Chapter 3

Propositional Logic

Logic has many applications in computer science. For example, in formal reasoning, the design and verification of algorithms (which form the basis of all programs), in the design of logic and switching circuits, in the specification of processes, and in a programming methodology called logic programming, PROLOG. Of course, logical expressions appear in programming language constructs after, for examples, `if` and `while` statements. In this course we introduce the basic concepts of propositional and predicate logic, and discuss the meaning and methods of proof.

Most of the material in this chapter can be found in Sections 1.1, 1.2, 1.3, and 3.1 of Rosen which you are strongly urged to read.

Logic consists of a set of rules for drawing inferences. We assume that certain statements, called *axioms*, are true and we have a set of rules for proving consequences of the axioms.

For example, we could have axioms

Anne Boleyn was the mother of Queen Elizabeth I
Mary Boleyn was the sister of Anne Boleyn

and the rule

If (A was the mother of B) and (C was the sister of A) then (C was the aunt of B).

Using these rules we can deduce that Mary Boleyn was the aunt of Queen Elizabeth I.

We can use the same rules with other axioms. If we have

Marge Simpson was the mother of Bart
Selma Bouvier was the sister of Marge Simpson

then we can deduce Selma Bouvier was the aunt of Bart.

In integer arithmetic we have axioms and rules

$$x + y = y + x$$
$$x + 0 = x$$
$$\text{if } A = B \text{ then } B = A$$
$$\text{if } (A = B \text{ and } B = C) \text{ then } A = C.$$

Then we can deduce that $x = 0 + x$. This is because
$0 + x = x + 0$ by the first rule, then
$0 + x = x$ by the second and fourth rules, and so
$x = 0 + x$ by the third rule.

3.1 Propositions

A *proposition* is a statement which is either true or false. For example,

CS1870 is a first year course at Royal Holloway.
A comes before B in the English alphabet.
$99 > 6$

are all statements. A statement which is true is said to have *truth value* true, and a false statement is said to have *truth value* false. The above statements all have truth value true.

CS1870 is a second year course at Royal Holloway.
A does not come before B in the English alphabet.
$6 = 2 \times 4$

are all statements whose truth value is false.

There are propositions which are either true or false, but we don't know which. For example,

There are life forms outside of earth's solar system.
On 1st November 2062 there will be an earthquake in India.
Given any integer N, there are primes p and $p + 2$ greater than N.

Logic is concerned with propositions but not with absolute truth values. *Axioms* are propositions which are *assumed* to be true. We use logic to determine consequences of axioms. We do not *prove* that the axioms are true, but if the axioms are true then so are the consequences. For example:

if I miss the train, then I shall be late for work.

I will be late for work will be true if I miss the train, but if I don't miss the train then I may or may not be late for work, we can't tell.

if it is true that for any N, there are primes p and $p + 2$ greater than N,
then there are infinitely many pairs of primes of the form $(p, p + 2)$.

Goldbach's conjecture: every even number greater than 2 is the sum of two primes.

If Goldbach's conjecture is true, we can write $2n$ in the form $p + q$, where p, q are primes.

Exercise 3.1 Which of these are propositions?
1. In 2018 Theresa May was UK's Prime Minister.
2. What is your name?
3. It is false that all grass is red.
4. 8 is a prime number.
5. Close the door!
6. If a bird is white, then it is a swan.
7. If 2+2=5 then 3-2=7.
8. Every even number greater than 2 is the sum of two primes.

Some sentences are not propositions because they cannot have a truth value. For example,
9. This sentence is false.

A sentence which cannot be a proposition is called a *paradox*. We often use P, Q, R etc. for propositional variables which can have truth value T (true) or F (false).

3.2 Logical Operations

We can join propositions together to get new propositions. We can construct compound propositions by applying logical operations (*connectives*). For example,

Ian Botham played cricket for England and 5>3

is a proposition which is true.

There is a set of logical operations which can be used to construct new propositions from existing ones, and there is a set of rules which determine the truth values of the new propositions. These connectives are called **and** , **or** , and **implies** . There is also an operator called **not**. The standard mathematical notation for these operations is $\wedge$, $\vee$, $\Rightarrow$, and $\neg$ respectively, and the study of compound propositions is known as *propositional calculus.*

We define each operation by stating the truth value of the new proposition in terms of its components. For convenience these values are given as a *truth table.* The truth table shows a value of a compound proposition for all possible values of it components.

NOT, $\neg$ (negation)

For any proposition P, the truth value of $\neg P$ is the opposite of the truth value of P. Thus the truth table for $\neg$ is

P	$\neg$ P
T	F
F	T

So for example

$\neg$(14>6) is false
$\neg$(Obama has been Prime Minister of the UK) is true
$\neg$(Obama has been President of the USA) is false

AND, $\wedge$ (conjunction)

P **and** Q is true if both P and Q are true, and false otherwise.

P	Q	P $\wedge$ Q
T	T	T
T	F	F
F	T	F
F	F	F

So for example

2 is an even prime number, i.e. (2 is even)$\wedge$(2 is prime) is true
(Jupiter is a planet) **and** (a week has 7 days) is true
(9.3 is positive)$\wedge$(9.3 is an integer) is false

OR, $\vee$ (disjunction)

P **or** Q is true if either P or Q is true, and false otherwise.

P	Q	P $\vee$ Q
T	T	T
T	F	T
F	T	T
F	F	F

So for example

$$(14>6)\vee(4 \text{ exactly divides } 8) \qquad \text{is true}$$
$$(\text{The moon is made of green cheese})\vee(\text{Grass is always red}) \qquad \text{is false}$$

In mathematics, ‘or’ always means inclusive: at least one of, but maybe both. There is a special operator for exclusive or called **xor**. We’ll meet this later.

IMPLIES, $\Rightarrow$ (implication)

This connective is equivalent to the programming language statement ‘if P then Q’. P$\Rightarrow$Q says that if P is true then Q must be true. So if P is true and Q is false then P$\Rightarrow$Q is false. The proposition does not say anything about the case when P is false, if P is false then there is no requirement on Q. So P$\Rightarrow$Q is always true when P is false.

P	Q	P $\Rightarrow$Q
T	T	T
T	F	F
F	T	T
F	F	T

So for example

$$(9>5) \Rightarrow(240 \text{ is an even number}) \qquad \text{is true}$$
$$(4 \text{ exactly divides } 5) \Rightarrow(14>6) \qquad \text{is true}$$
$$(4 \text{ exactly divides } 5) \Rightarrow \textbf{not}(14>6) \qquad \text{is true}$$
$$(\text{The moon is made of green cheese}) \Rightarrow (\text{Grass is always red}) \qquad \text{is true.}$$

Note, this is not the same as cause and effect.

LOGICALLY EQUIVALENT, $\Leftrightarrow$ (equal)

This says that P and Q have the same truth value.

P	Q	P $\Leftrightarrow$ Q
T	T	T
T	F	F
F	T	F
F	F	T

$$(N \text{ is an even number}) \Leftrightarrow (N \text{ is divisible by } 2) \qquad \text{is true}$$
$$(N \text{ is an odd number}) \Leftrightarrow (N \text{ is divisible by } 2) \qquad \text{is false}$$

Exercises 3.2 Given propositions
P = logic is fun for Jane
Q = David dislikes cabbage
R = David’s eyes are blue

Express the following as compound propositions:
1. Logic is not fun for Jane and David dislikes cabbage.
2. David's eyes are not blue, nor is logic fun for Jane.
3. Either logic is fun for Jane or David dislikes cabbage and David's eyes are blue.
4. If David's eyes are blue and David dislikes cabbage then logic is not fun for Jane.

Of course, we can build up propositions using several operators. For example,

$$\mathrm{P} \wedge (\mathrm{Q} \Rightarrow \mathrm{R}), \quad \neg(\mathrm{P} \vee (\mathrm{Q} \wedge \mathrm{P}))$$

These operations have precedence, $\neg$ is highest, $\wedge$ and $\vee$ are equal precedence, finally $\Rightarrow$and $\Leftrightarrow$ are of equal and lowest precedence.

We can use truth tables to give the truth values of a complex proposition in terms of its components.

P	Q	R	Q⇒R	P∧(Q⇒R)
T	T	T	T	T
T	T	F	F	F
T	F	T	T	T
T	F	F	T	T
F	T	T	T	F
F	T	F	F	F
F	F	T	T	F
F	F	F	T	F

P	Q	R	¬P	¬P∨ R	(¬P∨ R)∧ Q
T	T	T	F	T	T
T	T	F	F	F	F
T	F	T	F	T	F
T	F	F	F	F	F
F	T	T	T	T	T
F	T	F	T	T	T
F	F	T	T	T	F
F	F	F	T	T	F

Exercises 3.3
Draw the truth tables for
(i)$\neg P \vee Q$, (ii) $(R \wedge \neg Q) \Rightarrow \neg P$, (iii) $P \Rightarrow (Q \wedge \neg P)$, (iv) $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$

In any formal arena, such as engineering, design, computer programming or mathematics, it is important to give a precise definition of the terms being used. We now give a formal definition of what it means for two propositions to be equal.

Definition Two propositions are *equal* if they always have the same truth values. We can use truth tables to prove that propositions are (or are not) equal.

If P and Q have the same truth values, we often say that they are *logically equivalent* and write $P \Leftrightarrow Q$.

For example, it is easy to check that both $\neg(\mathrm{P} \wedge \mathrm{Q})$ and $(\neg\mathrm{P}) \vee (\neg\mathrm{Q})$ have the same truth table

P	Q	P∧Q	¬(P∧Q)	¬P	¬Q	(¬P)∨(¬Q)
T	T	T	F	F	F	F
T	F	F	T	F	T	T
F	T	F	T	T	F	T
F	F	F	T	T	T	T

Thus we have that $\neg(P \wedge Q) \Leftrightarrow (\neg P) \vee (\neg Q)$.

Some propositions are always true. For example,

P	Q	P∨Q	P ⇒(P∨Q)
T	T	T	T
T	F	T	T
F	T	T	T
F	F	F	T

so we see that P⇒(P∨Q) is always true, as is P∨¬P.

P	¬P	P ∨ ¬P
T	F	T
F	T	T

Propositions that are always true are called *tautologies*.

It is also the case that some propositions are always false. For example,

P	¬P	P∧ ¬P
T	F	F
F	T	F

Propositions that are always false are called *contradictions*. Propostions which are neither tautologies or contadictions are *contingent*.

We complete this introduction to propositional logic by returning to the 'exclusive or' operation mentioned above. We expect P **xor** Q to be true if exactly one of P or Q is true. So we expect a truth table of the form

P	Q	P **xor** Q
T	T	F
T	F	T
F	T	T
F	F	F

Considering the truth table for (P ∧ ¬Q) ∨ (¬P ∧ Q), we see that, as we might expect

$$(\mathrm{P} \wedge \neg\mathrm{Q}) \vee (\neg\mathrm{P} \wedge \mathrm{Q}) = \mathrm{P}\ \textbf{xor}\ \mathrm{Q}\ .$$

P	Q	(P ∧ ¬Q)	(¬P ∧ Q)	(P ∧ ¬Q) ∨ (¬P ∧ Q)
T	T	F	F	F
T	F	T	F	T
F	T	F	T	T
F	F	F	F	F

3.3 Normal forms

We can write propositional expressions in uniform ways.

3.3.1 Disjunctive normal form

A formula is said to be in *disjunctive normal form* (DNF) when it is a disjunction ($\vee$) of conjunctions ($\wedge$) of propositional variables or their negations. For example:

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

Every expression built up according to the rules of propositional calculus is equivalent to some formula in disjunctive normal form.

We can construct a DNF (of minterms see below) for a proposition from its truth table.

1. For each row whose truth value is true, write down, for each of the propositional variables P_i in the formula, either P_i if true in row or $\neg P_i$ if false. Then take the conjunction of these expressions.
2. Repeat 1 for each row in the truth table where the formula is true and write down the disjunction of all the conjunctions.

The result is a formula in DNF which is equivalent to the original formula.
(Note: A DNF is not necessarily unique, see the second example below.)

Example P $\wedge$ (Q$\Rightarrow$R)

The truth table for this expression is given above.
For the first row, $P = Q = R = T$ so we have $P \wedge Q \wedge R$.
The second row is false. For the third row we have $P = R = T$, $Q = F$ so we have $P \wedge \neg Q \wedge R$.
For the fourth row we have $R = Q = F$, $P = T$ so we have $P \wedge \neg Q \wedge \neg R$.
The other rows are false, so we have DNF

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

Example $\neg$(R $\vee$ ($\neg$Q$\wedge$ P))

P	Q	R	¬Q	¬Q∧ P	R ∨ (¬Q∧ P)	¬(R ∨ (¬Q∧ P))
T	T	T	F	F	T	F
T	T	F	F	F	F	T
T	F	T	T	T	T	F
T	F	F	T	T	T	F
F	T	T	F	F	T	F
F	T	F	F	F	F	T
F	F	T	T	F	T	F
F	F	F	T	F	F	T

For the second row, P=Q=T, R=F so we have $P \wedge Q \wedge \neg R$.
For the sixth row we have P=R=F, Q=T so we have $\neg P \wedge Q \wedge \neg R$.
For the eigth row we have P=Q=R=F so we have $\neg P \wedge \neg Q \wedge \neg R$.
The other rows are false, so we have DNF

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

Exercises 3.4
Find a DNF for (i) $P \Rightarrow (Q \wedge \neg R)$ (ii) $P \Rightarrow (Q \wedge \neg P)$

3.3.2 Conjunctive normal form

A formula is said to be in *conjunctive normal form* (CNF) when it is a conjunction ($\wedge$) of disjunctions ($\vee$) of propositional variables or their negations. For example:

$$(\neg P \vee Q \vee R \vee \neg S) \wedge (P \vee Q) \wedge \neg S \wedge (Q \vee \neg R \vee S)$$

Every expression built up according to the rules of propositional calculus is equivalent to some formula in conjunctive normal form. To construct a CNF of an expression we use logical equivalences.

3.3.3 Some logical equivalences

Suppose that P, Q and R are propositional expressions. The following logical equivalences can all be proved using truth tables.

Double negation
$\neg(\neg P) \Leftrightarrow P$

Commutative Laws
$P \vee Q \Leftrightarrow Q \vee P$
$P \wedge Q \Leftrightarrow Q \wedge P$

Associative Laws
$(P \vee Q) \vee R \Leftrightarrow P \vee (Q \vee R)$
$(P \wedge Q) \wedge R \Leftrightarrow P \wedge (Q \wedge R)$

Distributive Laws
$P \vee (Q \wedge R) \Leftrightarrow (P \vee Q) \wedge (P \vee R)$ $(Q \wedge R) \vee P \Leftrightarrow (Q \vee P) \wedge (R \vee P)$
$P \wedge (Q \vee R) \Leftrightarrow (P \wedge Q) \vee (P \wedge R)$ $(Q \vee R) \wedge P \Leftrightarrow (Q \wedge P) \vee (R \wedge P)$

De Morgan's Laws
$\neg(P \vee Q) \Leftrightarrow \neg P \wedge \neg Q$
$\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$

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

Excluded middle
$P \vee \neg P \Leftrightarrow T$

Contradiction
$P \wedge \neg P \Leftrightarrow F$

Idempotence
$P \vee P \Leftrightarrow P$
$P \wedge P \Leftrightarrow P$

Identity
$P \wedge T \Leftrightarrow P$
$P \vee F \Leftrightarrow P$

Domination
$P \vee T \Leftrightarrow T$
$P \wedge F \Leftrightarrow F$

3.3.4 Using logical equivalences to obtain a normal form

The general approach is to rewrite an expression to one that is logically equivalent until an expression in CNF, or DNF, is constructed.

The following strategy is likely to be helpful when trying to decide which equivalences to use.

1. Use Implication Laws to eliminate $\Rightarrow$ and $\Leftrightarrow$
2. Use Double Negation and De Morgan to bring $\neg$ immediately before propositional variables
3. Repeatedly use distributive laws (and, optionally, other laws) to obtain a normal form.

Example $Q \Rightarrow (\neg P \wedge (Q \vee R))$

Using logical equivalences, find a DNF and CNF for $Q \Rightarrow (\neg P \wedge (Q \vee R))$.

$$\begin{array}{rcl} Q \Rightarrow (\neg P \wedge (Q \vee R)) & \Leftrightarrow & \neg Q \vee (\neg P \wedge (Q \vee R)) \\ & \Leftrightarrow & \neg Q \vee ((\neg P \wedge Q) \vee (\neg P \wedge R)) \\ & \Leftrightarrow & \neg Q \vee (\neg P \wedge Q) \vee (\neg P \wedge R) \end{array}$$

This gives a DNF. We can apply other equivalences

$$\begin{array}{rcl} Q \Rightarrow (\neg P \wedge (Q \vee R)) & \Leftrightarrow & \neg Q \vee (\neg P \wedge (Q \vee R)) \\ & \Leftrightarrow & (\neg Q \vee \neg P) \wedge (\neg Q \vee (Q \vee R)) \\ & \Leftrightarrow & (\neg Q \vee \neg P) \wedge (\neg Q \vee Q \vee R) \end{array}$$

to get a CNF. In this case there is a simpler CNF, $\neg Q \vee \neg P$ which is also a DNF.

Example $(P \vee Q) \Rightarrow (\neg R \Rightarrow P)$

Using logical equivalences, find a CNF for $(P \vee Q) \Rightarrow (\neg R \Rightarrow P)$.

$$\begin{array}{rcl} (P \vee Q) \Rightarrow (\neg R \Rightarrow P) & \Leftrightarrow & (P \vee Q) \Rightarrow (\neg\neg R \vee P) \\ & \Leftrightarrow & (P \vee Q) \Rightarrow (R \vee P) \\ & \Leftrightarrow & \neg(P \vee Q) \vee (R \vee P) \\ & \Leftrightarrow & (\neg P \wedge \neg Q) \vee (R \vee P) \\ & \Leftrightarrow & (\neg P \vee (R \vee P)) \wedge (\neg Q \vee (R \vee P)) \\ & \Leftrightarrow & (\neg P \vee R \vee P) \wedge (\neg Q \vee R \vee P) \end{array}$$

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

Using logical equivalences, find DNF and CNF of $\neg(P \Rightarrow Q) \vee (\neg P \vee \neg Q)$.

$$\begin{array}{rcll} \neg(P \Rightarrow Q) \vee (\neg P \vee \neg Q) & \Leftrightarrow & \neg(\neg P \vee Q) \vee (\neg P \vee \neg Q) & \\ & \Leftrightarrow & (\neg\neg P \wedge \neg Q) \vee (\neg P \vee \neg Q) & \\ & \Leftrightarrow & (P \wedge \neg Q) \vee \neg P \vee \neg Q & (DNF) \\ & \Leftrightarrow & (P \wedge \neg Q) \vee \neg P \vee \neg Q & \\ & \Leftrightarrow & (P \vee \neg P \vee \neg Q) \wedge (\neg Q \vee \neg P \vee \neg Q) & (CNF) \end{array}$$

Sometimes these more general distributive laws are useful:

$$\begin{array}{rcl} (A \wedge B) \vee (C \wedge D) & \Leftrightarrow & (A \vee C) \wedge (A \vee D) \wedge (B \vee C) \wedge (B \vee D) \\ (A \vee B) \wedge (C \vee D) & \Leftrightarrow & (A \wedge C) \vee (A \wedge D) \vee (B \wedge C) \vee (B \wedge D) \end{array}$$

Exercise 3.5
Find a DNF and CNF for $P \Rightarrow (Q \wedge \neg R)$.

Exercise 3.6
Using logical equivalences, show that the expression
$(\neg Q \wedge (P \Rightarrow Q)) \Rightarrow \neg P$ is a tautology

If you are not sure whether your answer is correct you can always check by writing out the truth tables.

Don't forget, any disjunction ($\vee$) of propositional variables or their negations is both a DNF and a CNF, and any conjunction ($\wedge$) of propositional variables or their negations is both a DNF and a CNF.

3.4 Practice exercises

1. Assume that P and Q are true propositions, and that R and S are false propositions. Determine the truth values of
(a) $P \wedge Q$ (b) $P \vee S$ (c) $P \Rightarrow Q$ (d) $R \Rightarrow \neg S$.

2. Construct the truth table for $P \vee \neg Q$, and use it to determine the value of $P \vee \neg Q$ when P and Q are both false.

3. Let P be the proposition 'my umbrella is at home' and let Q be the proposition 'it will rain today'. Describe each of the following propositions in words:
(a) P **and** Q (b) P **or** Q (c) $P \Rightarrow Q$.

4. Identify the propositions in the following sentences, and rewrite the sentences using mathematical notation.
   a. Mary is John's mother and Michael is six.
   b. Mary is not John's mother or Michael is six.

5. Let P, Q, R be the following propositions
   P: John is 5
   Q: Michael is 6
   R: Mary is John's mother
Write each of the following sentences in mathematical notation, using P,Q,R and the logical operators.
   a. John is 5, and Mary is not John's mother.
   b. If Michael is 6, then John is 5.
   c. If John is not 5, then it is false that Michael is 6.
   d. It is false that Mary is John's mother, and that Michael is 6.
   e. Mary is not John's mother or Michael is not 6.

6. Construct truth tables for the following propositions.

a. $\neg(P \wedge Q)$	c. $(P \wedge Q) \wedge R$	e. $(P \wedge \neg Q) \Rightarrow F$
b. $\neg(P \vee (Q \wedge P))$	d. $P \wedge (Q \wedge R)$	f. $R \Rightarrow (\neg P \Rightarrow Q)$

7. Use truth tables to show that $(((P \wedge Q) \vee \neg P) \vee \neg Q)$ is a tautology.

8. Use logical equivalences to show that $(((P \wedge Q) \vee \neg P) \vee \neg Q)$ is a tautology.

9. Find a DNF and a CNF for $(((P \wedge (Q \Rightarrow R)) \wedge \neg R) \vee \neg(Q \wedge \neg P))$.