Goldsmiths
UNIVERSITY
OF LONDON

BSc and Diploma in **Computing and Information Systems**

Mathematics for computing volume 1

C.A. Whitehead

2004

2910102

This guide was prepared for the University of London by:

C.A. Whitehead, BA, MPhil, FTICA, Department of Mathematical and Computing Sciences, Goldsmith's College, University of London.

This is one of a series of subject guides published by the University. We regret that due to pressure of work the author is unable to enter into any correspondence relating to, or arising from, the guide. If you have any comments on this subject guide, favourable or unfavourable, please use the form at the back of this guide.

The External Programme
Publications Office
University of London
34 Tavistock Square
London WC1H 9EZ
United Kingdom
Web site: www.londonexternal.ac.uk

Published by: University of London Press

Printed by: Central Printing Service, University of London, England

Contents

Introduction

The aims and objectives of the unit

Computer science depends upon the science of mathematics and without the mathematical ideas that underpin it, none of the marvels of modern computer technology would be possible. In order to study for a degree in Computing and Information Systems, you need to understand and feel easy with some essential mathematical ideas. The topics in this course have been selected with that in mind and you will find that you use many of the ideas and skills introduced in this unit directly or indirectly in the other units in this degree programme. You will also gain experience of the way that mathematicians and computer scientists express their ideas using symbols and make their statements precise.

Although this unit does not go very deeply into any one topic, we hope that you will gain sufficient confidence from studying it to consult mathematical textbooks to pursue areas that particularly interest you or about which you need more information. Currently, there are two optional half units available at Level 3 of this degree programme that develop aspects of this unit directly.

Using this subject guide effectively

The subject guide for 2910102 [CIS102] is in two volumes, and the material is presented in a convenient order of study. Taken together, the two volumes contain a complete account of the examinable topics in the unit. The chapters are not all of equal length and the time you should allow for studying them will depend very much on your previous mathematical experience.

It is very important to understand that you only learn mathematics by *doing* it. Ideas and methods that may seem complicated at first sight will become more familiar and natural with practice. You will then be able to apply the ideas you learn in this course to your other units with more facility, and will find the exam questions easier to answer. The exercises at the ends of each chapter are therefore a crucially important part of the course and we strongly recommend you to try all of them. Answers to the exercises in this volume are included as an Appendix. You can get extra practice, particularly on topics that you find difficult, by trying additional questions from an appropriate textbook.

Textbooks

Although this subject guide gives a complete account of the course, we do encourage you to consult a textbook for alternative explanations, extra examples and exercises, and supplementary material. There are a number of books on Discrete Mathematics available. Many of them cover most (but not necessarily all) of the topics in the the syllabus. They vary in style and the level of mathematical maturity and expertise they presuppose on the part of the reader. We recommend that you obtain a copy of one of the following titles. References are given to them where appropriate in the subject guide.

Either

Susanna S. Epp, *Discrete Mathematics with Applications.* 2nd Edition (Brooks/Cole 1995) [ISBN 0-534-94446-9(hbk)]

or

Molluzzo, J. C. and Buckley, F. *A First Course in Discrete Mathematics.* (Waveland Press, reprinted 1997) [ISBN 0-88133-940-7]. Referred to as M&B throughout the text. Available in the USA only.

A list of other suitable textbooks known to the author of this guide and in print at the time of writing is included in this volume as an Appendix.

Discrete Mathematics is a comparatively modern area of study and the notation in some topics has not been completely standardised. This means that you may find occasional differences between the symbols and terminology used in textbooks by different authors. The examination papers will follow the usage in this subject guide, however.

Assessment

Important: the information and advice given in the following section is based on the examination structure used at the time this guide was written. However, the university can alter the format, style or requirements of an examination paper without notice. Because of this, we strongly advise you to check the rubric/instructions on the paper when you actually sit the examination. You should also read the examiner's report from the previous year for advice on this.

This unit is examined by a three hour written paper. Currently, ten questions are set and full marks for the paper are awarded for complete answers to *all* of them. Each question carries the same number of marks. In general, there is at least one question on the material in each chapter of the guide, but some questions may require knowledge or techniques from more than one chapter.

You may answer the questions in any order. If you feel nervous, it may help you to start with a question on a topic you feel really confident about. For the most part, the questions are very similar to worked examples or exercises in the subject guide, so that any student who has understood the material and revised thoroughly for the exam should be able to answer most of them quite easily. Some parts of questions may contain a new twist that requires more careful thought. If you get stuck on part of a question, do not spend *too* long trying to figure it out, because this may mean that you do not have time to attempt another question that you could answer quite easily. Leave a space in your script and come back to the bit you found difficult when you have attempted as much as you can of the rest of the paper. Some questions may also ask you for a definition or the statment of a result proved in the subject guide. These need to be carefully learnt. You may express a definition in your own words, but make sure that your words cover all the points in the definition in this guide. A detailed marking scheme is given on the paper and that can be used as a guide to the length of answer expected. If there is just 1 or 2 marks awarded for a definition, for example, only one sentence is expected, not a half page discussion.

Although most students find that attempting past examination papers is reassuring, you are strongly advised not to spend *all* your revision time in this way, because every year the questions will be different! It is much better to revise thoroughly the ideas and skills taught in each chapter until you are quite confident that you really *understand* the material. When you have finished your revision, test yourself by answering the specimen examination questions to time. Suggested solutions are also included, so that you can check your answers and see the level of detail required.

In conclusion, we hope you will enjoy studying this unit and find the material interesting and challenging, as well as increasing your understanding and appreciation of your other units.

Chapter 1

Numbers Systems

Summary

Number bases: the decimal, binary and hexadecimal number systems; calculating in the binary and hexadecimal systems; converting between number bases; factors, multiples, primes; rational numbers and their decimal representation; irrational numbers; inequalities; floating point notation.

References: Epp Sections 1.5 (pp 57-61, 70-73), 3.2 (pp 125-127) *or* M&B Sections 1.1, 1.2, 1.4.

Introduction

The history of mathematics began thousands of years ago with numbers and counting. As well as the practical importance of numbers in everyday life over the intervening centuries, the properties of numbers have fascinated mathematicians and laymen alike and been a focus for the development of deep mathematical theories. Today, numbers are more important in our lives than ever before, with information in every field, from pin numbers and barcodes to music and television transmissions recorded digitally. In this chapter, we consider various ways in which numbers can be represented.

1.1 Number Bases

Learning Objectives

After studying this section, you should be able to:

- convert a numeral given in any base to a decimal numeral;
- demonstrate understanding of the binary number system by performing simple arithmetical calculations in the binary system;
- convert a binary numeral to hexadecimal and vice-versa;
- express a decimal numeral in the binary and hexadecimal number systems.

1.1.1 Numbers in base 10

Let us start by reminding ourselves how a string of digits is used to represent a whole number in our familiar number system. Consider the number 73589. We can think of each digit as appearing in a labelled box, where each box has a **place value**, according to its position, as illustrated below.

$\cdots$	10^4	10^3	10^2	10^1	1
$\cdots$	7	3	5	8	9

The convention that the *place value* of each digit determines how much it contributes to the value of the number allows us to represent any whole number, however large, using only the ten different

digits $0, 1, 2, ..., 9$. The name *decimal*, coming from the Latin word for *ten*, reflects the fact that every place value in our number system is a power of 10. Note that the right hand *units* box also has a place value which is a power of 10, since $1 = 10^0$, but we shall represent this place value by 1 for convenience.

Definition 1.1 *Because each place value in the decimal system is a power of 10, we say that the decimal system has* **base** 10.

For the moment, we restrict our discussion to the *integers*, that is, whole numbers. We shall see later how the system of place value can be extended to fractional numbers by using *negative* powers of the base.

Instead of using boxes, we can write out the number 73589, for example, in **expanded notation**, as

$$7(10^4) + 3(10^3) + 5(10^2) + 8(10^1) + 9(1).$$

Example 1.1 Suppose that $a_3a_2a_1a_0$ represents a decimal number, where the symbols a_3, a_2, a_1, a_0 each represent one of the digits $0, 1, 2, \ldots, 9$ (and are not necessarily different from one another). Then we can write $a_3a_2a_1a_0$ in expanded notation in base 10, as

$$a_3(10^3) + a_2(10^2) + a_1(10^1) + a_0(1) \quad \square$$

Numbers in base 5

The decimal number system seems so natural and familiar that it is hard to realise that its use is entirely fortuitous, due presumably to the accident of our having ten fingers to count on. Other peoples and civilizations have based their number systems on other integers, such as 5, 12 or 20. The Babylonians based theirs on 60 and it is to them that we owe the division of both an hour and a degree of turn into 60 minutes and each minute into 60 seconds. There are vestiges of counting systems based on 12 and 20 in the old British systems of weights and measurement and in the special words *dozen* for 12, *gross* for 144 (or 12^2) and *score* for 20 that have remained in the language.

Let us consider first how numbers are represented in a number system based on 5 instead of 10. The digits are just $0, 1, 2, 3, 4$ and these are the *only* symbols we are allowed to use to record a number. The place value of each digit is a power of 5, so that we count in units, 5's, 25's and so on.

A natural way to think of the system in base 5 is that we are using the fingers of one hand only to count with, and 5 units makes a "hand". You may also have met a system of counting in 5's to keep a *tally* when making a count for a statistical survey. Thus to represent the decimal number 13, we would need 2 "hands" + 3 "fingers", which we record as $(23)_5$ in base 5. Working in this way, we obtain the table below, which records the base 5 representation of the numbers from 1 to 16. Note that when we write a number in a base other than 10, we enclose it in brackets and append the base as a suffix. Unless otherwise stated, numbers with no suffix are in base 10.

base 10	*base* 5	*base* 10	*base* 5
1	$(1)_5$	9	$(14)_5$
2	$(2)_5$	10	$(20)_5$
3	$(3)_5$	11	$(21)_5$
4	$(4)_5$	12	$(22)_5$
5	$(10)_5$	13	$(23)_5$
6	$(11)_5$	14	$(24)_5$
7	$(12)_5$	15	$(30)_5$
8	$(13)_5$	16	$(31)_5$

To convert a number from base 5 to decimal, we can use the box system.

Example 1.2 To convert the base 5 number $(3214)_5$ to decimal, we enter the digits in the boxes as shown below.

5^3	5^2	5^1	1
3	2	1	4

Thus, using expanded notation, $(3214)_5$ represents

$$3(5^3) + 2(5^2) + 1(5) + 4(1) = 434. \square$$

1.1.2 The binary system

The electronic circuitry of a computer easily lends itself to the representation of numbers using a two digit number system. This is because a digital computer is really just a gigantic collection of on/off switches, with an intricate set of connections between them. Thus it is natural to represent the two positions of a switch by the digits 0 and 1.

Definition 1.2 *The number system with base 2, having just the two digits 0 and 1, is called the* **binary** *number system. Each binary digit is called a* **bit** *(short for* **b**inary dig**it**).

A string of bits, like 10001011, for example, is called a **binary string**. The number of bits in the string is called the **length** of the string. A binary string of length 8 is called a **byte**. The computer uses bytes as the basic unit of information. We shall see later that since there are two choices, 0 or 1, for each of the eight bits in a byte, there are altogether $2^8 = 256$ different bytes.

We shall now consider how to represent numbers in the binary system. The box system in binary giving the place value of each bit is shown below.

$\cdots$	2^5	2^4	2^3	2^2	2^1	1

The following table shows the binary equivalent of each of the decimal numbers from 1 to 16.

base 10	*base* 2	*base* 10	*base* 2
1	$(1)_2$	9	$(1001)_2$
2	$(10)_2$	10	$(1010)_2$
3	$(11)_2$	11	$(1011)_2$
4	$(100)_2$	12	$(1100)_2$
5	$(101)_2$	13	$(1101)_2$
6	$(110)_2$	14	$(1110)_2$
7	$(111)_2$	15	$(1111)_2$
8	$(1000)_2$	16	$(10000)_2$

Example 1.3 To convert the binary integer $(1001011)_2$ to the decimal system, we write it in expanded notation. You may find it helpful to put it into boxes first, as shown below.

2^6	2^5	2^4	2^3	2^2	2^1	1
1	0	0	1	0	1	1

Thus we have

$$\begin{aligned}(1001011)_2 &= 1(2^6) + 0(2^5) + 0(2^4) + 1(2^3) + 0(2^2) + 1(2^1) + 1(1) \\ &= 2^6 + 2^3 + 2^1 + 1 = 75. \square\end{aligned}$$

In practice, when finding the decimal equivalent of a binary number, it is easiest to work from *right to left*. Thus in the previous example, we would start from the units and compute successively:

$$\begin{aligned}1 + 1(2) &= 3 \\ &+ 0(4) = 3 \\ &\quad + 1(8) = 11 \\ &\quad\quad + 0(16) = 11 \\ &\quad\quad\quad + 0(32) = 11 \\ &\quad\quad\quad\quad + 1(64) = 75.\end{aligned}$$

With a little practice, you will be able to do this directly from the binary form of the number without using expanded notation.

Example 1.4 Consider the binary number $(a_4a_3a_2a_1a_0)_2$, where each of the symbols a_i represents a bit (0 or 1). We can write this in expanded notation as

$$a_4(2^4) + a_3(2^3) + a_2(2^2) + a_1(2^1) + a_0(1). \square$$

1.1.3 Calculating in the binary system

In this section we shall drop the suffices on the binary numbers *in the body of the calculations*. Since these will only involve numbers in base 2, no confusion will arise.

Binary addition

First, recall that when we add a column of digits in the decimal system that sum to a number greater than 9, we record the units in the column we are working in and carry the number of 10's into the next column to the left. When a column sums to more than 99, we carry the number of hundreds into the column two places to the left, and the number of tens into the column next to the left.

We procede in a very similar way in binary. To obtain the rules for carrying, suppose the column to be added is as follows (with any zeros omitted):

1. $(1)_2 + (1)_2$. This gives 2, or $(10)_2$ in binary. We record the 0 in the current column and carry a 1 to the next column to the left;
2. $(1)_2 + (1)_2 + (1)_2$, giving $(11)_2$. We record 1 in the current column and carry a 1 to the next column to the left;
3. $(1)_2 + (1)_2 + (1)_2 + (1)_2$, giving $(100)_2$. We record 0 in the current column and carry a 1 to the column *two places* to the left;

and so on.

Example 1.5 We sum the binary numbers $(1011)_2 + (1111)_2 + (11)_2$. In the format below, the numbers we carry at each step are recorded above the horizontal line.

Step 1

$$\begin{array}{cccc} & & 1 & \\ \hline 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ & & 1 & 1 \\ \hline\hline & & & 1 \end{array}$$

Adding the right hand column, we obtain $(11)_2$, and so we record 1 and carry 1 into the next column (that is, the 2^1 column).

Step 2

$$\begin{array}{cccc} 1 & & 1 & \\ \hline 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ & & 1 & 1 \\ \hline\hline & & 0 & 1 \end{array}$$

Adding the 2^1 column, we obtain $(100)_2$, and so we record 0 and carry 1 into the column two places to the left (that is, the 2^3 column).

Step 3

$$\begin{array}{ccccc} 1 & 1 & & 1 & \\ \hline & 1 & 0 & 1 & 1 \\ & 1 & 1 & 1 & 1 \\ & & & 1 & 1 \\ \hline\hline 1 & 1 & 1 & 0 & 1 \end{array}$$

Adding the 2^2 column, we obtain $(1)_2$ and so this time there is nothing to carry. Adding the 2^3 column, we obtain $(11)_2$ and so we record 1 and carry 1 into the 2^4 column. Adding the 2^4 column, we obtain $(1)_2$ and the process terminates.

Thus we obtain $(1011)_2+(1111)_2+(11)_2 = (11101)_2$. In decimal numbers, this addition is 11+15+3, and the sum we obtained is 29. □

Binary subtraction

Let us first revise how we deal with "borrowing" in base 10.

Example 1.6 We perform the subtraction $4003 - 597$.

3	9	9	13
4	0	0	3
	5	9	7
3	4	0	6

Since 7 is greater than 3, we must borrow a 10. In this case, we do this from the 10^3 column. We rewrite the borrowed 1000 as $990 + 10$. So we reduce the entry in the 10^3 column from 4 to 3, replace the zeros by 9's in the next two columns and replace 3 by $10 + 3$ in the units column. The resulting row is recorded above the horizontal line. We can now obtain the answer by subtracting 597 from the row above the horizontal line. □

We shall now carry out the same process in binary, where we have to borrow a 2 when we have to perform the subtraction $(0)_2 - (1)_2$ in any column. The $(0)_2$ then becomes $(10)_2$, and we have

$$(10)_2 - (1)_2 = (1)_2.$$

This process is illustrated in the example below.

Example 1.7 We subtract $(1101000)_2 - (101011)_2$.

Step 1

1	1	0	0	1	1	10
1	1	0	1	0	0	0
	1	0	1	0	1	1

As in the previous example, we have to borrow to perform the subtraction in the units column. The first column with a non-zero entry is the 2^3 column. We borrow $(1000)_2$ from this column and rewrite it as $(110)_2 + (10)_2$ (effectively, we are borrowing an 8 and rewriting it as $6 + 2$). The revised row is recorded above the horizontal line.

Step 2 We replace the row representing the number we are subtracting *from* by the row above the horizontal line at the end of Step 1. We can now perform the subtraction in the first three columns as shown below. When we reach the 2^3 column, we have to borrow again. We repeat the procedure described in Step 1. The revised top row is shown above the horizontal line.

1	0	1	10			
1	1	0	0	1	1	10
	1	0	1	0	1	1
				1	0	1

Step 3 We replace the row representing the number we are subtracting *from* by the row above the horizontal line and perform the subtraction in the 2^3 and 2^4 columns. When we reach the 2^5 column, we have to borrow from the next column. The revised top row is shown above the horizontal line.

0	10					
1	0	1	10	1	1	10
	1	0	1	0	1	1
		1	1	1	0	1

Step 4 We replace the row representing the number we are subtracting *from* by the row above the horizontal line and complete the subtraction.

$$\begin{array}{cccccc} 10 & 1 & 10 & 1 & 1 & 10 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 0 & 1 \end{array}$$

Thus we have obtained $(1101000)^2 - (101011)^2 = (111101)_2$. In decimal numbers, this subtraction is $104 - 43$ and the answer we obtained is 61. □

Binary multiplication

The process of binary multiplication is very similar to long multiplication in the decimal system. A familiar feature in decimal is that when we multiply any whole number by 10, we move all the digits one place to the left and enter a zero in the units column. Let us pause a moment to see why this rule works.

Example 1.8 Consider 743×10. Writing 743 in expanded notation, we have

$$743 = 7(10^2) + 4(10^1) + 3(1).$$

Multiplying each term in this equation by 10, we obtain

$$\begin{aligned} 743 \times 10 &= 7(10 \times 10^2) + 4(10 \times 10^1) + 3(10 \times 1) \\ &= 7(10^3) + 4(10^2) + 3(10) \\ &= 7(10^3) + 4(10^2) + 3(10^1) + 0(1). \end{aligned}$$

Thus the *place value* of each digit has been multiplied by 10 and the the entry in the units column is now 0. □

A very similar thing happens when we multiply a binary number by the base number $2 = (10)_2$. As we have seen in Example 1.4, the binary number $(a_4a_3a_2a_1a_0)_2$ can be written in expanded notation as

$$a_4(2^4) + a_3(2^3) + a_2(2^2) + a_1(2^1) + a_0(1).$$

Multiplying this number by 2, we obtain

$$\begin{aligned} & a_4(2 \times 2^4) + a_3(2 \times 2^3) + a_2(2 \times 2^2) + a_1(2 \times 2^1) + a_0(2 \times 1) \\ = \; & a_4(2^5) + a_3(2^4) + a_2(2^3) + a_1(2^2) + a_0(2) \\ = \; & a_4(2^5) + a_3(2^4) + a_2(2^3) + a_1(2^2) + a_0(2^1) + 0(1). \end{aligned}$$

Thus the place value of each bit has been multiplied by 2 and so the bits all move one place to the left and the entry in the units column is 0. Thus we have the following rule:

Rule 1.3 *To multiply by $(10)_2$ in binary, we move each bit one place to the left and enter 0 in the 2^0 column.*

Example 1.9 Multiplying $(1101)_2 \times (10)_2$, the rule gives $(11010)_2$. Checking in decimal, we have calculated 13×2 and obtained 26. □

We can extend the rule for multiplying by 2 to multiplying by any power of 2. For example, since multiplying by 4 is the same as multiplying by 2 and then multiplying the result by 2 again, we move each bit *two* places to the left and enter zeros in both the 2^1 and the 2^0 columns. Thus the rule for multiplying by $4 = (100)_2$ in binary is exactly the same as the rule for multiplying by 100 in decimal. Similarly, the rule for multiplying by $8 = (1000)_2$ in binary is the same as the rule for multiplying by 1000 in decimal, and so on.

Example 1.10 We multiply $(1011)_2 \times (101)_2$. Note first that $(101)_2 = (100)_2 + (1)_2$. So we

perform two multiplications and add them together.

		1	0	1	1	
			1	0	1	
1	0	1	1	0	0	$(1011)_2 \times (100)_2$
		1	0	1	1	$(1011)_2 \times (1)_2$
	1					*bits carried forward in the sum*
1	1	0	1	1	1	$(1011)_2 \times (101)_2$

Thus we have $(1011)_2 \times (101)_2 = (110111)_2$. We leave you to check this is correct by finding the decimal equivalent. □

Binary division

We first note that dividing by 2 is just the reverse of the process of multiplying by 2 and so we have the following rule.

Rule 1.4 *To divide by $(10)_2$ in binary, we move each bit one place to the right and the entry in the units column becomes the remainder.*

Example 1.11 $(110111)_2 \div (10)_2 = (11011)_2$, remainder $(1)_2$. □

The number we are dividing by is called the **divisor** and the number of complete times it divides is called the **quotient**. In the example above, the divisor is $(10)_2$ and the quotient is $(11011)_2$.

We can easily extend Rule 1.4 to powers of 2, as illustrated in the following example.

Example 1.12 $(110111)_2 \div (1000)_2 = (110)_2$, remainder $(111)_2$. □

Note that to retrieve the number we are dividing *into*, we reverse the process by multiplying the quotient by the divisor and adding the remainder. Thus the reverse of the process in Example 1.11 is

$$(110111)_2 = (11011)_2 \times (10)_2 + (1)_2.$$

The process of long division in the binary system is very similar to long division in the decimal system, but simplified by the fact that when we divide a binary number by a divisor with the *same* number of bits, the quotient is either $(1)_2$ or $(0)_2$. In the latter case, when we bring down the next bit, we will always obtain a quotient of $(1)_2$. The process is illustrated in the following example.

Example 1.13 We calculate $(101001)_2 \div (11)_2$.

Step 1 We try dividing $(11)_2$ into the first two bits. Since $(11)_2$ is greater than $(10)_2$, the quotient is 0_2 and we bring down the next bit, so that we are now dividing $(11)_2$ into $(101)_2$. The quotient this time is $(1)_2$, which we record in the quotient line above the third bit. We now subtract $(11)_2 \times (1)_2 = (11)_2$ from the first three bits to obtain the remainder $(10)_2$.

			1			
11 ⟌	1	0	1	0	0	1
		1	1			
		1	0			

Step 2 We bring down the next bit. We are now dividing a 2-bit number into a 3-bit number, so the quotient must be $(1)_2$. We repeat the process described in Step 1.

			1	1		
11 ⟌	1	0	1	0	0	1
		1	1			
		1	0	0		
			1	1		
				1		

Step 3 We bring down the next bit. Then since $(11)_2$ is greater than $(10)_2$, we record (0_2) in the quotient row and bring down the last bit. The quotient must be $(1)_2$ this time, so we record $(1)_2$ above the last bit in the quotient row and subtract $(11)_2$ from $(101)_2$ to give the remainder.

$$\begin{array}{rcccccc} & & & 1 & 1 & 0 & 1 \\ \cline{3-7} 11 \,\big\lceil & 1 & 0 & 1 & 0 & 0 & 1 \\ & & 1 & 1 & & & \\ \cline{3-4} & & 1 & 0 & 0 & & \\ & & & 1 & 1 & & \\ \cline{4-5} & & & & 1 & 0 & 1 \\ & & & & & 1 & 1 \\ \cline{5-6} & & & & & 1 & 0 \end{array}$$

Thus we have obtained $(101001)_2 \div (11)_2 = (1101)_2$ with remainder $(10)_2$. Checking in decimal, we have divided 41 by 3 to obtain 13 with remainder 2. □

1.1.4 The hexadecimal number system

Another number system commonly used in computer science is the **hexadecimal** system based on 16. This system has the following advantages: it enables most numbers to be recorded using substantially fewer digits than are necessary in base 2; there is a very easy method of converting between hexadecimal and binary, which as we have seen is a "natural" language for computers to use.

In order to express a number in the hexadecimal system we need 16 digits. It is customary to use the digits $0, 1, \ldots, 9$ followed by the letters A,B,C,D,E, and F as symbols to represent the numbers $10, 11, 12, 13, 14$ and 15 respectively. The hexadecimal digits are known as **hexits** and the hexadecimal system is often called **hex**, for short.

We convert from hex to decimal in the usual way, using expanded notation in hexits.

Example 1.14 We first illustrate the number $(5AE7)_{16}$ using hexadecimal boxes.

16^3	16^2	16^1	1
5	A	E	7

Writing it in expanded notation, we have

$$\begin{aligned} (5AE7)_{16} &= 5(16^3) + 10(16^2) + 14(16^1) + 7(1) \\ &= 23271. \square \end{aligned}$$

Arithmetic in hex

Hexadecimal arithmetic is very similar to decimal arithmetic, except that we have 16 hexits and carry, or borrow, 16's instead of 10's. We work two illustrative examples here.

Example 1.15 We add $(9D7)_{16} + (4BA)_{16}$. Hexits carried into the next column are entered in the row above the horizontal line.

$$\begin{array}{ccc} 1 & 1 & \\ \hline 9 & D & 7 \\ 4 & B & A \\ \hline\hline E & 9 & 1 \end{array}$$

Thus $(9D7)_{16} + (4BA)_{16} = (E91)_{16}$. The decimal equivalent is $2519 + 1210 = 3729$. □

Example 1.16 We subtract $(F04)_{16} - (CD9)_{16}$. The revised row when we have borrowed a 16 is shown in the row above the horizontal line. Note that all the numbers in the table are in hex, so that the "14" in the revised row is $(14)_{16}$, representing the decimal number 20.

$$\begin{array}{ccc} E & F & 14 \\ \hline F & 0 & 4 \\ C & D & 9 \\ \hline\hline 2 & 2 & B \end{array}$$

Thus $(F04)_{16} - (CD9)_{16} = (22B)_{16}$. The decimal equivalent is $3844 - 3289 = 555$. □

Converting between hex and binary

There is a very easy procedure for converting between the binary number and hexadecimal number systems. First, we need a table giving the binary equivalents of each hexit. You will see that we have expressed each binary number as a 4-bit string, by adding zeros where necessary at the *front* of each binary number to make up the total number of bits to four.

hexadecimal	$(0)_{16}$	$(1)_{16}$	$(2)_{16}$	$(3)_{16}$	$(4)_{16}$	$(5)_{16}$	$(6)_{16}$	$(7)_{16}$
4-bit binary	$(0000)_2$	$(0001)_2$	$(0010)_2$	$(0011)_2$	$(0100)_2$	$(0101)_2$	$(0110)_2$	$(0111)_2$

hexadecimal	$(8)_{16}$	$(9)_{16}$	$(A)_{16}$	$(B)_{16}$	$(C)_{16}$	$(D)_{16}$	$(E)_{16}$	$(F)_{16}$
4-bit binary	$(1000)_2$	$(1001)_2$	$(1010)_2$	$(1011)_2$	$(1100)_2$	$(1101)_2$	$(1110)_2$	$(1111)_2$

To convert a number in binary into hex, we first split the binary string into blocks of four bits, *starting from the bit in the units column and working to the left.* If the total number of bits is not divisible by 4, then we add the necessary number of zeros to the *front* of the number to complete the last block on the left. Each block of four bits now corresponds to a hexit, as shown in the table above. We simply replace each 4-bit block by the corresponding hexit. To convert from hex to binary, we reverse this process and replace each hexit by a 4-bit binary string representing the corresponding binary number.

Example 1.17 We write the binary number $(1101101011)_2$ in hex. Splitting the binary string into blocks of four bits, working from the units column towards the left, we have

$$1101101011 = 11\ \ 0110\ \ 1011.$$

The first block is incomplete, so we add two zeros on the front to complete it. This gives

$$1101101011 = 0011\ \ 0110\ \ 1011.$$

Finally, replacing each block of four bits by the corresponding hexit, we have

$$(1101101011)_2 = (36B)_{16}.\ \square$$

Example 1.18 We express $(B35)_{16}$ in binary. Replacing each hexit by the corresponding 4-bit binary string, we obtain

$$1011\ \ 0011\ \ 0101$$

Thus we have

$$(B35)_{16} = (101100110101)_2.\ \square$$

1.1.5 Converting decimal integers into other bases

Consider the binary number $(a_4a_3a_2a_1a_0)_2$, where each of the symbols $a_0, a_1, \ldots$ represents a bit (0 or 1). By Rule 1.4, when we divide a binary number by $2 = (10)_2$, all the bits move one place to the right and the bit in the units column becomes the remainder. Thus

$$(a_4a_3a_2a_1a_0)_2 \div (10)_2 = (a_4a_3a_2a_1)_2,\ \textit{remainder}\ a_0.$$

So, to write a decimal number in binary, we start by dividing the decimal number by 2 and the remainder, 0 or 1, gives the value of the units bit a_0.

We now repeat this process with the quotient in place of the original number to find the value of the bit a_1. Thus

$$(a_4a_3a_2a_1)_2 \div (10)_2 = (a_4a_3a_2)_2,\ \textit{remainder}\ a_1,$$

so that the remainder at this second division by 2 gives the value of a_1. We continue this process until we have found the value of each bit.

Example 1.19 We express the decimal integer 25 in binary.

$$\begin{aligned} 25 \div 2 &= 12, \quad \textit{remainder } 1 \\ 12 \div 2 &= 6, \quad \textit{remainder } 0 \\ 6 \div 2 &= 3, \quad \textit{remainder } 0 \\ 3 \div 2 &= 1, \quad \textit{remainder } 1 \\ 1 \div 2 &= 0, \quad \textit{remainder } 1. \end{aligned}$$

Now the first remainder gives us the value of the bit in the units column, the second remainder gives us the value of the bit in the 2^1 column, the next remainder gives the bit in the 2^2 column, and so on. Thus, to find the binary representation of 25, we must read off the remainders in REVERSE ORDER. This gives

$$25 = (11001)_2. \square$$

We can use this method to convert a decimal number to any base.

Example 1.20 We convert 471 to base 5.

$$\begin{aligned} 471 \div 5 &= 94, \quad \textit{remainder } 1 \\ 94 \div 5 &= 18, \quad \textit{remainder } 4 \\ 18 \div 5 &= 3, \quad \textit{remainder } 3 \\ 3 \div 5 &= 0, \quad \textit{remainder } 3 \end{aligned}$$

Recording the remainders in reverse order, gives

$$471 = (3341)_5. \square$$

Example 1.21 We convert 2963 to hexadecimal.

$$\begin{aligned} 2963 \div 16 &= 185, \quad \textit{remainder } 3 \\ 185 \div 16 &= 11, \quad \textit{remainder } 9 \\ 11 \div 16 &= 0, \quad \textit{remainder } 11 \end{aligned}$$

Recording the remainders in reverse order, gives

$$2963 = (\mathrm{B93})_{16}. \square$$

1.2 Rational numbers

Learning Objectives

After studying this section, you should be able to:

- factorize a given decimal integer into its prime factors and use power notation;
- say what is meant by a *rational number* and express a recurring decimal as a fraction;
- interpret fractions in bases other than 10.

Introduction

Thus far, we have considered only the representation of positive integers in various number bases. In this section, we consider division of positive integers and the representation of fractions.

1.2.1 Factors, multiples and primes

Definition 1.5 *Suppose that m and n are positive integers. If n divides into m exactly, so that the quotient m/n is also an integer, then we say that* **n divides m**. *We also say that* **n is a factor** *(or a* **divisor***)* **of m**, *and that* **m is a multiple of n**.

Thus, for example, we can say that "4 divides 20", "4 is a factor of 20", "20 is a multiple of 4". These three sentences all have the same meaning.

Definition 1.6 *A* **prime number** *(or more simply, a* **prime***) is an integer greater than 1 whose only positive factors are itself and 1.*

Notice carefully that the number 1 is *not* a prime. It is called a **unit** and has the property that it divides *every* integer. A number greater than 1 that is not prime is called **composite**. The smallest primes are thus $2, 3, 5, 7, 11, 13, 17, \ldots$, whereas the numbers $4, 6, 8, 9, 10, 12, 14, 15, 16, \ldots$ are composite.

Example 1.22 The composite number 30 can be written as a product of two positive integers greater than 1 in several different ways. For example, we could write

$$30 = 2 \times 15, \; 30 = 3 \times 10, \text{ or } 30 = 5 \times 6.$$

Suppose we now repeat the process and express the composite integers in these products as products of factors, until every factor in each product is a prime. We would then obtain

$$\begin{aligned} 2 \times 15 &= 2 \times 3 \times 5 \\ 3 \times 10 &= 3 \times 5 \times 2 \\ 5 \times 6 &= 5 \times 2 \times 3. \end{aligned}$$

Note that each way of splitting up 30 gives the *same* set of prime factors; the final expressions differ only in the order which the primes are listed. □

Example 1.22 illustrates the following general result. You should be familiar with its statement, but the proof is beyond the scope of this course.

Theorem 1.7 The Fundamental Theorem of Arithmetic *Every positive integer greater than 1 can be written as the product of 1 and prime factors. Further, this expression is unique, except for the order in which the prime factors occur.*

The only reason for the appearance of 1 in this theorem is to accomodate the case when the integer is itself a prime number, such as 29 for example. It does not make sense to say that 29 is a product of 29, but we can say that 29 is a product 1 and 29. When asked to express a composite integer as a product of prime factors, we omit the unit 1. This theorem guarantees that however we go about splitting up an integer into prime factors, we will always obtain the same prime factors.

Example 1.23 We write 280 as the product of its prime factors. From Theorem 1.7, we can start by writing 280 as the product of any two of its factors. An obvious factor of 280 is 10, and so we can proceed as follows:

$$\begin{aligned} 280 &= 10 \times 28 \\ &= (2 \times 5) \times (4 \times 7) \\ &= 2 \times 5 \times (2 \times 2) \times 7. \end{aligned}$$

To simplify the expression, we gather together all occurences of the same prime and write the product as

$$280 = 2^3 \times 5 \times 7,$$

using index notation to deal with repeated factors, and listing the distinct primes in ascending order of size. □

Since every positive integer corresponds in a unique way to its prime factors, the primes can be regarded as the basic building blocks of the integers. The study of primes and the properties of

the integers has a long history, going back at least 2500 years to the early Greek mathematicians of the School of Pythagoras. An important modern application of prime numbers is to Public Key Cryptography. The details of this application are outside the scope of this course[1].

1.2.2 Representing fractions

Definition 1.8 *Numbers that can be expressed as a fraction (or ratio) m/n of two integers m and n, where $n \neq 0$, are called* **rational** *numbers.*

We have two ways of representing a rational number, whatever the number base. One is to write it in the form of a fraction m/n, where both m and n are integers (expressed in the relevant number base) and $n \neq 0$. Note that all integers are special cases of rational numbers, since we can write an integer m as the fraction $m/1$.

The other method of representing a rational number is to write it in the form of a decimal, using *negative* as well as positive and zero powers of the base as place values. We shall see in this section that the decimal form of a rational number has a very special property.

1.2.3 Decimal fractions

Let us look first at the representation of rational numbers in base 10 in expanded notation. The place values are illustrated by the following boxes.

$\cdots$	10^3	10^2	10^1	10^0	10^{-1}	10^{-2}	10^{-3}	$\cdots$

You will be familiar with the process of turning a fraction m/n into a decimal by dividing n into m. Since we are interested in what happens to the part of the expansion after the decimal point, we shall assume that m and n are positive integers with m less than n.

The decimal expansion of a fraction has an interesting feature: it either terminates after a finite number of places, as for example in the expansion $17/20 = 0.85$ or $3/8 = 0.375$; or *it has a recurring block of figures*, as for example the digit 6 in the expansion $5/12 = 0.41666...$ or the block 63 in $7/11 = 0.636363....$ We investigate why this occurs.

It can be shown that when we divide by a positive integer n, we can always choose the quotient so that the remainder is either 0 or one of the positive integers less than n. There are thus only n possible remainders when we divide by n. Now when we divide $m.000\cdots$ by n, the remainder at each division, becomes the "carrying number" for the next division. Since we can continue the divisions indefinitely and *there are only n possible carrying numbers*, we either obtain (eventually) a remainder 0, in which case the decimal expansion terminates, or *we eventually obtain a remainder that has occurred before*. From then on, the same sequence of quotients and remainders will occur again and again.

Example 1.24 Consider the decimal expansion of 4/7. On dividing by 7, the remainder is always one of the six possibilities: $0, 1, 2, 3, 4, 5$ or 6.

$$\begin{array}{r|l} & 0.\ 5\ 7\ 1\ 4\ 2\ 8\ 5\ 7\ \cdots \\ \hline 7 & 4.\ {}^{4}0\ {}^{5}0\ {}^{1}0\ {}^{3}0\ {}^{2}0\ {}^{6}0\ {}^{4}0\ {}^{5}0\ \cdots \end{array}$$

As you see, in this case, all six possible remainders arise: the first is 4, then we have successively remainders of $5, 1, 3, 2, 6$; then the remainder 4 occurs again. From this point on, the same sequence of six remainders and quotients repeats itself. Thus the decimal expansion of 4/7 has a repeated block of six digits. □

We now consider the converse problem. Suppose we are given a terminating or repeating decimal. Is it possible to write it as a fraction m/n, where m and n are integers? Clearly this is possible in the case of a terminating decimal, since we can write it with a denominator that is a power of 10. By a neat trick, we can show that any recurring decimal can also be expressed as a fraction m/n and find the values of m and n. We explain the method in the context of the following examples.

[1] If you are interested in reading about this application, there is a clear and interesting explanation in *Discrete Mathematics with Applications*, by M.O. Albertson and J.P. Hutchings (see Appendix A).

Example 1.25 We express the repeating decimals 0.272727... as a fraction in its lowest terms.

Step 1 Determine the number of digits in the repeating block. In this case, the repeating block is 27 and so the number of digits is 2.

Step 2 Write $x = 0.272727...$ and multiply x by 10^2, where the power of 10 is the length of the repeating block. This gives

$$100x = 27.272727... .$$

Step 3 Subtract x from $100x$.

$$\begin{array}{rrl} & 100x & = 27.272727... \\ - & x & = 0.272727... \\ \hline & 99x & = 27.000000... \end{array}$$

Step 4 Solve the equation $99x = 27$ to find x. This gives $x = 27/99$. Cancelling the common factor of 9, we have $x = 3/11$ as a fraction in its lowest terms. □

Example 1.26 We express the repeating decimal 0.15333... as a fraction in its lowest terms.

Step 1 Determine the number of digits in the repeating block. In this case, the repeating block is the single digit 3 and so the number of digits is 1.

Step 2 Write $x = 0.15333...$ and multiply x by 10, where the power of 10 is the length of the repeating block. This gives

$$10x = 1.53333... .$$

Step 3 Subtract x from $10x$.

$$\begin{array}{rrl} & 10x & = 1.53333... \\ - & x & = 0.15333... \\ \hline & 9x & = 1.38000... \end{array}$$

Step 4 Solve the equation $9x = 1.38$. We obtain $x = 1.38/9 = 138/900 = 23/150$. □

Since this process can be performed on any decimal with a recurring block, we see that any recurring decimal represents a fraction of two integers. Thus we have the following important result.

Result 1.9 *The rational numbers are just those numbers that can be represented as a terminating or recurring decimal.*

1.2.4 Fractions in bases other than 10

Binary system

We can represent rational numbers in any base by using negative powers of the base as place values. The place values in the full binary system are illustrated below.

...	2^3	2^2	2^1	$2^0 = 1$	2^{-1}	2^{-2}	2^{-3}	...

Example 1.27 The binary number $(0.1101)_2$ can be expressed in base 10 as follows.

$$1(2^{-1}) + 1(2^{-2}) + 0(2^{-3}) + 1(2^{-4}) = 1/2 + 1/4 + 1/16 = 13/16 = 0.8125.$$

An alternative method, is first to express $(0.1101)_2$ as a binary fraction, and then to convert both the numerator and denominator to base 10. Thus we have

$$(0.1101)_2 = (1101)/(10000)_2 = 13/16 = 0.8125.$$

The binary number $(101.11)_2$ represents

$$1(2^2) + 0(2^1) + 1(1) + 1(2^{-1}) + 1(2^{-2}) = 2^2 + 2^0 + 2^{-1} + 2^{-2} = 5.75.$$

Alternatively, we could work as follows.

$$(101.11)_2 = (101)_2 + (11/100)_2 = 5 + 3/4 = 5.75. \square$$

The digits to the *left* of the point are called the **integral part** of the number and the digits to the *right* of the point are called the **fractional part** of the number. Thus, for example, the integral part of $(101.11)_2$ is $(101)_2$ and the fractional part is $(0.11)_2$.

Hexadecimal system

The place values in the full hexadecimal system are illustrated below.

$\cdots$	16^3	16^2	16^1	$16^0 = 1$	16^{-1}	16^{-2}	16^{-3}	$\cdots$

Example 1.28 The number $(E7.3B)_{16}$ represents the decimal number

$$14(16^1) + 7(1) + 3(16^{-1}) + 11(16^{-2}) = 231 + 3/16 + 11/256 = 231.23046875. \square$$

To convert the fractional part of a binary number into the hexadecimal system, we split the binary string into groups of four bits *starting from the decimal point and working to the right*. If the last block to the right contains less than four bits, we add zeros as necessary *on the end of the number* to complete the block. We then use the binary-hex conversion table given in section 1.1.4.

Example 1.29 Converting the binary number $(101011101.11)_2$ to hex, we have

$$101011101.11 = 0001\ \ 0101\ \ 1101\ \ .1100 = (15D.C)_{16}.$$

Converting the hex number $(B3.5)_{16}$ to binary, we have

$$B3.5 = 1011\ 0011.0101 = (10110011.0101)_2. \square$$

1.3 Real numbers

Learning Objectives

After studying this section, you should be able to:

- give examples of the kinds of measurements for which the real number system is used;
- give examples of irrational numbers;
- use and interpret *greater than* and *less than* signs;
- express a rational number in floating point form and scientific notation.

Introduction

It is not easy to define exactly what we mean by a **real** number. We can visualize them geometically as a set of numbers that can be put into one-to-one correspondence with the points of a line, infinite in extent. This is the procedure we adopt when we mark a scale on a coordinate axis in order to graph a function. The real numbers have the important property that they form what is called a *continuum*. The set of real numbers is therefore used when we want to measure a quantity such as time, length or speed that can vary continuously.

1.3.1 Irrational numbers

The question arises as to whether all real numbers are rational, or whether there exist numbers that cannot be expressed as a fraction of two integers. The ancient Greek mathematicians answered this question in the 4th Century BC by showing that the square root of any integer that is not an exact square cannot be expressed as a fraction. This shows the existence of real numbers that are not rational. We call such numbers **irrational**. As well as numbers like $\sqrt{2}$, $\sqrt[3]{5}$, etc, numbers such as π and e (base of the natural logarithm) are irrational numbers. It follows from Result 1.9, that every irrational number has a decimal expansion that neither repeats nor terminates. Thus we can only ever give an approximation to the value of an irrational number in the decimal number system, or indeed in any number system with an integer base.

1.3.2 Inequality symbols

Suppose x and y are two different real numbers. The fact that x is not equal to y is expressed symbollically as

$$x \neq y.$$

Note that the word *distinct* is often used in mathematics, rather than "different", to describe two or more objects of the same class that can be distinguished from one another. So, in this case, we could have described x and y as "distinct numbers".

If $x \neq y$, then one of x and y must be the greater of the two, say x is greater than y. We write this as

$$x > y.$$

An equivalent statement is "y is less than x", written

$$y < x.$$

To avoid confusing these symbols, note that the pointed end (or small end) of either inequality sign always points towards the smaller number, while the open (or larger) end points to the larger number.

Example 1.30 The decimal expansion of π begins $3.1415\ldots$. Thus we can say that $\pi > 3.14$ and also that $\pi < 3.15$. Together, these two inequalities tell us that the true value of π is between 3.14 and 3.15. We can express this by concatenating the two inequalities as follows:

$$3.14 < \pi < 3.15.$$

Notice that we have turned the inequality $\pi > 3.14$ round and expressed it as $3.14 < \pi$, before combining it with the other inequality, so that π is sandwiched between the smallest and the largest of the three numbers and both inequality signs point in the same direction. □

Example 1.31 The decimal expansion of $17/9$ is $1.888\ldots$. Thus $17/9$ lies between 1.888 and 1.889. We can express this symbolically by writing

$$1.888 < 17/9 < 1.889. \square$$

The symbol "$>$" can be modified to express *at least* or, equivalently, *greater than or equal to*. For example,

$$x \geq 5.3$$

is read "x is at least 5.3", or "x is greater than or equal to 5.3". Similarly, the symbol $\leq$' reads "at most" or "less than or equal to".

Example 1.32 Suppose a real number x satisfies the inequalities $14 \leq x \leq 25$. The inequality $14 \leq x$ tells us that x is *at least* 14. The inequality $x \leq 25$ means that x is *at most* 25. Thus together they read: "x is at least 14 and at most 25".

The inequality sign "$\leq$" can also be concatenated with "$<$", in either order. So, for example, $14 \leq x < 25$ means that x is at least 14 and less than 25. □

1.3.3 Floating-point notation

A digital device, such as a hand-held calculator or digital computer, is only able to store a finite (and often fixed) number of digits to represent any numeral. On the other hand, all irrational numbers and rational numbers with repeating blocks in their decimal expansion require an infinite number of digits to represent them exactly in decimal form. Even a terminating decimal may have more digits in its decimal expansion than the device is able to store. Thus most real numbers will be represented in a digital computer only by an approximation, formed by cutting off the decimal expansion after a certain number of digits. This produces an error in calculations known as a **truncation error.**

Suppose that a calculator can store only 10 digits. Then, for example, the error in representing 5/12 as 0.416666666 is 0.000000000666... ., or a little less than 0.000000000667. This may not seem very grave, but in a complex calculation errors may accumulate, giving a substantial error in the solution.

It is hard to read and comprehend a number such as 0.000000000667 written in this form. There are more convenient ways of expressing very small or very large decimal numbers. The two main ones are called *scientific notation* and *floating-point notation*. Most computers can store real numbers in **floating-point notation**, so we shall describe this method in detail. In this system, a positive number is expressed as the product of a power of 10 called the **exponent**, and a number x such that $0.1 \leq x < 1$ called the **mantissa.**

Example 1.33 In floating point notation, the number 0.000000000667 is expressed as 0.667×10^{-9}, where -9 is the exponent and 0.667 is the mantissa; the number 146.75 is expressed as 0.14675×10^{3}, where 3 is the exponent and 0.14675 is the mantissa. □

Scientific notation is based on a similar principle, and differs only in that the mantissa is a number x such that $1 \leq x < 10$.

Example 1.34 In scientific notation, the number 0.000000000667 is expressed as 6.67×10^{-10}, where -10 is the exponent and 6.67 is the mantissa; the number 146.75 is expressed as 1.4675×10^{2}, where 2 is the exponent and 1.4675 is the mantissa. □

1.4 Exercises 1

1. (a) Write the decimal numbers 4037 and 40371 in expanded form as multiples of powers of 10.

   (b) Suppose that a is the 4-digit decimal number $a_3a_2a_1a_0$. Write a in expanded form as multiples of powers of 10. Suppose that x is the 5-digit decimal number $a_3a_2a_1a_01$. Can you express x in terms of a?

2. (a) A number is expressed in base 5 as $(234)_5$. What is it as a decimal number?

   (b) Suppose you multiply $(234)_5$ by 5. What would the answer be in base 5?

3. Express the following binary numbers as decimal numbers:

$$(11011)_2; \qquad (1100110)_2; \qquad (111111111)_2.$$

   Can you think of a quick way of doing the last one?

4. Perform the binary additions

$$(10111)_2 + (111010)_2; \qquad (1101)_2 + (1011)_2 + (1111)_2.$$

5. Perform the binary subtractions

$$(1001)_2 - (111)_2; \qquad (110000)_2 - (10111)_2.$$

6. Perform the following binary multiplications

$$(1101)_2 \times (101)_2; \qquad (1101)_2 \times (1101)_2.$$

7. Perform the binary division $(111011)_2 \div (101)_2$, giving a quotient and remainder.

8. Find the decimal equivalents of each of the binary numbers in the previous questions and check your answers by decimal arithemetic.

9. (a) Write the hexadecimal numeral $(\mathrm{A5D})_{16}$ in decimal.

   (b) Suppose that a is represented in hex by $(a_2a_1a_0)_{16}$. Write a in expanded form as multiples of powers of 16. Suppose that x and y are represented in hex by $(a_2a_1a_0 0)_{16}$ and $(a_2a_1a_0\mathrm{A})_{16}$, respectively. Can you express x and y in terms of a in base 10?

10. Calculate $(\mathrm{BBB})_{16} + (\mathrm{A5D})_{16}$ and $(\mathrm{BBB})_{16} - (\mathrm{A5D})_{16}$, working in hex.

11. (a) Write the hex number $(\mathrm{EC4})_{16}$ in binary.

    (b) Write the binary number $(1111011010)_2$ in hex.

12. Express the decimal number 753 (a) in binary; (b) in base 5; (c) in hex.

13. Express 42900 as a product of its prime factors, using index notation for repeated factors.

14. Express the recurring decimals 0.126126126... and 0.7545454... as fractions *in their lowest terms.*

15. Given that π is an irrational number, can you say whether $\frac{\pi}{2}$ is rational or irrational? Or is it impossible to tell?

16. (a) Express the binary number $(0.101)_2$ in base 10 and in base 16.

    (b) Express the hex number $(\mathrm{B}.25)_{16}$ in base 10 and in base 2.

    (c) Write the decimal fraction 3/8 in base 2 (as a "bicimal").

    (d) Can you work out how to express the number $(0.32)_5$ in base 10?

17. Express the following inequalities in symbols:

    (a) 5/7 lies between 0.714 and 0.715;

    (b) $\sqrt{2}$ is at least 1.41;

    (c) $\sqrt{3}$ is at least 1.732 and at most 1.7322.

18. Write the numbers 0.0000526 and 429000000 in floating point form. How is the number 1 expressed in this notation?

19. Let $n = ab$ be a composite integer, where a, b are proper factors of n and $a \leq b$.

    (a) Say why $a \leq \sqrt{n}$.

    (b) Deduce that every composite integer n has a prime factor p such that $p \leq \sqrt{n}$.

    (c) Use this result to prove that 89 is prime by testing it for divisibility by just four prime numbers.

    (d) Decide whether 899 is prime.

20. (a) What is the *maximum* number of digits that a decimal fraction with denominator 13 could have in a recurring block in theory?

    (b) How many digits does 1/13 actually have?

    (c) Do each of the decimal fractions $1/13, 2/13, 3/13, 4/13, 5/13$ have the same number of digits in a recurring block? What do you notice about the digits in the recurring blocks of the fractions $1/13, 3/13, 4/13$?

    (d) Can you predict which other fractions with denominator 13 will have the same digits as 1/13 in their recurring block?

Chapter 2

Sets and Binary Operations

Summary

Set notation, specifying sets by the listing method and rules of inclusion method; special sets of numbers; empty set; cardinality; subsets, power set; set complement; binary operations on sets: union, intersection, difference and symmetric difference; Venn diagram, membership table; laws of set algebra.

References: Epp Sections 5.1 (pp 231-237), 5.2, 5.3 (pp 258-264) *or* M&B Sections 2.1, 2.2, 2.3, 2.4.

Introduction

By a set we simply mean a collection or class of objects. The objects in the set are called its **members** or **elements**. Sets have become the basic language in which most results in mathematics and computer science are expressed. In this chapter, we look at ways in which sets are specified, how they may be represented and how they are combined to make other sets.

2.1 Specifying sets

Learning Objectives

When you have completed your study of this section, you should be able to:

- use set notation for specifying sets by the listing method and rules of inclusion method;
- use and interpret the standard symbols for special sets of numbers and for the empty set.

2.1.1 Listing method

We usually use an upper case letter to denote a set and a lower case letter to denote a member of the set. To specify a set, we must describe its members in an unambiguous way. One way of doing this is to *list* the members of the set, separated by commas, and enclose the list in a pair of brace brackets.

Example 2.1 The set D of decimal digits can be expressed as

$$D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\},$$

and the set B of bits can be expressed as

$$B = \{0, 1\}. \ \square$$

Definition 2.1 *We use the symbol* $\in$ *to mean* **belongs to** *and* $\notin$ *to mean* **does not belong to**. *Thus we write* $y \in X$ *to denote that (the element)* y *belongs to (the set)* X *and* $y \notin X$ *to indicate that* y *is not a member of the set* X.

Example 2.2 Referring to the sets D and B defined in Example 2.1, we can say that $5 \in D$, but $5 \notin B$. □

You will meet quite a lot of new notation in this chapter. In order for it to become familiar, it is important to *verbalise* the symbol as you read it. So, in Example 2.2, we *write* "$5 \in D$", but we *read* this as "5 belongs to (set) D", or "5 is in D". Similarly, we write "$5 \notin B$", but read this as "5 does not belong to (set) B" or "5 is not in B".

Sometimes the elements in the set are in a sequence that is easily recognised when we are given the first few terms. In this case, instead of listing *every* member of the set, we can use a sequence of three dots (called an **elipsis**) to mean "et cetera" or "and so on", as the following example illustrates.

Example 2.3 Let H be the set of integers from 1 to 100. We write

$$H = \{1, 2, 3, \ldots, 100\}.$$

In a similar way, we can express the set of positive odd integers as

$$K = \{1, 3, 5, 7, \ldots\}. \square$$

Special sets of numbers

It is convenient to denote certain key sets of numbers by a standard letter.

Definition 2.2 *The symbol* $\mathbb{Z}$ *is used to denote the set of integers;* $\mathbb{Z}^+$ *denotes the set of positive integers;* $\mathbb{R}$ *denotes the set of real numbers; and* $\mathbb{Q}$ *denotes the set of* **rational numbers** *(the letter* $\mathbb{Q}$ *stands for "quotient").*

Of these sets, we can specify the set of positive integers and the set of integers by the listing method using elipses, as follows:

$$\begin{aligned} \mathbb{Z}^+ &= \{1, 2, 3, \ldots\} \\ \mathbb{Z} &= \{0, 1, -1, 2, -2, 3, -3, \ldots\} \end{aligned}$$

Note that $\mathbb{Z}$ includes 0 and the negative whole numbers as well as the positive ones. Similarly, $\mathbb{R}$ contains 0 and the negative reals and $\mathbb{Q}$ contains 0 and the negative rationals, as well as the positive ones.

2.1.2 Rules of inclusion method

Another way of specifying a set is by giving **rules of inclusion** that distinguish members of the set from objects not in the set.

Definition 2.3 *The context of the problem in which a set arises determines an underlying set, called the* ***universal set*** *for the problem, from which the elements of the set will be drawn.*

For example, if our subject is a set of leopards, the universal set, explicitly stated or implied by the context, might be *all wild animals in Africa* or *all animals in London Zoo* or *all animals belonging to the cat family.*

Example 2.4 To specify the set H of Example 2.3, we could write

$$H = \{n \in \mathbb{Z} : 1 \leq n \leq 100\}.$$

This tells us that the *universal set* is $\mathbb{Z}$ and the *rule of inclusion* is that n must be between 1 and 100 inclusive; the colon stands for the words *such that*, and so we would read this description of H as

$$H \text{ is the set of integers } n \text{ such that } 1 \leq n \leq 100.$$

Similarly, we could specify the set K of Example 2.3 as

$$K = \{m \in \mathbb{Z}^+ : m \text{ is odd}\}. \square$$

Example 2.5 Let X be the set of real numbers that satisfy the equation $x^2 - x = 0$. Then we could write

$$X = \{x \in \mathbb{R} : x^2 - x = 0\},$$

read as

$$X \text{ is the set of real numbers } x \text{ such that } x^2 - x = 0. \square$$

The empty set

Another set which has a special letter to denote it is the set containing no elements. This is called the **empty** or **null set** and denoted by the symbol $\emptyset$.

Example 2.6 The set of integers m such that $m^2 = 5$ is the empty set. We could write

$$\{m \in \mathbb{Z} : m^2 = 5\} = \emptyset.$$

Even and odd integers

Given an integer, you could probably say immediately whether it is odd or even, but how do we define the set of even integers and the set of odd integers? To test whether an integer is even we divide it by 2: the even integers are just those that are divisible by 2; the odd integers are just those that are not divisible by 2. Hence the set of **even integers** is

$$\{0, 2, -2, 4, -4, 6, -6 \ldots\}$$

Notice that 0 is even, and even integers can be negative as well as positive. The set of **odd integers** is

$$\{1, -1, 3, -3, 5, -5, \ldots\}.$$

As we saw in Chapter 1, another way of saying that an integer is divisible by 2 is to say that it is a multiple of 2. Thus an even integer is a number that can be expressed in the form $2m$, where $m \in \mathbb{Z}$. The set of even integers can therefore be expressed by the *rules of inclusion* method as

$$\{2m : m \in \mathbb{Z}\}.$$

This is read

"the set of all numbers of the form $2m$, where m is an integer."

Any odd integer can be obtained by adding 1 to (or subtracting 1 from) some even integer. Thus the set of odd integers can be expressed as

$$\{2m + 1 : m \in \mathbb{Z}\} \text{ or } \{2m - 1 : m \in \mathbb{Z}\}.$$

Other sets of integers which have as defining property that they are all multiples or powers of a given fixed integer can be expressed in a similar way.

Example 2.7 The set $\{\ldots -20, -10, 0, 10, 20, \ldots\}$ of multiples of 10 can be expressed by the *rules of inclusion method* as

$$\{10a : a \in \mathbb{Z}\},$$

and the set $\{1, 10, 100, 1000, \ldots\}$ can be expressed as

$$\{10^r : r \in \mathbb{Z}, r \geq 0\}. \square$$

2.2 Subsets

Learning Objectives

When you have completed your study of this section, you should be able to:

- state the condition for a set to be contained in another set as a subset and the condition for two sets to be equal;
- use subset notation correctly;
- say what is meant by the *cardinality* and the *power set* of a finite set;
- find the power set of a given finite set.

Introduction

When we are considering a set, it often arises that we would like to pick out just those objects in the set which satisfy a given condition. When we do this, we are picking a *subset* of the given set. Of course, if all the objects in the original set satisfy the condition, the subset will contain the same members as the original set. At the other extreme, it may turn out that none of the members of the set satisfy the condition, and then our subset will be the empty set. More typically though, some but not all of the members of the original set will satify our condition to be in the subset. In this section, we shall introduce a formal test for a subset and some important notation.

2.2.1 Notation for subsets

Definition 2.4 *Given two sets A and B, the set A is said to be a* **subset** *of B if every element of A is also an element of B. When this is the case, we write $A \subseteq B$.*

Notice that $A \subseteq A$, for all sets A. We also regard the empty set $\emptyset$ as a subset of every set.

Example 2.8 Let $K = \{1, 3, 5, 7, ...\}$. Then we can say that $K \subseteq \mathbb{Z}^+$. We can also say that $\mathbb{Z}^+ \subseteq \mathbb{Z}$ and that $\mathbb{Z} \subseteq \mathbb{R}$.
We can concatenate these relations between the sets, as follows:

$$K \subseteq \mathbb{Z}^+ \subseteq \mathbb{Z} \subseteq \mathbb{R}. \square$$

The previous example illustrates a general rule that follows from the definition of *subset*.

Rule 2.5 *If A, B, C are sets such that A is a subset of B, and B is a subset of C, then A is also a subset of C. In symbols, this becomes:*

$$\textit{If } A \subseteq B \textit{ and } B \subseteq C, \textit{ then } A \subseteq C.$$

If it is true that both $A \subseteq B$ and $B \subseteq A$, then A and B must contain the same elements. This observation leads to the following definition of *equality* of two sets.

Definition 2.6 *If it is true that both $A \subseteq B$ and $B \subseteq A$, then we say that the sets A and B are* **equal** *and write $A = B$.*

Example 2.9 Suppose $X = \{0, 1\}$, $Y = \{1, 0\}$ and $Z = \{1, 1, 0, 1, 0\}$. Then since each of these sets contain just the numbers 0 and 1, we have $X = Y = Z$. $\square$

These equalities illustrate the fact that a set is *determined by its elements*, the method of specification is not important; further, we may ignore repetitions of elements and also the order in which the elements are written.

Definition 2.7 *If $A \subseteq B$ but $A \neq B$, then A we say that A is a* **proper** *subset of B. In this case, B contains at least one element that is not in A. This is written symbolically as $A \subset B$.*

You will have noticed that the relation "$\subseteq$" for sets has some of the properties of "$\leq$" for real numbers. For example, if we have numbers x, y, z such that $x \leq y$ and $y \leq z$, then we know that $x \leq z$; also, if we have numbers a, b such that $a \leq b$ and $b \leq a$, then we know that $a = b$. However, there is an important difference. For any two real numbers x, y, we know that *at least one* of the statements $x \leq y$ and $y \leq x$ must be true. But in the case of sets, it is easy to find examples of pairs of subsets X, Y of the same universal set for which *neither* of the statements $X \subseteq Y$ and $Y \subseteq X$ is true.

Example 2.10 The sets $X = \{1, 3, 5\}$ and $Y = \{1, 2, 6\}$ are both subsets of the set of decimal digits D. Clearly, neither the statement $X \subseteq Y$ nor the statement $Y \subseteq X$ is true. □

Definition 2.8 *Let A, B be sets such that neither of the statements $A \subseteq B$ nor the statement $B \subseteq A$ is true. Then we say that the sets A, B are* **noncomparable**.

2.2.2 Cardinality of a set

Definition 2.9 *A set is called finite when it contains a finite number of elements and otherwise it is called infinite. The number of distinct elements in a finite set S is called its cardinality.*

Notice that we use the word *distinct* in mathematics to mean "distinguishable" or"different".

Example 2.11 The sets X, Y, Z of Example 2.9 each have cardinality 2, the set $H = \{n \in \mathbb{Z} : 1 \leq n \leq 100\}$ has cardinality 100, whereas $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$ are all examples of *infinite* sets. □

Note that since the empty set contains no elements, it has cardinality 0.

2.2.3 Power set

Counting subsets

Now suppose we are given a finite set S. We might want to ask how many distinct subsets does S contain? In order to count all possible subsets of a set of given cardinality, we shall show how we can code each of them with a binary string. Consider the following problem.

Example 2.12 The College Refectory is offering three vegetables, beans, carrots and tomatoes to accompany your main dish. You may, of course, not wish to order any of these, but if you do, you may choose any one, two or three of them. How many different possibilities arise?

Suppose we code the possible choices for your order using a 3-bit binary string. We record 1 if you choose an item and a 0 if you reject it. The first bit represents your decision on beans, the second on carrots and the third on tomatoes. But each possible order can also be regarded as a subset of the set $V = \{b, c, t\}$ (where b = beans, c = carrots and t = tomatoes). The correspondence between the subsets and the 3-bit codes is given in the table below.

subset	$\emptyset$	$\{b\}$	$\{c\}$	$\{t\}$	$\{b,c\}$	$\{b,t\}$	$\{c,t\}$	$\{b,c,t\}$
code	000	100	010	001	110	101	011	111

Thus there are in all 8 choices, each represented by a different 3-bit binary string. Further, each possible 3-bit binary string corresponds to a different subset, so that the table above shows a *one-to-one* correspondence between the subsets of V and the set of all 3-bit binary strings. □

We can extend this method to code the subsets of any finite set, by using binary strings of an appropriate length.

Example 2.13 In order to code the subsets of the set $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, we must use the set of 10-bit binary strings. Then, for example, the subset $\{0, 1, 2, 3, 4\}$ of D is represented by

the string 1111100000; and the subset of D represented by the string 1100010101 is $\{0, 1, 5, 7, 9\}$. □

Now we can generalise this idea, to give a formula for the number of subsets of a set containing n elements, where n is any positive integer. Every subset of a set containing n elements can be represented in a unique way as an n-bit binary string. Conversely, every n-bit binary string represents a unique subset. So there are the same number of subsets as there are n-bit binary strings. We shall see later in the course that this number is 2^n. This implies the following result:

Theorem 2.10 *A set with n elements has exactly 2^n subsets.*

Definition 2.11 *We call the* collection *or* family *of all subsets of a set S the* **power set of** *S and denote it by $\mathcal{P}(S)$.*

The term *power set* derives from the fact that when the cardinality of S is n, then the cardinality of $\mathcal{P}(S)$ is 2^n. Notice that both $\emptyset$ and S are members of $\mathcal{P}(S)$; they correspond respectively to the binary strings in which each bit is 0 and in which each bit is 1.

The situation in which we have a set whose elements are themselves sets calls for some care in the use of notation.

Example 2.14 Let $D = \{0, 1, 2, ..., 9\}$. Then we have seen that we write the statement "5 is an element of S" as

$$5 \in S.$$

The set $\{5\}$ denotes the subset of D containing just the element 5. We express the fact that $\{5\}$ is a subset of D by writing

$$\{5\} \subseteq D.$$

However, $\{5\}$ is a member of the family of subsets of D, that is, the powerset of D. We express this fact by writing

$$\{5\} \in \mathcal{P}(D).$$

Notice also, that the subset $\{0\}$ is *not* the empty set: it is the subset of D containing just the digit 0. □

2.3 Operations on Sets

Learning Objectives

When you have completed your study of this section, you should be able to:

- define the *set complement* of a set and find the complement of a given set;
- define the binary operations of union, intersection, set difference and symmetric difference of two sets and illustrate each of these operations by a Venn diagram;
- find the union, intersection, set difference and symmetric difference of two given sets;
- state and use the commutative, associative and distributive laws for set union and intersection;
- use membership tables and Venn diagrams to establish relations between given sets.

Introduction

In this section, we suppose that A, B, C are subsets of some universal set $\mathcal{U}$. We shall define operations that can be performed on these subsets to yield other subsets of $\mathcal{U}$. We shall make use of a pictorial representation of sets, called **Venn diagrams** and also an equivalent, but more general, method of defining subsets of a universal set by using **membership tables**. We consider how both these methods of representation can be used to verify relations between sets.

2.3.1 The complement of a set

Definition 2.12 *The set of elements of $\mathcal{U}$ that are* not *in A is called the* **complement of A,** *denoted by A'. Thus*

$$A' = \{x : x \notin A\}.$$

Common alternative notations found in textbooks for the complement of A are $\sim A$ and $\overline{A}$.

We illustrate the relation between the sets A and A' in a Venn diagram in Figure 2.1. In drawing a Venn diagram, we assume that all members of the universal set are contained within the rectangular frame of the picture. We subdivide the rectangle to depict subsets of the universal set. There are two conventions:

1. no element is depicted as lying on any boundary line of a set;
2. some regions of the diagram may contain no elements.

We usually shade the region of the diagram containing all elements in the indicated set.

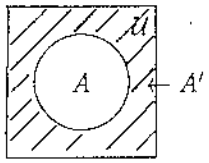


Figure 2.1.

Note the following two special cases of complements.

$$\mathcal{U}' = \emptyset \text{ and } \emptyset' = \mathcal{U}.$$

Notice also that for any subset $A \subseteq \mathcal{U}$, we have

$$(A')' = \{x : x \notin A'\} = A.$$

2.3.2 Binary operations on sets

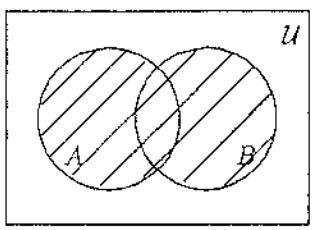
The remaining operations that we define in this section combine *two* subsets and are in consequence known as **binary operations.**

Definition 2.13 *The set of elements that are in A or in B (including the elements that are in both sets) is called the* **union of** *A and B, and denoted by $A \cup B$. Thus we have*

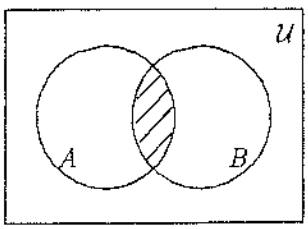
$$A \cup B = \{x : x \in A \text{ or } x \in B \text{ (or both)}\}.$$

Definition 2.14 *The set of elements that are in both A and B is called the* **intersection of** *A and B, and denoted by $A \cap B$. Thus we have*

$$A \cap B = \{x : x \in A \text{ and } x \in B\}.$$



$A \cup B$ is shaded region



$A \cap B$ is shaded region

Figure 2.2.

Definition 2.15 *The set of elements that are in A but not in B is called the set difference of A and B, and denoted by $A - B$. Thus we have*

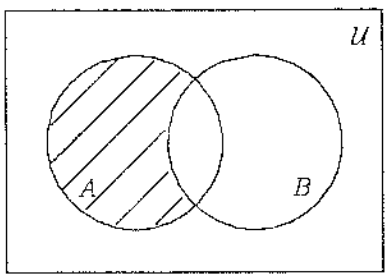
$$A - B = \{x : x \in A \text{ and } x \notin B\}.$$

Definition 2.16 *The set of elements that are in A or in B, but not in both, is called the symmetric difference of A and B, and denoted by $A \oplus B$. Thus we have*

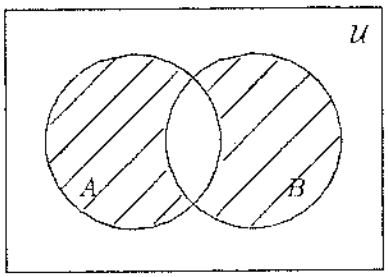
$$A \oplus B = \{x : x \in A \text{ or } x \in B, \text{ but not both }\}.$$

There are several ways of expressing the symmetric difference in terms of the other binary operations we have defined. For example

$$\begin{aligned} A \oplus B &= (A - B) \cup (B - A) \\ &= (A \cup B) - (B \cap A). \end{aligned}$$



$A - B$ is shaded region



$A \oplus B$ is shaded region

Figure 2.3.

Notice that in drawing a Venn diagram to illustrate two sets in general, we depict the sets as overlapping, as shown in Figures 2.2 and 2.3. Drawn in this way, the boundaries of the sets partition the whole area (representing $\mathcal{U}$) into *four* discrete regions. We are not saying that in every example there are necesarily elements in each of these regions; in any particular example it may happen that one or more of the regions is empty. In particular, if the region representing $A \cap B$ is empty, then A and B are said to be **disjoint** subsets.

Example 2.15 Suppose $\mathcal{U} = \mathbb{Z}$, $A = \{1, 3, 5, 7, 9\}$, $B = \{2, 4, 5, 6, 7, 8\}$ and $C = \{2, 4\}$. Then $A \cup B = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$; $A \cap B = \{5, 7\}$; $A - B = \{1, 3, 9\}$; $B - A = \{2, 4, 6, 8\}$ and $A \oplus B = \{1, 3, 9, 2, 4, 6, 8\}$. However, $A \cap C = \emptyset$, so we can say that the subsets A and C are *disjoint*. □

It follows directly from the definitions of union and intersection that $A \cap B$ is a subset of each of the sets A and B; similarly, both A and B are subsets of $A \cup B$. You can easily verify these statements from the Venn diagrams shown above.

2.3.3 Laws for binary operations

Rule 2.17 *(Identity and complement laws). Let A be a subset of a universal set $\mathcal{U}$. Then*

(i) $A \cap \emptyset = \emptyset$ *and* $A \cup \emptyset = A$;

(ii) $A \cap \mathcal{U} = A$ *and* $A \cup \mathcal{U} = \mathcal{U}$;

(iii) $A \cap A' = \emptyset$ *and* $A \cup A' = \mathcal{U}$. □

Examples of **binary operations** on the real numbers are addition, multiplication and subtraction (division is a binary operation on the non-zero reals). Addition and multiplication obey certain rules that are so familiar that we tend to take them for granted. For example, we know that for any two real numbers x, y,

$$x + y = y + x \text{ and } xy = yx.$$

We say that the operations of addition and multiplication are **commutative**, because we can commute (or interchange) the relative positions of x and y without changing the value of the sum or product. On the other hand, subtraction is *not* a commutative operation because the statement "$x - y = y - x$" is not true for *all* values of x and y. You will encounter another example of a non-commutative operation when we come to study matrix multiplication later in this course.

In the case of the set operations union and intersection, however, it is clear from their definitions, and also from the Venn diagrams above, that $A \cup B = B \cup A$ and $A \cap B = B \cap A$, for all sets A, B. So we have:

Rule 2.18 *(Commutative laws). For all subsets A and B of a universal set $\mathcal{U}$, we have*

$$\text{(i) } A \cap B = B \cap A; \text{ (ii) } A \cup B = B \cup A. \ \square$$

2.3.4 Membership tables

Before considering binary operations on more than two sets, it is useful to develop an alternative method to Venn diagrams for illustrating and verifying our results.

We have already noted that in drawing a Venn diagram to illustrate two subsets A, B of a universal set $\mathcal{U}$, the boundaries of A and B partition the whole area (representing $\mathcal{U}$) into *four* regions, labelled R_a, R_b, R_c and R_d respectively in Figure 2.4 below.

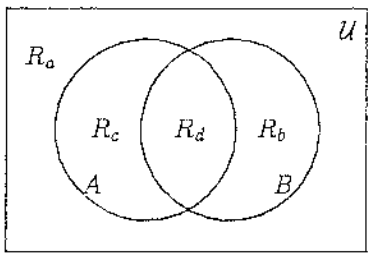


Figure 2.4.

We now give each of these regions a unique 2-bit binary code, using the following rule:

- the first bit is 1 if the region is inside the set A, and is 0 otherwise;
- the second bit is 1 if the region is inside the set B, and is 0 otherwise.

For example, the region R_a is not in A and not in B, so we give it the binary code 00; R_b is not in A, but it is in B, so we give region R_b the code 01, and so on. We record the codes for these four regions in the following table (Figure 2.5). It is easy to interpret a given 2-bit binary code as a region; for example, 10 codes the region that is in A but not in B.

region	A	B
R_a	0	0
R_b	0	1
R_c	1	0
R_d	1	1

Figure 2.5.

Notice that in the column indexed by set A in Figure 2.5, the entry 1 appears just in the rows for regions R_c and R_d, the two regions that comprise set A. Similarly, the entry 1 in the B column occurs just in the rows corresponding to regions R_b and R_d, the regions comprising set B. We call the column corresponding to set A the **membership table for** A.

We can construct membership tables for each of the subsets formed by combining the sets A and B by one of the binary operations $\cup$, $\cap$, $-$ or $\oplus$ in a similar way. To determine the membership table for $A \cap B$, note that this subset corresponds just to the region coded 11 (see Figure 2.2). Thus the column for $A \cap B$ has a 1 in the row coded 11 and a zero in each of the other rows. The set

$A \cup B$ contains all elements in A or in B or in both. Hence we enter a 1 in the rows corresponding to each of the regions coded 10, 01 and 11, and enter 0 in the row coded 00 (see Figure 2.2). The resulting membership tables are shown in Figure 2.6.

A	B	$A \cap B$	$A \cup B$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Figure 2.6.

De Morgan's laws

We can use either Venn diagrams or membership tables to prove the following two laws relating to set complements. They are due to the British mathematician Augustus De Morgan, who was a Professor of Mathematics at London University in the later part of the nineteenth century.

Rule 2.19 *(De Morgan's laws). Let A, B be subsets of a universal set $\mathcal{U}$, then*

(i) $(A \cap B)' = A' \cup B'$;

(ii) $(A \cup B)' = A' \cap B'$.

These laws are very important and students often get them wrong, so let us also express them in words. The first law says that the complement of the *intersection* of two sets is the *union* of their complements; the second law says that the complement of the *union* of two sets is the *intersection* of their complements.

As an illustration of how we use membership tables to prove a relation between two sets, we use this method to prove (i).

Example 2.16 We need to construct and compare columns for $(A \cap B)'$ and $A' \cup B'$. To find the column for $(A \cap B)'$, we first obtain the column for $A \cap B$ (see Figure 2.6) and then take its complement. Recollect that the complement of a given set comprises just those regions of the universal set that are not in the given set. Thus to obtain the membership table for $(A \cap B)'$, we enter 1 in each row in which $A \cap B$ has 0, and 0 in each row in which the $A \cap B$ has a 1. This gives the following table.

A	B	$A \cap B$	$(A \cap B)'$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

To construct a column for $A' \cup B'$, we first construct columns for A' and B' by taking the complements of the rows for A and B respectively. Now the union of two sets contains any region that is in at least one of these sets. Thus we construct the column for $A' \cup B'$ by putting a 1 in any row in which either A' or B' has a 1, and 0 just in the row(s) where both A' and B' have a 0 (the "set union rule" in Figure 2.6). This gives the following table.

A	B	A'	B'	$A' \cup B'$
0	0	1	1	1
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

Since the columns corresponding to $(A \cap B)'$ and $A' \cup B'$ are identical, the Venn diagram for each of these subsets will comprise precisely the same regions of $\mathcal{U}$. Hence if $x \in (A \cap B)'$, then $x \in A' \cup B'$ and conversely. Thus these sets are equal, proving De Morgans law (i). □

Combinations of three sets

We now consider combinations of more than two sets. Figure 2.6 shows how three sets can be represented in the most general way by a Venn diagram. You will see that their boundaries subdivide the area representing the universal set into *eight* discrete regions, labelled $R_a, R_b, \ldots, R_h$ in Figure 2.7.

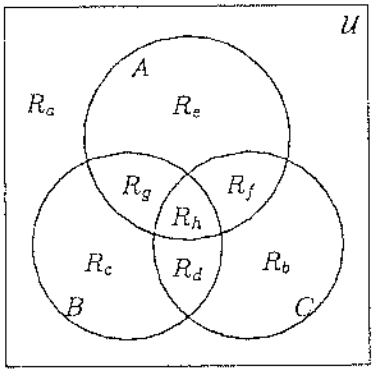


Figure 2.7.

We code each of these eight regions with a unique 3-bit binary string, by the following rule:

- the first bit is 1 if the region is inside the set A, and is 0 otherwise;
- the second bit is 1 if the region is inside the set B, and is 0 otherwise;
- the third bit is 1 if the region is inside the set C, and is 0 otherwise;

For example, the region R_a is not in A, not in B and not in C, so we give it the binary code 000; R_b is not in A, not in B, but it is in C, so we give region R_b the code 001, and so on. A membership table for a subset expressed as a combination of three sets A, B, C must therefore have *eight* rows, one corresponding to each of the eight regions in Figure 2.7.

The membership table for each of the sets A, B, C is given below. Check that each row in the table corresponds to the region of the Venn diagram labelled with the same letter in Figure 2.7.

	A	B	C
R_a	0	0	0
R_b	0	0	1
R_c	0	1	0
R_d	0	1	1
R_e	1	0	0
R_f	1	0	1
R_g	1	1	0
R_h	1	1	1

As in the case of regions defined by the intersection of two sets, it is easy to interpret one of these binary codes: for example, the code 101 defines a region that is in A and C, but not in B. Thus we can dispense with the literal labels $R_a, R_b, R_c, \ldots, R_h$ and index the rows of a membership table involving three sets with the eight possible 3-bit binary codes. Note that as each code uniquely defines a row of such a membership table, the rows can be given in any order. However, it is a good idea to develop a systemtic way of listing them, so that you can be sure that you have included all eight different codes.

2.3.5 Laws for combining three sets

We now give two important pairs of laws concerned with the use of *brackets* when we combine three (or more) sets by intersection or union operations. Note that the same general rule holds in

set algebra as in arithmetic - when simplifying an expression in which brackets have been inserted, deal with the terms in brackets first.

Associative laws

In general, when we combine three or more sets by binary operations, we need to use brackets to say which pair of sets should be combined first. However, in one special situation, when we combine three (or more) sets using the *same* operation (either union or intersection) between each pair of sets, as for example $A \cup B \cup C$ or $A \cap B \cap C$, then it is not necessary to use brackets. This is a consequence of the **associative laws** for union and intersection. These say that whichever way the brackets are inserted in these expressions, we obtain the same set.

Rule 2.20 *(Associative laws). For any three subsets A, B, C of a universal set $\mathcal{U}$,*

(i) $(A \cap B) \cap C = A \cap (B \cap C)$;

(ii) $(A \cup B) \cup C = A \cup (B \cup C)$.

We can prove this pair of laws either by Venn diagrams or by membership tables. We give a proof of part (i) using membership tables. We construct *two* tables, one for each side of the expression. The rows of each table are indexed by the eight 3-bit binary codes in columns headed by A, B, C. In the table for the left side, we construct a column for $A \cap B$, by the "intersection rule", that is, we put a 1 in each row in which both A and B have a 1, and put a 0 in all other rows (see Figure 2.8). We then combine this column with column C by the "intersection rule " to produce a column for $(A \cap B) \cap C$.

The table for the right side is developed in the same way. Using the "intersection rule", we first construct a column for $B \cap C$ and then combine this with column A to obtain the membership table for $A \cap (B \cap C)$.

A	B	C	$A \cap B$	$(A \cap B) \cap C$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	1	0
1	1	1	1	1

A	B	C	$B \cap C$	$A \cap (B \cap C)$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Figure 2.8.

Since the columns for $(A \cap B) \cap C$ and $(A \cap B) \cap C$ have identical entries, we deduce these two sets are equal. □

Binary operations that satisfy the associative law are said to be **associative**. Not all operations are associative, however. For example, the operation *subtraction* on the set of real numbers is not associative, because it is easy to find examples of real numbers x, y, z, such that

$$x - (y - z) \neq (x - y) - z.$$

For example, if we take $x = 6$, $y = 4$ and $z = 3$, then $x - (y - z) = 6 - 1 = 5$, while $(x - y) - z = 2 - 3 = -1$. We leave it as an exercise for you to verify that set difference is also not an associative operation.

Distributive laws

The second law dealing with the use of brackets relates to expressions involving two *different* binary operations. It gives a rule for expanding a bracket or, equivalently, for factorizing an expression. This is the rule of arithmetic that says that, for all real numbers a, x, y, we have

$$a(x + y) = ax + ay.$$

It is known as the **distributive law**, and we say that *multiplication is distributive over addition.* We have two distributive laws in set theory, since union and intersection are each distributive over the other. They can be proved using either Venn diagrams or membership tables. The proofs are left as exercises.

Rule 2.21 *(Distributive laws). For any three subsets A, B, C of a universal set $\mathcal{U}$,*

(i) $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$;

(ii) $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$. □

2.4 Exercises 2

1. Describe the following sets by the *listing* method.
   (a) $\{r \in \mathbb{Z}^+ : r \leq 5\}$
   (b) $\{b \in \mathbb{Z} : -1 \leq b \leq 4\}\}$
   (c) $\{s : s$ is an odd integer and $2 \leq s \leq 10\}$
   (d) $\{2m : m \in \mathbb{Z}$ and $5 \leq m \leq 10\}$
   (e) $\{2^t : t \in \mathbb{Z}$ and $0 \leq t \leq 5\}$.

2. Describe the following sets by giving a suitable universal set and rules of inclusion (note: there is more than one correct answer in each case).
   (a) $\{12, 13, 14, 15, 16, 17\}$
   (b) $\{0, 5, -5, 10, -10, 15, -15, \ldots\}$
   (c) $\{-1, -2, -3, -4, \ldots -10\}$
   (d) $\{6, 8, 10, 12, 14, 16, 18\}$
   (e) $\{1, 5, 5^2, 5^3, 5^4, \ldots\}$
   (f) $\{0.1, 0.01, 0.001, 0.0001, \ldots\}$.

3. Which of the following sets are equal?
   $$\{x, y, z\}\ ; \qquad \{z, x, y\}\ ; \qquad \{x, y, x, y, z\}\ ; \qquad \{y, z, z, x\}.$$
   Give the cardinality of each.

4. For the set $V = \{a, e, i, o, u\}$, give the 5-bit binary string that codes each of the following subsets:
   $$\{a, i, o\}; \qquad \{e\}; \qquad V; \qquad \emptyset.$$
   Which subset is represented by the 5-bit string 10001?

5. Let $S = \{0, 1, 2, 3, 4, 5\}$, $A = \{2, 4, 5\}$. Put the correct sign, $\in$ or $\subseteq$, between each of the following pairs:
   $$A, S; \qquad 3, S; \qquad 0, S; \qquad \emptyset, S; \qquad S, \mathcal{P}(S); \qquad A, \mathcal{P}(S).$$

6. Draw a Venn diagram to represent the universal set $\mathcal{U} = \{0, 1, 2, 3, 4, 5, 6\}$ with subsets $A = \{2, 4, 5\}$ and $B = \{1, 4, 5, 6\}$. Find each of the following:
   $$A \cup B; \qquad A \cap B; \qquad (A \cup B)'; \qquad A - B; \qquad B - A; \qquad A \oplus B.$$

7. Let A, B be subsets of a universal set $\mathcal{U}$. Construct a membership table for the sets A and B and add columns for $A - B$, $(A - B)'$, A' and $A' \cup B$. Hence prove that $(A - B)' = A' \cup B$. Illustrate this result on a Venn diagram.

8. Let A, B be subsets of a universal set $\mathcal{U}$.

   (a) Use membership tables to prove De Morgan's Law that

   $$(A \cup B)' = A' \cap B'.$$

   (b) Use this law to show that

   $$(A' \cup B')' = A \cap B.$$

9. (a) Draw a Venn diagram to show three subsets A, B, C of a universal set $\mathcal{U}$ intersecting in the most general way. Shade the region corresponding to the subset X defined by the membership table below.

   (b) Repeat part (a) for each of the subsets Y and Z in place of X.

   (c) How are the sets X and Z related?

   (d) Can you describe each of the subsets X, Y and Z in terms of the sets A, B, C, using the operations union, intersection and set complement?

A	B	C	X	Y	Z
0	0	0	0	1	0
0	0	1	0	1	1
0	1	0	0	0	1
0	1	1	1	0	1
1	0	0	0	1	0
1	0	1	0	1	0
1	1	0	1	0	1
1	1	1	1	0	1

10. Let A, B, C be subsets of a universal set $\mathcal{U}$.

    (a) Construct membership tables for each of the sets $(A - B) - C$ and $A - (B - C)$.

    (b) Which, if any, of the following statements are true for all subsets A, B, C?

    (i) $(A - B) - C = A - (B - C)$

    (ii) $(A - B) - C \subseteq A - (B - C)$

    (iii) $A - (B - C) \subseteq (A - B) - C$.

Chapter 3

Logic

Summary

Proposition, truth set, tautology, contradiction; negation; joining propositions by *and*, *or*, *exclusive or*; truth table; conditional connectives and their truth tables; contrapositive; laws of logic; logic gate, logic network.

References: Epp Sections 1.1, 1.2, 1.4 *or* M&B Sections 2.5, 2.6, 2.7.

Introduction

Logical argument and deductive reasoning are central to mathematics, and we could not write or test the validity of a computer program without them. In this chapter, we consider the symbolic representation of statements and the laws of logic.

3.1 Symbolic Statements and Truth Tables

Learning Objectives

After studying this section, you should be able to:

- define the truth set of a given proposition;
- recognise when a given proposition is a tautology or a contradiction;
- state the negation of a given proposition;
- construct the truth table for the connectives *not*, *and*, *or* and *exclusive or*.

3.1.1 Propositions

The statements with which we are concerned are known as **propositions**. These are statements that are either true or false. Statements that could be considered true by one observer but simultaneously considered false by another observer are *not* propositions.

Example 3.1 The following sentences are propositions.

(a) This animal is a cat.

(b) This program is in C.

(c) The positive integer n is prime.

(d) The real number $\mathbf{x}$ is greater than 5.

The following sentences are not propositions.

(e) Are you coming to the Disco?

(f) Hurry up, then!

(g) My bag is heavy.

The statements (a) to (d) are either true or false, depending on the circumstances. The important thing to understand is that any two observers would agree about whether each of these statements is true or false in the same circumstances. The sentence (e) is a question and (f) is a command: these cannot be either true or false and so are not propositions. The sentence (g) is not a proposition because one observer might consider it true and another consider it false.

We shall denote propositions by lower case letters, such as p and q. We can define a **truth set** for each proposition. The truth set P for the proposition p contains all the circumstances under which p is true; its complement P' contains all the circumstances under which p is false.

Example 3.2 Suppose we toss a coin three times. Let p denote the proposition "The first toss is a head", and q denote the proposition "The third toss is a tail". The set of all possible outcomes (or results) of this experiment is

$$U = \{HHH, HHT, HTH, HTT, THH, THT, TTH, TTT\}.$$

The truth set for p is

$$P = \{HHH, HHT, HTH, HTT\}$$

and the truth set for q is

$$Q = \{HHT, HTT, THT, TTT\}.$$

Tautologies and contradictions

Propositions that are always true are called **tautologies** and those that are always false are called **contradictions**.

Example 3.3 The following propositions are tautologies.

(a) The result of tossing a fair coin is either a head or a tail.

(b) A square has four sides.

(c) $(x+1)^2 = x^2 + 2x + 1$.

The following propositions are contradictions.

(d) The number x is less than 3 and greater than 10.

(e) $x = x + 1$.

Proposition (a) is true for all properly minted coins, and is an an intrinsic property of such. Similarly, proposition (b) is true for all squares because it is part of the definition of the term *square*; all definitions are tautologies. Proposition (c) is an example of an algebraic identity. An **identity** is an equation where the right side is just a rearrangement of the left side, and hence all identities are examples of tautologies. The truth set for a tautology is therefore the universal set of the objects under discussion.

There is no value of x for which either of the propositions (d) or (e) is true and hence these propositions are both examples of *contradictions*. The truth set for a contradiction is always the empty set $\emptyset$.

3.1.2 The negation of a proposition

The **negation** of a proposition p, is the proposition that is true when p is false and false when p is true. We denote the negation of p by $\neg p$, read "not-p". The truth set for $\neg p$ is the complement of the truth set for p.

Example 3.4

(a) Let p denote the proposition "The integer n is prime"; then $\neg p$ is the proposition "The integer n is not prime".

(b) Let p denote the proposition "$x = x + 1$". Then $\neg p$ is the proposition "$x \neq x + 1$".

Notice that in Example 3.4(b), p is a contradiction and its negation is a tautology. This is a particular example of a general rule. Similarly, the negation of a tautology will always be a contradiction.

3.1.3 Compound statements

We can join two or more propositions together by such words as "and", "or", "if ... then", to form **compound** statements.

Example 3.5 We combine the propositions p and q of Example 3.2 to give the following compound statements.

(a) "The first toss is a head **and** the third toss is a tail" is denoted symbolically by $p \wedge q$.

(b) "The first toss is a head **or** the third toss is a tail" is denoted symbolically by $p \vee q$.

(c) "The first toss is a head **or** the third toss is a tail, **but not both**" is denoted symbolically by $p \oplus q$. □

Notice that **or** in mathematics is used in its inclusive sense, unless we specify otherwise, so that (b) includes the possibility that the first toss is a head *and* the third toss is a tail, whereas (c) does not. The symbol $\oplus$ is known as **exclusive or**.

Example 3.6 We have already found above the truth sets for the propositions p and q defined in Example 3.2. We now find the truth set for each of the compound statements (a), (b) and (c).

(a) The truth set for $p \wedge q$ is $\{HHT, HTT\} = P \cap Q$.

(b) The truth set for $p \vee q$ is $\{HHH, HHT, HTH, HTT, THT, TTT\} = P \cup Q$.

(c) The truth set for $p \oplus q$ is $\{HHH, HTH, THT, TTT\} = P \ominus Q$. □

3.1.4 Truth tables

We give each proposition a **truth value**, either 1 for true, or 0 for false. We may then determine the truth value of each of the compound statements $p \vee q$, $p \wedge q$, $p \oplus q$, from the truth values of their constituent propositions, p and q, by considering each combination of p true or false with q true or false. This is most easily done in the form of a **truth table**, as shown below.

p	q	$p \wedge q$	$p \vee q$	$p \oplus q$
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Figure 3.1.

We see that each of the compound statements is logically distinct, because it has its own distinct pattern of 0's and 1's. Further, this pattern is identical to the pattern of 0's and 1's in the

membership tables of $P \cap Q$ in the case of $p \wedge q$, of $P \cup Q$ in the case of $p \vee q$ and of $P \oplus Q$ in the case of $p \oplus q$. Thus we have:

Result 3.1 *Let P, Q be the truth sets for the propositions p and q respectively. Then the truth set for $p \wedge q$ is $P \cap Q$, for $p \vee q$ is $P \cup Q$ and for $p \oplus q$ is $P \oplus Q$.*

Definition 3.2 *When two statements p and q have the* same *truth table, they are called* ***logically equivalent*** *and we write $p = q$.*

Statements involving negation

Result 3.3 (Double negative law) $\neg(\neg p) = p$.

Proof. To prove this result, we construct the truth table for $\neg(\neg p)$ and compare it with the truth values of p.

p	$\neg p$	$\neg(\neg p)$
0	1	0
1	0	1

Figure 3.2.

We see that whatever the truth value of p, the truth value of $\neg(\neg p)$ is the same. This proves that $\neg(\neg p) = p$. □

Example 3.7 Let p denote the proposition "This program is in Pascal". Then $\neg p$ denotes the proposition "This program is not in Pascal", and $\neg(\neg p)$ denotes the proposition "It is not true that this program is not in Pascal". This means the same as "This program is in Pascal". □

The double negative in Example 3.7 makes the sentence awkward and its meaning less obvious. Rule 3.3 says that we can always avoid double negatives, and in the interests of clarity we should do so.

We also have to take great care when negating expressions involving the connectives *and* or *or*, as ordinary English usage is often rather inexact. We have the following equivalents of De Morgan's Laws for $\vee$ and $\wedge$.

Result 3.4 (De Morgan's laws) *For all propositions p and q, we have*

(i) $\neg(p \wedge q) = \neg p \vee \neg q$;

(ii) $\neg(p \vee q) = \neg p \wedge \neg q$.

Proof. These laws can be proved using truth tables. The general method is similar to the way we proved set laws using membership tables. As an example, we prove law (i) by constructing a truth table for each side of the equation. For the left side, we first construct a column for $p \wedge q$ and from this deduce the column for $\neg(p \wedge q)$. Similarly, for the right side, we first construct columns for $\neg p$ and $\neg q$, and from these deduce the column for $\neg p \vee \neg q$.

p	q	$p \wedge q$	$\neg(p \wedge q)$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

p	q	$\neg p$	$\neg q$	$\neg p \vee \neg q$
0	0	1	1	1
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

Figure 3.3

Since the columns corresponding to the statements $\neg(p \wedge q)$ and $\neg p \vee \neg q$ are identical, these two statements are logically equivalent. □

Result 3.4 states that

"not (p and q)" is the same as saying "not p or not q;"

and

"not (p or q)" is the same as saying "not p and not q."

Example 3.8 Let p be the proposition: "the last digit of n is 5" and q be the proposition: "the last digit of n is 0". Then $\neg(p \vee q)$ is the statement

"the last digit of n is not 0 or 5."

This has the same meaning the as:

"the last digit of n is not 0 and the last digit of n is not 5"

which is the statement $\neg p \wedge \neg q$. □

Tautologies and contradictions

All tautologies are logically equivalent because every tautology has only the truth value 1. Similarly, all contradictions are logically equivalent, since each takes only the truth value 0. We can therefore use the one symbol T to denote any tautology and the symbol F to denote any contradiction.

Example 3.9 The truth table for $p \vee T$ is shown in Figure 3.4. Note that we need only two rows for this table, because T has only one truth value.

p	T	$p \vee T$
0	1	1
1	1	1

Figure 3.4.

The table shows that $p \vee T = T$. This means that the compound statement

(proposition p) or (tautology)

is always true, whatever the truth value of p. □

3.2 The Conditional Connectives

Learning Objectives

After studying this section, you should be able to:

- construct the truth table for the conditional connectives *if*, *only if* and *if and only if*;
- interpret alternative ways of wording conditional statements;
- state the contrapositive of a given statement.

Introduction

In mathematics and when writing computer programs, as well as in everyday language, we often use the word "if" to connect two statements. However, "if" can be used in more than one way. Consider, for example, the following statements concerning an integer n.

(a) *If* $n = 17$, *then* n is greater than 10.

We sometimes say this with the "if" clause second, as:

(b) n is greater than 10 *if* $n = 17$.

But notice that in either case, the "if" introduces the statement "$n = 17$".

A connective with a different meaning is *only if*. In order to link the two propositions "$n = 17$" and "n is greater than 10" using *only if* with the same meaning as the statements (a) and (b), the "only if" must introduce the proposition "n is greater than 10". Thus we would say:

(c) $n = 17$ *only if* n is greater than 10.

Let p denote the proposition "$n = 17$" and q denote the proposition "n is greater than 10". Then the statements above can be written as

(a) *If* p, *then* q; (b) q *if* p ; (c) p *only if* q.

Without altering the meaning, we can rewrite each of these statements using the word *implies* as:

$n = 17$ *implies* n is greater than 10.

This statement is written symbolically as

$$p \rightarrow q.$$

In general, the statements $p \rightarrow q$ and $q \rightarrow p$ have different meanings. When we want to make both of these statements, we often use the connectives *if* and *only if* together. For example, the following statement concerns an integer n expressed in base 10:

(d) n is divisible by 10 *if and only if* the last digit of n is 0.

Let r denote the proposition "the last digit of n is 0" and s denote the proposition "n is divisible by 10". Then (d) can be written as

r *if and only if* s.

In this case, each of the propositions r and s imply the other, so (d) has the same meaning as

$$r \rightarrow s \text{ and } s \rightarrow r.$$

This compound statement is written as

$$r \leftrightarrow s.$$

3.2.1 Truth tables for $p \rightarrow q$ and $p \leftrightarrow q$

We consider the truth value of $p \rightarrow q$ first in the special case of the propositions concerning a positive integer n discussed in the previous section. Thus we let p denote the proposition "$n = 17$", and q denote the proposition "$n > 10$".

For this pair of propositions, the statement $p \rightarrow q$ is always true; that is, the statement "$n = 17$ implies $n > 10$" is true *whatever value of* n *we choose*. For example, if n happens to have the value 1, this does not make the statement "$n = 17$ implies $n > 10$" false.

To construct the truth table for $p \rightarrow q$, we need to find the truth value of $p \rightarrow q$ for each combination of p false or true with q false or true. Can we find an example of a value of n to illustrate each of these four possibilities?

Now p is true when $n = 17$ and false when $n \neq 17$; while q is true when $n > 10$ and false when $n \leq 10$. Thus we can construct the following table.

p	q	*example of* n
0	0	2
0	1	14
1	0	*none*
1	1	17

As we have observed, the statement $p \rightarrow q$ is true for all integers n. Thus $p \rightarrow q$ is true when both p and q are false, when p is false and q is true and when both p and q are true. However, it is

impossible to find a value of n that makes p true and q false. Thus this is the ONLY combination of truth values of p and q that would make the implications *false*.

Although we have determined the truth value of $p \to q$ only for a particular pair of propositions p and q, our conclusions hold in general: for any propositions p and q, $p \to q$ is false *only when* p *is true and* q *is false*. Thus we have the following truth table for $p \to q$.

p	q	$p \to q$
0	0	1
0	1	1
1	0	0
1	1	1

Figure 3.5.

To most people, this table does not seem to follow our intuition in the same way as the tables for $\neg$, $\wedge$ and $\vee$. You are therefore strongly advised to learn it carefully.

An alternative expression for $p \to q$

The truth table for $p \to q$ enables us to express this conditional statement by means of negations and the connective "or".

Result 3.5 $p \to q = \neg p \vee q$.

Proof. We use truth tables to show that $p \to q \equiv \neg p \vee q$. The first table is for the left side of this equation and the second table is for the right side.

p	q	$p \to q$
0	0	1
0	1	1
1	0	0
1	1	1

p	q	$\neg p$	$\neg p \vee q$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

Comparing the columns corresponding to the statements $p \to q$ and $\neg p \vee q$, we see they are identical. Hence these two expressions are logically equivalent. □

Truth tables for $q \to p$ and $p \leftrightarrow q$

From the table in Figure 3.5, we can deduce the truth table for $q \to p$, by interchanging the roles of p and q. Thus $q \to p$ is false only when q is true and p is false.

From the truth tables for $p \to q$ and $q \to p$, we can deduce the truth table for $p \leftrightarrow q$, because we have defined $p \leftrightarrow q$ as $(p \to q) \wedge (q \to p)$. Thus $p \leftrightarrow q$ is true only when either *both* p and q are true or *both* p and q are false.

p	q	$p \to q$	$q \to p$	$p \leftrightarrow q$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	1	1	1

Figure 3.6.

Result 3.5 gives us a way of expressing the truth set for the conditional statement $p \to q$ and hence for $q \to p$ and $p \leftrightarrow q$.

Result 3.6 *Let* P, Q *be the truth sets for propositions* p *and* q *respectively. Then the truth set for* $p \to q$ *is* $P' \cup Q$ *and the truth set for* $p \leftrightarrow q$ *is* $(P' \cup Q) \cap (P \cup Q')$. □

3.2.2 The contrapositive

Consider the following two statements.

1. If this rectangle is a square, then its sides are all equal.
2. If the sides are not all equal, then this rectangle is not a square.

The second statement is called the **contrapositive** of the first statement. We see that the first statement has the form

$$p \rightarrow q$$

and its contrapositive has the form

$$\neg q \rightarrow \neg p.$$

Note that since $\neg(\neg p) = p$ and $\neg(\neg q) = q$, the contrapositive of the statement $\neg q \rightarrow \neg p$, is the statement $p \rightarrow q$.

We can show that every conditional statement is logically equivalent to its contrapositive. This means that we can make either a statement or its contrapositive, with the same meaning.

Result 3.7 $\neg q \rightarrow \neg p = p \rightarrow q$.

Proof. We can prove this result by constructing a truth table for each side of the equation.

p	q	$p \rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	1

p	q	$\neg q$	$\neg p$	$\neg q \rightarrow \neg p$
0	0	1	1	1
0	1	0	1	1
1	0	1	0	0
1	1	0	0	1

Figure 3.7.

Since the columns for $p \rightarrow q$ and $\neg q \rightarrow \neg p$ in Figure 3.7 contain the same entries, these two statements are logically equivalent. □.

Example 3.10

(a) The contrapositive of the statement "If your ticket has been drawn, then you win a prize" is the statement: "If you don't win a prize, then your ticket has not been drawn".

(b) The contrapositive of the statement "If $n = 17$, then $n > 10$" is "If $n \leq 10$, then $n \neq 17$ ", where we have expressed "not greater than" as "less than or equal to".

3.3 Laws of logic

Learning Objectives

After studying this section, you should be able to:

- use the laws of logic to simplify a given expression.

Introduction

By applying the laws of set algebra to truth sets, we can deduce equivalent laws for manipulating compound statements; these are known as *the laws of logic*. These laws can be used to prove the logical equivalence of two statements as an alternative to constructing truth tables.

Sets	Propositions
Commutative Laws	
$P \cap Q = Q \cap P$	$p \wedge q \equiv q \wedge p$
$P \cup Q = Q \cup P$	$p \vee q \equiv q \vee p$
Associative Laws	
$(P \cap Q) \cap R = P \cap (Q \cap R)$	$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
$(P \cup Q) \cup R = P \cup (Q \cup R)$	$(p \vee q) \vee r \equiv p \vee (q \vee r)$
Distributive Laws	
$P \cap (Q \cup R) = (P \cap Q) \cup (P \cap R)$	$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
$P \cup (Q \cap R) = (P \cup Q) \cap (P \cup R)$	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
De Morgan's Laws	
$(P \cap Q)' = P' \cup Q'$	$\neg(p \wedge q) \equiv \neg p \vee \neg q$
$(P \cup Q)' = P' \cap Q'$	$\neg(p \vee q) \equiv \neg p \wedge \neg q$
Identity Laws	
$P \cap \emptyset = \emptyset;\ P \cup \emptyset = P$	$p \wedge F \equiv F;\ p \vee F \equiv p$
$P \cap \mathcal{U} = P;\ P \cup \mathcal{U} = \mathcal{U}$	$p \wedge T \equiv p;\ p \vee T \equiv T$
Absorption and Complement Laws	
$P \cap P = P;\ P \cup P = P$	$p \wedge p \equiv p;\ p \vee p \equiv p$
$P \cap P' = \emptyset;\ P \cup P' = \mathcal{U}$	$p \wedge \neg p \equiv F;\ p \vee \neg p \equiv T$

Example 3.11 We use the laws of logic to prove

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

First, by De Morgan's Law, we have

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

However, $\neg(\neg q) \equiv q$, by Result 3.3. Thus we have

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

as required. □

3.4 Logic Gates

Learning Objectives

After studying this section, you should be able to:

- draw block diagrams to represent the **NOT-gate**, the **AND-gate** and the **OR-gate**;
- construct a logic network to represent a given symbolic statement;
- obtain an expression for the output from a given logic network.

Introduction

A modern digital computer is often envisaged as a super-fast adding machine. It would be more accurate, however, to think of it as a logic machine, because it performs all the arithmetical operations by means of the logical rules summarized in the previous section. It does this by means of simple electronic circuits called **gates**. These are designed to operate on one or more input

values to output a corresponding value. Each input and output usually corresponds to either a high or a low voltage. We shall use the bit 1 to represent a high voltage, or the truth value "1" of a statement; and the bit 0 to represent a low voltage, or the truth value "0" of a statement. Recollect that all contradictions have truth value 0 and all tautologies have truth value 1.

We shall consider three basic gates, corresponding to the logical connectives $\neg$, $\wedge$ and $\vee$. They are called, respectively, the **NOT-gate**, the **AND-gate** and the **OR-gate**. They are usually depicted by block diagrams, as illustrated in Figure 3.8 below.

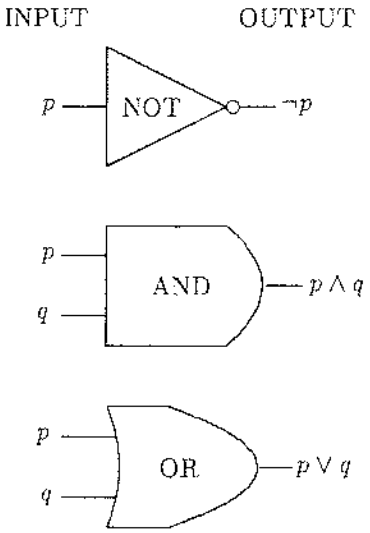


Figure 3.8.

The value of the output (0 or 1) for each value, or combination of values, for the input(s) can be derived from the logic tables for $\neg p$, $p \wedge q$ and $p \vee q$ (see Figures 3.1 and 3.2).

3.4.1 Designing logic networks

The logic gates can also be concatenated to represent more complicated compound statements. The circuits so formed are known as **logic networks**.

Example 3.12 We design a logic network that has two inputs p and q and outputs $p \rightarrow q$. To do this, we must use a compound statement that is logically equivalent to $p \rightarrow q$, but uses only the connectives $\neg$, $\wedge$ and $\vee$. In Result 3.5, we proved that $p \rightarrow q \equiv \neg p \vee q$. Using this fact, we can design a logic network with output $p \rightarrow q$, as shown in Figure 3.9. □

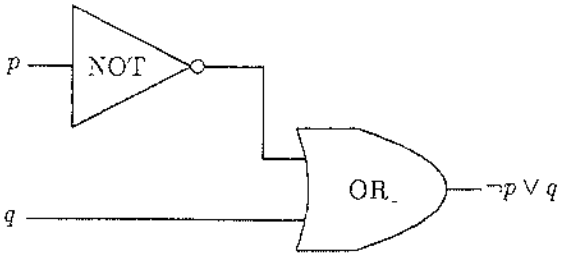


Figure 3.9.

Example 3.13 Suppose we require to devise a logic network with four inputs p, q, r, s and output

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

We use the same principle as in elementary algebra: deal with the brackets first. Suppose we denote the output of $p \wedge q$ by x and the output of $r \vee s$ by y. Then the required output is $x \rightarrow y$. The AND-gate gives output x for inputs p, q; the OR-gate gives output y for inputs r, s; the network shown in Figure 3.9 gives output $x \rightarrow y$, for inputs x and y.

Concatenating these components gives the network shown in Figure 3.10, where the final output $z = (p \wedge q) \rightarrow (r \vee s)$. □

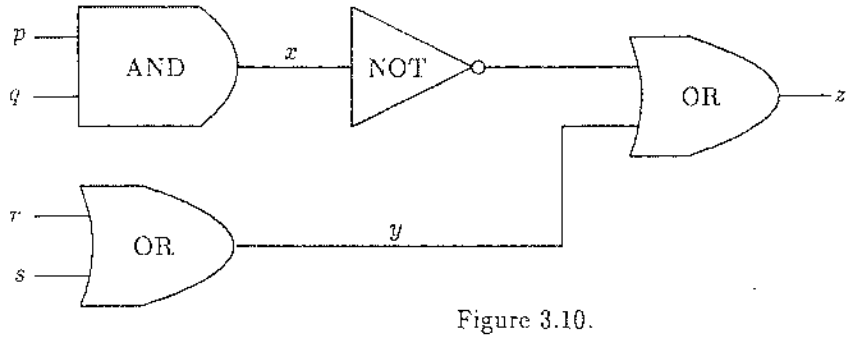


Figure 3.10.

3.4.2 Output of a given network

Suppose we are given the diagram of a logic network. We can reverse the process described above to obtain an expression for the output.

Example 3.14 We determine the output of the logic network shown in Figure 3.11. To do this, we work from left to right across the diagram, determining the output of each gate in turn. Note that the filled circles represent junctions where the inputs branch, whereas other points where lines cross represent bridges, where the inputs are insulated from one another.

We obtain:

1. Output $r = \neg q$.
2. Output $s = p \wedge \neg q$.
3. Output $t = p \wedge q$.
4. Output $w = (p \wedge \neg q) \vee (p \wedge q)$.

Hence the output of this network is $(p \wedge \neg q) \vee (p \wedge q)$. □

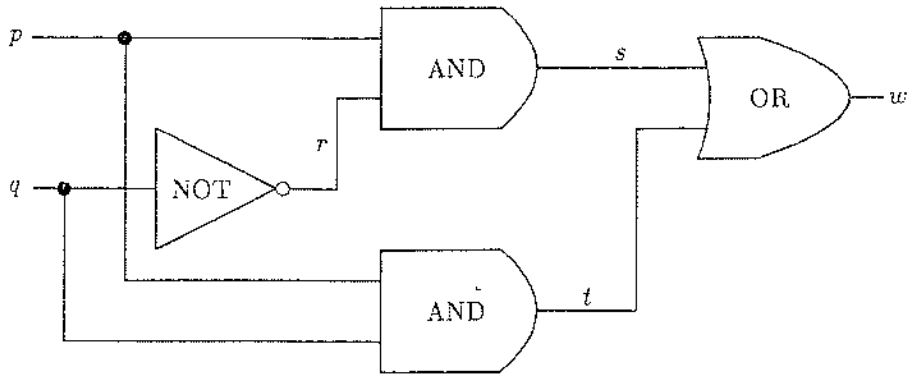


Figure 3.11.

We can next ask whether it is possible to simplify the network shown in Figure 3.11, by finding a network with fewer gates that gives the same output for every combination of inputs p and q. There

are two general methods for tackling this type of problem; they are illustrated in the following two examples.

Example 3.15 We use truth tables to simplify $w = (p \wedge \neg q) \vee (p \wedge q)$.

p	q	$\neg q$	$p \wedge \neg q$	$p \wedge q$	$(p \wedge \neg q) \vee (p \wedge q)$
0	0	1	0	0	0
0	1	0	0	0	0
1	0	1	1	0	1
1	1	0	0	1	1

From the table, we see that the column for the output $w = (p \wedge \neg q) \vee (p \wedge q)$ is identical to the column for p. Hence this network can be replaced by the input p, and no input q or gates are necessary. □

Example 3.16 We use the laws of logic to simplify the expression for the final output $w = (p \wedge \neg q) \vee (p \wedge q)$.
First note that we have "$p\wedge$(something)" in both brackets. By the distributive law, we can take "$p\wedge$" out of these brackets as "a common factor" and write:

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

Now $(\neg q \vee q) = (q \vee \neg q) = T$, by the commutative law and the complement law. Thus

$$w = p \wedge T.$$

However, $p \wedge T = p$, by the identity law. Hence we obtain $w = p$, as in the previous example. □

3.5 Exercises 3

1. The following propositions relate to a 3-bit binary string s.

   p: *Only one bit of s is 0.*
   q: *The first two bits of s are the same.*

   Find the truth set for each of the following statements:

   $$p; \qquad q; \qquad p \wedge q; \qquad p \vee q.$$

2. Let p, q denote the following propositions concerning an integer n.

   $$p\text{: } n \leq 50; \qquad q\text{: } n \geq 10.$$

   Express in words, as simply as you can, the following statements as conditions on n.

   $$\neg p; \qquad p \wedge q; \qquad \neg(\neg q); \qquad \neg p \vee \neg q.$$

3. Let p, q be the following propositions.

   p: *This book is on Databases.*
   q: *This book is on Programming.*

   Express each of the following compound statements symbolically in TWO different ways:

   (a) This book is not on Databases or Programming.

   (b) This book is not on Databases and Programming.

4. Use truth tables to prove that $(p \wedge q) \vee (\neg p \wedge q) = q$. (Hint: you will need to construct columns for p, q and $p \wedge q, \neg p, \neg p \wedge q, (p \wedge q) \vee (\neg p \wedge q)$. Remember to make a comment at the end to say why the table proves that the two statements are logically equivalent.)

5. Construct a truth table for each of the following compound statements and hence find simpler propositions to which each is equivalent.

$$\text{(a) } p \vee F; \qquad \text{(b) } p \wedge T.$$

6. Let p, q, r be the following propositions concerning an integer n.

$$p\text{: } n = 20; \qquad q\text{: } n \text{ is even}; \qquad r\text{: } n \text{ is positive.}$$

Express each of the following conditional statements symbolically, using the symbol $\rightarrow$.

(a) If $n = 20$, then n is positive.

(b) n is even if $n = 20$.

(c) $n = 20$ only if n is even.

7. Let q and r be the propositions defined in the previous exercise. Complete the following table by giving the truth value of each of the statements $q, r, q \rightarrow r, r \rightarrow q$ and $q \leftrightarrow r$ corresponding to each value of n.

n	q	r	$q \rightarrow r$	$r \rightarrow q$	$q \leftrightarrow r$
-8					
-3					
10					
17					

8. Use truth tables to prove that $\neg p \leftrightarrow \neg q$ is logically equivalent to $p \leftrightarrow q$.

9. Write the contrapositive of each of the following statements.

(a) If $n = 12$, then n is divisible by 3.

(b) If $n = 5$, then n is positive.

(c) If the quadrilateral is a square, then its four sides are equal.

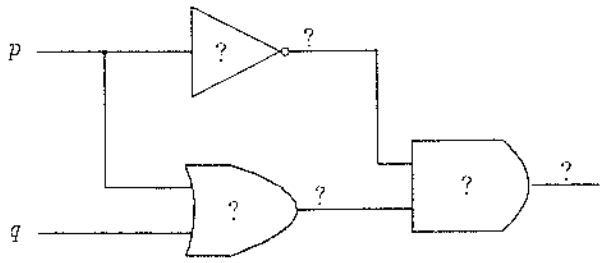
10. The basis for logical argument is that given propositions p, q, r such that p imples q and q implies r, then we can deduce that p implies r. The validity of this argument depends upon the fact that the statement $[(p \rightarrow q) \wedge (q \rightarrow r)] \rightarrow (p \rightarrow r)$ is always true, that is, it is a tautology.

Construct a truth table with columns for p, q, r and $p \rightarrow q, q \rightarrow r, (p \rightarrow q) \wedge (q \rightarrow r), p \rightarrow r, [(p \rightarrow q) \wedge (q \rightarrow r)] \rightarrow (p \rightarrow r)$. Hence prove that

$$[(p \rightarrow q) \wedge (q \rightarrow r)] \rightarrow (p \rightarrow r)$$

is indeed a tautology.

11. The following logic network accepts inputs p and q, which may each independently have the value 0 or 1.



(a) Copy the network and label each of the gates appropriately with one of the words "NOT", "AND" or "OR". Label the diagram also with a symbolic expression for the output from *each* gate.

(b) Construct a logic table to show the value of the output corresponding to each combination of values (0 or 1) for the inputs p and q.

(c) Find a simpler expression that is logically equivalent to the final output.