetor? We can show that $(C \vee D) \wedge (C \Rightarrow D) \wedge (D \Leftrightarrow S) \not\models C$ but also $(C \vee D) \wedge (C \Rightarrow D) \wedge (D \Leftrightarrow S) \not\models \neg C$. Therefore, we cannot conclude anything about the carburetor from the statements we know.

In fact, we can show that $(C \vee D) \wedge (C \Rightarrow D) \wedge (D \Leftrightarrow S) \models\!\models D \wedge S$, meaning that the three statements can be replaced by one saying that the sparking plugs and

the distributor need changing. This statement says nothing about the carburetor—it may be working fine, or not working at all. ◇

Example If we were to draw the truth table for the multiplexer circuit, we would then want to check whether the circuit behaviour guarantees the properties:

$$\left(\begin{array}{ll} & (out \Leftrightarrow m_1 \vee m_2) \\ \wedge & (m_0 \Leftrightarrow \neg sel) \\ \wedge & (m_1 \Leftrightarrow sel \wedge h) \\ \wedge & (m_2 \Leftrightarrow m_0 \wedge l) \end{array}\right) \models \left(\begin{array}{ll} & (sel \Rightarrow (out \Leftrightarrow h)) \\ \wedge & (\neg sel \Rightarrow (out \Leftrightarrow h)) \\ \wedge & ((l \Leftrightarrow h) \Rightarrow (out \Leftrightarrow h)) \end{array}\right)$$

Unfortunately, as we have already seen, this would require drawing a truth table with 128 entries to confirm. ◇

Recall that a tautology is a proposition which is true no matter what the value of the propositional variables in the expression. Therefore, a tautology E should satisfy $true \models E$—since *true* always holds, all the entries of E should be true. We sometimes abbreviate this to $\models E$.

Exercises

2.28 Show that the following semantic entailments hold:

(i) $(P \Rightarrow Q) \wedge P \models Q$
(ii) $P \wedge Q \models P$
(iii) $(P \vee Q) \wedge \neg P \models Q$

2.29 Write a computer program to check whether the multiplexer circuit is correct, by generating and looking at all 128 possibilities.

2.30 Show that $\models P \vee (\neg P \wedge \neg Q) \vee Q$.

2.31 $P \not\models Q$ is not equivalent to $P \models \neg Q$. Give an example to show that these two statements are not the same.

2.3.3 Conclusion

This concludes this section about model theory for propositional logic. Reasoning about propositions using truth tables is rather straightforward, if laborious at times. However, these notions give us means of reasoning about statements in a precise and unambiguous manner.

Exercises

2.32 The exclusive-or operator (usually referred to simply as *xor*) is a binary operator which is true if exactly one of its operands is true. Thus, "Paul loves men xor Paul loves women" is true if Paul is purely heterosexual or homosexual. It would be false if Paul is asexual or bisexual. The symbol we will use for xor is $\bowtie$.

(i) Draw the basic truth table for the $\bowtie$ operator.
(ii) Draw the derived truth table for $(P \bowtie Q) \vee Q$.
(iii) Show that $P \bowtie P$ is a contradiction, while $P \bowtie \neg P$ is a tautology.
(iv) Show that $\bowtie$ is commutative and associative.
(v) Show that $P \bowtie Q \models P \vee Q$.
(vi) Give a well-formed formula using P, Q, $\vee$, $\wedge$ and $\neg$ which is equivalent to $P \bowtie Q$. Show that they are semantically equivalent.

2.4 Proof Theory

Propositional model theory is just one way of reasoning about propositions. In model theory, we explicitly build every possible model of the system (using truth tables or similar techniques) and painstakingly verify our claims for every case. We have already noted that the number of entries in a truth table with n variables is 2^n. This made the manual checking of the multiplexer example with 128 entries in the truth table impractical. But it rapidly gets worse—with 10 variables, we will have over 1,000 rows to fill and with 20 variables it is already over 1,000,000. Even using a computer program to check all rows quickly becomes unfeasible as the number of variables increases. The sheer magnitude of these tables means that maybe there are more effective techniques, which at least in certain cases, reduce the work required to show that a proposition is a tautology or equivalent to another. Furthermore, when we move on to numbers and mathematical systems which have an infinite number of possible values, performing an exhaustive, case-by-case analysis is impossible. The definitions used in model theory gave us the meaning of the operators—we will now look at how we can build a proof theory to reason about propositions.

2.4.1 What Is a Proof?

The concept of proof is possibly the most fundamental notion in mathematics. Everyone is familiar with the concept of a proof from an everyday viewpoint—whether it is from lawsuits or normal conversational argumentation. In mathematics, a proof is a way of justifying a mathematical statement based on a number of basic truths, using acceptable rules to move from one true statement to the next.

We thus have two important elements in any mathematical system in which we want to write proofs:

Basic truths: Basic truths, or *axioms* as they are usually called, are statements in the mathematical system under analysis which we accept to be true without justification. For example, in the case of propositional logic, we may choose to take $P \vee \neg P$ to be one of the basic truths, meaning that $P \vee \neg P$ needs no justification to be proved. When defining the odd numbers, we can have an axiom that 1 is an odd number. One may ask why, if we are using such an approach, do we not define

all true statements as axioms. In the first place, we would like our axioms to be *basic* truths. The proposition $((P \Rightarrow \neg Q) \wedge Q) \Rightarrow \neg P$ is true; however, not everyone would readily accept it as a basic truth which needs no justification to believe. Secondly, we want to be able to list all basic truths in a finite manner. Unfortunately, there exist an infinite number of true statements, and we limit ourselves to start with only a finite collection of basic truths to avoid unnecessary complexity.

Acceptable rules: *Rules of inference* tell us how statements follow from one another. Just as in the case of axioms, we will accept their validity as is. In the case of propositional logic, for example, we may use a rule saying that knowledge of proposition P follows directly from knowledge of proposition $P \wedge Q$. If we were talking about odd numbers, we would say that, from the fact that n is an even number, we can conclude that $n + 2$ is also an odd number. Again, here one could adopt complex rules which allow us to do everything in one step. However, this defeats the purpose of the exercise. We will define a simple format of rules which are allowed, and all rules are to be expressed in terms of this format. Furthermore, it is considered good mathematics to have a small number of rules whose truth is self-evident.

The notion of *self-evident* is not very precise. A discussion as to what should be self-evident will ensue later on in this chapter.

We call a statement with an accompanying proof justifying it a *theorem.* To avoid long-winded proofs, we will also accept steps in a proof which are the application of a theorem proved earlier.

Definition 2.11 *A* proof *that a statement Y follows from statement X in a mathematical system S is a finite sequence of statements finishing with Y, such that every line of the proof is either* (i) *statement X (called the hypothesis), or* (ii) *an axiom of S, or* (iii) *follows from previous lines in the proof and a rule of inference of S, or* (iv) *follows from previous lines in the proof and a theorem already proved in S.* ■

2.4.2 Example Axioms and Rules of Inference

Let us take a look at the notation we will use before we start giving the axioms and rules of inference of propositional logic. Let us take a look at a system in which we can prove that a number is odd. Instead of using the normal representation of numbers, we will use a number of blobs to stand for numbers—to represent a number n, we will write n blobs in line. For example, 3 would be written as •••, 1 as • and 7 as •••••••. Furthermore, we will use the notation $\mathcal{O}(n)$ to mean that n is an odd number.

To represent a logic in which we can prove numbers to be odd, it suffices to use one axiom and one rule of inference:

Axiom: The only axiom of the system will assert that 1 is an odd number. We write this as follows:

$$\frac{}{\mathcal{O}(\bullet)}$$

Sometimes this is also written (more concisely) as $\vdash \mathcal{O}(\bullet)$. Since we need to refer to this axiom in our proofs, we will name this axiom *1-odd.*

Rule of inference: The rule of inference we will now need states that, if we know that a string of n blobs is an odd number, then adding another two blobs to n will still give us an odd number:

$$\frac{\mathcal{O}(n)}{\mathcal{O}(\bullet\bullet n)}$$

The rule says that, if the top formula can be proved, then the formula below follows directly (in one step). We will name this rule *odd-$n+2$*. Note that the formulae in the rule use a variable to stand for any number of blobs (of course, the constraint allows only an odd number of blobs, but we know that only because we have an intuitive understanding of $\mathcal{O}$). The important thing is that n is the same collection of blobs below and above the inference line. We will also use this in our rules for propositional logic. In general, rules can have more than one formula above the line. These formulae are called the antecedents of the rule of inference.

Let us now proceed to prove that 5 is an odd number using these rules. Recall that a formal proof consists of a sequence of lines, each containing (i) a well-formed formula, and (ii) justification in terms of either an axiom name, the name of a rule of inference and the lines (earlier in the proof) where we can find the antecedents, or the name of a theorem we proved earlier. Here is what the proof would look like:

1.	$\mathcal{O}(\bullet)$	(axiom *1-odd*)
2.	$\mathcal{O}(\bullet\bullet\bullet)$	(rule *odd-$n+2$* on line 1)
3.	$\mathcal{O}(\bullet\bullet\bullet\bullet\bullet)$	(rule *odd-$n+2$* on line 2)

Notice the three columns: the first is just the line number; the second contains a well-formed formula; and the third contains the axiom used, or rule applied and line numbers on which it is applied.

We have therefore just proved that $\mathcal{O}(\bullet\bullet\bullet\bullet\bullet)$ based on just the axioms and rules of inference. This is called a *theorem*.

Sometimes, proofs are of statements of the form "From formulae $E_1, E_2, \ldots, E_n$, it follows that F". In this case, the Es are called the hypotheses and can be used in the proof. Here is a simple example: From $\mathcal{O}(\bullet\bullet)$, it follows that $\mathcal{O}(\bullet\bullet\bullet\bullet\bullet\bullet)$.

1.	$\mathcal{O}(\bullet\bullet)$	(hypothesis)
2.	$\mathcal{O}(\bullet\bullet\bullet\bullet)$	(rule *odd-$n+2$* on line 1)
3.	$\mathcal{O}(\bullet\bullet\bullet\bullet\bullet\bullet)$	(rule *odd-$n+2$* on line 2)

"But wait," you may argue, "we have just proved that 6 is an odd number. There is something going wrong!" Not really. We have only proved that if 2 is an odd

number, then we can prove that 6 is also an odd number. Anything follows from a false statement. For instance from $1 = 2$, it should follow that I am the Pope.[5]

Definition 2.12 *If from a number of well-formed formulae $\mathcal{X}$, we can prove a well-formed formula Y, we say that Y* is provable from *X, written as $\mathcal{X} \vdash Y$. $\mathcal{X}$ are called the* hypotheses *and Y the* conclusion. *$\mathcal{X} \vdash Y$ is said to be a theorem.*

If no hypotheses are needed (the proof uses only the axioms and rules of inference) we write $\vdash Y$, and we simply say that Y is a theorem *of the system.*

We say that X and Y are equivalent, written $X \dashv\vdash Y$, if both $X \vdash Y$ and $Y \vdash X$ hold. ■

The above two proofs thus enable us to write that $\vdash \mathcal{O}(\bullet\bullet\bullet\bullet\bullet)$ and that $\mathcal{O}(\bullet\bullet) \vdash \mathcal{O}(\bullet\bullet\bullet\bullet\bullet\bullet)$. Let us now give the rules for propositional logic.

2.4.2.1 Conjunction

Note that we will need rules for all the operators in the logic. Let us start by looking at conjunction. We will need rules to remove conjunctions appearing in the hypotheses, and others to produce conjunction to appear in the conclusion. The rules are rather straightforward. Let us start with the introduction rule:

$$\frac{A,\ B}{A \wedge B}$$

This says that, if we can prove A and also B, then it follows in one additional step that $A \wedge B$. We call this the $\wedge$-*introduction* rule.

What if we have a conjunction which we would like to break into its constituent parts? Well, from $A \wedge B$, it follows both that A and also B. We write this as two separate rules:

$$\frac{A \wedge B}{A} \qquad \frac{A \wedge B}{B}$$

We call these rules $\wedge$-*elimination 1* and $\wedge$-*elimination 2,* respectively. Note that this is not the only way to axiomatise conjunction. Other approaches exist, which you may find in other books. The important thing is that we are identifying one set of axioms and rules of inference and sticking to them as our basic truths.

Example Let us start by giving a simple example of a proof using only these rules of inference. We would like to prove that conjunction is commutative: $P \wedge Q \vdash Q \wedge P$.

[5] Here is the proof: Either I am the Pope, or I am not. If I am, then fine, we have proved what was required. If I am not, then the Pope and I are two persons. But $1 = 2$, and therefore we are one person. Hence, I am the Pope.

The proof is quite straightforward:

1.	$P \wedge Q$	(hypothesis)
2.	P	($\wedge$-elimination 1 on line 1)
3.	Q	($\wedge$-elimination 2 on line 1)
4.	$Q \wedge P$	($\wedge$-introduction on lines 3 and 2)

◇

Example Here is another proof of associativity of conjunction shown in one direction: $P \wedge (Q \wedge R) \vdash (P \wedge Q) \wedge R$.

1.	$P \wedge (Q \wedge R)$	(hypothesis)
2.	P	($\wedge$-elimination 1 on line 1)
3.	$Q \wedge R$	($\wedge$-elimination 2 on line 1)
4.	Q	($\wedge$-elimination 1 on line 3)
5.	R	($\wedge$-elimination 2 on line 3)
6.	$P \wedge Q$	($\wedge$-introduction on lines 2 and 4)
7.	$(P \wedge Q) \wedge R$	($\wedge$-introduction on lines 6 and 5)

◇

One important rule of applying the rules of inference is that the antecedents must match the whole formula in the quoted line. For instance, from the formula $(P \wedge Q) \wedge R$, we cannot directly conclude that $R \wedge R$ using the second rule of $\wedge$-elimination on the subexpression $P \wedge Q$. The known formula is a conjunction of $(P \wedge Q)$ and R, and thus, the elimination rules can only conclude these results. To prove Q requires applying the elimination rules twice. This may seem extremely pedantic, but as we introduce new operators, applying rules of inference on subexpressions can invalidate proofs.

Exercises

2.33 Prove the following:

(i) $(P \wedge Q) \wedge R \vdash P \wedge (Q \wedge R)$
(ii) $P \wedge P \vdash P$
(iii) $P \vdash P \wedge P$

2.4.2.2 Implication

Let us move on to the next operator: implication. To eliminate an implication $A \Rightarrow B$, we first have to prove the left-hand side of the implication A, from which we can then conclude the right-hand side B:

$$\frac{A \Rightarrow B,\ A}{B}$$

We will call this rule $\Rightarrow$*-elimination*, although the classical name which you will find in most logic books is *modus ponens*.[6]

Example A simple rule involving implication: $(A \wedge (A \Rightarrow B)) \wedge (B \Rightarrow C) \vdash C$

1.	$(A \wedge (A \Rightarrow B)) \wedge (B \Rightarrow C)$	(hypothesis)
2.	$A \wedge (A \Rightarrow B)$	($\wedge$-elimination 1 on line 1)
3.	A	($\wedge$-elimination 1 on line 2)
4.	$A \Rightarrow B$	($\wedge$-elimination 2 on line 2)
5.	$B \Rightarrow C$	($\wedge$-elimination 2 on line 1)
6.	B	($\Rightarrow$-elimination on lines 4 and 3)
7.	C	($\Rightarrow$-elimination on lines 5 and 6)

$\diamond$

What about introducing an implication? What knowledge would allow us to conclude that $A \Rightarrow B$? Looking at the meaning of implication—if A holds then B must also hold—provides a way of approaching implication introduction. If we manage to write a proof which has A as an additional hypothesis, and which concludes proposition B, then we can cite the *whole proof* as evidence of $A \Rightarrow B$.

Let us look at the introduction rule $\Rightarrow$*-introduction* for implication:

$$\frac{\mathcal{S},\ A \vdash B}{\mathcal{S} \vdash A \Rightarrow B}$$

This rule is more complex than the ones we have seen, so let us look at it step by step. Unlike the conjunction rules we have seen, the statement that needs to be proved is not a single well-formed formula, but a theorem. Ignoring the $\mathcal{S}$ for the moment, the rule says that, if we write a formal proof to show that $A \vdash B$, then we can conclude that $A \Rightarrow B$. The addition of $\mathcal{S}$ is to be able to enrich the rule in the following manner: if we have already proved a number of statements $\mathcal{S}$, from which we would like to conclude that $A \Rightarrow B$, then we can also use statements from $\mathcal{S}$ to prove B from A—in other words, we require a proof of $\mathcal{S},\ A \vdash B$ to be able to conclude $A \Rightarrow B$ when we already know $\mathcal{S}$.

How do we write such a proof? Well, from the rule it seems like we would have to write a separate small proof, and then apply it in the main proof to conclude $A \Rightarrow B$. That would not be very practical, so we incorporate subproofs into our proofs directly. The proof will be written as part of the main proof but marked with

[6] *Modus ponens* is Latin for 'the method that affirms'.

a separator to show it is a subproof and should thus be treated as such. This is the structure of such a proof:

	⋮	
8.	A	(subhypothesis)
	⋮	
12.	B	(?)
13.	$A \Rightarrow B$	($\Rightarrow$-introduction on subproof 8–12)
	⋮	

As you can see, lines 8 to 12 are marked as a separate subproof with the hypothesis (line 8) marked as a subhypothesis to avoid confusion with the hypotheses of the main proof. The propositions to be found on lines 1 to 6 are what we were calling $\mathcal{S}$, and can therefore be used within the subproof in lines 8 to 12. Lines 8 to 12 should be seen as a separate proof and can only be referred to as a whole to conclude that $A \Rightarrow B$ using implication introduction, as we did on line 13. This means that, beyond line 12, we are not allowed to refer to individual lines that appear within the subproof.

Definition 2.13 *A* subproof *is a proof appearing within another proof or subproof. It has a scope ranging over a number of lines, and can be* discharged *using implication introduction. No individual lines within the scope of a subproof can be used outside its scope.* ■

Example The best way to understand this business about subproofs is to see a couple of examples. Let us start by proving that implication is transitive: $(P \Rightarrow Q) \wedge (Q \Rightarrow R) \vdash P \Rightarrow R$.

The best way to understand a proof is to build it from scratch, rather than read the full proof from top to bottom. We will see how this proof is constructed. We start by copying the hypothesis and conclusion of the proof, and apply conjunction elimination to break down the hypothesis into its constituent parts:

1.	$(P \Rightarrow Q) \wedge (Q \Rightarrow R)$	(hypothesis)
2.	$P \Rightarrow Q$	($\wedge$-elimination 1 on line 1)
3.	$Q \Rightarrow R$	($\wedge$-elimination 2 on line 1)
	⋮	
7.	$P \Rightarrow R$	(?)

Since we need to conclude the implication $P \Rightarrow R$, we need to open a subproof with subhypothesis P and concluding with R:

1.	$(P \Rightarrow Q) \wedge (Q \Rightarrow R)$	(hypothesis)
2.	$P \Rightarrow Q$	($\wedge$-elimination 1 on line 1)
3.	$Q \Rightarrow R$	($\wedge$-elimination 2 on line 1)
4.	P	(subhypothesis)
	$\vdots$	
6.	R	(?)
7.	$P \Rightarrow R$	($\Rightarrow$-introduction on subproof 4–6)

The proof can now be completed using implication elimination:

1.	$(P \Rightarrow Q) \wedge (Q \Rightarrow R)$	(hypothesis)
2.	$P \Rightarrow Q$	($\wedge$-elimination 1 on line 1)
3.	$Q \Rightarrow R$	($\wedge$-elimination 2 on line 1)
4.	P	(subhypothesis)
5.	Q	($\Rightarrow$-elimination on lines 2 and 4)
6.	R	($\Rightarrow$-elimination on lines 3 and 5)
7.	$P \Rightarrow R$	($\Rightarrow$-introduction on subproof 4–6)

Now that was easy, wasn't it? Note that lines 5 and 6 use information which was known before the subproof was opened, namely lines 2 and 3. ◇

Example Rather surprisingly, $P \Rightarrow (Q \Rightarrow R) \dashv\vdash P \wedge Q \Rightarrow R$. Let us prove just one direction of the equivalence here: $P \Rightarrow (Q \Rightarrow R) \vdash P \wedge Q \Rightarrow R$, leaving the other as an exercise.

1.	$P \Rightarrow (Q \Rightarrow R)$	(hypothesis)
2.	$P \wedge Q$	(subhypothesis)
3.	P	($\wedge$-elimination 1 on line 2)
4.	$Q \Rightarrow R$	($\Rightarrow$-elimination on lines 1 and 3)
5.	Q	($\wedge$-elimination 2 on line 2)
6.	R	($\Rightarrow$-elimination on lines 4 and 5)
7.	$P \wedge Q \Rightarrow R$	($\Rightarrow$-introduction on subproof 2–6)

Build the proof yourself to understand it better. ◇

Example What if we were asked to prove something very similar: $P \Rightarrow (Q \Rightarrow R) \vdash Q \wedge P \Rightarrow R$. The proof should be easy if we use the previous theorem and the theorem of commutativity of conjunction.

1.	$P \Rightarrow (Q \Rightarrow R)$	(hypothesis)
2.	$P \wedge Q \Rightarrow R$	(theorem $P \Rightarrow (Q \Rightarrow R) \vdash P \wedge Q \Rightarrow R$ on line 1)
3.	$Q \wedge P$	(subhypothesis)
4.	$P \wedge Q$	(theorem $A \wedge B \vdash B \wedge A$ on line 3)
5.	R	($\Rightarrow$-elimination on lines 2 and 4)
6.	$Q \wedge P \Rightarrow R$	($\Rightarrow$-introduction on subproof 3–5)

Note that every time we use a theorem we must clearly state what we are applying, and make sure that it has already been proved elsewhere. ◇

Example Subproofs can appear within other subproofs. As an example of a proof which uses nested subproofs, consider a formal proof of: $A \Rightarrow C \vdash A \Rightarrow (B \Rightarrow C)$. We start by quoting the hypothesis and adding the conclusion as the last line of the proof:

1.	$A \Rightarrow C$	(hypothesis)
	⋮	
6.	$A \Rightarrow (B \Rightarrow C)$	(?)

Since the last line is an implication, we use implication introduction, introducing a subproof:

1.	$A \Rightarrow C$	(hypothesis)
2.	A	(subhypothesis)
	⋮	
5.	$B \Rightarrow C$	(?)
6.	$A \Rightarrow (B \Rightarrow C)$	($\Rightarrow$-introduction on subproof 2–5)

Once again, the line at the bottom is an implication, thus forcing us to open a further subproof to be able to prove the bottom line using implication introduction.

Using implication introduction, we obtain:

1.	$A \Rightarrow C$	(hypothesis)
2.	A	(subhypothesis)
3.	B	(subhypothesis)
	⋮	
4.	C	(?)
5.	$B \Rightarrow C$	($\Rightarrow$-introduction on subproof 3–4)
6.	$A \Rightarrow (B \Rightarrow C)$	($\Rightarrow$-introduction on subproof 2–5)

The proof can now be concluded using implication elimination:

1.	$A \Rightarrow C$	(hypothesis)
2.	A	(subhypothesis)
3.	B	(subhypothesis)
4.	C	($\Rightarrow$-elimination on lines 1 and 2)
5.	$B \Rightarrow C$	($\Rightarrow$-introduction on subproof 3–4)
6.	$A \Rightarrow (B \Rightarrow C)$	($\Rightarrow$-introduction on subproof 2–5)

◇

Exercises

2.34 Prove the other direction of the equivalence in the last example.

2.35 Prove that $(P \Rightarrow Q) \Rightarrow R \vdash P \Rightarrow (Q \Rightarrow R)$.

2.36 Using previous results, prove that $(P \Rightarrow Q) \Rightarrow R \vdash P \wedge Q \Rightarrow R$.
2.37 Prove that $P \Rightarrow Q \wedge R \vdash (P \Rightarrow Q) \wedge (P \Rightarrow R)$.

2.4.2.3 Bi-implication

Bi-implication has very simple rules. We can either break up a bi-implication into two implications, or combine two implications into a bi-implication:

$$\frac{A \Rightarrow B, \quad B \Rightarrow A}{A \Leftrightarrow B} \qquad \frac{A \Leftrightarrow B}{A \Rightarrow B} \qquad \frac{A \Leftrightarrow B}{B \Rightarrow A}$$

We call these rules $\Leftrightarrow$*-introduction*, $\Leftrightarrow$*-elimination 1* and $\Leftrightarrow$*-elimination 2*, respectively.

Example Let us prove that $P \wedge Q \vdash P \Leftrightarrow Q$

1.	$P \wedge Q$	(hypothesis)
2.	P	(subhypothesis)
3.	Q	($\wedge$-elimination 2 on line 1)
4.	$P \Rightarrow Q$	($\Rightarrow$-introduction on subproof 2–3)
5.	Q	(subhypothesis)
6.	P	($\wedge$-elimination 1 on line 1)
7.	$Q \Rightarrow P$	($\Rightarrow$-introduction on subproof 5–6)
8.	$P \Leftrightarrow Q$	($\Leftrightarrow$-introduction on lines 4 and 7)

$\diamond$

Exercises
2.38 Prove the following:

(i) $A \Leftrightarrow B \vdash B \Leftrightarrow A$
(ii) $\vdash (A \wedge B) \wedge C \Leftrightarrow A \wedge (B \wedge C)$
(iii) $\vdash A \wedge B \Leftrightarrow B \wedge A$
(iv) $\vdash A \Leftrightarrow A$

2.4.2.4 Disjunction

Let us move on to disjunction. Introducing a disjunction is straightforward:

$$\frac{A}{A \vee B} \qquad \frac{B}{A \vee B}$$

We call these rules $\vee$*-introduction 1* and $\vee$*-introduction 2*, respectively.

Example To prove that disjunction follows from conjunction: $P \wedge Q \vdash P \vee Q$, a short proof does the job:

1.	$P \wedge Q$	(hypothesis)
2.	P	($\wedge$-elimination 1 on line 1)
3.	$P \vee Q$	($\vee$-introduction 1 on line 2)

$\diamond$

What about eliminating a disjunction? If we know that $P \vee Q$, can we conclude something which does not involve a disjunction? At first sight, not much. However, consider a scenario in which we know that (i) if a car is rusty then it is cheap, and (ii) if a car is old, then it is cheap. Now, if I own a car which is either old or rusty (possibly both), I should be able to conclude that my car is cheap. Therefore, if we know that $P \Rightarrow R$ and $Q \Rightarrow R$, then the proposition R must follow from $P \vee Q$.

This is the elimination rule $\vee$-*elimination* for disjunction:

$$\frac{A \Rightarrow C,\ B \Rightarrow C,\ A \vee B}{C}$$

Consider a partially finished proof, in which we have to conclude C from a statement $A \vee B$:

	$\vdots$	
9.	$A \vee B$	$(\dots)$
	$\vdots$	
23.	C	(?)
	$\vdots$	

To be able to apply disjunction elimination, we need two additional results, namely that $A \Rightarrow C$ and $B \Rightarrow C$:

	$\vdots$	
9.	$A \vee B$	$(\dots)$
	$\vdots$	
21.	$A \Rightarrow C$	(?)
22.	$B \Rightarrow C$	(?)
23.	C	($\vee$-elimination on lines 9, 21 and 22)
	$\vdots$	

This leaves two implications to be proved. To prove them, we need to open a subproof for each:

	⋮	
9.	$A \vee B$	
10.	A	(subhypothesis)
	⋮	
15.	C	(?)
16.	B	(subhypothesis)
	⋮	
20.	C	(?)
21.	$A \Rightarrow C$	($\Rightarrow$-introduction on subproof 10–15)
22.	$B \Rightarrow C$	($\Rightarrow$-introduction on subproof 16–20)
23.	C	($\vee$-elimination on lines 9, 21 and 22)
	⋮	

This structure of proofs in which we need to eliminate disjunction is thus rather uniform, as we will see in the coming examples.

Example Let us start by proving that disjunction is commutative:

$$P \vee Q \vdash Q \vee P$$

1.	$P \vee Q$	(hypothesis)
2.	P	(subhypothesis)
3.	$Q \vee P$	($\vee$-introduction 2 on line 2)
4.	Q	(subhypothesis)
5.	$Q \vee P$	($\vee$-introduction 1 on line 4)
6.	$P \Rightarrow (Q \vee P)$	($\Rightarrow$-introduction on subproof 2–3)
7.	$Q \Rightarrow (Q \vee P)$	($\Rightarrow$-introduction on subproof 4–5)
8.	$Q \vee P$	($\vee$-elimination on lines 1, 6 and 7)

The proof follows the structure we have seen for proofs which eliminate a disjunction, with the remaining parts being rather straightforward applications of disjunction introduction. ◇

Example An important law of propositional logic is that of distributivity of conjunction over disjunction: $P \wedge (Q \vee R) \dashv\vdash (P \wedge Q) \vee (P \wedge R)$. Let us look at one direction of this equivalence: $P \wedge (Q \vee R) \vdash (P \wedge Q) \vee (P \wedge R)$.

The proof of this direction of the law is the following:

1.	$P \wedge (Q \vee R)$	(hypothesis)
2.	P	($\wedge$-elimination 1 on line 1)
3.	$Q \vee R$	($\wedge$-elimination 2 on line 1)
4.	Q	(subhypothesis)
5.	$P \wedge Q$	($\wedge$-introduction on lines 2 and 4)
6.	$(P \wedge Q) \vee (P \wedge R)$	($\vee$-introduction 1 on line 5)
7.	R	(subhypothesis)
8.	$P \wedge R$	($\wedge$-introduction on lines 2 and 7)
9.	$(P \wedge Q) \vee (P \wedge R)$	($\vee$-introduction 2 on line 8)
10.	$Q \Rightarrow ((P \wedge Q) \vee (P \wedge R))$	($\Rightarrow$-introduction on subproof 4–6)
11.	$R \Rightarrow ((P \wedge Q) \vee (P \wedge R))$	($\Rightarrow$-introduction on subproof 7–9)
12.	$(P \wedge Q) \vee (P \wedge R)$	($\vee$-elimination on lines 3, 10 and 11)

◇

Example Let us now look at a slightly more complex example: associativity. Only one direction is proved, $P \vee (Q \vee R) \vdash (P \vee Q) \vee R$.

1.	$P \vee (Q \vee R)$	(hypothesis)
2.	P	(subhypothesis)
3.	$P \vee Q$	($\vee$-introduction 1 on line 2)
4.	$(P \vee Q) \vee R$	($\vee$-introduction 1 on line 3)
5.	$Q \vee R$	(subhypothesis)
6.	Q	(subhypothesis)
7.	$P \vee Q$	($\vee$-introduction 2 on line 6)
8.	$(P \vee Q) \vee R$	($\vee$-introduction 1 on line 7)
9.	R	(subhypothesis)
10.	$(P \vee Q) \vee R$	($\vee$-introduction 2 on line 9)
11.	$Q \Rightarrow ((P \vee Q) \vee R)$	($\Rightarrow$-introduction on subproof 6–8)
12.	$R \Rightarrow ((P \vee Q) \vee R)$	($\Rightarrow$-introduction on subproof 9–10)
13.	$(P \vee Q) \vee R$	($\vee$-elimination on lines 5, 11 and 12)
14.	$P \Rightarrow ((P \vee Q) \vee R)$	($\Rightarrow$-introduction on subproof 2–4)
15.	$(Q \vee R) \Rightarrow ((P \vee Q) \vee R)$	($\Rightarrow$-introduction on subproof 5–13)
16.	$(P \vee Q) \vee R$	($\vee$-elimination on lines 1, 14 and 15)

Although it is easy to check that it is correct, this proof is practically impossible to understand if you try to read it from top to bottom. Instead, build the proof gradually, using the rules and creating subproofs as necessary, to ensure that you understand it. ◇

Exercises

2.39 Give a different three-line proof that disjunction follows from conjunction from the one given in the example.

2.40 Complete the proof of distributivity of conjunction over disjunction.
2.41 Prove that disjunction is distributive over conjunction: $P \vee (Q \wedge R) \dashv\vdash (P \vee Q) \wedge (P \vee R)$.
2.42 Complete the proof that disjunction is associative.
2.43 Prove the other distributivity law that $P \vee (Q \wedge R) \dashv\vdash (P \vee Q) \wedge (P \vee R)$.

2.4.2.5 Negation

It is not immediately obvious what rules of inference to use for negation. Let us start with negation elimination. Eliminating a single negation is not simple to do; however, we can eliminate negations in pairs, concluding A from $\neg(\neg A)$:

$$\frac{\neg\neg A}{A}$$

As for the introduction of negation, how can we prove that something is false: $\neg A$. A technique which philosophers developed to be able to deal with this is that of *reductio ad absurdum* or *proof by contradiction*.

Example Consider the following scenario. An operating system ensures two properties: (i) system files cannot be owned by a normal user, and (ii) a user may only delete files owned by him- or herself. One would expect that, if the operating system satisfies these two properties, then normal users may not delete system files. An informal argument would be as follows: If a normal user may delete a system file then, by rule (ii), the user must own that file. However, by rule (i), the user does not own it. Therefore, from the statement that a normal user can delete a system file, we concluded two contradictory things—that the user owns the file, and that the user does not own it. This means that it is impossible for the statement that a normal user may not delete a system file to be true. This is a proof by contradiction. ◇

We will embody this reasoning in a rule of inference: if a statement A implies two contradictory statements ($A \Rightarrow B$ and $A \Rightarrow \neg B$) then A cannot be true, and we can conclude $\neg A$:

$$\frac{A \Rightarrow B,\ A \Rightarrow \neg B}{\neg A}$$

This rule (*¬-introduction*) allows us to prove negations. The biggest challenge is that B appears only in the antecedents of the rule. Therefore, if we need to prove that $\neg A$, we have to guess what is the appropriate B that will allow us to use the rule.

Let us see some applications of these rules of inference.

Example The first example is a rather straightforward application of the rule of inference. If both a statement P and its inverse hold, then we should be able to prove anything: $P \wedge \neg P \vdash Q$. As usual, we start by writing the desired proof structure:

1.	$P \wedge \neg P$	(hypothesis)
	$\vdots$	
9.	Q	(?)

Since there is nothing we can do with the conclusion (and breaking down the hypothesis does not help), we add negations to be able to use negation introduction:

1.	$P \wedge \neg P$	(hypothesis)
	$\vdots$	
8.	$\neg\neg Q$	(?)
9.	Q	($\neg$-elimination on line 8)

Now, the result we have to prove, $\neg\neg Q$, is a negation, which means that we can use negation introduction, by proving that $\neg Q$ implies some statement and its opposite:

1.	$P \wedge \neg P$	(hypothesis)
	$\vdots$	
6.	$\neg Q \Rightarrow ?$	(?)
7.	$\neg Q \Rightarrow \neg ?$	(?)
8.	$\neg\neg Q$	($\neg$-introduction on lines 6 and 7)
9.	Q	($\neg$-elimination on line 8)

What statement to choose is the key to the proof. Looking at the hypothesis, we realise, however, that both P and $\neg P$ are known, and would thus be good candidates for the statements following from $\neg Q$:

1.	$P \wedge \neg P$	(hypothesis)
	$\vdots$	
6.	$\neg Q \Rightarrow P$	(?)
7.	$\neg Q \Rightarrow \neg P$	(?)
8.	$\neg\neg Q$	($\neg$-introduction on lines 6 and 7)
9.	Q	($\neg$-elimination on line 8)

The proof can now be easily completed:

1.	$P \wedge \neg P$	(hypothesis)
2.	$\neg Q$	(subhypothesis)
3.	P	($\wedge$-elimination 1 on line 1)
4.	$\neg Q$	(subhypothesis)
5.	$\neg P$	($\wedge$-elimination 2 on line 1)
6.	$\neg Q \Rightarrow P$	($\Rightarrow$-introduction on subproof 2–3)
7.	$\neg Q \Rightarrow \neg P$	($\Rightarrow$-introduction on subproof 4–5)
8.	$\neg\neg Q$	($\neg$-introduction on lines 6 and 7)
9.	Q	($\neg$-elimination on line 8)

◇

Example When talking about model theory, we mentioned the rule of the excluded middle: $\vdash P \vee \neg P$. We can now prove it using our rules of inference.

1.	$\neg(P \vee \neg P)$	(subhypothesis)
2.	P	(subhypothesis)
3.	$P \vee \neg P$	($\vee$-introduction 1 on line 2)
4.	P	(subhypothesis)
5.	$\neg(P \vee \neg P)$	(copy line 1)
6.	$P \Rightarrow (P \vee \neg P)$	($\Rightarrow$-introduction on subproof 2–3)
7.	$P \Rightarrow \neg(P \vee \neg P)$	($\Rightarrow$-introduction on subproof 4–5)
8.	$\neg P$	($\neg$-introduction on lines 6 and 7)
9.	$P \vee \neg P$	($\vee$-introduction 2 on line 9)
10.	$\neg(P \vee \neg P)$	(subhypothesis)
11.	$\neg(P \vee \neg P) \Rightarrow (P \vee \neg P)$	($\Rightarrow$-introduction on subproof 1–9)
12.	$\neg(P \vee \neg P) \Rightarrow \neg(P \vee \neg P)$	($\Rightarrow$-introduction on subproof 10–10)
13.	$\neg\neg(P \vee \neg P)$	($\neg$-introduction on lines 11 and 12)
14.	$P \vee \neg P$	($\neg$-elimination on line 13)

The proof is not straightforward, and it helps to reconstruct it step by step to understand exactly how it works. ◇

These last two theorems we have proved, $P \wedge \neg P \vdash Q$ and $\vdash P \vee \neg P$, sometimes allow us to prove results in a more direct way than using the rules of inference for negation directly. Consider the following proof:

Example Implication can be expressed in terms of negation and disjunction—$P \Rightarrow Q$ is equivalent to $\neg P \vee Q$. The following proof shows one direction, namely that $P \Rightarrow Q \vdash \neg P \vee Q$.

The key to this proof is that, when we open the hypothesis and conclusion, nothing much can be done, so we apply the law of the excluded middle to be able to conclude that $P \vee \neg P$, after which we can use disjunction elimination to complete the proof.

1.	$P \Rightarrow Q$	(hypothesis)
2.	$P \vee \neg P$	(theorem $\vdash A \vee \neg A$)
3.	P	(subhypothesis)
4.	Q	($\Rightarrow$-elimination on lines 1 and 3)
5.	$\neg P \vee Q$	($\vee$-introduction 2 on line 4)
6.	$\neg P$	(subhypothesis)
7.	$\neg P \vee Q$	($\vee$-introduction 1 on line 6)
8.	$P \Rightarrow (\neg P \vee Q)$	($\Rightarrow$-introduction on subproof 3–5)
9.	$\neg P \Rightarrow (\neg P \vee Q)$	($\Rightarrow$-introduction on subproof 6–7)
10.	$\neg P \vee Q$	($\vee$-elimination on lines 2, 8 and 9)

The proof of $\neg P \vee Q \vdash P \Rightarrow Q$ is left as an exercise. ◇

Example Here are a couple of lemmata that we will need for the theorem that follows. The first says that a proposition which holds follows from anything: $P \vdash Q \Rightarrow P$:

1.	Q	(subhypothesis)
2.	P	(hypothesis)
3.	$Q \Rightarrow P$	($\Rightarrow$-introduction on subproof 1–2)

The second says that anything follows from a false statement: $P \vdash \neg P \Rightarrow Q$:

1.	P	(hypothesis)
2.	$\neg P$	(subhypothesis)
3.	$P \wedge \neg P$	($\neg$-introduction on lines 1 and 2)
4.	Q	(theorem $A \wedge \neg A \vdash B$ on line 3)
5.	$\neg P \Rightarrow Q$	($\Rightarrow$-introduction on subproof 2–4)

◇

Example The contrapositive law of implication says that $P \Rightarrow Q \dashv\vdash \neg Q \Rightarrow \neg P$. Let us prove this in one direction $P \Rightarrow Q \vdash \neg Q \Rightarrow \neg P$.

1.	$P \Rightarrow Q$	(hypothesis)
2.	$\neg(\neg Q \Rightarrow \neg P)$	(subhypothesis)
3.	P	(subhypothesis)
4.	Q	($\Rightarrow$-elimination on lines 1 and 3)
5.	$\neg Q \Rightarrow \neg P$	(lemma $A \vdash \neg A \Rightarrow B$ on line 4)
6.	P	(subhypothesis)
7.	$\neg(\neg Q \Rightarrow \neg P)$	(copy line 2)
8.	$\neg P$	($\neg$-introduction on lines 3–5 and 6–7)
9.	$\neg Q \Rightarrow \neg P$	(lemma $A \vdash B \Rightarrow A$ on line 8)
10.	$\neg(\neg Q \Rightarrow \neg P)$	(subhypothesis)
11.	$\neg\neg(\neg Q \Rightarrow \neg P)$	($\neg$-introduction on lines 2–9 and 10–10)
12.	$\neg Q \Rightarrow \neg P$	($\neg$-elimination on line 11)

◇

As you can see from these examples, proofs which involve negation are the most challenging ones we have encountered. When there seems to be no way forward in a proof, an approach which frequently works is to use the rule of the excluded middle: $\vdash P \vee \neg P$, and then use disjunction elimination to break the proof into two parts.

Exercises

2.44 Negation elimination allows us to conclude P from $\neg\neg P$. Prove the opposite direction: $P \vdash \neg\neg P$.

2.45 Prove that $\neg P \wedge (P \vee Q) \vdash Q$.

2.46 Prove that $\neg P \vdash \neg(P \wedge Q)$.

2.47 Prove that $\neg(P \vee Q) \vdash \neg P$ and that $\neg(P \vee Q) \vdash \neg Q$.

2.48 Prove the opposite direction of the contrapositive law.

2.49 *Modus tollens*, meaning 'the method that denies,' says that $(P \Rightarrow Q) \wedge \neg Q \vdash \neg P$. Prove it.

2.50 One way of decomposing bi-implication in terms of other operators is to rewrite $P \Leftrightarrow Q$ as $(P \wedge Q) \vee (\neg P \wedge \neg Q)$. Prove that $P \Leftrightarrow Q \vdash (P \wedge Q) \vee (\neg P \wedge \neg Q)$ and $(P \wedge Q) \vee (\neg P \wedge \neg Q) \vdash P \Leftrightarrow Q$.

2.51 Prove that $\neg P \vee Q \vdash P \Rightarrow Q$.

2.52 De Morgan's laws show how conjunction can be expressed in terms of disjunction and negation, and similarly, how disjunction can be expressed in terms of conjunction and negation. Prove them:

(i) $P \wedge Q \dashv\vdash \neg(\neg P \vee \neg Q)$
(ii) $P \vee Q \dashv\vdash \neg(\neg P \wedge \neg Q)$

2.4.2.6 Truth and Falsity

The only remaining terms in the language of propositional calculus which we do not know how to handle are *true* and *false*.

The well-formed formula *true* is easy to introduce, because it always holds:

$$\frac{}{true}$$

true-introduction is the only axiom we will be using. You may have observed that the others were all rules of inference.

What about eliminating *true*? Since nothing follows from just *true* (apart from *true* itself), we need no such rule of inference.

As for *false*, we can introduce it only when we reach a contradiction (by proving that P and $\neg P$ both hold). But we have proved that $P \wedge \neg P \vdash Q$ (for any Q), and thus we can already introduce *false* without the need of a new inference rule. What about the elimination of *false*? Using the dictum that anything follows from falsity, we express *false-elimination* as follows:

$$\frac{false}{P}$$

Example The proof that $\vdash P \vee \mathit{true}$ turns out to be straightforward:

1. true (true-introduction)
2. $P \vee \mathit{true}$ ($\vee$-introduction 2 on line 1)

◇

Exercises

2.53 Prove the following:

(i) $P \vee \mathit{true}$ is equivalent to true
(ii) $P \wedge \mathit{true}$ is equivalent to P
(iii) $P \vee \mathit{false}$ is equivalent to P
(iv) $P \wedge \mathit{false}$ is equivalent to false

2.4.3 Why Proofs?

One may be put off using this approach to prove things due to its complexity. However, as we have noted earlier an enumerative way of showing things is not possible for most domains. Furthermore, axioms and rules of inference allow us, or rather force us, to make explicit our underlying assumptions. Proofs can also be much shorter than the exponentially sized truth tables. For example, we can prove that $P \wedge (Q_1 \wedge \cdots \wedge Q_{1000}) \vdash P$ in just one line, rather than by drawing a table with 2^{1000} rows.

The proof approach is applicable to all areas of mathematics, from geometry and topology to propositional calculus and arithmetic. It was mentioned earlier, that this is sometimes called 'syntactic reasoning', and the reason is that we can apply the axioms and rules of inference without bothering to understand the real meaning of the symbols. You may have found yourself reading a proof, and understanding each and every line, without really understanding how each line contributes towards the theorem. This is an advantage, since it means that once we agree on the axioms and rules of inference, once someone produces a proof, there is no disagreeing about it. Each step can be easily checked (even by a computer), since it must match one of a small number of axioms or rules of inference. On the other hand, the level of detail one has to go through is usually daunting, and even mathematicians rarely prove theorems at this level. We will discuss this later in this chapter, when we will compare different types of proofs.

2.5 Comparing Truth Tables and Proofs

So we now have two different mathematical descriptions of propositional calculus: one model based, the other proof based. In the first we encoded the whole meaning of the symbols in the mathematical system, while in the second we tried to mimic

the way we would reason about formulae. The obvious question is now how the two systems compare. Is every semantic entailment a theorem? If $P \models Q$, can we always find a proof of $P \vdash Q$? In other words, can every true statement (in terms of truth tables) be proved? Vice versa, is every theorem a semantic entailment? In other words, is everything that can be proved true?

2.5.1 Completeness

Definition 2.14 *A mathematical system S is said to be* complete *with respect to another mathematical system S', if every truth in S' is a truth in S.* ■

In our case, we ask whether for every P and Q such that $P \models Q$, it is also the case that $P \vdash Q$. Propositional calculus turns out to be complete with respect to model theory. The proof of completeness is beyond the scope of the book, however its implications are not. If we are trying in vain to prove that $P \vdash Q$, we can use a truth table, and if we can show that $P \models Q$, it then follows by completeness that $P \vdash Q$.

2.5.2 Soundness

Definition 2.15 *A mathematical system S is said to be* sound *with respect to another mathematical system S', if every truth in S is also a truth in S'.* ■

With propositional logic, the question is whether every proof yields a semantic entailment: if $P \vdash Q$, does it follow that $P \models Q$? Again, propositional calculus turns out to be sound, and we can thus avoid working out huge truth tables by using proofs. The proof of soundness is also beyond the scope of the book.

Note that saying that S is complete with respect to S' is equivalent to saying that S' is sound with respect to S. Therefore, it follows that $\models$ and $\vdash$ are both sound and complete with respect to each other.

2.6 More About Propositional Logic

2.6.1 Boolean Algebra

You have probably been using the word algebra in mathematics for quite a few years. But have you ever wondered what an algebra really is? In mathematics, an algebra is a collection of identities (equalities). Any operators which satisfy these identities are called an instance of the algebra.

George Boole came up with what is today called a Boolean algebra, when he observed some laws of sets and realised that they are also satisfied by logic sentences and numbers. He chose a number of such laws and went on to prove things about the objects which satisfy the laws. Any class of objects which satisfies the basic laws automatically satisfied his results.

A Boolean algebra is a collection of objects X, together with binary operators $\odot$ and $\oplus$, unary operator $\bar{\cdot}$ (a bar over the operand) and a notion of equality $=$, which satisfy the following laws:

Commutativity: Both $\odot$ and $\oplus$ are commutative.

$$x \odot y = y \odot x$$
$$x \oplus y = y \oplus x$$

Associativity: Both $\odot$ and $\oplus$ are associative.

$$x \odot (y \odot z) = (x \odot y) \odot z$$
$$x \oplus (y \oplus z) = (x \oplus y) \oplus z$$

Identities: There are two special values of S, namely 1 and 0, which act as identities of $\odot$ and $\oplus$ respectively:

$$x \odot 1 = x$$
$$x \oplus 0 = x$$

Distributivity: $\odot$ and $\oplus$ distribute over each other:

$$x \odot (y \oplus z) = (x \odot y) \oplus (x \odot z)$$
$$x \oplus (y \odot z) = (x \oplus y) \odot (x \oplus z)$$

Complement: The unary operator interacts with $\odot$ and $\oplus$ as a complement operator:

$$x \odot \bar{x} = 0$$
$$x \oplus \bar{x} = 1$$

What operators satisfy these laws? At first numbers with addition and multiplication seem to fit the bill, but some laws fail (for example the distributivity laws). One solution is to take S to be just the numbers 0 and 1, $\odot$ to be multiplication, $\oplus$ to be addition modulo 2,[7] and the unary operator $\bar{x}$ to be defined to be $1 - x$. All the identities of Boolean algebra are satisfied by these operators.

Another way of satisfying these laws is to take 1 to be *true*, 0 to be *false*, $\odot$ to be $\wedge$, $\oplus$ to be $\vee$ and $\bar{x}$ to be $\neg x$. If we take equality to be either $=\!\!|\!\!=$ or $\dashv\vdash$,

[7] In which we have only the numbers 0 and 1, and taking $1 \oplus 1 = 1$.

propositional logic turns out to be an instance of Boolean algebra. Various other interesting domains turn out to be Boolean algebrae.

What is interesting about these laws, is that whatever we prove based only on the laws of the algebra, can be concluded for propositional logic, addition modulo 2, or any other instance of a Boolean algebra. Various interesting laws can be derived from the basic identities of the algebra.

The following table gives a number of these laws.

Idempotency laws:	$x \oplus x = x$
	$x \odot x = x$
Dominance laws:	$x \oplus 1 = 1$
	$x \odot 0 = 0$
De Morgan's laws:	$x \oplus y = \overline{\bar{x} \odot \bar{y}}$
	$x \odot y = \overline{\bar{x} \oplus \bar{y}}$
Absorption laws:	$x \oplus (x \odot y) = x$
	$x \odot (x \oplus y) = x$
Double complement:	$\bar{\bar{x}} = x$

Just to give an idea of how these are proved, we will show the proof of one of the idempotency laws:

$$
\begin{aligned}
& x \\
= \quad & \text{identities law } (x = x \odot 1) \\
& x \odot 1 \\
= \quad & \text{complement law } (x \oplus \bar{x} = 1) \\
& x \odot (x \oplus \bar{x}) \\
= \quad & \text{distributivity law } (x \odot (y \oplus z) = (x \odot y) \oplus (x \odot z)) \\
& (x \odot x) \oplus (x \odot \bar{x}) \\
= \quad & \text{complement law } (x \odot \bar{x} = 0) \\
& (x \odot x) \oplus 0 \\
= \quad & \text{identities law } (x \oplus 0 = x) \\
& x \odot x
\end{aligned}
$$

Note the different style of proof. This is called an algebraic proof, where we replace a term by an equal term until we reach the desired result.

Since propositional logic satisfies the Boolean algebra basic laws, we immediately know that anything that follows from these laws is also satisfied by propositional logic.

Exercises

2.54 Prove the other idempotency law.

2.55 Prove the dominance laws.

2.6.2 Doing with Less

For convenience, we have defined our logic in terms of a rather large number of operators. Can we do with less? We have already seen how some operators can be

expressed in terms of simpler ones. The equivalences below show how we can make do with just negation and conjunction or negation and disjunction.

$$P \wedge Q \dashv\vdash \neg(\neg P \vee \neg Q)$$
$$P \vee Q \dashv\vdash \neg(\neg P \wedge \neg Q)$$
$$P \Rightarrow Q \dashv\vdash \neg P \vee Q$$
$$P \Leftrightarrow Q \dashv\vdash (P \Rightarrow Q) \wedge (Q \Rightarrow P)$$
$$\mathit{true} \dashv\vdash P \vee \neg P$$
$$\mathit{false} \dashv\vdash P \wedge \neg P$$

The last four laws show how all the operators can be expressed in terms of just conjunction, disjunction and negation. The first line then shows how one can convert conjunction to disjunction and negation, thus reducing any formula into these two operators. Alternatively, one may apply the law given in the second line to convert the operators into just conjunction and negation.

Another related question you may ask is whether someone can define a new interesting operator which cannot be expressed in terms of our operators. It turns out not to be possible. The operators are universal in the sense that, for any truth table, we can express the last column in terms of our operators.

Exercises

2.56 A propositional formula is said to be a *literal* if it consists of either just a propositional variable, or the negation of a propositional variable. It is said to be a *clause* if it is the disjunction of a number of literals (possibly just one). Finally, it is said to be in *conjunctive normal form* if it is a conjunction of a number of clauses (again, possibly just one). For example, $\neg P$ and Q are literals, while $(\neg P \vee Q \vee \neg R)$ is a clause and $(\neg P \vee Q \vee \neg R) \wedge (P \vee Q) \wedge \neg Q$ is in conjunctive normal form. It can be shown that all propositional formulae can be transformed into conjunctive normal form. Using the results we proved in this chapter, transform the formula $P \Rightarrow (\neg Q \wedge R)$ into an equivalent formula in conjunctive normal form.

2.6.3 Doing with Even Less

Can we do with even less? Consider the nor operator (written as $\triangledown$). $P \triangledown Q$ is true only if neither P nor Q are true (both are false).

P	Q	$P \triangledown Q$
true	*true*	*false*
true	*false*	*false*
false	*true*	*false*
false	*false*	*true*

Now look at the following truth table for $P \bigtriangledown P$:

P	$P \bigtriangledown P$
true	*false*
false	*true*

Therefore, we can write $\neg P$ as $P \bigtriangledown P$.

As its name suggests, nor behaves much like $\neg$ and $\vee$: $P \bigtriangledown Q = \neg(P \vee Q)$. It thus follows that $P \vee Q = \neg(P \bigtriangledown Q)$. But we already know how to write $\neg$ in terms of $\bigtriangledown$. We can thus express $\vee$ in terms of $\bigtriangledown$.

But we have shown earlier that all our operators can be expressed in terms of $\neg$ and $\vee$. We can thus express all expressions in terms of $\bigtriangledown$.

Exercises

2.57 Write an expression equivalent to $P \vee Q$ using only $\bigtriangledown$. Verify it is correct using truth tables.

2.58 Nor is not the only universal operator. Another operator is the *nand* operator written as $\bigtriangleup$. $P \bigtriangleup Q$ is false only when both P and Q are true. Give the basic truth table of nand and use the same kind of arguments we used with nor to show that nand is a universal operator.

2.6.4 What if the Axioms Are Wrong?

You may be asking why we should believe the axioms and rules of inference. Of course, the fact that we have shown them to be sound and complete guarantees that they are correct with respect to the results obtainable using truth tables. If we accept truth tables to be the 'truth', our rules of inference are correct.

What about systems where we are not sure of their soundness? Well, let us look at propositional logic. We were calling the symbols and, or, not etc. However, the conclusions are true for all systems which obey the axioms and rules of inference. In fact, the short section about Boolean algebra should have convinced you that the results hold for addition and multiplication modulo 2. Therefore, we should see our results as holding not just on propositions, but also any system which happens to satisfy the axioms.

Over 2,000 years ago, Euclid gave an axiomatised geometry. Amongst his axioms was a rather complex one which people tried for 2,000 years to prove in terms of the other axioms. In the 1700s, mathematicians showed that, if we take the opposite of the axiom to be true, not only can no contradiction be reached, but the resulting system models a different kind of geometry which has a physical (albeit non-Euclidian) instance. In other words, there is nothing holy about axioms. Their opposite may equally well describe another phenomenon.

One such case in propositional logic is intuitionistic logic, where the law of the excluded middle is no longer a theorem of the system. The result is a logic, not

unlike the one we have explored, but where all proofs are constructive (hence sometimes called *constructive logic*) and no proofs by contradiction are possible. It turns out to be a better logic to reason about certain systems.

2.7 Some Notes

Now that we have explored an area of mathematics in depth, we can look back at the subject as a whole and raise some issues which were not evident during the exposition, or which are peripheral to the area.

2.7.1 Notation and Symbols

We started this chapter by presenting the language of propositional logic—its syntax, from which we can derive meaningful sentences in the language, or well-formed formulae as we have called them. One frequently overlooked aspect of mathematics is the notation and symbols used. Indeed, notation in itself is not a prerequisite to good mathematics.[8] However, mathematicians all agree that elegance plays a strong role in mathematics. Einstein is said to have noted that, given the choice between two mathematical models, the simpler, more elegant one turns out to be the correct one. Although elegance is usually applied to the axioms and proofs, notation can also be elegant or not.

Designers have an important rule of thumb when it comes to designing functional (as opposed to decorative) devices: *the user should never have to stop and think how to use a device.* Have you ever came across a door which you were unsure whether it should be pushed or pulled? Or does it slide sideways? One solution is to put up a sign saying 'Turn clockwise, then push'; however, the best solution is to have the device suggesting its use: a flat bar as a handle immediately indicates that you have to push. You wouldn't even stop to think what you should do. The same should hold with mathematical notation. Here are a few things to keep in mind when defining new operators:

- A symmetric symbol should be used for commutative functions or symmetric relations. This indicates to the reader a property of the operator. Good examples of this rule are the symbols $+$ and $\times$. A bad example of an operator which gives the visual illusion of symmetry is subtraction.

[8]The celebrated mathematician John Conway was once attending a presentation where the mathematician noted, in passing, that he had not yet managed to prove one particular statement which he believed to be true. At the end of the presentation, Conway stood up and sketched out the proof of the statement. 'Yes, your notation is much better, which allows you to derive the proof easily' was the presenter's first comment, to which Conway is said to have tersely reprimanded 'Mathematics is about notions, not notations.'

- In operators where the order of the operands is important, the use of an asymmetric symbol can be useful. Not only does it allow the user to remember better how the operands are applied, but also allows him or her to write them the other way round if more convenient. Consider the less-than operator. Not only does $x < y$ help us remember which is the smaller, but it also allows us to write $y > x$ (reversing the operator symbol) as a syntactic equivalent of $x < y$. In programming, the assignment symbol `:=` is much more appropriate than `=`. Indeed, in some circumstances one might prefer to write `sin(x)+cos(y) =: z;` instead of `z := sin(x)+cos(y);` because it would make the meaning of a program more evident to the reader of the code. Alas, mainstream programming languages do not support this.[9]
- Operators which are related, duals or inverses of each other, should be graphically related. In propositional logic, conjunction and disjunction are duals (see De Morgan's laws) and are thus visually related: $\wedge$ and $\vee$. We will later see that set intersection and union are related to conjunction and disjunction (respectively), and will use the symbols $\cap$ and $\cup$ to represent them. The similarity is not coincidental. It indicates a similarity between the operators, and suggests that they obey similar laws.
- We say that a symbol is *overloaded* when it is used to represent more than one operation. As we have mentioned in Sect. 2.6.1 about Boolean algebra, some texts use $+$ for disjunction and $\cdot$ for conjunction. At first, one may find this confusing. However, once the similarity of the laws obeyed by disjunction and conjunction with the ones obeyed by addition (counting up to 1) and multiplication becomes evident, we realise that the choice of symbols is appropriate since it hints at these similarities. It is not surprising that $x \times (y + z) = x \times y + x \times z$ is true in both interpretations of the operator.
- However clear and intuitive your symbols are, always tell the reader how expressions which use your symbol are read.

2.7.2 Proof Elegance

Some proofs are more elegant than others. It is not a matter of simplicity or comprehensibility, but one of style, even if reaching a conclusion using simpler arguments and theorems is usually a prevalent element in elegant proofs. Such a proof usually depends on a twist or insight which, although at first may seem tangential to the result, leads to the desired conclusion in an unexpected, yet more direct manner. You will know one when you see one. For example, proofs by exhaustive case analysis rarely fall under this category.

[9]Most programming languages are notoriously bad when it comes to notation and meaning. You may have always believed that `+` is commutative in C or Java, but think again. If you write a function `incX()` which increments the value of a global variable `x` and returns the value `0`, the value of `incX()+x` is now clearly different from `x+incX()`.

Paul Erdös, one of the most influential mathematicians of this past century, had more than 1,000 papers published under his name, each of which contained numerous proofs of varying complexity. He frequently talked about the Book—a book with a perfect proof on each of its (infinite number of) pages possessed by God. Mathematicians were allowed one glance of a page before being born, which they strive throughout their life to reproduce. These were what he called 'proofs from the Book'.

It takes time to learn how to write an elegant proof. You may still be wondering why, given the soundness and completeness results we gave, we do not simply prove everything by constructing the truth table. Well, firstly, the size of the truth table of an expression grows exponentially with the number of variables. Propositional formulae are used to model the behaviour of circuits, which can run into thousands of variables. Despite the size of the truth table, a proof of the formula may be much shorter. Secondly, from the next chapter we will start reasoning about infinite domain systems. Proving that every number divisible by four is also divisible by two is quite easy using axioms and rules of inference, while showing its truth by testing it on all multiples of four is impossible. Thirdly, it is a matter of elegance. Never go for brute force when more elegant approaches exist.

2.7.3 How to Come up with a Proof

Proofs are not always easy to come up with, being mostly difficult because there seems to be no obvious pattern, method or approach that will always work. So is there one? Can we write a computer program which outputs whether a given conjecture is true or not?

For propositional logic, we have shown that the axioms and rules are complete and sound. We can thus try out all combinations of the truth table. As noted before, however, this is of exponential size, and thus very inefficient. Is there a better way? Interestingly enough, this question turns out to be a very difficult one. It was posed in 1970, and no one has yet managed to show whether it is possible to write a program which does not require exponential time to solve the problem.

For more complex systems, the situation is even worse. Following the work of Kurt Gödel and Alan Turing in the 1930s, we know that (i) a mathematical system which can handle arithmetic (addition and multiplication) cannot be both sound and complete, and (ii) it is not possible to write an algorithm which, given a statement in such a system, tells us whether or not it is true.

So to conclude, sorry, but there is no surefire easy method of coming up with a proof.

2.7.4 Not All Proofs Are Equal

We have already mentioned that the proofs we gave in this chapter are extremely detailed and tedious to write. Writing every proof in such a manner is not much

fun, and in fact, most proofs in mathematics and computer science are given in a less detailed fashion. Proofs written in this style are called *formal proofs,* with every single argument going back to the axioms or rules of inference. We will be proving things formally in this and the following chapter. What is particularly enticing about these proofs is that, since they leave no details out, they are easy to check, even by machine.

Less fastidious are *rigorous proofs,* where the proof is given in an exact mathematical manner, but not going back to the axioms. Most of the proofs in this book starting from Chap. 4 are rigorous. Rigorous does not mean leaving out parts of the proof, or stating that parts are obvious, but still proving everything, possibly taking several obvious steps at a time to make the proof easier to write and comprehend.

At the bottom level are *informal*, or *hand-waving* proofs. These can be useful to convey the general idea of how the actual proof works. Informal arguments will only be used to given an intuition of material beyond the scope of this book, and should not be considered as proofs.

http://www.springer.com/978-3-642-29839-4

Mathematics of Discrete Structures for Computer Science

Pace, G.J.

2012, XVI, 296 p., Hardcover
ISBN: 978-3-642-29839-4