CHAPTER 4

Logic

4.1 Introduction

The example at the end of the previous chapter which discovered whether or not a person was on board an aircraft used the *logical operators and* and *or*. This chapter gives a fuller explanation of these and other logical operators.

4.2 Propositional calculus

Propositional calculus is also known as *Boolean algebra* and is named after the mathematician George Boole. It is concerned with statements, called *propositions*, which may be either true or false. The values that the propositions may take are denoted by the words *true* and *false* in Z.

4.3 Operators

Since there are only two possible values that a proposition can take it is practical to explain the action of operators by enumerating all results. This is done using a table called a *truth table*. The following operators can be applied to propositions:

4.3.1 Negation

The *negation* operator is pronounced 'not' and is written:

$\neg$

For any proposition P, the truth table for negation is:

P	$\neg P$
false	true
true	false

When P is *false*, $\neg P$ is *true*; when P is *true*, $\neg P$ is *false*.

4.3.2 Conjunction

The *conjunction* operator is pronounced 'and' and is written:

$\wedge$

(To remember this symbol: $\wedge$ looks like the 'A' of 'AND')
Given propositions P and Q, the truth table for conjunction is:

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

The proposition $P \wedge Q$ is true only when both P is true and Q is true, otherwise it is false.

4.3.3 Disjunction

The *disjunction* operator is pronounced 'or' and is written:

$\vee$

Given propositions P and Q, the truth table for disjunction is:

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

The proposition $P \vee Q$ is true only when either P is true or Q is true or both are true. It is false when both P and Q are false.

4.3.4 Implication

The *implication* operator is pronounced 'implies' and is written:

$\Rightarrow$

The implication

$P \Rightarrow Q$

can be read as: 'if P is true then so is Q'.
Given propositions P and Q, the truth table for implication is:

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

The proposition $P \Rightarrow Q$ is false only when P is true and Q is false.
A useful relationship between implication and disjunction is

$P \Rightarrow Q$ is equivalent to $\neg P \vee Q$

This is useful for removing implications when manipulating expressions.

4.3.5 Equivalence

The *equivalence* operator is pronounced 'is equivalent to' or 'if and only if' and is written:

$$\Leftrightarrow$$

Given propositions P and Q, the truth table for equivalence is:

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

The proposition $P \Leftrightarrow Q$ is true if P always has the same truth value as Q. The equivalence given above for implication can be written:

$$P \Rightarrow Q \Leftrightarrow \neg P \vee Q$$

A useful law relating implication to equivalence is:

$$(P \Rightarrow Q) \wedge (Q \Rightarrow P) \Leftrightarrow (P \Leftrightarrow Q)$$

4.4 De Morgan's laws

The following important laws relating negation, conjunction and disjunction are due to the mathematician Augustus de Morgan:

$$\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$$
$$\neg(P \vee Q) \Leftrightarrow \neg P \wedge \neg Q$$

These are useful for simplifying expressions.

4.5 Demonstrating laws

A *law* is a relationship which holds good irrespective of the actual values of the propositions involved. Truth tables can be used to demonstrate the validity of a law. For example, to show the validity of the first of de Morgan's laws given above:

$$\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$$

we complete the truth table, building towards the expression to be compared:

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

$\neg P$	$\neg Q$	$\neg P \vee \neg Q$	$\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$
true	true	true	true
true	false	true	true
false	true	true	true
false	false	false	true

4.6 Using laws

Laws are used to prove that two statements in the propositional calculus, which are not necessarily identical, are equivalent. In formal specification laws are used in chains of transformations called *proofs* which can verify that a specification is consistent and make deductions about the behaviour of a system from its specification.

4.7 Example proof – exclusive-or

The *exclusive-or* operator (often written *xor*) is sometimes needed to relate two propositions. It gives true if one or other proposition is true but not both. (The disjunction operator is an *inclusive* or.) It is not a basic operator of propositional calculus but it can be formulated from the existing operators in (at least) two ways:

$$P \text{ xor } Q \Leftrightarrow (P \vee Q) \wedge \neg(P \wedge Q)$$
$$P \text{ xor } Q \Leftrightarrow (P \wedge \neg Q) \vee (\neg P \wedge Q)$$

These could be shown to be equivalent by means of a truth table, but a proof by application of laws taken from the list which follows is also possible and such a proof is more practical in cases where more than two propositions are involved.

The proof which follows is deliberately made very laborious so that every transformation will be visible and justified. In practice, many simple transformations are made from one line to the next and simple laws such as commutativity are not cited.

In general, deriving proofs is a mathematical skill which must be learned and which is beyond the scope of this book. Software tools are now available to assist in the derivation of proofs.

$(P \vee Q) \wedge \neg(P \wedge Q)$	first formulation
$\Leftrightarrow$	
$(P \vee Q) \wedge (\neg P \vee \neg Q)$	by de Morgan's *and*

$\Leftrightarrow$
$((P \vee Q) \wedge \neg P)$
$\vee ((P \vee Q) \wedge \neg Q)$ by distribution of *or* over *and*

$\Leftrightarrow$
$(\neg P \wedge (P \vee Q))$
$\vee (\neg Q \wedge (P \vee Q))$ by commutativity of *and* (twice)

$\Leftrightarrow$
$((\neg P \wedge P) \vee (\neg P \wedge Q))$
$\vee ((\neg Q \wedge P) \vee (\neg Q \wedge Q))$ by distribution of *or* over *and*

$\Leftrightarrow$
$((P \wedge \neg P) \vee (\neg P \wedge Q))$
$\vee ((\neg Q \wedge P) \vee (Q \wedge \neg Q))$ commutative *and* (twice)

$\Leftrightarrow$
$(\text{false} \vee (\neg P \wedge Q))$
$\vee ((\neg Q \wedge P) \vee \text{false})$ by contradiction (twice)

$\Leftrightarrow$
$((\neg P \wedge Q) \vee \text{false})$
$\vee ((\neg Q \wedge P) \vee \text{false})$ by commutativity of *or*

$\Leftrightarrow$
$(\neg P \wedge Q)$
$\vee (\neg Q \wedge P)$ by *or* simplification 3 (twice)

$\Leftrightarrow$
$(\neg P \wedge Q)$
$\vee (P \wedge \neg Q)$ by commutativity of *and*

$\Leftrightarrow$
$(P \wedge \neg Q) \vee (\neg P \wedge Q)$ by commutativity of *or*

4.8 Laws about logical operators

Given propositions P, Q and R:

Law	*Name*
$(P \wedge Q) \Leftrightarrow (Q \wedge P)$	commutativity of *and*
$(P \vee Q) \Leftrightarrow (Q \vee P)$	commutativity of *or*
$(P \Leftrightarrow Q) \Leftrightarrow (Q \Leftrightarrow P)$	commutativity of *equivalence*
$P \wedge (Q \wedge R) \Leftrightarrow (P \wedge Q) \wedge R$ $\Leftrightarrow P \wedge Q \wedge R$	associativity of *and*
$P \vee (Q \vee R) \Leftrightarrow (P \vee Q) \vee R$ $\Leftrightarrow P \vee Q \vee R$	associativity of *or*
$P \wedge (Q \vee R) \Leftrightarrow (P \wedge Q) \vee (P \wedge R)$	distribution of *or* over *and*
$P \vee (Q \wedge R) \Leftrightarrow (P \vee Q) \wedge (P \vee R)$	distribution of *and* over *or*
$\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$	de Morgan's *and*

$\neg(P \vee Q) \Leftrightarrow \neg P \wedge \neg Q$	de Morgan's *or*
$\neg(\neg P) \Leftrightarrow P$	negation
$P \vee \neg P \Leftrightarrow true$	excluded middle
$P \wedge \neg P \Leftrightarrow false$	contradiction
$P \Rightarrow Q \Leftrightarrow \neg P \vee Q$	implication
$(P \Leftrightarrow Q) \Leftrightarrow (P \Rightarrow Q) \wedge (Q \Rightarrow P)$	equality
$P \vee P \Leftrightarrow P$	*or* simplification 1
$P \vee true \Leftrightarrow true$	*or* simplification 2
$P \vee false \Leftrightarrow P$	*or* simplification 3
$P \vee (P \wedge Q) \Leftrightarrow P$	*or* simplification 4
$P \wedge P \Leftrightarrow P$	*and* simplification 1
$P \wedge true \Leftrightarrow P$	*and* simplification 2
$P \wedge false \Leftrightarrow false$	*and* simplification 3
$P \wedge (P \vee Q) \Leftrightarrow P$	*and* simplification 4

Precedence (highest to lowest)

$\neg$

$\wedge$

$\vee$

$\Rightarrow$

$\Leftrightarrow$

4.9 Summary of notation

true, false	logical constants
$\neg P$	negation: 'not P'
$P \wedge Q$	conjunction: 'P and Q'
$P \vee Q$	disjunction: 'P or Q'
$P \Rightarrow Q$	implication: 'P implies Q' or 'if P then Q'
$P \Leftrightarrow Q$	equivalence: 'P is logically equivalent to Q'
$t_1 = t_2$	equality between terms
$t_1 \neq t_2$	$\neg(t_1 = t_2)$

EXERCISES

1. Show by truth table that:

   $(P \Rightarrow Q) \Leftrightarrow (\neg P \vee Q)$

2. Show by truth table that:

   $((P \Rightarrow Q) \wedge (Q \Rightarrow P)) \Leftrightarrow (P \Leftrightarrow Q)$

3. By using laws from this chapter simplify:

   $\neg(p \notin onboard \wedge \#onboard < capacity)$

4. By using laws from this chapter simplify:

   $(a \wedge b) \vee (a \wedge c) \vee (a \wedge \neg c)$

5. Given

   $$p \in \text{loggedIn} \Rightarrow p \in \text{user}$$

   convince yourself that

   $$p \in \text{loggedIn} \wedge p \in \text{user}$$

   can be simplified to

   $$p \in \text{loggedIn}$$

6. Use de Morgan's laws to simplify the following expression:

   $x \neq 2 \vee x \neq 6$

7. Simplify the following expression:

   $s = t \wedge s \neq \text{EOF} \wedge t \neq \text{EOF}$

8. Simplify the following expression:

   $x = x \wedge (x \leq y \vee x = y)$

9. Simplify the following expression:

   $x = 0 \wedge x \geq 0$

10. Simplify the following expression:

    $\neg(\text{age} \geq 16 \vee \text{student})$