UNIVERSITY
OF LONDON

Mathematics for computing
Volume 1

C.A. Whitehead

CO1102

2004

Undergraduate study in
Computing and related programmes

Goldsmiths
UNIVERSITY OF LONDON

This guide was prepared for the University of London by:

C.A. Whitehead

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.

In this and other publications you may see references to the 'University of London International Programmes', which was the name of the University's flexible and distance learning arm until 2018. It is now known simply as the 'University of London', which better reflects the academic award our students are working towards. The change in name will be incorporated into our materials as they are revised.

University of London
Publications Office
32 Russell Square
London WC1B 5DN
United Kingdom

london.ac.uk

Published by: University of London

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

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

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**i**n**ary 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

			1	
1	0	1	1	
1	1	1	1	
		1	1	
				1

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

1			1	
1	0	1	1	
1	1	1	1	
		1	1	
			0	1

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

1	1			1	
	1	0	1	1	
	1	1	1	1	
				1	1
1	1	1	0	1	

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.

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

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 $100x$.

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

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 $10x$.

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

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.

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

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

$$1.888 < 17/9 < 1.889. \text{ □}$$

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 $(A5D)_{16}$ 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 $(a_2a_1a_00)_{16}$ and $(a_2a_1a_0A)_{16}$, respectively. Can you express x and y in terms of a in base 10?

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

11. (a) Write the hex number $(EC4)_{16}$ 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 $(B.25)_{16}$ 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.

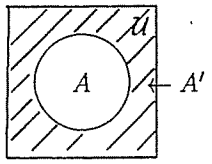


Figure 2.1.

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

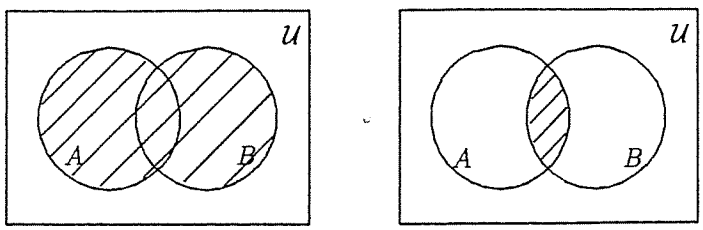
The remaining operations that we define in this section combine *two* subsets and are in consequence known as **binary operations**.

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*

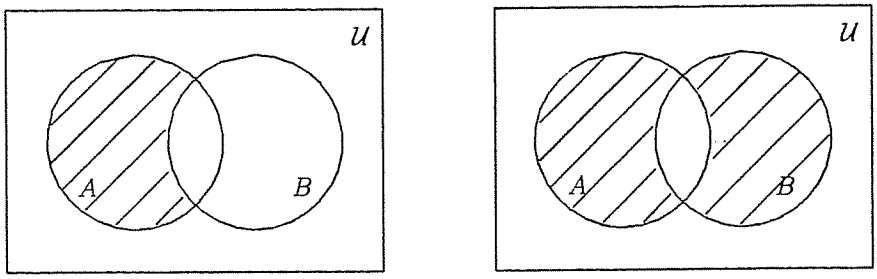
$$A - B = \{x : x \in A \text{ and } x \notin B\}.$$

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.

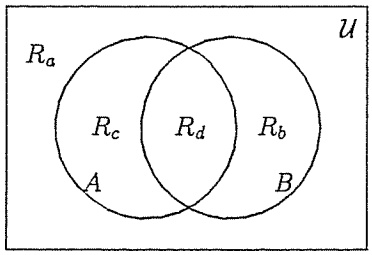


Figure 2.4.

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.

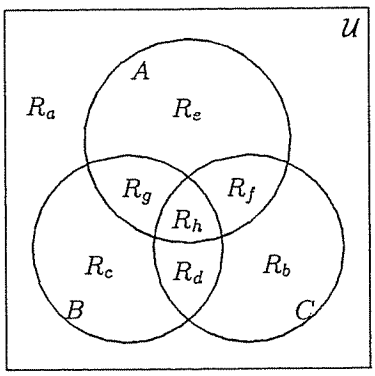


Figure 2.7.

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 2.6). 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) $\{s : s \text{ is an odd integer and } 2 \leq s \leq 10\}$
   (d) $\{2m : m \in \mathbb{Z} \text{ and } 5 \leq m \leq 10\}$
   (e) $\{2^t : t \in \mathbb{Z} \text{ and } 0 \leq t \leq 5\}$.

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 $\{HHH, HTH, THT, TTT\} = P \oplus Q$. □

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

Example 3.11 We use the laws of logic to prove

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

First, by De Morgan's Law, we have

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

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

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

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.

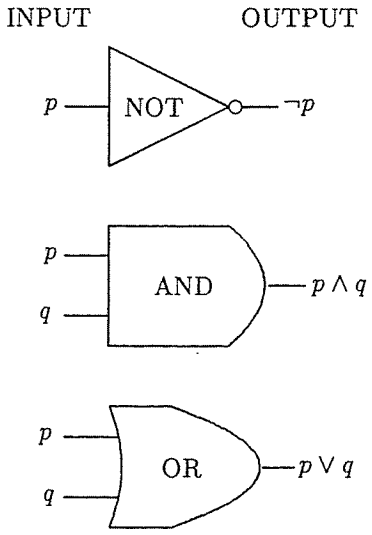


Figure 3.8.

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 $p \rightarrow q = \neg p \vee q$. Using this fact, we can design a logic network with output $p \rightarrow q$, as shown in Figure 3.9. □

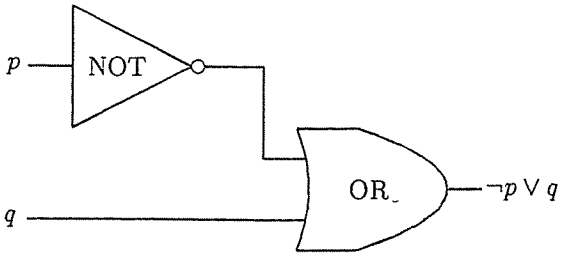


Figure 3.9.

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

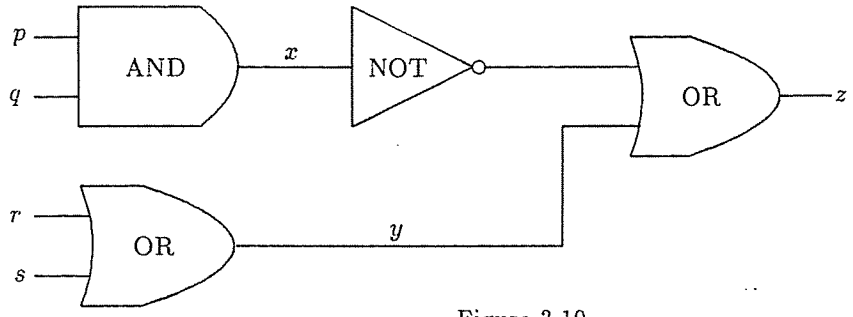


Figure 3.10.

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

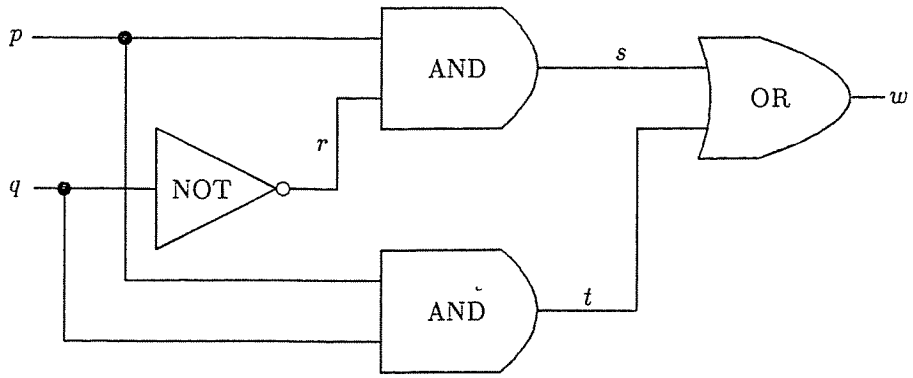


Figure 3.11.

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.

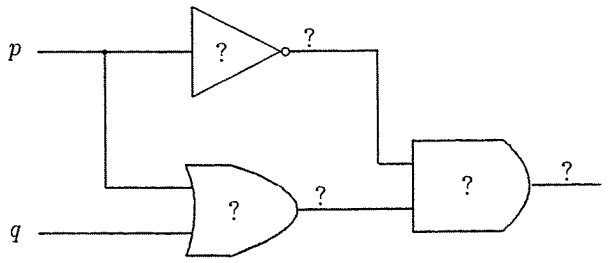
10. The basis for logical argument is that given propositions p, q, r such that p imples q and q implies r, then we can deduce that p implies r. The validity of this argument depends upon the fact that the statement $[(p \rightarrow q) \wedge (q \rightarrow r)] \rightarrow (p \rightarrow r)$ is always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

Chapter 4

Functions

Summary

Definition of function, domain, co-domain, image, ancestor, range; represention of functions by tables, formulae, arrow diagrams; Boolean functions; absolute value function; floor and ceiling functions; polynomial functions; *onto* and *one-to-one* properties, one-to-one correspondence; inverse function; exponential and logarithmic functions; O-notation; power functions; finding the order of a polynomial function; comparison of algorithms.

References: Epp Sections 7.1, 3.5, 5.1 (pp 237-239), 7.3, 9.1, 9.2, 9.4 *or* M&B Sections 5.2, 5.3, 5.4, 5.5.

Introduction

Functions are important in all branches of mathematics and computer science and you will come across many examples of them. In this chapter we shall introduce a variety of useful functions and discuss some of their properties.

4.1 What is a function?

Learning Objectives

After studying this section, you should be able to

- say what is meant by a *function* and determine whether a given relation represents a function;
- define the terms *domain*, *co-domain*, *image*, *ancestor* and *range* of a function and interpret these terms for a given function represented by a table, formula or arrow diagram;
- define and interpret the absolute value function;
- define and interpret the floor and ceiling functions and deduce some simple properties of these functions;
- interpret the output from a symbolic expression as a Boolean function;
- interpret and use ordered pairs and ordered n-tuples;
- be able to say whether a given polynomial function is a constant, linear or quadratic function, or none of these.

Introduction

An intuitive way to think of a *function* is to imagine that it is a *machine* that accepts various *inputs* and transforms each input into a uniquely determined *output*.

Different functions will accept different types of input. We can think of each function as acting on a clearly defined set of inputs. These may be numbers, but can be many other quantities as well.

Example 4.1 A "machine" that accepts as inputs the family names of all the students on Goldsmiths College database and outputs the initial letter of each family name is an example of a *function*. For each family name, there is just one initial letter and so for each input, there is a uniquely determined output. A convenient set containing all possible outputs is the set of letters of the alphabet and the set of actual outputs is some subset of this. Note that it often happens with a function that several different inputs each give the *same output*, as will certainly occur in this example. □

Example 4.2 A Maths test is scored out of 60, where each score awarded is an integer. The following table is used to transform the students' scores into grades.

score	0-20	21-23	24-29	30-35	36-41	42-60
grade	F	E	D	C	B	A

This table defines a function because given any score between 0 and 60 as input, the table determines just one possible grade as output. The set of inputs for this function is the set of integers $\{0, 1, 2, ..., 60\}$ and the set of outputs is $\{A, B, C, D, E, F\}$. □

These examples illustrate that the definition of a function has three parts:

1. a set of inputs;
2. a rule that determines the unique output corresponding to each input;
3. a set containing all outputs.

We usually denote the rule for a function by a single letter, such as f or g, for example, but some common functions are denoted by character strings such as log, exp, ABS.

Definition 4.1 *Suppose we are given two sets X and Y and a rule f such that given any element $x \in X$, f assigns to x a unique element $y \in Y$. Then we call f a* **function of X into Y**, *written $f : X \to Y$. We denote the unique element y assigned to x by $y = f(x)$. The set X (the set of inputs) is called the* **domain** *of the function and the set Y (containing the outputs) is called the* **co-domain** *of the function. The set of actual outputs is a subset of Y called the* **range** *of the function. Thus*

$$\text{range of } f = \{f(x) : x \in X\}.$$

Definition 4.2 *Let $f : X \to Y$ be a function of X into Y. We call $f(x)$ the* ***image of x under f*** *and we call x an* ***ancestor (or pre-image)*** *of $f(x)$. We sometimes say that f maps x* **onto** *$f(x)$.*

Example 4.3 Suppose S is the set of all 3-bit binary strings. Let $SUM : S \to \mathbb{Z}$ be the rule defined by

$$SUM(s) = \text{ the number of ones in } s.$$

Then SUM accepts any 3-bit binary string as input and outputs the number of ones in the string. The rule SUM is a function with domain S and co-domain $\mathbb{Z}$. We find

(a) the image of the strings 010 and 111;

(b) the set of ancestors of each of the outputs 0, 2 and 5;

(c) the range of SUM.

(a) The image 010 is the number of ones in the string 010, which is 1. Similarly, the image of 111 is the number of ones in 111, which is 3.

(b) The only 3-bit string s containing no ones is 000. Hence the set of ancestors of 0 is $\{000\}$. There are three 3-bit strings containing 2 ones, and so 2 has the set of ancestors $\{110, 101, 011\}$. There is no 3-bit string containing 5 ones, and hence the set of ancestors of 5 is $\emptyset$.

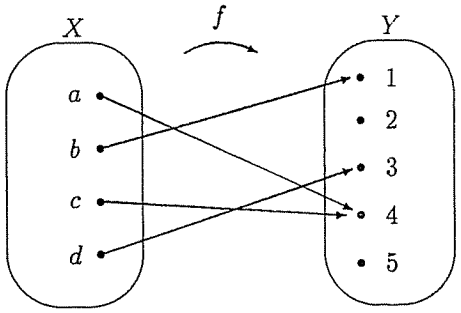
(c) The outputs of this function with the given set of inputs are $0, 1, 2, 3$. Hence the range of SUM is the set $\{0, 1, 2, 3\}$. □

4.1.1 Arrow diagram of a function

We have seen already that a function rule can be specified in several different ways. It can be stated in words, such as the rule in Example 4.1; or given by a table as in Example 4.2; or it can be given by a formula, such as $f(x) = 3x^2$, for example. Another method is to draw an *arrow diagram.*

We draw two disjoint sets to represent the domain and co-domain of the function and an arrow to link each element (or a typical element) in the domain with its image in the codomain.

Example 4.4 The arrow diagram below defines a function f with domain $X = \{a, b, c, d\}$ and co-domain $Y = \{1, 2, 3, 4, 5\}$.



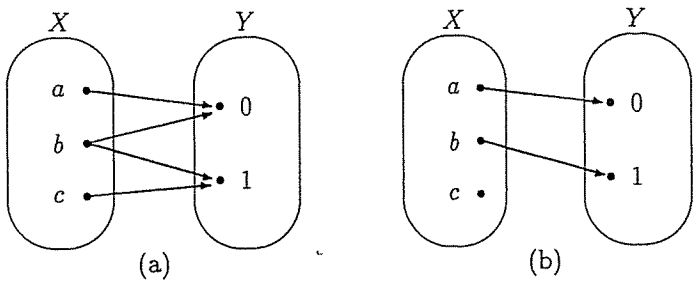
Note that the range of f is the subset of elements in the co-domain that receive an incoming arrow. To find the ancestors of a point in the range, we trace the arrow(s) it receives backwards. For example:

(a) the image of a is 4 and the image of d is 3;

(b) the set of ancestors of 4 is $\{a, c\}$ and of 5 is $\emptyset$;

(c) the range of f is the set $\{1, 3, 4\}$. □

Example 4.5 The following arrow diagrams do *not* represent functions.



The left hand diagram does not represent a function because the element b in the domain *has more than one image.* The right hand diagram does not represent a function because the element c *has no image.* □

The preceding examples illustrate that an arrow diagram needs to possess the following features in order to represent a *function.*

1. Every point in the domain set must have an arrow directed out of it.
2. No point in the domain set can have two arrows directed out of it going to different points of the co-domain.

Defining a function by a table of values

Every function defined by an arrow diagram could equally well be defined by a table of values.

Example 4.6 The function in Example 4.4 could be defined by the following table of values.

x	a	b	c	d
$f(x)$	4	1	4	3

. □

A table represents a function if it has the following properties:

1. the top row of the table must contain *every* element of the domain;
2. there must be exactly one entry in the second row of the table corresponding to each entry in the top row.

When a table defines a function, the set of elements appearing in the second row of the table is the range of the function.

4.1.2 Boolean functions and ordered n-tuples

In our study of logic, we used a table to define the output from a logic network in terms of certain inputs, represented by variables which can each independently take the values 0 or 1. A variable that is restricted to taking just two values is called a **Boolean variable**. We now describe how the input/output table for a logic network can be used to define a function.

Example 4.7 The simplest example is the NOT-gate, that accepts as input a single Boolean variable p and outputs $\neg p$. Let B denote the set of bits $\{0,1\}$. Then we can think of this gate as representing the function $NOT : B \to B$, defined by $NOT(p) = \neg p$, or equivalently, by the table shown below.

p	0	1
$NOT(p)$	1	0

. □

The input to a logic network generally depends on the value of more than one variable. Suppose, for example, that the input consists of the values of two variables p and q. Then we can represent each input by a *pair* of numbers, the first giving the value of p and the second giving the value of q, very much as we represent a point on a cartesian diagram by a pair of co-ordinates. Thus the input ($p = 1$ and $q = 0$) can be represented by the pair $(1,0)$, for example.

As we have seen, the order in which the elements of a set are listed inside brace brackets does not matter and we eliminate repetitions. Thus, for example, $\{0,1\} = \{1,0\}$ and $\{0,0\} = \{0\}$. It is therefore important to use a different notation for paired inputs. We reserve *round brackets* to denote a set where the *order* in which the elements are listed is important and repetitions of values may occur.

Definition 4.3 *Given two elements x, y (not necessarily distinct), the* **ordered pair** (x, y) *denotes the ordered list: first x, then y. In general, given n elements, $x_1, x_2, ..., x_n$, not necessarily distinct, the* **ordered n-tuple** $(x_1, x_2, ..., x_n)$ *denotes the ordered list: first x_1, then x_2, and so on up to x_n. When $n = 3$, an ordered n-tuple is called an* **ordered triple**.

Example 4.8 Suppose a logic network accepts inputs p and q and gives the output $\neg p \vee q$. We can denote the set of inputs as the set of ordered pairs $S = \{(0,0),(0,1),(1,0),(1,1)\}$, where the first component of each pair gives the value of p and the second component gives the value of q. The

output is always an element of the set $B = \{0, 1\}$.. The table below defines a function $f : S \to B$ that has the same output for each input as the logic network for $\neg p \vee q$.

(p, q)	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
$f(p, q)$	1	1	0	1

. □

Definition 4.4 *Let S denote the set of all ordered n-tuples $(x_1, x_2, ..., x_n)$, where $x_1, x_2, ..., x_n$ are Boolean variables. A function with domain S and co-domain $\{0, 1\}$ is called a* **Boolean function**.

The functions described in Example 4.7 and Example 4.8 are examples of *Boolean functions.*

4.1.3 Absolute value function

The *absolute value* of a real number x is the *size* of the number x irrespective of its *sign*. We often denote this value by $|x|$. Thus, for example, $|-6.5| = 6.5$ and $|6.5| = 6.5$, also. We obtain the absolute value of a negative number simply by changing its sign to positive, whereas the absolute value of a positive real number and zero is just the number itself. Given any real number x, we can visualise $|x|$ geometrically as representing the *distance* of the point representing x from the point representing 0 on the number line.

Example 4.9 We can define a function $ABS : \mathbb{R} \to \mathbb{R}$ by the rule

$$ABS(x) = \begin{cases} x, \text{ when } x \geq 0; \\ -x, \text{ when } x < 0. \end{cases}$$

Thus the function ABS has domain $\mathbb{R}$, co-domain $\mathbb{R}$, and its range is the set of non-negative real numbers. □

4.1.4 Floor and Ceiling Functions

In discrete mathematics, we often require an integer as an answer to a problem where perhaps we have calculated the exact solution as a fraction or even as an irrational number.

Every real number x that is not itself an integer lies strictly between two integers: one just below x in value, which we call the **floor** of x; and one just above x in value, which we call the **ceiling** of x. When x is an integer, then the floor of x and the ceiling of x are both equal to x itself. The floor of x is denoted by $\lfloor x \rfloor$ and the ceiling of x by $\lceil x \rceil$. Formally, the floor and ceiling functions are defined as follows.

Definition 4.5 *Given any real number x, $\lfloor x \rfloor$ is the unique integer n such that*

$$n \leq x < n + 1.$$

Definition 4.6 *Given any real number x, $\lceil x \rceil$ is the unique integer n such that*

$$n - 1 < x \leq n.$$

Example 4.10 We illustrate the definitions by finding the floor and ceiling of each of the numbers (a) π; (b) $-7/3$; (c) 25.

(a) The value of π is given by 3.14, correct to 2 decimal places. Therefore $3 < \pi < 4$, giving $\lfloor \pi \rfloor = 3$ and $\lceil \pi \rceil = 4$.

(b) We have $-3 < -7/3 < -2$. Hence $\lfloor -7/3 \rfloor = -3$ and $\lceil 7/3 \rceil = -2$.

(c) Since 25 is an integer, we have $\lfloor 25 \rfloor = \lceil 25 \rceil = 25$. □

Example 4.11 The coaches for a company's annual excursion have a maximum capacity of 45 passengers. Suppose n people want to go on the excursion. How many coaches should the company hire?

Since each coach takes 45 people, we divide n by 45. However, if this does not go exactly, we shall have to round up the number of coaches hired to the nearest integer above $45/n$, in order to accommodate the remaining passengers. Thus the number of coaches required is $\lceil n/45 \rceil$. □

Example 4.12 How many of the integers $1, 2, 3, \ldots, 99$ are even? How many are divisible by 5?

First note that starting from 1, every second integer in the list is even. Hence the number of even integers in the list is found by dividing 99 by 2 and *discarding* any remainder. The required number is therefore $\lfloor 99/2 \rfloor = \lfloor 49.5 \rfloor = 49$.

Similarly, every fifth integer in the list is divisible by 5. Hence to find the number of integers divisible by 5, we calculate $\lfloor 99/5 \rfloor = \lfloor 19.8 \rfloor = 19$. □

The floor and ceiling of a real number can be used to define two functions, each with domain $\mathbb{R}$ and range $\mathbb{Z}$, called the **floor function** and the **ceiling function** respectively. Formally, $FLOOR : \mathbb{R} \to \mathbb{Z}$ and $CEIL : \mathbb{R} \to \mathbb{Z}$ are defined by the rules $FLOOR(x) = \lfloor x \rfloor$ and $CEIL(x) = \lceil x \rceil$.

4.1.5 Polynomial functions

Example 4.13 The function $f : \mathbb{R} \to \mathbb{R}$ defined by the rule $f(x) = 5$ accepts each real number x as input and outputs the number 5. This is an example of a **constant function**, so called because the output has the same value (in this case 5), no matter what the input. Thus the range of f is $\{5\}$. □

Example 4.14 The function $f : \mathbb{R} \to \mathbb{R}$ defined by the rule $f(x) = 2x + 5$ accepts each real number x as input and outputs the number $2x + 5$. This is an example of a **linear function**. As an illustration, we find the following:

(a) the image of 4.3;

(b) the ancestors of 17.

(a) The image of 4.3 is $f(4.3) = 2(4.3) + 5 = 13.6$.

(b) To find the ancestors of 17, we solve the equation

$$f(x) = 17.$$

Since $f(x) = 2x + 5$, this equation becomes 2x+5=17, giving x=6. Since this is the only solution to this equation, 6 is the only ancestor of 17. □

Definition 4.7 *A function f defined by the rule $f(x) = ax + b$, where a and b are fixed numbers and $a \neq 0$, is called a* **linear function**. *It accepts a number x from some given domain as input and transforms x into the number $ax + b$.*

Example 4.15 The function $f : \mathbb{Z} \to \mathbb{Z}$ that accepts as input an integer n and outputs the integer $2n + 1$ is another example of a linear function. Its domain is $\mathbb{Z}$, the co-domain is also $\mathbb{Z}$, and its range is the set of odd integers. □

Definition 4.8 *A function f which accepts a number x as input and transforms x into an output $f(x) = ax^2 + bx + c$, where a, b and c are fixed numbers and $a \neq 0$, is called a* **quadratic function**.

Example 4.16 The function $f : \mathbb{R} \to \mathbb{R}$ defined by the formula $f(x) = x^2$ takes each real number x as input and outputs x^2. This is an example of a quadratic function with domain $\mathbb{R}$. The range of this function is $\{x \in \mathbb{R} : x \geq 0\}$. □

Example 4.17 The function $f : \mathbb{Z} \to \mathbb{Z}$ defined by the rule $f(r) = r^2 + r$ takes each integer r as input and outputs $r^2 + r$. This is another example of a quadratic function, in this case with domain $\mathbb{Z}$. The image of each integer will also be an integer by this rule, and hence the range is a subset of $\mathbb{Z}$. The co-domain is therefore chosen as $\mathbb{Z}$. □

Definition 4.9 *A function f which accepts a number x as input and transforms x into an output calculated using an expression of the form $a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$, where $n \geq 0$ is an integer and $a_0, a_1, \ldots a_{n-1}, a_n$ are fixed numbers with $a_n \neq 0$, is called a* **polynomial function.** *The output $a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$ is called a* **polynomial** *in x. The integer n is called the* **degree** *of the polynomial.*

We have just met three different types of polynomial functions that have special names: *non-zero constant functions* are polynomial functions of degree 0; *linear functions* are polynomial functions of degree 1 and *quadratic functions* are polynomial functions of degree 2. Polynomial functions of degree 3 also have a special name: they are called **cubic** functions. Polynomial functions are important in computer science for calculating the number of steps it takes to run a given algorithm when we know the size of the set of inputs. We shall discuss this in more detail in the last section of this chapter.

4.1.6 Equality of functions

In order to specify a function, it is necessary to state the domain and co-domain as well as giving the rule for determining the output corresponding to each input.

Definition 4.10 *We shall say that two funtions f, g are* **equal** *if and only if they have the same domain, the same co-domain and, for each element x in the domain,*

$$f(x) = g(x).$$

A consequence of this definition is that if we change either the domain or the co-domain of a function, then we change the function itself, and it is necessary to denote it by a new name or letter.

4.2 Functions with Special Properties

Learning Objectives

After studying this section, you should be able to

- state the conditions for a function to be *onto*, *one-to-one*, a *one-to-one correspondence* and determine whether a given function has any of these properties;
- state the conditions for a function to be invertible and find the inverse of a given invertible function in simple cases.

Introduction

We often want to reverse a function, so that starting from an element of the co-domain, we can track it back and find its ancestor in the domain. In this section, we shall see that this cannot always be done and, when we can find an ancestor of the given element, it may not be unique. However, when the function has two very useful properties, we can always reverse it and work back to a unique ancestor for each element of the co-domain. Before we discuss these properties, we look at the problem of reversing a function in the context of sending secret messages.

4.2.1 Encoding and decoding functions

A simple example of a secret code can be formed by assigning a unique number (or other symbol) to each letter of the English alphabet. Such an assignment is an example of a function. It has the set A of 26 letters of the English alphabet as its domain and the set of numbers (or other symbols) used for the codes as its co-domain. We shall call such a function an *encoding* function. To keep things simple in the following examples, we use the set of 26 two digit numbers $S = \{01, 02, \ldots 26\}$ to encode the letters of the alphabet.

Example 4.18 The function $\alpha : A \to S$ specified by the following table is an example of an encoding function.

x	a	b	c	d	e	f	g	h	i	j	k	l	m
$\alpha(x)$	21	16	11	02	15	12	01	13	18	25	04	09	17
x	n	o	p	q	r	s	t	u	v	w	x	y	z
$\alpha(x)$	05	24	19	07	22	08	23	10	14	20	06	26	03

Example 4.19 In order to save time deciphering messages sent using the encoding function α defined in Example 4.18, we can compile a *decoding* function $\beta : S \to A$, which inputs each number in S and outputs the letter it encodes. The table for β (with the elements of S listed in their natural order) is given below.

y	01	02	03	04	05	06	07	08	09	10	11	12	13
$\beta(y)$	g	d	z	k	n	x	q	s	l	u	c	f	h
y	14	15	16	17	18	19	20	21	22	23	24	25	26
$\beta(y)$	v	e	b	m	i	p	w	a	r	t	o	j	y

Example 4.20 Now consider the function $\gamma : A \to S$ defined by the following table.

x	a	b	c	d	e	f	g	h	i	j	k	l	m
$\gamma(x)$	21	16	16	02	15	12	01	13	18	25	04	09	17
x	n	o	p	q	r	s	t	u	v	w	x	y	z
$\alpha(x)$	05	24	19	07	22	08	23	10	14	20	06	26	03

The function $\gamma : A \to S$ differs from $\alpha : A \to S$ in two respects: whereas each element of S has a *distinct* image under α, we have $\gamma(b) = \gamma(c) = 16$; furthermore, whereas every pair of digits between 01 and 26 occur in the range of α, 11 does not occur in the range of γ. Now this may lead us into difficulties when we come to decode a message containing either of the pairs of digits 11 or 16. In the first case, we do not know what letter is represented by 11 and in the second case, we do not know whether 16 is coding b or c.

These ideas illustrate why, for a given function, there are two questions that we often wish to ask.

1. Does every element of the co-domain have an ancestor in the domain? That is, given any element of the co-domain, is there an input that will give this element as an output? This is the same as asking whether the range equals the co-domain.
2. Do distinct elements in the domain have distinct images, or can we find two different elements in the domain that have the *same* image? That is, can we find two different inputs that give the same output?

4.2.2 Onto functions

Definition 4.11 *If every element in the co-domain of the function has an ancestor, then the function is said to be* **onto**. *An onto function has the property that its range equals its co-domain.*

Example 4.21 The function $\alpha : A \to S$ defined in Example 4.18 is an onto function, since every element of S is in its range (that is, every element of S occurs in the second row of the table). However, the function $\gamma : A \to S$ defined in Example 4.20 is not onto, because 11 has no ancestor.

Example 4.22 The function for translating scores into literal grades defined in Example 4.2 is an example of an onto function. The Boolean functions described in Example 4.7 and Example 4.8 are also both onto.

To test whether the function $f : X \to Y$ is an onto function, we must see whether it is possible to find an ancestor of *every* element in the domain. If we can find just one example of an element $y \in Y$ that does not have an ancestor, then we can say that f is not an onto function. Such an example is called a **counter-example**.

Testing functions with a finite domain for the onto property

The best way to investigate a function with a finite domain is usually to find the range of the function from a table or arrow diagram and then to compare it with the co-domain.

Example 4.23 The function illustrated by an arrow diagram in Example 4.4 is not an onto function. A counter-example is provided by the element 3 in the co-domain which has no ancestor (an alternative counter-example is given by the element 5). □

Testing functions with an infinite domain for the onto property

In this case, a general method for deciding whether each element in the co-domain has an ancestor is to solve an equation. We let y denote a "typical" element in the co-domain and try to find $x \in X$ such that $f(x) = y$. If it is possible to find such an x *whatever the value of* y, then f is **onto**.

Example 4.24 Consider the function $FLOOR : \mathbb{R} \to \mathbb{Z}$, defined by $FLOOR(x) = \lfloor x \rfloor$. In this case, a typical element of the co-domain is an integer $n \in \mathbb{Z}$. Now we can always find (at least one) real number $x \in \mathbb{R}$ such that $FLOOR(x) = n$, since we could have $x = n$, for example. Thus $FLOOR$ is an *onto* function.

Example 4.25 Consider the linear function $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) = 2x + 5$, introduced in Example 4.14. A typical element of the co-domain of this function is a real number, y say. If y has an ancestor x, then x satisfies the equation $f(x) = y$, or

$$2x + 5 = y.$$

Solving this equation for x, we obtain $2x = y - 5$ and hence $x = (y - 5)/2$. Thus given a real number y, this formula calculates a corresponding real number x which is the ancestor of y. Hence *every* element of the co-domain has an ancestor in the domain and f is onto.

In proving that this function has the onto property, we have also shown that its range is $\mathbb{R}$. □

Example 4.26 Consider the function $f : \mathbb{Z} \to \mathbb{Z}$ defined by $f(n) = 3n - 1$. Does this function have the onto property?

A typical element in the co-domain is an integer k, say. To find an ancestor n for k, we solve the equation

$$f(n) = k$$

for n. This gives $3n - 1 = k$ so that $n = (k + 1)/3$.

Now the rational number $(k + 1)/3$ is an integer only if $k + 1$ is divisible by 3. Putting $k = 0$, for example, we get $n = 1/3$, which is not in the domain $\mathbb{Z}$. Hence 0 has no ancestor and we conclude that f is not onto. □

Investigating polynomial functions with domain set $\mathbb{Z}$

To investigate whether a polynomial function with $\mathbb{Z}$ or $\mathbb{Z}^+$ as domain has the onto property, we can construct part of the range by calculating the images of a sequence of *consecutive* integers. This may sometimes indicate the existence, or otherwise, of an element that does not have an ancestor.

Example 4.27 We consider the function $f : \mathbb{Z} \to \mathbb{Z}$ defined by $f(n) = 3n - 1$ (see Example 4.26) by constructing the following table.

x	...	-2	-1	0	1	2	3	...
$f(x)$	...	-7	-4	-1	2	5	8	...

From the second row of the table, we see that the range is a sequence of integers increasing in steps of 3. Thus the range is not equal to the co-domain $\mathbb{Z}$. One possible counter-example to the onto property is 0, which we can see from the table has no ancestor. Hence, as before, we conclude this function is not onto. □

4.2.3 One-to-one functions

Definition 4.12 *A function is called* **one-to-one** *if distinct elements in the domain have distinct images in the co-domain. Another way of saying this is that no two different inputs give the same output. Thus a one-to-one function matches each element in the domain with a different element in the co-domain.*

Testing functions with a finite domain for the one-to-one property

We can investigate whether functions with a finite domain have the one-to-one property from an arrow diagram. A function is one-to-one if *no element in the co-domain receives arrows from two or more different elements in the domain.* If we can find an element of the co-domain that receives arrows from two or more elements of the domain, this is a counter-example to show that the function is *not* one-to-one.

Equivalently, if the function is specified by a table, then we look to see whether any element of the co-domain appears in the second row of the table as the image of two or more distinct elements in the domain. In this case, we have a counter-example to the one-to-one property. If all the elements in the second row of the table are distinct however, we can conclude that the function is one-to-one.

Example 4.28

1. The function illustrated by an arrow diagram in Example 4.4 is not a one-to-one function, because the elements a and c both have the *same* image 4.
2. The Boolean function $NOT : B \to B$ described in Example 4.7 is one-to-one.
3. The Boolean function $f : S \to B$ described in Example 4.8 is not one-to-one. The elements $(0,0)$, $(0,1)$ and $(1,1)$ all have the *same* image 1. (Note that it is sufficient to give *just two* of the elements $(0,0), (0,1), (1,1)$ to establish the counter-example.)
4. The encoding function $\alpha : A \to S$ defined in Example 4.18 is one-to-one, but the function $\gamma : A \to S$ defined in Example 4.20 is not one-to-one, because the elements b and c have the same image 16. □

Testing functions with an infinite domain for the one-to-one property

In this case, a general method for deciding whether a given function f has the one-to-one property is to use algebra to investigate whether it is possible for two different elements of the domain, say x_1 and x_2, to have the *same* image in the co-domain. This would happen if

$$f(x_1) = f(x_2).$$

We now substitute $x = x_1$ in the function rule on the left side and $x = x_2$ in the function rule on the right side, and solve the resulting equation. One solution will always be that $x_1 = x_2$. If this is the *only* possible solution, then the function is one-to-one. If, however, we can find two *distinct* elements x_1 and x_2 such that $f(x_1) = f(x_2)$, then we have a counter-example to the one-to-one property.

Example 4.29 Consider the function $FLOOR : \mathbb{R} \to \mathbb{Z}$, defined by $FLOOR(x) = \lfloor x \rfloor$. In this case, there are many pairs of distinct real numbers x_1, x_2 for which $FLOOR(x_1) = FLOOR(x_2)$. For example, when $x_1 = 1.1$ and $x_2 = 1$, we have $\lfloor x_1 \rfloor = \lfloor x_2 \rfloor = 1$. Hence $FLOOR$ is *not* a one-to-one function. □

Example 4.30 Consider the linear function $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) = 2x + 5$, introduced in Example 4.14. Suppose x_1, x_2 are real numbers such that $f(x_1) = f(x_2)$. Then we have

$$\begin{aligned} 2x_1 + 5 &= 2x_2 + 5 \\ 2x_1 &= 2x_2 \\ x_1 &= x_2. \end{aligned}$$

Thus we cannot find two *distinct* elements of the domain that have the *same* image in the co-domain, and we conclude that f is a one-to-one function. □

Example 4.31 Consider the quadratic polynomial function $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) = 2x^2 + 1$. Suppose x_1, x_2 are real numbers such that $f(x_1) = f(x_2)$. Then we have

$$\begin{aligned} 2x_1^2 + 1 &= 2x_2^2 + 1 \\ 2x_1^2 &= 2x_2^2 \\ x_1^2 &= x_2^2 \\ x_1 &= x_2 \text{ or } -x_2. \end{aligned}$$

In this example, $x_1 = x_2$ is not the only solution. If we choose any non-zero value for x_1 and put $x_2 = -x_1$, then we obtain a counter-example to show that f is not one-to-one. For example, let $x_1 = 1$ and $x_2 = -1$. Checking, we find $f(1) = 3$ and $f(-1) = 3$. Hence $f(1) = f(-1)$ and so f is not one-to-one. □

4.2.4 Inverse functions

We started this section by discussing encoding functions. We saw that the function α defined in Example 4.18 has a decoding function that allows us to decode any message coded by α unambiguously. This is because α has both the onto property, so that every possible code has an ancestor, and also the one-to-one property, so that different letters have different codes. The decoding function reverses the coding procedure and maps each symbol back into its (unique) corresponding letter.

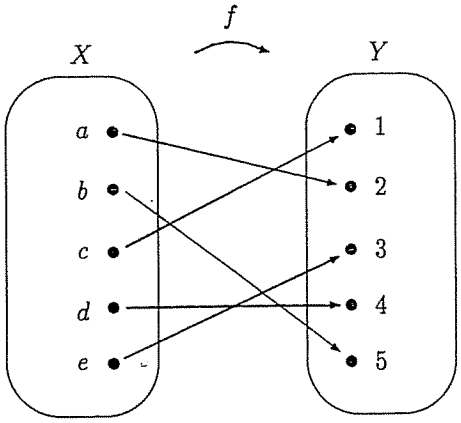
In fact, there is nothing special in this respect about encoding functions that have the letters of the alphabet as domain. Corresponding to any function that has both the onto and one-to-one properties, we can define a "decoding" function that takes each element of the co-domain back into its (unique) ancestor in the domain. We now define how this is done in general.

Let $f : X \to Y$ be a function. Suppose we define a rule v with domain Y and co-domain X by

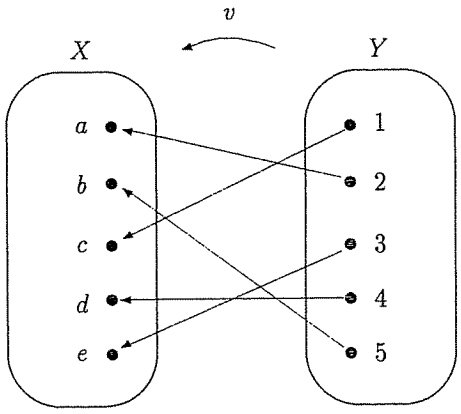
$$v(y) = x \text{ if and only if } f(x) = y.$$

You will see that v is the rule that starts from the co-domain of f and maps each element *back into its ancestor(s)*. In general v will not be a function, but when f is *both one-to-one and onto*, then v satisfies the conditions for a function.

Example 4.32 Let $X = \{a, b, c, d, e\}$ and $Y = \{1, 2, 3, 4, 5\}$. The arrow diagram below illustrates a one-to-one and onto function $f : X \to Y$.



The rule v described above is obtained by *reversing* each arrow so that it relates each image in Y back to its ancestor in X. This is illustrated by the arrow diagram below and it is easily seen that v satisfies the definition of a function. □



Theorem 4.13 *Let $f : X \to Y$ be one-to-one and onto. Then the rule v with domain Y and co-domain X defined by*

$$v(y) = x \text{ if and only if } f(x) = y$$

is a function from Y onto X.

Proof. Since f is *onto*, every element of Y has an ancestor in X and so $v(y)$ is defined for every element of Y. Further, since f is *one-to-one*, no two distinct elements of Y have the same ancestor in X and hence there is only one element $x \in X$ such that $v(y) = x$. Thus v satisfies both the conditions for it to be a *function* from Y into X. □

Definition 4.14 *When $f : X \to Y$ is both onto and one-to-one, the function $v : Y \to X$ defined in Theorem 4.13 is called the* **inverse function** *of f and the function f is said to be* **invertible**.

An invertible function f can be viewed as an encoding function and its inverse function as the corresponding decoding function. Note that the inverse function of f (when it exists) is often denoted by f^{-1}.

Finding the inverse of an invertible function given by a table

Example 4.33 Let $X = \{a, b, c, d\}$ and $Y = \{1, 2, 3, 4\}$, and let $f : X \to Y$ be the function defined by the following table.

x	a	b	c	d
$f(x)$	4	3	1	2

Since f is one-to-one and onto, f is invertible. We find its inverse function $f^{-1} : Y \to X$.
When $y = f(x)$, then $f^{-1}(y) = x$. Hence $f(a) = 4$ gives $f^{-1}(4) = a$; similarly, $f(b) = 3$ gives $f^{-1}(3) = b$; $f(c) = 1$ gives $f^{-1}(1) = c$; and $f(d) = 2$ gives $f^{-1}(2) = d$. Thus the inverse function $f^{-1} : Y \to X$ is defined by the following table:

x	4	3	1	2
$f^{-1}(x)$	a	b	c	d

□

Thus when a one-to-one and onto function is defined by a table, the table for the inverse function is found by *interchanging the position of each element of the domain with its image.*

Finding the inverse of an invertible function defined by a function rule

Example 4.34 We have seen that the linear function $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) = 2x + 5$, is both one-to-one and onto (see Example 4.25 and Example 4.30) and hence it is invertible. We find its inverse function $f^{-1} : \mathbb{R} \to \mathbb{R}$. The method is essentially the same as the one we used to check that f has the onto property.

Let y be any real number in the co-domain. We know there is a unique real number x such that $f(x) = y$, because f is one-to-one and onto. Then $f^{-1}(y) = x$, so we need to solve the equation $f(x) = y$ to give a formula for x in terms of y, just as in Example 4.25. We have

$$\begin{aligned} 2x + 5 &= y \\ 2x &= y - 5 \\ x &= (y - 5)/2. \end{aligned}$$

Thus the inverse function f^{-1} takes an input y from $\mathbb{R}$ and transforms it into the output $(y-5)/2$ in $\mathbb{R}$. Hence $f^{-1} : \mathbb{R} \to \mathbb{R}$ is defined by $f^{-1}(y) = (y - 5)/2$.

We can use any variable name to stand for the input, but the output must be a formula written in the *same* variable. If you want to change the variable name from y to x in the formula for f^{-1}, then you must do so in both input and output, so that we would have

$$f^{-1}(x) = (x - 5)/2.$$

Notice that the inverse function *reverses* each of the operations of the original function to get back from an image to its ancestor. Thus to get from x to $f(x)$, the function f multiplies x by 2 and then adds 5 to the result. To go backwards from $y = f(x)$ to x, f^{-1} subtracts 5 from y and then divides by 2. □

4.2.5 One-to-one correspondence

Let $f : X \to Y$ be a one-to-one and onto function. We have seen above that f has an inverse function $f^{-1} : Y \to X$. Such a function f pairs the elements of X and Y, so that each element of X is paired with its unique image in Y and each element of Y is paired with its unique ancestor in X. For this reason, such a function is sometimes called a **one-to-one correspondence** from X onto Y.

Definition 4.15 *When a function $f : X \to Y$ is both one-to-one and onto, we say that f is a* **one-to-one correspondence from X onto Y.**

You will be familiar with the idea of a one-to-one correspondence in an intuitive way. In fact, whenever you count a finite set of n objects you are mentally pairing the objects with the numbers $1, 2, \ldots, n$. In other words, you are putting the set into one-to-one correspondence with the set $\{1, 2, \ldots, n\}$. It is also intuitively obvious that if we can find a way of matching the elements of two finite sets, then these sets must each contain the same number of elements. This simple idea can be surprisingly useful in some counting problems. It is stated formally below.

Result 4.16 The Correspondence Principle. *Let X, Y be finite sets and suppose we can find a one-to-one correspondence $f : X \to Y$. Then X and Y have the same cardinality.*

4.3 Exponential and Logarithmic functions

Learning Objectives

After studying this section, you should be able to:

- use the laws of exponents and interpret rational indices;
- define and apply the exponential and logarithmic functions with given positive base b and state the relationship between them;
- state and use the laws of logarithms.

Introduction

There are two further families of functions that have an important role in computing: the exponential functions and the logarithmic functions. We have seen that if we have a finite set containing n elements, then the number of subsets of this set is 2^n. We say that 2^n is an *exponential function* of n, and that the number of subsets *grows exponentially* as n increases. We shall see in the next section that if an algorithm takes an exponential number of operations to implement, then this has very alarming implications for the time the algorithm takes to run on even the fastest modern computer. The logarithmic functions are the inverse functions of the exponential functions, and these functions grow very slowly.

We start with a formal definition of an exponential function.

Definition 4.17 *A function* $exp_b : \mathbb{R} \to \mathbb{R}$ *defined by a rule of the form*

$$exp_b(x) = b^x,$$

where b *is a given positive number, is called an* **exponential function** *with base* b. *Note that we define*

$$b^{-x} = 1/b^x \text{ and } b^0 = 1.$$

In the branches of mathematics where we use a continuous variable (such as all applications of calculus, for example), the base b is usually the irrational number denoted by e which has a value of approximately 2.71828, correct to 5 decimal places. However, in discrete mathematics and computing, the constant b is usually a positive integer. In computing, powers of 2 are particularly important. The next example illustrates an exponential function with an integer input.

Example 4.35 The function $exp_2 : \mathbb{Z} \to \mathbb{Q}$ defined by the rule $exp_2(n) = 2^n$ accepts an integer n as input and outputs the value 2^n. Then exp_2 has domain $\mathbb{Z}$, co-domain $\mathbb{Q}$ and range the set $\{\ldots, 1/8, 1/4, 1/2, 1, 2, 4, 8, 16, \ldots\}$. □

4.3.1 Laws of exponents

You should be familiar with the rules of indices, for multiplying numbers raised to integer powers. The *laws of exponents* are generalisations of these rules to the case where the indices are any real numbers (note that *exponent* is an alternative term for *index*).

Rule 4.18 The laws of exponents. *Let* b, c *be any positive real numbers and let* s, t *be any real numbers. Then*

(a) $b^s b^t = b^{s+t}$;

(b) $(b^s)^t = b^{st}$;

(c) $(bc)^s = b^s c^s$.

Rule 4.18(b) enables us to interpret rational indices in terms of nth roots of the base number b. Suppose we put $s = \frac{1}{2}$ and $t = 2$. Then rule 4.18(b) gives:

$$(b^{\frac{1}{2}})^2 = b^{\frac{1}{2} \times 2} = b^1 = b.$$

Putting $s = 2$ and $t = \frac{1}{2}$ gives the same result:

$$(b^2)^{\frac{1}{2}} = b^{2 \times \frac{1}{2}} = b^1 = b.$$

Hence $b^{\frac{1}{2}}$ is a square root of b. We define $b^{\frac{1}{2}}$ to be $\sqrt{b}$, the *positive* square root of b.

Definition 4.19 *Let* p, q *be integers with* $q > 0$. *We define*

1. $b^{\frac{1}{q}} = \sqrt[q]{b}$, *the positive* q*th root of* b;

2. $b^{\frac{p}{q}} = \sqrt[q]{b^p} = (\sqrt[q]{b})^p$.

Example 4.36 We simplify $16^{-0.5}$ and $128^{\frac{3}{7}}$.

1. $16^{-0.5} = 1/16^{0.5} = 1/\sqrt{16} = 1/4$. Alternatively, we can write $16 = 2^4$ and work in powers of 2, using rule (b) directly. We then have $16^{-0.5} = (2^4)^{-0.5} = 2^{4\times(-0.5)} = 2^{-2} = 1/2^2 = 1/4$.
2. $128^{\frac{3}{7}} = (\sqrt[7]{128})^3$. Now, since $128 = 2^7$, we have $\sqrt[7]{128} = 2$. Thus $128^{\frac{3}{7}} = 2^3 = 8$. Alternatively, we can use rule (b) directly and write $128^{\frac{3}{7}} = (2^7)^{\frac{3}{7}} = 2^{7\times 3/7} = 2^3 = 8$, as before.

It is not nearly so easy to interpret an irrational exponent. For example, what is meant by $8^{\sqrt{2}}$? We cannot write $\sqrt{2}$ as a fraction, but we can approximate to it by a sequence of fractions that give its value with increasing accuracy. For example, the first six significant figures in the decimal expansion of $\sqrt{2}$ are 1.41421. Thus the sequence of rational numbers 1.4, 1.41, 1.414, 1.4142, 1.41421, ..., gives an increasingly good approximation to $\sqrt{2}$, and can be continued as far as we like by calculating more and more decimal places of $\sqrt{2}$. It follows that the sequence $8^{1.4}$, $8^{1.41}$, $8^{1.414}$, ... gives an increasingly good approximation to $8^{\sqrt{2}}$. It can be shown that we get a value as close as we please to the actual value of $8^{\sqrt{2}}$ by continuing this sequence sufficiently far.

The following results can be proved by calculus.

Result 4.20 *The exponential function* $exp_b : \mathbb{R} \to \mathbb{R}$ *defined by the rule*

$$exp_b(x) = b^x$$

is a one-to-one function, for every positive number $b \neq 1$.

Result 4.21 *The range of the exponential function* $exp_b : \mathbb{R} \to \mathbb{R}$ *defined the rule*

$$exp_b(x) = b^x$$

is $\mathbb{R}^+$, *for every positive number* $b \neq 1$.

From Result 4.21 and Result 4.20, we see that if we define the exponential function with base b to have co-domain $\mathbb{R}^+$, then $exp_b : \mathbb{R} \to \mathbb{R}^+$ defined by the rule

$$exp_b(x) = b^x$$

is one-to-one and onto, for every positive number $b \neq 1$.

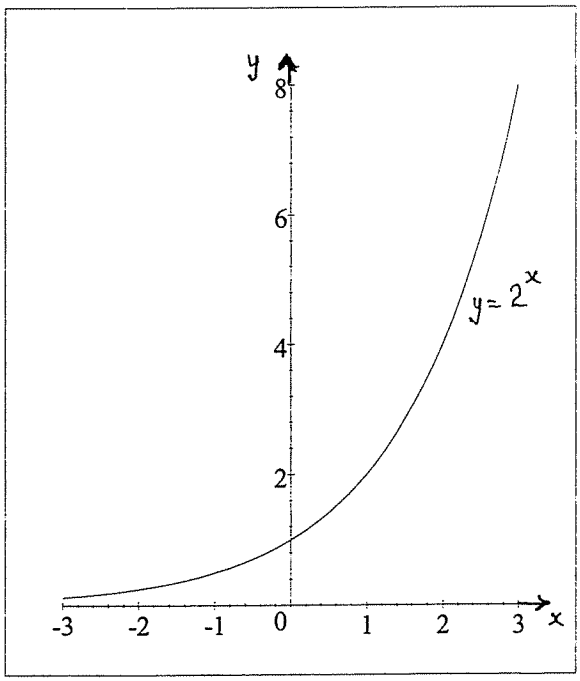


Figure 4.1.

The graph of the function $exp_2(x) = 2^x$, when $-3 \leq x \leq 3$, is shown in Figure 4.1.

4.3.2 Logarithmic functions

Suppose that the function $exp_b : \mathbb{R} \to \mathbb{R}^+$ is defined by $exp_b(x) = b^x$, where $b \neq 1$. Then, from Results 4.20 and 4.21, the function exp_b is both one-to-one and onto and hence is invertible. Its inverse function is called the **logarithm of x to the base b** and is denoted by $\log_b x$. Thus we have the following definition.

Definition 4.22 *Let $b \neq 1$ be a positive number. For each* positive *real number x, the exponent to which b must be raised in order to give x is called the* **logarithm of x to the base b** *and written $\log_b x$. Thus we have*

$$y = \log_b x \text{ when } x = b^y.$$

The domain of $\log_b x$ is $\mathbb{R}^+$ and its co-domain is $\mathbb{R}$.

The graph of the function $y = \log_2 x$, when $0 < x \leq 16$ is shown in Figure 4.2. Note that as x increases, $\log_2 x$ also increases.

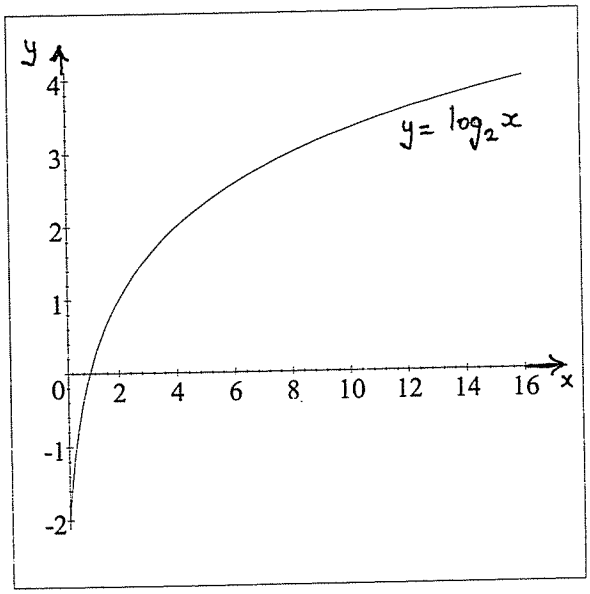


Figure 4.2.

Example 4.37 We find the value of $\log_2 x$ when x has the values 8, 128, 1, 1/16.

The first step is to write each number as 2 raised to an exponent. Thus we have

$$8 = 2^3; \qquad 128 = 2^7; \qquad 2^0 = 1; \qquad 1/16 = 2^{-4}.$$

Then the logarithm of each number to the base 2 is given by the exponent to which 2 has been raised. Hence $\log_2 8 = 3$, $\log_2 128 = 7$, $\log_2(1/16) = -4$ and $\log_2 1 = 0$. □

Since logarithms are *exponents*, they obey laws which can be derived from the laws of exponents. Suppose that x, y are any positive real numbers, and let s, t be real numbers such that $b^s = x$ and $b^t = y$, where $b \neq 1$. Then $s = \log_b x$ and $t = \log_b y$ and the law $b^s b^t = b^{s+t}$ becomes $xy = b^{s+t}$. Hence

$$\log_b(xy) = s + t = \log_b x + \log_b y.$$

Similarly, the law $(b^s)^w = b^{sw}$ becomes $x^w = b^{sw}$ and hence

$$\log_b(x^w) = sw = w \log_b x.$$

These rules are summarized below.

Rule 4.23 The laws of logarithms. *Let x, y be any positive real numbers and let s, t be any real numbers. Then*

(a) $\log_b(xy) = \log_b x + \log_b y$;

(b) $\log_b(x^w) = w \log_b x$.

Before the invention of the hand-held calculator and the personal computer, logarithms were a very important tool for multiplying, dividing or exponentiating large numbers. Logarithms to the base 10 were calibrated in tables and used for performing calculations by most people studying or using mathematics. Although they have lost that once vital role, logs still have a theoretical importance. In computer science we are particularly interested in logs to the base 2. However, not all hand-held calculators have a key to compute $\log_2 x$ directly, although most have a key to compute $\log_{10} x$. Happily, there is a very simple rule for converting logs to the base 10 into logs to the base 2.

Rule 4.24 *Let x be any positive real number. Then*

$$\log_2 x = \frac{\log_{10} x}{\log_{10} 2}.$$

Proof. Let $y = \log_2 x$. Then $x = 2^y$. Taking logs to the base 10 of both sides of this equation, we have

$$\log_{10} x = \log_{10}(2^y) = y(\log_{10} 2),$$

by Rule 4.23(b). Thus we have $y(\log_{10} 2) = \log_{10} x$, giving

$$y = \frac{\log_{10} x}{\log_{10} 2}. \quad \square$$

Example 4.38

$$\log_2 13 = \frac{\log_{10} 13}{\log_{10} 2} \approx \frac{1.113943352}{0.301029995} \approx 3.700439718.$$

As a rough check on this answer, note that $2^3 < 13 < 2^4$ and hence we would expect $\log_2 13$ to lie between 3 and 4. $\square$

4.4 Comparing the size of functions

Learning Objectives

After studying this section, you should be able to

- compare a polynomial function with a power function using the O-notation, giving the steps in the argument;
- use O-notation to state the relationship between the exponential and logarithmic functions and a given power function.

Introduction

It often happens that the computer algorithms available to perform a certain task differ widely in their time or memory space requirements when the data set is very large. In this case, ratios of a constant factor are relatively unimportant. The O-notation (read "big-oh notation") provides a way of comparing the relative sizes of the outputs from different functions when the input (to each) is a very large number, while ignoring the behaviour of the functions for low values of the input. It is based on the following idea.

4.4.1 O-notation

Let X be a subset of $\mathbb{R}$ and suppose that $f : \mathbb{X} \to \mathbb{R}$ and $g : \mathbb{X} \to \mathbb{R}$ are two functions of a real variable x. If, for all large values of $x \in X$, the graph of the function f lies closer to the x-axis than the graph of some fixed positive multiple of the function g, then we say that f is *of order* g and we write "$f(x)$ is $O(g(x))$".

The distance of the point $(x, f(x))$ on the graph $y = f(x)$ from the x-axis is the absolute value of $f(x)$, which we denote by $|f(x)|$ (see Example 4.9). Thus we can express the definition of *order of a function* more formally as follows.

Definition 4.25 *Let $f : \mathbb{X} \to \mathbb{R}$ and $g : \mathbb{X} \to \mathbb{R}$ be two functions with a common domain $X \subseteq \mathbb{R}$. Then we say that f* **is of order** *g, written "$f(x)$ is $O(g(x))$", if we can find a positive real number M and a real number x_0 such that*

$$|f(x)| \leq M|g(x)|,$$

for all $x > x_0$.

4.4.2 Power functions

In using the O-notation, we often compare a given function f with one of the following family of functions.

Definition 4.26 *Let s be any positive rational number. Then the function $f : \mathbb{R} \to \mathbb{R}$ defined by*

$$f(x) = x^s$$

is called the **power function** *of x with exponent s.*

The following table compares the values of the power functions $f(x) = x^s$, where $s = 0.5, 1, 1.5, 2, 3$ (values of $f(x)$ have been entered correct to 2 decimal places).

x	0	0.5	1	2	3	4
$x^{0.5}$	0	0.71	1	1.41	1.73	2
x^1	0	0.5	1	2	3	4
$x^{1.5}$	0	0.35	1	2.83	5.20	8
x^2	0	0.25	1	4	9	16
x^3	0	0.13	1	8	27	64

The graphs of the power functions $y = x^r$, when $0 \leq x \leq 2$ and $r = 1/3, 1/2, 1, 2, 3$ are compared in Figure 4.3. They illustrate the following result.

Result 4.27 *Let r, s be any rational numbers such that $r < s$. Then when $x > 1$, we have*

$$x^r < x^s. \ \square$$

We see that for any pair of rational numbers r, s with $r < s$, the graph of $y = x^r$ lies closer than the graph of $y = x^s$ to the x-axis, for all values of $x > 1$. Reasoning analytically, when $x > 1$, x^r and x^s are both positive, and hence $|x^r| = x^r$ and $|x^s| = x^s$. Taking $x_0 = 1$ and $M = 1$ in the defintion of the O-notation, we have:

Result 4.28 *Let r, s be any rational numbers such that $r < s$. Then*

$$x^r \text{ is } O(x^s). \ \square$$

Example 4.39 Let $f(x) = x\sqrt{x}$. Then since $x\sqrt{x} = x^{1.5}$, we have

$$f(x) < x^2$$

for all $x > 1$. We can say that $f(x)$ is $O(x^2)$. $\square$

4.4.3 Orders of polynomial functions

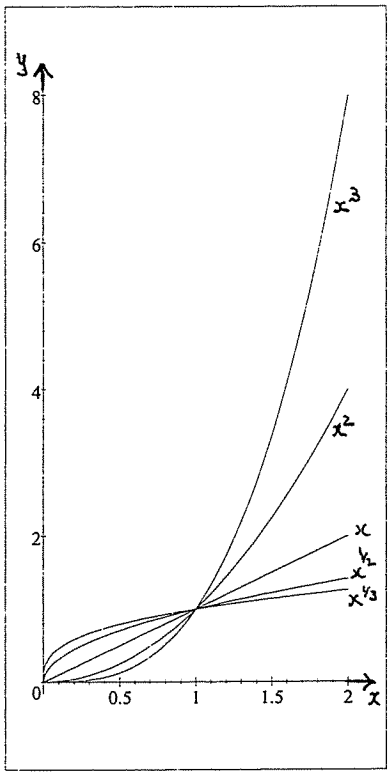


Figure 4.3.

Example 4.40 We show that $f(x) = 5x^2 + 2x + 9 < 16x^2$, for all $x > 1$.

First note that to find the value of $f(x)$ for any given x, we have to sum multiples of the three power functions x^2, x^1 and x^0. Now suppose that $x > 1$. Then from Result 4.27, we have

$$x^1 < x^2, \text{ and } x^0 < x^2.$$

Multiplying the first inequality by 2 and the second by 9, we have

$$2x^1 < 2x^2, \text{ and } 9x^0 < 9x^2.$$

Thus summing the terms gives

$$f(x) = 5x^2 + 2x + 9 < 5x^2 + 2x^2 + 9x^2 = 16x^2.$$

Hence $f(x) < 16x^2$. □

Example 4.41 We show that $f(x) = 5x^2 + 2x + 9$ is $O(x^2)$.

Suppose that $x > 1$. Then each of the terms in the polynomial $f(x)$ is positive. Hence the absolute value of $f(x)$ is given by the sum of the terms. Thus

$$|f(x)| = |5x^2 + 2x + 9| = 5x^2 + 2x + 9.$$

But in the previous example, we showed that

$$5x^2 + 2x + 9 < 5x^2 + 2x^2 + 9x^2 = 16x^2.$$

Since x^2 is positive, $|x^2| = x^2$. Thus we have

$$|f(x)| < 16|x^2|,$$

for all $x > 1$. Hence $f(x)$ is $O(x^2)$, where in the formal definition, we have $M = 16$, $g(x) = x^2$ and $x_0 = 1$. □

Example 4.42 We show that $f(x) = 2x^3 - 5x^2 + 27$ is $O(x^3)$.

Suppose that $x > 1$. This polynomial contains a negative coefficient. In this case we can say that the absolute value of $f(x)$, calculated at any given value of x, is at most the value we would get by *adding* all the terms, instead of adding some and subtracting others. Thus we have

$$|f(x)| \leq |2x^3| + |5x^2| + |27|.$$

Now each of the terms $2x^3$, $5x^2$, 27 is positive when $x > 1$. Hence

$$|2x^3| + |5x^2| + |27| = 2x^3 + 5x^2 + 27,$$

and using Result 4.27 again, we have

$$2x^3 + 5x^2 + 27 < 2x^3 + 5x^3 + 27x^3 = 34x^3,$$

for all $x > 1$. Thus $|f(x)| < 34|x^3|$, for all $x > 1$, and hence $f(x)$ is $O(x^3)$. □

Following the method of the previous two examples, we can prove the following result.

Result 4.29 *Suppose that m is the highest exponent of x present in a polynomial function $f(x)$, then $f(x)$ is $O(x^m)$.* □

4.4.4 Comparing the exponential and logarithmic functions with the power functions

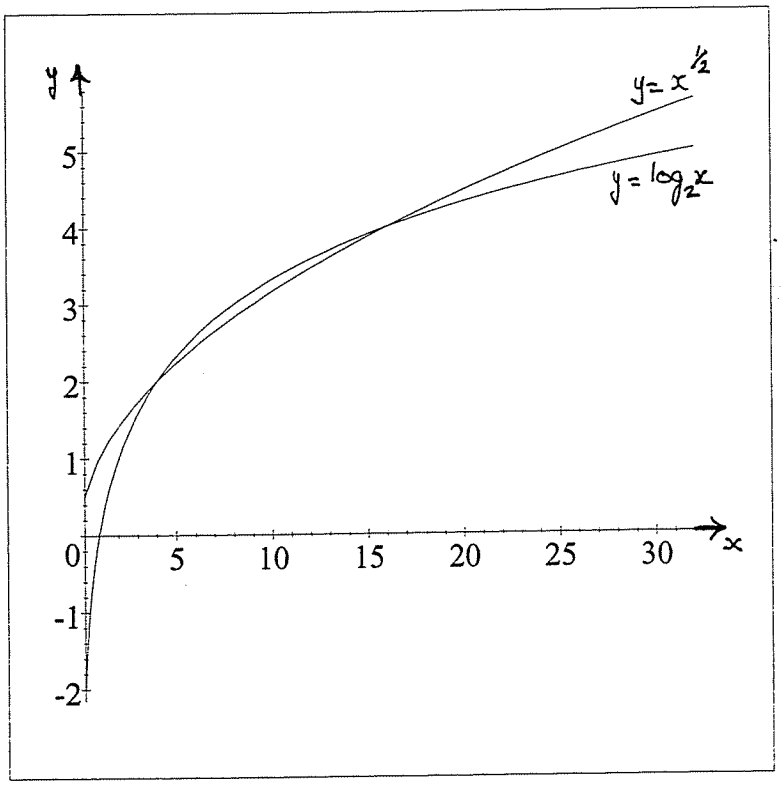


Figure 4.4.

It remains to compare the sizes of the exponential functions and the logarithmic functions with the power functions for large values of x. The following result can be established by calculus.

Result 4.30 *Let b be any real number such that $b > 1$ and r be any positive rational number. Then for all sufficiently large values of x, we have*

$$\log_b x < x^r \text{ and } x^r < b^x. \ \square$$

Because of the importance of the base 2 in computer science, we interpret this result with $b = 2$. Suppose we want to compare the behaviour of $\log_2 x$ with a power function of x, say $x^{\frac{1}{2}}$ for example. Then Result 4.30 says that we can find a real number x_0 such that $\log_2 x < x^{\frac{1}{2}}$, whenever $x > x_0$. In Figure 4.4, the graph of $y = \log_2 x$ is compared with the graph of $y = x^{\frac{1}{2}}$, when $0 < x \leq 32$. You can see that when $x > 16$, then $x^{\frac{1}{2}} > \log_2 x$, so that in this case the least value for x_0 is 16. Even if we compare the value of $\log_2 x$ with the power function $x^{\frac{1}{100}}$, for example, it would still be possible to find a real number x_0 such that $\log_2 x < x^{\frac{1}{100}}$ for all $x > x_0$.

In particular, we have the following result.

Result 4.31 $\log_2 x < x$, *for all* $x > 0$. $\square$

In a similar way, if we compare the behaviour of 2^x with any power function of x, say with x^{10} for example, then Result 4.30 says that we can find a real number x_0 such that $2^x > x^{10}$, for all $x > x_0$. The graph of $y = 2^x$ is compared with the graphs of $y = x$ and $y = x^2$, for $0 \leq x \leq 5$, in Figure 4.5. You can see from these graphs that $2^x > x$ for all $x > 0$ and $2^x > x^2$ when $x > 4$.

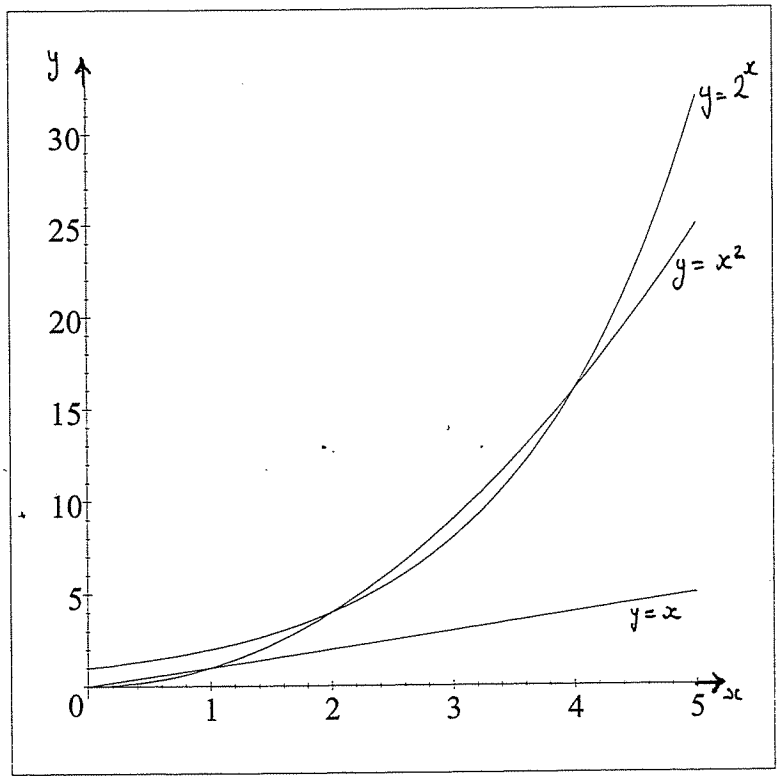


Figure 4.5.

4.4.5 Comparison of algorithms

When studying how long a given computer algorithm will take to perform some task, an important factor is the size of the data set that will be input into the algorithm. Since this will be measured as a positive integer, the domain X of a function measuring the time the algorithm takes to run will be a subset of $\mathbb{Z}^+$.

Suppose, for example, that a measurement of the running time of an algorithm is given for input size n by the formula $f(n) = 5n^2 + 8n + 3$. Then when n is *large*, terms such as $8n + 3$ are

tiresome and irrelevant compared with the size of the term in n^2. Now for $n \geq 1$, we have $f(n) \leq 5n^2 + 8n^2 + 3n^2 = 16n^2$, and thus

$$5n^2 + 8n + 3 = O(n^2).$$

The use of the O-notation strips a rather complicated expression for the running time down to big-oh of a simple expression (such as a power or log function), enabling easy comparison between algorithms to be made.

You might imagine that because each operation performed by a modern digital computer takes only a tiny fraction of a second, it will not be very important whether we are running an $O(n)$ algorithm or an $O(n^2)$ or even an $O(n^3)$ algorithm. Unfortunately that is not the case, as the table below will show you. It compares the approximate time to perform $f(n)$ operations for the functions $f(n)$ with which algorithms are commonly compared, assuming that each operation takes about one microsecond.

Approximate time to perform $f(n)$ operations

$f(n)$	$n = 10^2$	$n = 10^3$	$n = 10^4$	$n = 10^5$	$n = 10^6$
$\log_2 n$	7×10^{-6}s	1×10^{-5}s	1.3×10^{-5}s	1.7×10^{-5}s	2×10^{-5}s
n	0.0001s	0.001s	0.01s	0.1s	1s
$n \log_2 n$	0.0007s	0.01s	0.13s	1.7s	20s
n^2	0.01s	1s	1.67 mins	2.78 hrs	11.6 days
n^3	1s	16.7 mins	11.6 days	31.7 yrs	31,710 yrs

There are some important conclusions to be drawn from this table.

1. There is not much difference between an $O(n)$ algorithm and an $O(n \log_2 n)$ algorithm, but the latter takes slightly longer.
2. An $O(n^2)$ algorithm takes significantly longer to run than an $O(n \log_2 n)$ algorithm and this difference will be important if the data set is large.
3. An $O(n^3)$ algorithm takes significantly longer to run than an $O(n^2)$ algorithm and is completely impractical for input sizes of much more than 1000.

Although an $O(n^3)$ algorithm takes a long time to run when $n > 1000$, this is as nothing compared with an exponential algorithm. The following table gives the times to perform 2^n operations, again assuming one operation per microsecond.

Approximate time to perform 2^n operations

n	10	10^2	10^3	10^4
2^n	0.001s	4×10^{17}yrs	3.4×10^{287}yrs	6.3×10^{2996}yrs

It goes without saying that an exponential algorithm should be avoided at all costs! However, there are tasks for which no polynomial algorithm is known and some for which a polynomial algorithm exists but is $O(n^3)$. These tasks typically arise in areas such as operational research, where an optimal (that is, very best) solution to a particular problem is sought. An important aspect of current research by mathematicians and computer scientists is in developing for this kind of problem $O(n)$ or $O(n \log_2 n)$ algorithms that find a nearly optimal solution, but one that is not necessarily the very best.

4.5 Exercises 4

1. Let $X = \{1, 2, 3, 4\}$ and $Y = \{1, 2, 3, \ldots, 10\}$. A function $f : X \to Y$ is defined by the following table.

x	1	2	3	4
$f(x)$	1	4	7	10

Find

(a) the domain of f; (b) the codomain of f; (c) $f(2)$;

(d) the ancestor of 10; (e) the range of f.

Illustrate f by drawing an arrow diagram.

2. Let S denote the set of all 3-bit binary strings and $B = \{0, 1\}$. The function $f : S \to B$ is defined by the rule:

$$f(x) = \text{the last bit of } x,$$

for each binary string $x \in S$. Find the following:

(a) the domain of f, specified by the listing method;

(b) $f(011)$;

(c) the set of ancestors of 0;

(d) the range of f.

3. Let x and y be any two real numbers. Express $|x - y|$ in terms of x and y (a) when $x > y$; (b) when $y > x$. What is the value of $|x - y|$ when $x = y$?

Suppose that x and y are marked on the real number line. What does $|x - y|$ represent geometrically?

4. Find the set of real numbers x for which $|x - 10| = 2$.

5. Find the floor and the ceiling of the following numbers:

$$\text{(a) } -1.4\text{; (b) } 2.3\text{; (c) } 7/9\text{; (d) } -16/3\text{; (e) } 0.$$

6. Let x be a real number. Suppose that $\lfloor x \rfloor = n$, where n is an integer.

(a) Find in terms of n the values of $\lfloor x + 1 \rfloor$ and $\lfloor x - 3 \rfloor$.

(b) Use the definition of $\lfloor x \rfloor$ to justify your answers to part (a).

(c) Can you find examples of real numbers x, y such that $\lfloor x + y \rfloor \neq \lfloor x \rfloor + \lfloor y \rfloor$? (Remember to justify your example by finding the value of both sides of this inequation for the numbers you give.)

7. For each of the following functions, decide whether they have the onto property or the one-to-one property, justifying your answers.

(a) the function $f : X \to Y$ defined in Question 1 above;

(b) the function $f : S \to B$ defined in Question 2 above;

(c) the function $f : X \to X$, where $X = \{1, 2, 3, 4, 5\}$ defined by the following table

x	1	2	3	4	5
$f(x)$	5	3	1	2	4

(d) the function $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) = 5x - 1$;

(e) the function $g : \mathbb{Z} \to \mathbb{Z}$ defined by $g(r) = 5r - 1$;

(f) the function $h : \mathbb{R} \to \mathbb{R}$ defined by $h(x) = 5x^2 - 1$.

8. Decide whether any of the functions in the previous question are invertible and find the inverse of each invertible function.

9. Without using your calculator, express each of the following as an integer or fraction (with exponent 1).

$$8^{\frac{1}{3}}; \qquad 8^{-\frac{2}{3}}; \qquad 16^{1.25}; \qquad 32^{1.4}; \qquad 32^{-0.6}.$$

Then check your answers using your calculator.

10. Without using your calculator, find the value of

$$\log_2 128; \qquad \log_2 4\sqrt{2}; \qquad \log_2 \frac{1}{8}; \qquad \log_2 \frac{\sqrt{2}}{16}.$$

11. Let X denote the set of non-negative real numbers. Then the power function $f : X \to X$ defined by $f(x) = x^s$ is a one-to-one correspondence for every positive rational number s. Hence each power function has an inverse function. For example, the inverse function of $f(x) = x^2$ is $v(x) = \sqrt{x} = x^{\frac{1}{2}}$. Find the inverse function of

$$f(x) = x^3; \qquad f(x) = x^{\frac{1}{4}}; \qquad f(x) = x^{\frac{3}{2}}.$$

12. On the same graph and with the same axes, sketch the graphs of $f(x) = \log_2 x$ for $\frac{1}{4} \leq x \leq 32$, and $g(x) = \sqrt{x}$ for $0 \leq x \leq 32$. Why can we say that $\log_2 x = O(x^{\frac{1}{2}})$?

13. Let x be a positive real number and let $f(x) = 2x\sqrt{x} + 5x + 24$. Show that $f(x) < 31x\sqrt{x}$ when $x > 1$. Deduce that $f(x) = O(x^{\frac{3}{2}})$.

14. Let k be a positive integer. Find a relationship between the number of bits in the binary representation of k and $\log_2 k$ (hint: the number of bits is always an integer, so you might think about using the floor or ceiling function). Show that your formula will give the correct number of bits for any $k \in \mathbb{Z}^+$.

Chapter 5

Introduction to Graph Theory

Summary

Terminology of graph theory; parallel edges, loop; degree of a vertex; degree sequence; regular graphs; complete graphs; path, cycle; connectivity; isomorphism of graphs; adjacency matrix, adjacency list.

References: Epp Sections 11.1, 11.2 (pp 619-625), 11.3 (pp 641-643), 11.4 *or* M&B Sections 8.1, 8.2 and pp 419, 425.

Introduction

The term *graph* is used in discrete mathematics to describe the kind of structure that you might think of as a "network". Graph Theory is a modern, fast growing area of mathematics with many applications to practical problems and to such disciplines as computer science, telecommunications, chemistry and the behavioural and environmental sciences. Many problems that need to be solved in computing can be modelled by a graph. In this chapter, you will be introduced to the elementary mathematical analysis of graphs. This will enable you to reason effectively about the types of problems that graphs model.

Note that because it is such a modern subject, not all the terminology of graph theory has been standardized. You may find that text books differ both in the way they define terms and in the notation they use.

5.1 What is a graph?

Learning objectives

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

- define the terms *incidence, adjacency, degree of a vertex*;
- state and apply a result concerning the connection between the sum of the degrees of the vertices of a given graph and the number of its edges;
- say what is meant by an *r-regular graph*;
- draw a *complete graph* on n vertices for small integers n.

Introduction

We start with an example to illustrate how a graph can be used to model a communications network.

Example 5.1 Suppose that A, B, C, D, E, F represent six cities. Although it is possible to travel between any pair of these cities by rail, in some cases the journey involves changing trains. There are direct links between pairs of cities (travelling in either direction) as follows:

A is linked to B, D, E;	B is linked to A, C, D, F;
C is linked to B and D;	D is linked to B and C;
E is linked to A;	F is linked to B and D.

The diagram in Figure 5.1 gives a convenient way of illustrating this information. □

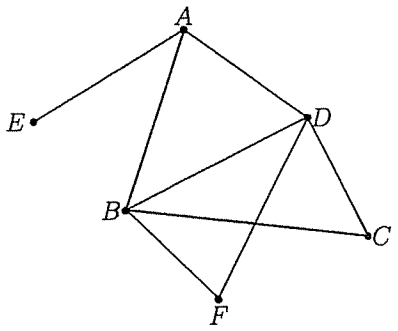


Figure 5.1.

This diagram is an example of a *graph*. The dots representing the cities are called the *vertices* of the graph and the line segments connecting pairs of vertices are called *edges*. (Note that the singular of *vertices* is *vertex*.) It is possible for pairs of edges to cross at a point which is not a vertex, as for example the edges joining B to C and F to D in Figure 5.1. It is thus very important when drawing a graph to mark the vertices with a definite dot or small ring. The edges may be drawn straight or curved; their only function is to show that a *connection* exists between the pair of vertices they join. The following formal definitions do not make use of pictorial ideas.

5.1.1 Some definitions

Definition 5.1 *A graph consists of a pair of finite sets, V and E. The elements of V are called vertices; the elements of E are subsets of V of size 2 and are called* **edges**. *We shall often denote a graph by a letter such as G, and then we denote its set of vertices by $V(G)$ and its set of edges by $E(G)$.*

Example 5.2 For the graph shown in Figure 5.2, we have $V(G) = \{u, v, w, x\}$ and $E(G) = \{\{u, v\}, \{v, w\}, \{v, x\}, \{w, x\}\}$. □

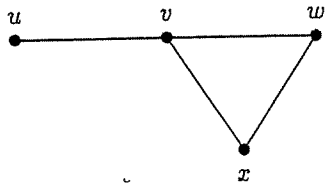


Figure 5.2.

Definition 5.2 *Suppose that G is a graph. Let $u, v \in V(G)$ and suppose that $\{u, v\} \in E(G)$. We shall call u, v the* **endpoints** *of the edge $e = \{u, v\}$. A vertex and an edge are said to be* **incident** *if the vertex is an endpoint of the edge. Two vertices are said to be* **adjacent** *if they are endpoints of the same edge. Two edges are said to be* **adjacent** *if they are incident with the same endpoint.*

Example 5.3 In the graph described in Example 5.2, v and w are the endpoints of the edge $e = \{v, w\}$; e is incident with both v and w; w is incident with the edges e and $f = \{w, x\}$. The vertex x is adjacent to v and w; the edge $\{v, w\}$ is adjacent to $\{u, v\}$, $\{v, x\}$ and $\{w, x\}$. □

To increase the usefulness of a graph G as a model, it is convenient to allow it to contain more than one edge with the same pair of endpoints. Such a graph is illustrated in Figure 5.3. In this graph the edges e_1 and e_2 have the same endpoints. Such sets of edges are called **parallel edges**. It will also be convenient to allow the two endpoints of an edge to coincide. In this case, we call the edge a **loop**. The edge e_3 of the graph in Figure 5.3 is an example of a loop incident with the vertex v_4. A graph that has neither loops nor parallel edges is known as a **simple** graph. It is also possible to have a vertex that is not adjacent to any other vertex of the graph, such as the vertex v_5 in Figure 5.3. Such a vertex is called an **isolated vertex**.

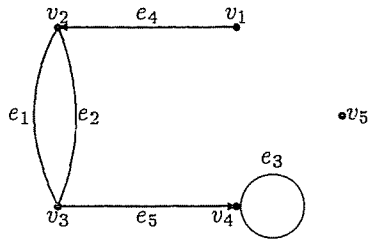


Figure 5.3.

In a *simple* graph, we often denote an edge $\{x, y\}$ by xy or yx. In a graph containing parallel edges joining some pairs of vertices, this notation cannot be used since it would be ambiguous.

5.1.2 Degree of a vertex

The **degree** of a vertex v, denoted by $\deg(v)$, is the number of edges incident with v. Note that in a graph with loops, each loop contributes 2 towards the degree of the vertex incident with it.

Example 5.4 In the graph depicted in Figure 5.3, $\deg(v_1) = 1$, $\deg(v_2) = deg(v_3) = 3$; $\deg(v_4) = 3$, since v_4 is incident with the loop e_3; $\deg(v_5) = 0$. □

Example 5.5 A graph can be used in chemistry to model the structure of a molecule. In this model, the atoms of the elements composing the molecule are taken as the vertices of the graph and the edges represent the chemical bonds between pairs of atoms. So for example, a graph model of the water molecule H_2O would have three vertices and two edges. The degree of a vertex is the *valency* of the atom it represents. Since an early use of graph theory (by Arthur Cayley, in the mid 19th century) was to count the number of different isomers of some of the hydrocarbons, the term *valency* was initially used instead of *degree* of a vertex, and you will still find this usuage in some text books. □

We next investigate the connection between the number of *edges* in a graph and the *degrees* of the vertices. If, for each of the graphs shown Figures 5.1, 5.2 and 5.3, you sum the degrees of the vertices and compare this sum with the number of edges of the graph, you should find that the following result holds.

Result 5.3 *Let G be a graph. Then the sum of the degrees of the vertices of G is equal to twice the number of edges of G.*

Proof. Let e be any edge of G and suppose that the endpoints of e are u and v. Then e contributes 1 both to $\deg(u)$ and to $\deg(v)$. (If e is a loop, then $u = v$ and e contributes 2 to $\deg(u)$). Thus every edge contributes 2 to the sum of the degrees of the vertices, giving the result. □

Result 5.3 can be used to find the number of edges in a graph when we know the vertex degrees. It can also be used to show that it is not possible to construct graphs with some combinations of vertex degrees.

Example 5.6 A graph can be used to model acquaintanceship between pairs of people in a group. In this model, each person in the group is represented by a vertex. Two vertices in the graph are joined by an edge if and only if the two people they represent know one another. Thus the number of edges in this graph gives the number of pairs of people in the group who know one another. (Note that in order to be useful, acquaintance graphs are *simple* graphs; that is, we do not record the obvious fact that each person knows himself or herself by putting a loop at each vertex.)

Suppose that there are ten people in the group and we ask each of them how many other people in the group they are acquainted with. Suppose they give us the following numbers:

$$2, 5, 6, 6, 4, 3, 9, 1, 7, 5.$$

We cannot construct the acquaintance graph from this information alone, because we do not know *which* other people in the group each person knows. However, we can calculate the *number of edges* in the graph from just the information given. This is because the number of acquaintances of a given member of the group gives us the *degree* of the vertex representing her or him. Summing the sequence of numbers given above, we obtain

$$2 + 5 + 6 + 6 + 4 + 3 + 9 + 1 + 7 + 5 = 48.$$

Hence, using Result 5.3, the number of different pairs of acquaintances in the group is $48/2 = 24$. □

Definition 5.4 *The degree sequence of a graph G is the sequence formed from the degrees of its vertices, usually arranged in descending order of size.*

Example 5.7 The degree sequence for the acquaintance graph described in Example 5.6 is:

$$9, 7, 6, 6, 5, 5, 4, 3, 2, 1.$$

Example 5.8 Is it possible to have a group of 7 people who each know 3 other members of the group?

The sum of the degrees of the vertices in the acquaintance graph for this group of people is $7 \times 3 = 21$. But this sum should give us twice the number of edges in the graph, and hence it must be an *even* integer. Thus it is not possible for a group of 7 people to each know 3 of the others. □

In solving this example, we have used a deduction from Result 5.3 that is true for any graph.

Result 5.5 *In any graph, the sum of the degrees of the vertices is an even integer.* □

5.1.3 Some special graphs

Definition 5.6 *A graph in which every vertex has the same degree r is called r-regular.*

The graph depicted in Figure 5.4 is an example of a 3-regular graph with 8 vertices.

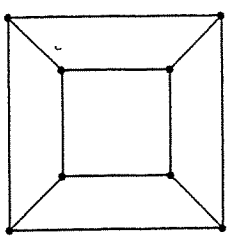


Figure 5.4.

Definition 5.7 *A simple graph in which every vertex is joined to every other vertex is called a complete graph. A complete graph with n vertices is denoted by K_n.*

Figure 5.5 shows the complete graphs K_1, K_2, K_3 and K_4.

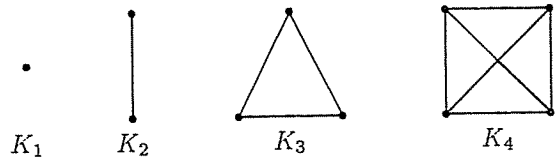


Figure 5.5.

5.2 Paths, cycles and connectivity

Learning objectives

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

- define the terms *path* and *cycle* in a graph;
- state the condition for a graph to be *connected*;
- find an example of a path or cycle of a given length in a graph.

Introduction

In several applications of graph theory, the main problems relate to the existence of chains of edges between given vertices of the graph. Examples of this occur in the routing of messages through a communications network. We might want to check whether there exists a route between a given pair of vertices and, if so, to find the shortest path between them. These considerations motivate the definitions in this section.

5.2.1 Paths

Definition 5.8 *A path is an alternating sequence of vertices and edges of the form*

$$v_1 e_1 v_2 e_2 v_3 \ldots e_{k-1} v_k,$$

where e_i is the edge joining v_i to v_{i+1}, and all the vertices and all the edges are distinct.

Note that in a simple graph, there is *at most* one edge joining any pair of vertices and so a path can be specified just by a sequence of distinct vertices.

Example 5.9 The sequence $uvwx$ denotes a path from u to x in the graph depicted in Figure 5.2. However, if we wanted to specify a path from v_1 to v_3 in the graph shown in Figure 5.3, then we would have to say which of the parallel edges between v_2 and v_3 should be used. Thus we would denote the two paths from v_1 to v_3 in this graph by

$$v_1 e_4 v_2 e_1 v_3 \text{ and } v_1 e_4 v_2 e_1 v_3. \quad \square$$

Definition 5.9 *The length of a path is the number of* edges *in it.*

A common mistake is to take the number of vertices in a path as its length instead of the number of edges.

Example 5.10 The path $uvwx$ in the graph shown in Figure 5.2 has length 3, and the paths $v_1 e_4 v_2 e_1 v_3$ and $v_1 e_4 v_2 e_1 v_3$ in the graph shown in Figure 5.3 each have length 2.

5.2.2 Cycles

Definition 5.10 *Suppose that $v_1e_1v_2e_2v_3\ldots e_{k-1}v_k$ is a path in a graph G. If v_kv_1 is also an edge e_k, say, of G, then we say that the vertex-edge sequence*

$$v_1e_1v_2e_2v_3\ldots e_{k-1}v_ke_kv_1$$

forms a cycle. The **length** *of the cycle is the number of edges in it. We sometimes refer to a cycle of length k as a k-***cycle***.*

As with paths, we can omit the edges when specifying a cycle in a simple graph. Thus a cycle is a sequence of distinct vertices and edges that begins and ends at the same vertex. Two paths (or two cycles) are different if they differ in at least one edge. Thus we can specify a cycle starting at *any of its vertices* and travelling *either way round*.

Example 5.11 In the graph shown in Figure 5.1, the cycle $ABFDA$ can be also be specified as $ADFBA$; or, starting at B, for example, we could write it as $BFDAB$. The length of this cycle is 4, so we call it a 4-cycle. Note that $BCDFB$ is also a 4-cycle in this graph. □

5.2.3 Connectivity

In any communications network, it is often important to know whether or not there exists a route between each pair of vertices.

Definition 5.11 *Two vertices u, v in a graph G are said to be* ***connected*** *if there is a path in G from u to v. The graph G is said to be* ***connected*** *if each pair of vertices is connected; otherwise, G is said to be* ***disconnected.***

Example 5.12 The graphs shown in Figure 5.1 and Figure 5.2 are both connected, whereas the graph shown in Figure 5.3 is disconnected because, for example, there is no path to v_5 from any of the other vertices. □

You can visualize a connected graph as being one that is "all in one piece". The separate connected parts of a disconnected graph are called its **connected components**. Thus the graph shown in Figure 5.3 has 2 connected components; every connected graph has just one connected component, the graph itself.

5.3 Isomorphism of graphs

Learning objectives

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

- state the definition of *graph isomorphism*;
- decide whether two given graphs on n vertices (for small values of n) are isomorphic and justify your decision.

Introduction

A graph G is determined by its vertex set $V(G)$ and its edge set $E(G)$. However, given this information, two people might draw the graph differently, so that it is not immediately obvious that their pictures represent the same graph. An example is shown in Figure 5.6, where both diagrams correctly depict the graph G defined by $V(G) = \{u, v, w, x\}$ and $E(G) = \{uv, uw, ux, vw, vx, wx\}$.

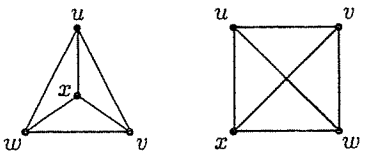


Figure 5.6.

Two graphs which are the same except for the way their vertices have been labelled or the way the graphs have been drawn are called *isomorphic.* Thus two graphs G and H are isomorphic if we can label their vertices with the same set of labels in such a way that any pair of vertices u, v are joined by the same number of edges in G as they are in H. Now "labelling the vertices with the same set of labels" is just a way of establishing a one-to-one correspondence between the vertex sets $V(G)$ and $V(H)$. This is the idea behind the following definition.

Definition 5.12 *The graphs G and H are called* ***isomorphic*** *if there is a one-to-one correspondence $f : V(G) \to V(H)$ such that the number of edges joining any pair of vertices u, v in the graph G is the same as the number of edges joining the vertices $f(u)$, $f(v)$ in H.*

Notice that when G and H are *simple* graphs, then there is at most one edge joining any pair of vertices. Thus the condition for a one-to-one correspondence between $f : V(G) \to V(H)$ to be an isomorphism is that vertices u and v are adjacent in G if and only if their images $f(u)$ and $f(v)$ are adjacent in H. We express this fact by saying that an isomorphism **preserves adjacency**.

Definition 5.13 *Let u, v be adjacent vertices. We say that u is a* **neighbour** *of v and v is a neighbour of u.*

5.3.1 Showing that two graphs are isomorphic

Suppose we are given two graphs G and H and we require to establish whether or not they are isomorphic. If an isomorphism $f : V(G) \to V(H)$ exists, then G and H must certainly have *the same number of vertices.* Moreover, since the number of edges joining any pair of vertices u, v in G is the same as the number of edges joining $f(u)$ and $f(v)$ in H, then G and H must also have *the same number of edges.* Further, the number of edges incident with a given vertex u in G must be the same as the number of edges incident with its image $f(u)$ in H, so that *u and $f(u)$ must have the same degree.* Since this applies to *every* vertex of G, graphs G and H must have *the same degree sequence.* This last condition is the most powerful, since if two graphs have the same degree sequence, then they must have the same number of vertices (because this is the number of terms in the degree sequence) and by Result 5.3, they must also have the same number of edges.

Thus if G and H do *not* have the same degree sequence, we can immediately conclude that they are *not* isomorphic. If they do have the same degree sequence, it is possible they are isomorphic but *this is not certain*, and so further investigation must take place. The next example illustrates how we might go about trying to establish an isomorphism between two graphs with the same degree sequence.

Example 5.13 Both graphs shown in Figure 5.7 have degree sequence $3, 3, 2, 1, 1, 1, 1$, but they have been drawn so that they appear to be different graphs. In fact they are isomorphic. To show this, we must construct a one-to-one correspondence between $V(G_1)$ and $V(G_2)$ that preserves adjacency.

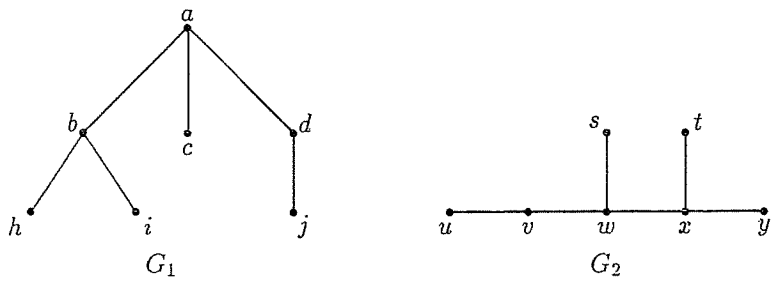


Figure 5.7.

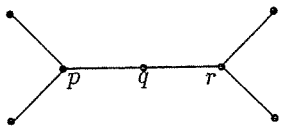
First note that each graph has just one vertex of degree 2; thus in any isomorphism, this pair of vertices must correspond. So we start by defining $f(d) = v$. The vertices d and v each have a neighbour of degree 1 and a neighbour of degree 3; so to preserve adjacency, we must define $f(j) = u$ and $f(a) = w$. Comparing the neighbours of a and w, we see that we must define $f(b) = x$ and $f(c) = s$. Finally, the two neighbours of b of degree 1 must be paired with the two neighbours of x of degree 1. This can be done in two different ways and so there are two possible isomorphisms of G_1 and G_2, one of which is given in the table below.

z	a	b	c	d	h	i	j
$f(z)$	w	x	s	v	t	y	u

□

5.3.2 Showing that two graphs are not isomorphic

Example 5.14 Figure 5.8 illustrates a graph H that has the same degree sequence as the graphs G_1 and G_2 shown in Figure 5.7. Can we conclude that H is isomorphic to G_1 and G_2? No, we cannot so conclude. To see this, note that in G_1 and G_2, the two vertices of degree 3 are *adjacent*, whereas in H they are separated by the vertex q with degree 2. Any isomorphism from H to G_1 must pair the vertex q of degree 2 in H with the vertex d of degree 2 in G_1. But then d has a neighbour of degree 1 in G_1, whereas the neighbours of q both have degree 3 in H. Thus no isomorphism between H and G_1 is possible. Similarly, no isomorphism between H and G_2 is possible. □



H

Figure 5.8.

Example 5.14 illustrates that to show that two graphs are *not* isomorphic, we must look for some property depending upon adjacencies that is possessed by one graph and not by the other.

Since the vertices in a path (or cycle) form a sequence of *adjacent* vertices in a given graph G, their images must also lie in a path (or cycle) in any graph isomorphic to G. Thus when comparing two graphs to see whether they are isomorphic, we can consider the existence of paths or cycles of a given length. In particular, a connected graph can never be isomorphic to a disconnected graph, because in one graph there is a path between each pair of vertices and in the other there is no path between a pair of vertices in different components. Reasoning in this way, we can produce a "check list" of differences that we might look for to show that two graphs are *not* isomorphic. They are listed below in the order in which they are usually easiest to check.

Result 5.14 *Two given graphs are NOT isomorphic if:*

1. *they have a different number of connected components;*
2. *they have a different number of vertices;*
3. *they have different degree sequences;*
4. *they have a different number of paths of any given length;*
5. *they have a different number of cycles of any given length.* □

Example 5.15 The longest path in each of the graphs G_1 and G_2 (see Figure 5.7) and H (see Figure 5.8) has length 4. The graphs G_1 and G_2 each have two different paths of length 4, while H has four different paths of length 4. This gives another way of showing that H is not isomorphic to either of G_1 and G_2. □

If you cannot find any differences between the two graphs, then they may be isomorphic and you should start trying to construct a one-to-one correspondence between the vertex sets that preserves adjacency.

5.4 Adjacency matrices and adjacency lists

Learning objectives

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

- explain how to construct the adjacency matrix of a graph;
- explain how to construct a set of adjacency lists for a simple graph;
- find the degree of a given vertex and the total number of edges in a loopless graph from its adjacency matrix.

Introduction

If we want to use a computer to analyse the properties of a graph, we must have some way of representing it in the computer. This section describes two methods by which this can be achieved.

5.4.1 Adjacency matrix of a graph

One method is to record the number of edges joining each pair of vertices in a square array of numbers, called a *matrix*.

Definition 5.15 *Suppose that G is a graph with n vertices, numbered $1, 2, \ldots, n$. Then the* **adjacency matrix** *$\mathbb{A}(G)$ of G is a square $n \times n$ array, with rows and columns numbered $1, 2, \ldots, n$, such that the entry in row i and column j is the number of edges joining vertex i to vertex j.*

Example 5.16 The adjacency matrix of the graph shown in Figure 5.3 is as follows.

	v_1	v_2	v_3	v_4	v_5
v_1	0	1	0	0	0
v_2	1	0	2	0	0
v_3	0	2	0	1	0
v_4	0	0	1	1	0
v_5	0	0	0	0	0

Definition 5.16 *The diagonal line of cells from the top left hand cell to the bottom right hand cell of a square matrix is called the* **main diagonal.** *A square matrix is called* **symmetric** *if the entries in row i are the same as the entries in column i, for $i = 1, 2, \ldots, n$.*

The adjacency matrix of a graph is always *symmetric* because the number of edges joining vertex i to vertex j is equal to the number of edges joining vertex j to vertex i. Geometrically, the symmetry of the matrix is about the main diagonal. Note that the entries on the main diagonal record the number of *loops* at each vertex.

5.4.2 Adjacency lists

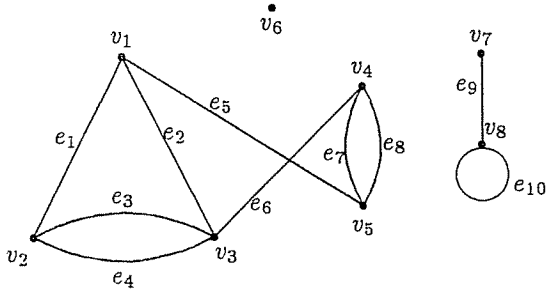
In some large graphs, each vertex may only be adjacent to relatively few other vertices. In this case, a high proportion of the entries in the adjacency matrix will be zero, which is wasteful of storage space. An alternative method of storing a *simple* graph is to use a set of **adjacency lists**. First, we list the vertices of the graph; then, after each vertex, we put a colon and list the vertices that it is adjacent to. The procedure is illustrated in the example below.

Example 5.17 The adjacencey lists for the graph shown in Figure 5.2 are as follows.

u : v
v : w, x
w : v, x
x : v, w. □

5.5 Exercises 5

1. For the graph shown below, find:
   (a) all the edges incident with v_1;
   (b) all the vertices adjacent to v_3;
   (c) all the edges adjacent to e_2;
   (d) all the loops;
   (e) the number of its connected components;
   (f) $\deg(v_4)$, $\deg(v_6)$ and $\deg(v_8)$;
   (g) the degree sequence of the graph;
   (h) all the cycles of lengths 2, 3 and 4 respectively.



2. In each of the following cases, either construct a graph with the specified properties or say why it is not possible to do so.
   (a) A graph with degree sequence $3, 3, 2, 1$.
   (b) A simple graph with degree sequence $3, 3, 2, 1, 1$.
   (c) A simple graph with degree sequence $4, 3, 2, 1$.
   (d) A simple 3-regular graph with 6 vertices.
3. Say why every graph has an *even* number of vertices of *odd* degree.
4. Suppose that eight sites are connected in a network. The number of other sites to which each site has a direct connection is given by the following sequence.

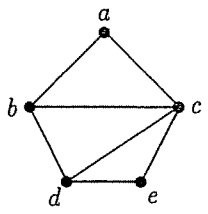
$$5, 3, 2, 7, 1, 2, 6, 4.$$

   (a) Describe how a communications network such as this can be modelled by a *graph*, saying what the vertices represent and when two vertices are adjacent.
   (b) What information about the graph is given by the sequence of numbers above?
   (c) Find how many pairs of sites have a direct connection between them, giving a brief explanation of your method.
   (d) Say why it is impossible to construct a network with 9 sites, in which each site has a direct connection to exactly 5 of the other sites.

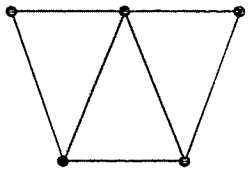
5. (a) Sketch the complete graph K_5. What is the degree of each vertex? How many edges does it have?

   (b) What is the degree of each vertex of the complete graph K_n? How many edges does it have?

6. Are the graphs G and H shown below isomorphic? If you think they are isomorphic, label each vertex of H with the same letter as the corresponding vertex in G. Otherwise, give a reason why the two graphs are not isomorphic.



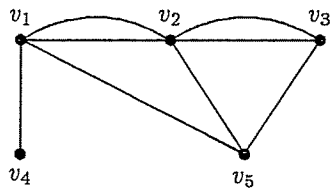
G



H

7. Draw two simple non-isomorphic graphs with degree sequence $3, 2, 2, 1, 1, 1$. Give a reason why the graphs you have drawn are not isomorphic.

8. Construct an adjacency matrix $\mathbb{A}(G)$ for the graph G shown below.



   (a) What information does the sum of the elements in any row of $\mathbb{A}(G)$ give you about the graph G?

   (b) What information does the sum of all the elements in the matrix tell you about G?

   (c) Would these rules hold in the case of a graph with loops?

Appendix A

Additional References

There are a number of books available on Discrete Mathematics. This list contains a selection of those known to the authors. Other editions (earlier or later) can be used instead of the edition listed below.

The books in this list are additional to the main text books for the course (by Epp and by Molluzzo and Buckley), for which details have been given in the Introduction.

Albertson, M.O. and Hutchinson, J.P. *Discrete Mathematics with Algorithms.* (John Wiley and Sons, 1998) [ISBN 0-471-61278-2(pbk) or 0-471-84902-2(hbk)].
Good background book, showing links with computer science.

Barnet, Stephen *Discrete Mathematics: Numbers and Beyond.* (Addison Wesley, 1988) [ISBN 0-201-34292-8(pbk)].
Does not cover all the syllabus, but useful for some sections, particularly number bases, counting methods, applications of graphs.

Eccles, Peter J. *An Introduction to Mathematical Reasoning: Numbers, sets and functions.* (Cambridge University Press, 1997) [ISBN 0-521-59718-8].

Grimaldi, R.P. *Discrete and Combinatorial Mathematics* (Addison-Wesley World Student Series, 1999) 4^{th} Ed. [ISBN 0-201-30424-4(pbk)] or (Addison-Wesley, 1998) [ISBN 0-201-19912-2(hbk)] (earlier editions are just as useful).
A comprehensive coverage of the whole subject area, but more demanding than Epp.

Garnier, Rowan and Taylor, John. *Discrete Mathematics for new Technology.* (The Institute of Physics, 1997) [ISBN 0-7503-0135-X(pbk)].
Good coverage and simply explained.

Goodaire, Edgar G. and Parmenter Michael M. *Discrete Mathematics with Graph Theory.* (Prentice Hall, 1998) [ISBN 0-13-602798-8(hbk)] or (Prentice Hall, 2001) [ISBN 0-13-092000-2(hbk)].

Johnsonbaugh, Richard *Discrete Mathematics.* 5^{th} Ed. (Prentice Hall, 2000) [ISBN 0-13-089008-1(hbk)].

Mattson Jr., Harold F. *Discrete Mathematics with Applications.* (John Wiley and Sons (WIE), 1993) [ISBN 0-471-59966-2(pbk)] or (John Wiley and Sons, 1993) [ISBN 0-471-60672-3(hbk)].
A more demanding treatment, but good for links with computer science.

Ross, Kenneth A. and Wright, Charles R.B. *Discrete Mathematics* 4^{th} Ed. (Prentice Hall, 1999) [ISBN 0-13-218157] An excellent book, covering the whole syllabus and much more besides, with a good selection of exercises.

Simpson, Andrew *Discrete Mathematics by Example.* (McGraw-Hill Education, 2001) [ISBN 0-07-709840-4(pbk)].

Appendix B

Solutions to Exercises

Exercises 1

1. (a) $4(10^3)+3(10)+7$; $4(10^4)+3(10^2)+7(10)+1(1)$; (b) $a = a_3(10^3)+a_2(10^2)+a_1(10)+a_0(1)$; $x = a_3(10^4)+a_2(10^3)+a_1(10^2)+a_0(10)+1$; $x = 10a+1$.

2. (a) $2(5^2)+3(5)+4 = 69$; (b) $(2340)_5$.

3. 27; 102; 511. The quick way is to see that $(111111111)_2 = 2^9 - 1$.

4.

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

Thus (a) $(10111)_2 + (111010)_2 = (1010001)_2$; (b) $(1101)_2 + (1011)_2 + (1111)_2 = (100111)_2$.

5.

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

Hence (a) $(1001)_2 - (111)_2 = (10)_2$; (b) $(110000)_2 - (10111)_2 = (11001)_2$.

6.

$$\begin{array}{ccccccc} & & & 1 & 1 & 0 & 1 \\ & & & & 1 & 0 & 1 \\ \hline & & 1 & 1 & 0 & 1 & 0 & 0 \\ & & & & 1 & 1 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \qquad \begin{array}{cccccccc} & & & & 1 & 1 & 0 & 1 \\ & & & & 1 & 1 & 0 & 1 \\ \hline & & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ & & & 1 & 1 & 0 & 1 & 0 & 0 \\ & & & & & 1 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}$$

Hence (a) $(1101)_2 \times (101)_2 = (1000001)_2$; (b) $(1101)_2 \times (1101)_2 = (10101001)_2$.

7.

$$\begin{array}{r|cccccc} & & & 1 & 0 & 1 & 1 \\ \hline 101 & 1 & 1 & 1 & 0 & 1 & 1 \\ & 1 & 0 & 1 & & & \\ \hline & & 1 & 0 & 0 & 1 & \\ & & & 1 & 0 & 1 & \\ \hline & & & 1 & 0 & 0 & 1 \\ & & & & 1 & 0 & 1 \\ \hline & & & & 1 & 0 & 0 \end{array}$$

Hence $(111011)_2 \div (101)_2 = (1011)_2$, remainder $(100)_2$.

8. $23 + 58 = 81$, $13 + 11 + 15 = 39$; $9 - 7 = 2$, $48 - 23 = 25$; $13 \times 5 = 65$, $13 \times 13 = 169$; $59 \div 5 = 11$, remainder 4.

9. (a) 2653; (b) $a = a_2(16^2) + a_1(16) + a_0(1)$; $x = 16a$, $y = 16a + 10$.

10.

1	1	1				A	1B
	B	B	B		B	B	B
	A	5	D		A	5	D
1	6	1	8		1	5	E

Hence $(BBB)_{16} + (A5D)_{16} = (1618)_{16}$; $(BBB)_{16} - (A5D)_{16} = (15E)_{16}$.

11. (a) $(111011000100)_2$; (b) $(3DA)_{16}$.

12. (a) $(1011110001)_2$; (b) $(11003)_5$; (c) $(2F1)_{16}$.

13. $42900 = 2^2 \times 3 \times 5^2 \times 11 \times 13$.

14. (a) Let $x = 0.126126126\ldots$. Then $1000x = 126.126126126\ldots$. Subtracting x gives $999x = 126$, so that $x = 126/999 = 14/111$.

(b) Let $x = 0.7545454\ldots$. Then $100x = 75.454545\ldots$. Subtracting x gives $99x = 74.7$. Hence $x = 74.7/99 = 747/990 = 83/110$.

15. $\frac{\pi}{2}$ is irrational. For, suppose $\frac{\pi}{2}$ is rational. Then we can write $\frac{\pi}{2} = \frac{a}{b}$, where a, b are integers. But this gives $\pi = \frac{2a}{b}$, which implies π is rational also, which we know is false.

16. (a) $5/8 = 0.625$; $(0.A)_{16}$; (b) $11 + \frac{2}{16} + \frac{5}{256} = 11.14453125$; $(1011.00100101)_2$; (c) $(0.011)_2$; (d) $\frac{3}{5} + \frac{2}{25} = 0.68$.

17. (a) $0.714 < 5/7 < 0.715$; (b) $\sqrt{2} \geq 1.41$; (c) $7.32 \leq \sqrt{3} \leq 7.322$.

18. 0.526×10^{-4}; 0.429×10^{9}.

19. (a) If $a > \sqrt{n}$, then $b > \sqrt{n}$ also, so that $ab > n$, which is impossible. So the smaller factor a must be at most $\sqrt{n}$. (Note that if $a = \sqrt{n}$, then $b = \sqrt{n}$ also, and so $n = a^2$.)

(b) If a is prime, then a is the required prime factor of n; otherwise, a is composite. In this case a has a prime factor p such that $p < a$. Then p is a prime factor of n such that $p \leq \sqrt{n}$.

(c) If 89 is composite, then it must have a prime factor p such that $p \leq \sqrt{89}$. Since $9 < \sqrt{89} < 10$, we test 89 for divisibility by each prime less than 10, that is, by 2,3,5 and 7. Since none of these divides 89, we know 89 is prime.

(d) First note that $29 < \sqrt{899} < 30$. Hence if 899 is composite, it is divisible by one of the primes $2, 3, 5, \ldots, 29$. You will find one of these primes is a factor of 899, so that 899 is composite.

20. (a) 12. (b) 6.

(c) Yes, they each have exactly 6 digits in the recurring block. In the case of 1/13, 3/13 and 4/13 these digits are the same, but each fraction starts at a different point of the sequence.

(d) One method is to use the fact that the fraction $a/13$ has the same digits as $1/13$ in its recurring block if a is one of the sequence of remainders obtained when $1.000\ldots$ is divided by 13.

Exercises 2

1. (a) $\{1,2,3,4,5\}$; (b) $\{-1,0,1,2,3,4\}$; (c) $\{3,5,7,9\}$; (d) $\{10,12,14,16,18,20\}$; (e) $\{1,2,4,8,16,32\}$.

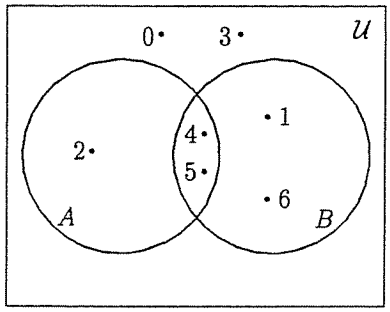
2. (a) $\{r : r \in \mathbb{Z} \text{ and } 12 \leq r \leq 17\}$; (b) $\{5m : m \in \mathbb{Z}\}$; (c) $\{-a : a \in \mathbb{Z} \text{ and } 1 \leq a \leq 10\}$; (d) $\{2n : n \in \mathbb{Z} \text{ and } 3 \leq n \leq 9\}$; (e) $\{5^k : k \in \mathbb{Z} \text{ and } k \geq 0\}$; (f) $\{(0.1)^s : s \in \mathbb{Z}^+\}$.

3. They are all equal; 3.

4. 10110; 01000; 11111; 00000; $\{a, u\}$.

5. $A \subseteq S$; $3 \in S$; $0 \in S$; $\emptyset \subseteq S$; $S \in \mathcal{P}(S)$; $A \in \mathcal{P}(S)$.

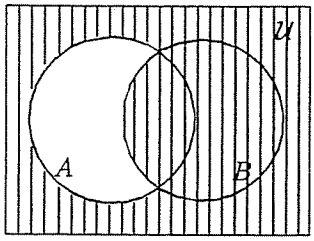
6. $A \cup B = \{1,2,4,5,6\}$; $A \cap B = \{4,5\}$; $(A \cup B)' = \{0,3\}$; $A - B = \{2\}$; $B - A = \{1,6\}$; $A \oplus B = \{1,2,6\}$.



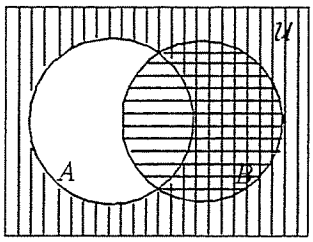
7.

A	B	$A-B$	$(A-B)'$	A'	$A' \cup B$
0	0	0	1	1	1
0	1	0	1	1	1
1	0	1	0	0	0
1	1	0	1	0	1

Since the columns for $(A-B)'$ and $A' \cup B$ have the same entries, these sets are equal.



$(A-B)'$ is shaded region



$A' \cup B$ is total shaded region

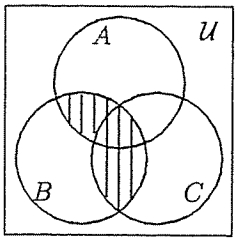
8. (a)

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

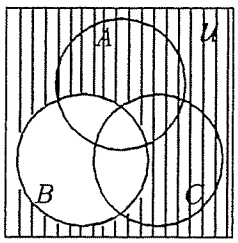
Since the columns for $(A \cup B)'$ and $A' \cap B'$ have the same entries, these sets are equal.

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

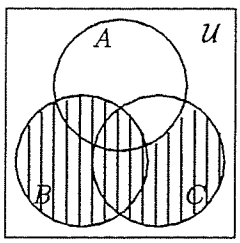
9. (a) and (b)



X is shaded region



Y is shaded region



Z is shaded region

(c) $X \subseteq Z$; (d) $X = (A \cap B) \cup (B \cap C)$; $Y = B'$; $Z = B \cup (C \cap A')$ (there is more than one correct answer).

10.

(a)

A	B	C	$A-B$	$(A-B)-C$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	0	0

A	B	C	$B-C$	$A-(B-C)$
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	0	0
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1

(b) The only statement that is true for *all* sets A, B, C is (ii) $(A-B)-C \subseteq A-(B-C)$.

Exercises 3

1. $\{011, 101, 110\}$; $\{000, 001, 110, 111\}$; $\{110\}$; $\{011, 101, 110, 000, 001, 111\}$

2. n is greater than 50; n is between 10 and 50 inclusive (*or* n is at least 10 and at most 50); n is not less than 10; n is greater than 50 or less than 10.

3. (a) $\neg(p \vee q)$, $\neg p \wedge \neg q$; (b) $\neg(p \wedge q)$, $\neg p \vee \neg q$.

4.

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

Since the columns for $(p \wedge q) \vee (\neg p \wedge q)$ and q have the same entries, these statements are equivalent.

5.

(a)

p	F	$p \vee F$
0	0	0
1	0	1

(b)

p	T	$p \wedge T$
0	1	0
1	1	1

Hence $p \vee F = p$ and $p \wedge T = p$.

6. (a) $p \to r$; (b) $p \to q$; (c) $p \to q$.

7.

n	q	r	$q \to r$	$r \to q$	$q \leftrightarrow r$
-8	1	0	0	1	0
-3	0	0	1	1	1
10	1	1	1	1	1
17	0	1	1	0	0

8.

p	q	$p \leftrightarrow q$	$\neg p$	$\neg q$	$\neg p \leftrightarrow \neg q$
0	0	1	1	1	1
0	1	0	1	0	0
1	0	0	0	1	0
1	1	1	0	0	1

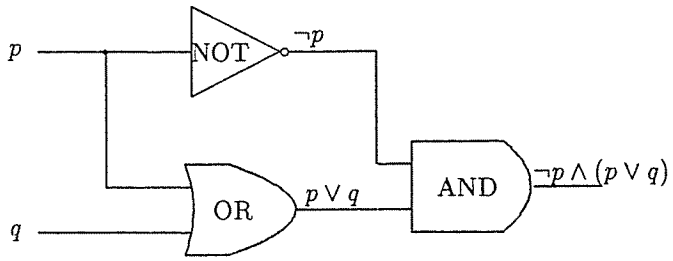
Since the columns for $p \leftrightarrow q$ and $\neg p \leftrightarrow \neg q$ have the same entries, these statements are equivalent.

9. (a) If n is not divisible by 3, then $n \neq 12$; (b) if n is not positive, then $n \neq 5$ (notice that "If n is *negative*, then $n \neq 5$" is an incorrect solution, because it excludes the number 0); (c) if its four sides are not equal, then the quadrilateral is not a square.

10.

p	q	r	$p \to q$	$q \to r$	$(p \to q) \wedge (q \to r)$	$p \to r$	$[(p \to q) \wedge (q \to r)] \to (p \to r)$
0	0	0	1	1	1	1	1
0	0	1	1	1	1	1	1
0	1	0	1	0	0	1	1
0	1	1	1	1	1	1	1
1	0	0	0	1	0	0	1
1	0	1	0	1	0	1	1
1	1	0	1	0	0	0	1
1	1	1	1	1	1	1	1

11. (a)



Notice that in labelling the output from the final gate, the brackets in the expression $\neg p \wedge (p \vee q)$ are essential.

(b)

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

(c) $\neg p \wedge q$

Exercises 4

1. (a) X *or* $\{1,2,3,4\}$; (b) Y *or* $\{1,2,3,\ldots,10\}$; (c) $f(2) = 4$; (d) 4; (e) $\{1,4,7,10\}$.

2. (a) $\{000,001,010,011,100,101,110,111\}$; (b) 1; (c) $\{000,010,100,110\}$; (d) B *or* $\{0,1\}$.

3. (a) $x - y$; (b) $y - x$; 0.

4. $\{8,12\}$

5. (a) $\lfloor -1.4 \rfloor = -2$, $\lceil -1.4 \rceil = -1$; (b) $\lfloor 2.3 \rfloor = 2$, $\lceil 2.3 \rceil = 3$; (c) $\lfloor 7/9 \rfloor = 0$, $\lceil 7/9 \rceil = 1$; (d) $\lfloor -16/3 \rfloor = -6$, $\lceil -16/3 \rceil = -5$; (e) $\lfloor 0 \rfloor = \lceil 0 \rceil = 0$.

6. (a) $n+1$, $n-3$.

(b) From the definition of $\lfloor x \rfloor$, we know that $n \leq x < n+1$. Adding 1 to each part of these inequalities gives $n+1 \leq x+1 < n+2$ and hence $\lfloor x+1 \rfloor = n+1$. Similarly, subtracting 3 from each part of the inequalities gives $n-3 \leq x-3 < n-2$, implying $\lfloor x-3 \rfloor = n-3$.

(c) For example, let $x = y = 1.8$. Then $\lfloor x \rfloor + \lfloor y \rfloor = 1+1 = 2$. But $\lfloor x+y \rfloor = \lfloor 3.6 \rfloor = 3$.

7. (a) f is not onto, because e.g. 2 has no pre-image; f is one-to-one, because distinct elements in the domain have distinct images.

(b) f is onto, because every element of the codomain has a pre-image; f is not one-to-one, because e.g. $f(000) = f(100) = 0$.

(c) f is onto, because every element of the codomain has a pre-image; f is one-to-one, because distinct elements of the domain have distinct images.

(d) f is onto: let $y \in \mathbb{R}$; then putting $x = (y+1)/5$, we have $f(x) = y$ and $x \in \mathbb{R}$; hence every element y in the codomain has a pre-image in the domain.

f is one-to-one: suppose $f(x_1) = f(x_2)$, then $5x_1 - 1 = 5x_2 - 1$, giving $x_1 = x_2$. Hence distinct elements in the domain have distinct images.

(e) g is not onto, because e.g. 2 has no pre-image; g is one-to-one, with similar proof to (d) above.

(f) h is not onto, because e.g. -2 has no pre-image; h is not one-to-one, because e.g. $h(-1) = h(1) = 4$.

8. The functions in 7(a),(e),(f) are not invertible because they are not onto; the function in 7(b) is not invertible because it is not one-to-one. The functions in 7(c) and 7(d) are invertible.

The inverse of the function f in 7(c) is defined by the table below.

x	5	3	1	2	4
$f^{-1}(x)$	1	2	3	4	5

or

x	1	2	3	4	5
$f^{-1}(x)$	3	4	2	5	1

To find the inverse of the function f in 7(d), let $f(x) = y$, so that $y = 5x - 1$. Solving for x gives $x = \frac{y+1}{5}$. Hence $f^{-1}(y) = \frac{y+1}{5}$.

9. $8^{\frac{1}{3}} = 2$; $8^{-\frac{2}{3}} = (8^{\frac{1}{3}})^{-2} = 2^{-2} = \frac{1}{4}$; $16^{1.25} = 16^{\frac{5}{4}} = (16^{\frac{1}{4}})^5 = 2^5 = 32$; $32^{1.4} = 32^{\frac{7}{5}} = (32^{\frac{1}{5}})^7 = 2^7 = 128$; $32^{-0.6} = 32^{-\frac{3}{5}} = (32^{\frac{1}{5}})^{-3} = 2^{-3} = \frac{1}{8}$.

10. $128 = 2^7$, so $\log_2 128 = 7$; $4\sqrt{2} = 2^2 \times 2^{\frac{1}{2}} = 2^{2.5}$. Hence $\log_2 4\sqrt{2} = 2.5$; $\frac{1}{8} = 2^{-3}$ and so $\log_2 \frac{1}{8} = -3$; $\frac{\sqrt{2}}{16} = 2^{\frac{1}{2}} \times 2^{-4} = 2^{-3.5}$. Hence $\log_2 \frac{\sqrt{2}}{16} = -3.5$.

11. $f^{-1}(x) = x^{\frac{1}{3}}$; $f^{-1}(x) = x^4$; $f^{-1}(x) = x^{\frac{2}{3}}$.

12. Table of values (entries rounded to 2 d.p.) is shown below.

x	$\frac{1}{4}$	$\frac{1}{2}$	1	2	4	8	12	16	20	24	28	32
$f(x)$	-2	-1	0	1	2	3	3.58	4	4.32	4.58	4.81	5
$g(x)$	0.5	0.71	1	1.41	2	2.83	3.46	4	4.7	4.90	5.29	5.66

Plotting the graphs of $y = f(x)$ and $y = g(x)$ (see Figure 4.4), we see that $f(x)$ is nearer to the x-axis than $g(x)$ when $x > 16$, implying $\log_2 x = O(x^{\frac{1}{2}})$.

13. When $x > 1$, $x\sqrt{x} > x > 1$, so that $f(x) < 2x\sqrt{x} + 5x\sqrt{x} + 24x\sqrt{x} = 31x\sqrt{x} = 31x^{\frac{3}{2}}$. Hence when $x > 1$, $|f(x)| < 31|x^{\frac{3}{2}}|$, giving $f(x) = O(x^{\frac{3}{2}})$.

14. The following table shows the number of bits in the binary representation of k when k lies in the given range.

k	1	2 – 3	4 – 7	8 – 15	16 – 31	...
number of bits	1	2	3	4	5	...

From the table, we see that when $2^r \leq k < 2^{r+1}$, the number of bits in the binary representation of k is $r + 1$. However, $2^r \leq k < 2^{r+1}$ implies that $r \leq \log_2 k < r + 1$. Hence $r = \lfloor \log_2 k \rfloor$. Thus the number of bits required is $r + 1 = \lfloor \log_2 k \rfloor + 1$.

Exercises 5

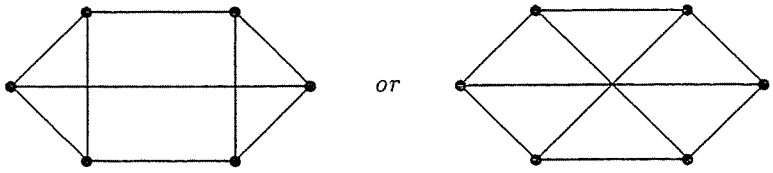
1. (a) e_1, e_2, e_5; (b) v_1, v_2, v_4; (c) e_1, e_3, e_4, e_5, e_6; (d) e_{10}; (e) 3; (f) $\deg(v_4) = 3$, $\deg(v_6) = 0$, $\deg(v_8) = 3$; (g) $4, 3, 3, 3, 3, 3, 1, 0$; (h) length 2 : $v_2e_3v_3e_4v_2$, $v_4e_7v_5e_8v_4$; length 3 : $v_1e_1v_2e_3v_3e_2v_1$, $v_1e_1v_2e_4v_3e_2v_1$; length 4 : $v_1e_2v_3e_6v_4e_7v_5e_5v_1$, $v_1e_2v_3e_6v_4e_8v_5e_5v_1$.

2. (a) Not possible. The sum of the vertex degrees in a graph must be *even*, because it is twice the number of edges. Here it is odd.

(b) A simple connected graph with the given degree sequence is shown below.

(c) Not possible. Since there are only 4 vertices, no vertex can have more than 3 neighbours. Hence the maximum vertex degree in a simple graph with four vertices is 3.

(d) There are two possibilities:



3. The sum of all the vertex degrees is even. The sum of the even degrees is even and hence the sum of the odd degrees must also be even. But this is only possible if we have an even number of vertices of odd degree. (Note: the solution to this problem depends on the following facts: *the sum of two even integers is even*, *the sum of two odd integers is even*, but *the sum of an even integer and an odd integer is odd*.)

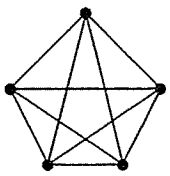
4. (a) The vertices represent the 8 sites; two vertices are adjacent just when the sites they represent have a direct connection between them.

(b) The degrees of the vertices (or the degree sequence).

(c) The number of pairs of sites which have a direct connection between them is the number of edges in the graph. This is half the sum of the degrees of the vertices. Hence number of connections $=(5+3+2+7+1+2+6+4)/2 = 15$.

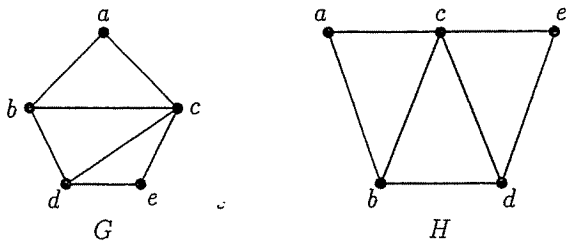
(d) The sum of the vertex degrees would be $9 \times 5 = 45$, which is odd (see question 3).

5. (a) Each vertex of K_6 has degree 5 (i.e. K_6 is 5-regular); sum of vertex degrees is therefore $6 \times 5 = 30$. Hence the number of edges is $= 30/2 = 15$.



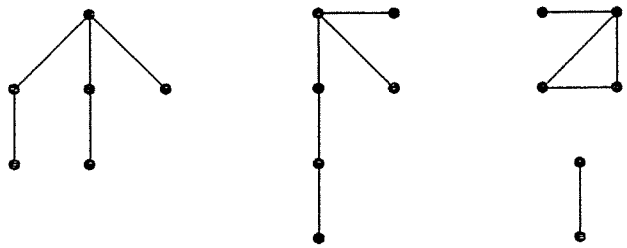
(b) $n-1$; the sum of vertex degrees is $n(n-1)$. Hence number of edges $= n(n-1)/2$.

6. The graphs G and H are isomorphic. A labelling of corresponding vertices is shown below. An alternative labelling is achieved by interchanging a with e and b with d.



7. You should have drawn any two of the graphs G_1, G_2, G_3 shown below. G_3 is not isomorphich to G_1 or G_2 because it has two components, whereas G_1 and G_2 are connected. To explain why G_1 and G_2 are not isomophic, you should use adjacency properties. For example, the vertex of

degree 3 is adjacent to only one vertex of degree 1 in G_1 and to two vertices of degree 1 in G_2, or you could observe that the two vertices of degree 2 are adjacent in G_2 but not in G_1.



G_1 G_2 G_3

8.

	v_1	v_2	v_3	v_4	v_5
v_1	0	2	0	1	1
v_2	2	0	2	0	1
v_3	0	2	0	0	1
v_4	1	0	0	0	0
v_5	1	1	1	0	0

(a) It gives the degree of the corresponding vertex;

(b) it gives the sum of all the vertex degrees (or twice the number of edges);

(c) these rules do not apply if the graph has a loop, because a loop at a vertex v contributes 2 to the degree of v, while it contributes only 1 to the sum of the elements in the row of the matrix corresponding to v (see Example 5.16).

Appendix C

Specimen Examination Questions

Note that you will be expected to be able to answer questions on all the material covered in the two Volumes of the Subject Guide. The following specimen questions are on the topics covered in this volume of the Subject Guide to give an idea of the standard expected.

Question 1

(a) Calculate the value of $(1010)_2 \times (1010)_2$ in the binary system, showing your working. [2]

(b) Express the hexadecimal number $(A5F)_{16}$ (i) as a binary number; (ii) as a decimal number. [3]

(c) (i) What is a *rational* number? [1]

(ii) Give one example of an *irrational* number. [1]

(iii) Express the recurring decimal $0.84545\ldots$ in the form a/b where a, b are positive integers. [3]

Question 2

(a) Let S denote the set of odd integers and X denote the set of integers greater than 17. Express the *set of even integers less than 18* in terms of the sets S and X using the set operations

(i) complementation and intersection; [1]

(ii) complementation and union. [1]

(b) Let $X = \{a, b\}$. Describe the *power set* of X by the listing method. [2]

(c) Let A, B, C be three subsets of the universal set $\mathcal{U}$.

(i) Draw Venn diagrams to illustrate each of the sets $X = A - (B \cup C)$ and $Y = (A - B) \cap (A - C)$. [3]

(ii) Decide which, if any, of the following statements are true for all subsets A, B, C:

$$X \subset Y, \quad X = Y, \quad Y \subset X.$$

[1]

(iii) Taking $\mathcal{U} = \{a, b, c\}$, construct a counter-example to show that $A - (B \cap C) \neq (A - B) \cap (A - C)$ for all sets A, B, C. [2]

Question 3

(a) Let p, q be propositions. Construct truth tables for the compound statements

$$\text{(i) } p \wedge q; \quad \text{(ii) } p \vee q; \quad \text{(iii) } p \rightarrow q.$$

[3]

(b) Suppose that $n = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ and let p, q denote the following propositions concerning n.

$$p\text{: } n \text{ is odd}; \qquad q\text{: } n > 5.$$

Find the truth set for each of the following compound statements.

$$\text{(i) } \neg q; \quad \text{(ii) } p \wedge q; \quad \text{(iii) } \neg p \vee q; \quad \text{(iv) } p \to q; \quad \text{(v) } p \leftrightarrow q.$$

[5]

(c) Is any pair of statements (i)–(v) in part (b) logically equivalent? Justify your answer. [2]

Question 4

(a) Consider the function

$$f : \mathbb{Z} \to \mathbb{Z} \text{ defined by } f(r) = r^2 + 1.$$

(i) Find the image of -3 under f. [1]

(ii) Find the set of preimages of 5. [1]

(iii) Decide, giving reasons whether the function f is onto. [2]

(iv) Decide, giving reasons whether the function f is one-to-one. [2]

(b) (i) The function $f : \mathbb{R} \to \mathbb{R}$ is defined by $f(x) = 5x^2 - 3x + 31$. Prove that $f(x)$ is $O(x^2)$. [3]

(ii) Is it true to say that $f(x) = O(x)$? [1]

Question 5

(a) (i) Given any real number x, say what is meant by $\lfloor x \rfloor$. [1]

(ii) Give the value of $\lfloor 13/5 \rfloor$, $\lfloor -7/3 \rfloor$. [2]

(iii) Suppose that $\lfloor x \rfloor = n$, where n is an integer. Express in terms of n, the values of $\lfloor x - 5 \rfloor$ and $\lfloor -x \rfloor$. [2]

(iv) Find examples of real numbers x, y for which $\lfloor x - y \rfloor \neq \lfloor x \rfloor - \lfloor y \rfloor$. [2]

(b) Define the inverse function for each of the following functions

(i) $f : \mathbb{R} \to \mathbb{R}$, where $f(x) = 4x - 1$;

(ii) $g : \mathbb{R} \to \mathbb{R}^+$, where $g(x) = 2^x$,

being careful to specify the domain and codomain in each case. [3]

Question 6

(a) Let G be a graph. Prove that the sum of the degrees of the vertices of G is equal to twice the number of edges. [2]

(b) Calculate the number of edges of the complete graph K_7 with 7 vertices. [2]

(c) Why is it impossible to construct a 3-regular graph with 9 vertices? [2]

(d) Draw two different (that is, non-isomorphic) *connected* graphs, each having the degree sequence

$$3, 3, 2, 2, 1, 1.$$

Give one reason why the graphs you have drawn are not isomorphic. [4]

Appendix D

Solutions to Specimen Examination Questions

Question 1

(a)

$$\begin{array}{r} 1010 \\ \times 1010 \\ \hline 10100 \\ 1010000 \\ \hline 1100100 \\ \hline {\scriptstyle 1} \end{array}$$

Answer: 1100100

(b) $(10100101111)_{16}$; 2655.

(c) (i) A rational number is one that can be expressed as a fraction m/n (or ratio) of two integers m, n.

(ii) $\sqrt{2}$ (or π etc).

(iii)

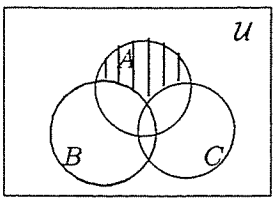
$$\begin{aligned} x &= 0.84545\ldots \\ 100x &= 84.545\ldots \\ 100x - x &= 83.7 \\ 99x &= 83.7 \\ x &= 83.7/99 \\ x &= 837/990 = 93/110. \end{aligned}$$

Question 2

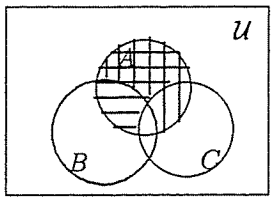
(a) (i) $S' \cap X'$; (ii) $(S \cup X)'$.

(b) $\mathcal{P}(X) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$.

(c) (i) In the Venn diagrams below, X is shaded and Y is cross-hatched.



X is shaded region



Y is cross-hatched region

(ii) $X = Y$ is true.

(iii) Several solutions are possible, for example:

$$A = \{a, b, c\}, B = \{b\}, C = \{c\},$$

giving $A - (B \cap C) = \{a, b, c\}$. But $A - B = \{a, c\}$, $A - C = \{a, b\}$, and hence $(A - B) \cap (A - C) = \{a\}$.

Question 3

(a)

		(i)	(ii)	(iii)
p	q	$p \wedge q$	$p \vee q$	$p \rightarrow q$
f	f	f	f	t
f	t	f	t	t
t	f	f	t	f
t	t	t	t	t

(b) (i) $\{1, 2, 3, 4, 5\}$; (ii) $\{7, 9\}$; (iii) $\{2, 4, 6, 7, 8, 9\}$; (iv) $\{2, 4, 6, 7, 8, 9\}$; (v) $\{2, 4, 7, 9\}$.

(c) (iii) and (iv) are logically equivalent, because they have the same truth sets.

Question 4

(a) (i) $f(-3) = 10$;

(ii) $\{-2, 2\}$;

(iii) f is not onto because e.g. 0 is in the co-domain, but 0 is not in the range.

(iv) f is not one-to-one because e.g. the element 5 in the co-domain has two different pre-images.

(b) (i)

$$\begin{aligned} f(x) &= 5x^2 - 3x + 31 \\ |f(x)| &\leq |5x^2| + |3x| + |31|. \end{aligned}$$

Hence, when $x > 1$, $|f(x)| < |5x^2| + |3x^2| + |31x^2| = |39x^2|$, and so $f(x)$ is $O(x^2)$.

(ii) It is not true to say that $f(x) = O(x)$.

Question 5

(a) (i) $\lfloor x \rfloor$ is the integer n such that $n \leq x < (n + 1)$.

(ii) $\lfloor 13/5 \rfloor = 2$; $\lfloor -7/3 \rfloor = -3$.

(iii) $\lfloor x - 5 \rfloor = n - 5$; $\lfloor -x \rfloor = -n - 1$.

(iv) For example, let $x = 5$ and $y = 0.9$. Then $\lfloor x - y \rfloor = \lfloor 4.1 \rfloor = 4$; but $\lfloor 5 \rfloor - \lfloor 0.9 \rfloor = 5 - 0 = 5$.

(b) (i) $f^{-1}(x) = \frac{x+1}{4}$; where $f^{-1} : \mathbb{R} \to \mathbb{R}$.
(ii) $g^{-1}(x) = \log_2 x$, where $g^{-1} : \mathbb{R}^+ \to \mathbb{R}$.

Question 6

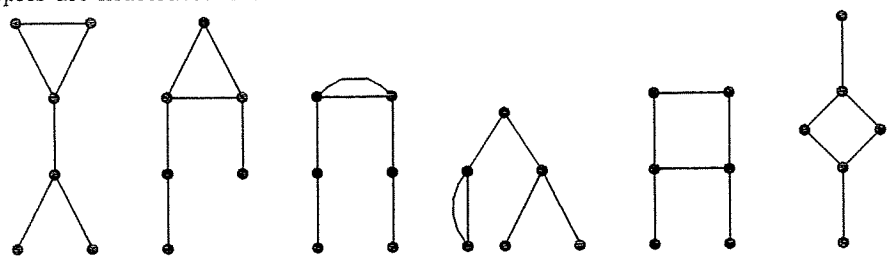
(a) Each edge has two ends. Each end contributes 1 to the sum of the degrees of the vertices. Therefore, each edge contributes 2 to the sum of the degrees of the vertices, and hence this sum is twice the number of edges.

(b) (i) In a 3-regular graph, each vertex has degree 3. The sum of the degrees of a 3-regular graph with 8 vertices is 3×8. So the number of edges is $(3 \times 8)/2 = 12$

(ii) K_7 has 7 vertices, each of degree 6. Hence the number of edges in K_7 is $(6 \times 7)/2 = 21$.

(c) The number of edges in such a graph would be $(9 \times 3)/2 = 13.5$, but it is impossible to have half an edge.

(d) There are a number of non-isomorphic connected graphs with this degree sequence: six examples are illustrated below.



The following are examples of the kind of argument that could be used to show that the two graphs you constructed are not isomorphic:

(a) *Cycles of different lengths*: e.g. (a) has a cycle of length 3 and hence cannot be isomorphic to any of (c), (d), (e) or (f), which do not contain such a cycle.

(b) *Parallel edges*: e.g. (c) has a pair of parallel edges and so cannot be isomorphic to any of (a), (b), (e) or (f), which are simple graphs.

(c) *Vertex adjacency*: e.g. In (a), both vertices of degree 1 are adjacent to the *same* vertex, while in (b) they are adjacent to *different* vertices; or: in (e) the two vertices of degree 3 are adjacent to one another, but in (f) they are not adjacent to one another (the same comment proves a difference between (c) and (d)).

(d) *Path length*: e.g. the longest path in (a) has length 4, but the longest path in (b) has length 5.

Appendix E

Glossary of Symbols

Number Systems

$(n)_5$	number n written in base 5	[Section 1.1.1]
$(n)_2$	number n written in binary	[Section 1.1.2]
$(n)_{16}$	number n written in hexadecimal	[Section 1.1.4]
$x \neq y$	x is not equal to y	[Section 1.3.2]
$x > y$	x is greater than y	[Section 1.3.2]
$x \geq y$	x is greater than or equal to y	[Section 1.3.2]
$x < y$	x is less than y	[Section 1.3.2]
$x \leq y$	x is less than or equal to y	[Section 1.3.2]

Sets

$\{a, b, c\}$	set with elements a, b, c	[Section 2.1.1]
$y \in X$	y is an element of X	[Section 2.1.1]
$y \notin X$	y is not an element of X	[Section 2.1.1]
$\mathbb{Z}$	set of integers	[Section 2.1.1]
$\mathbb{Z}^+$	set of positive integers	[Section 2.1.1]
$\mathbb{Q}$	set of rational numbers	[Section 2.1.1]
$\mathbb{R}$	set of real numbers	[Section 2.1.1]
$\{x : x \text{ has property } p\}$	set of elements with property p	[Section 2.1.2]
$\emptyset$	empty (or null) set	[Section 2.1.2]
$A \subseteq B$	A is a subset of B	[Section 2.2.1]
$A \subset B$	A is a proper subset of B	[Section 2.2.1]
$\mathcal{P}(S)$	power set of S	[Section 2.2.3]
A'	complement of set A	[Section 2.3.1]
$A \cup B$	A union B	[Section 2.3.2]
$A \cap B$	A intersection B	[Section 2.3.2]
$A - B$	set difference of A minus B	[Section 2.3.2]
$A \oplus B$	symmetric difference of A and B	[Section 2.3.2]

Logic

$\neg p$	not p	[Section 3.1.2]
$p \wedge q$	p and q	[Section 3.1.2]
$p \vee q$	p or q	[Section 3.1.2]
$p \oplus q$	p or q but not both	[Section 3.1.2]
$p \rightarrow q$	p implies q	[Section 3.2]
$p \leftrightarrow q$	p if and only if q	[Section 3.2]
	NOT-gate	[Section 3.4]
	AND-gate	[Section 3.4]
	OR-gate	[Section 3.4]

Functions

$f : X \rightarrow Y$	f is a function from X into Y	[Section 4.1]
$f(x)$	image of x under f	[Section 4.1]
$(x_1, x_2, \ldots, x_n)$	ordered n-tuple	[Section 4.1.2]
$f(p, q)$	image of ordered pair (p, q) under f	[Section 4.1.2]
$\lvert x \rvert$	absolute value of x	[Section 4.1.3]
$\lfloor x \rfloor$	floor of x	[Section 4.1.4]
$\lceil x \rceil$	ceiling of x	[Section 4.1.4]
$a_n x^n + \cdots + a_1 x + a_0$	polynomial of degree n	[Section 4.1.5]
f^{-1}	inverse function of f	[Section 4.2.4]
exp_b	exponential function with base b	[Section 4.3]
$b^{\frac{1}{q}}$	the qth root of b	[Section 4.3.1]
$b^{\frac{p}{q}}$	the qth root of b^p	[Section 4.3.1]
$\log_b x$	logarithm of x to the base b	[Section 4.3.2]
$O(f(x))$	order of $f(x)$	[Section 4.4.1]

Graph Theory

$V(G)$	vertex set of graph G	[Section 5.1.1]
$E(G)$	edge set of graph G	[Section 5.1.1]
$\deg(v)$	degree of vertex v	[Section 5.1.2]
K_n	complete graph with n vertices	[Section 5.1.3]
$u_1 u_2 \ldots u_r$	path through distinct vertices $u_1, \ldots, u_r$	[Section 5.2.1]
k-cycle	cycle of length k	[Section 5.2.2]
$\mathbb{A}(G)$	adjacency matrix of graph G	[Section 5.4.1]

Notes

Notes

Comment form

We welcome any comments you may have on the materials which are sent to you as part of your study pack. Such feedback from students helps us in our effort to improve the materials produced for the University of London.
If you have any comments about this guide, either general or specific (including corrections, non-availability of Essential readings, etc.), please take the time to complete and return this form.

Title of this subject guide: .....

.....

Name .....

Address .....

.....

Email .....

Student number .....

For which qualification are you studying? .....

Comments

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Please continue on additional sheets if necessary.

Date: .....

Please send your completed form (or a photocopy of it) to:
Publishing Manager, Publications Office, University of London, Stewart House, 32 Russell Square, London WC1B 5DN, UK.