

# 1

## *Coding and its uses*

### 1.1 Messages

The first task is to set up a simple mathematical model of a message. We do this by looking at some examples and extracting some common features from them.

#### Example 1.1

Many messages are written in a natural language, such as English. These messages contain symbols, and the symbols form words, which in turn form sentences, such as this one. The messages may be sent from one person to another in several ways: in the form of a handwritten note or an email, for example. A text message is essentially the same, but it is often expressed in an unnatural language.

#### Example 1.2

Devices such as scanners and digital cameras produce messages in the form of electronic impulses. These messages may be sent from one device to another by wires or optic fibres, or by radio waves.

Formal definitions based on these examples will be given in Section 1.3. For the time being, we shall think of a message as a sequence of symbols, noting

that the order of the symbols is clearly important.

The function of a message is to convey information from a sender to a receiver. In order to do this successfully, the sender and receiver must agree to use the same set of symbols. This set is called an *alphabet*.

### Example 1.3

We denote by  $\mathbb{A}$  the alphabet which has 27 symbols, the letters A, B, C, ..., Z, and a ‘space’, which we denote by  $\sqcup$ . We shall often use the alphabet  $\mathbb{A}$  to represent messages written in English. This is convenient for the sake of exposition, but obviously some features are ignored. Thus we ignore the distinction between upper and lower case letters, and we omit punctuation marks. Of course, there may be some loss in reducing an English message into a string of symbols in this alphabet. For example the text

The word ‘hopefully’ is often misused.

is reduced to the following message in  $\mathbb{A}$ .

THE $\sqcup$ WORD $\sqcup$ HOPEFULLY $\sqcup$ IS $\sqcup$ OFTEN $\sqcup$ MISUSED

### Example 1.4

The alphabet  $\mathbb{B}$  has 2 symbols, 0 and 1, which are called *binary digits* or *bits*. Because the bits 0 and 1 can be implemented electronically as the states OFF and ON, this is the underlying alphabet for all modern applications. In practice, the bits are often combined into larger groups, such as ‘32-bit words’. But any message that is transmitted electronically, whether it originates as an email from me or as an image from a satellite orbiting the earth, is essentially a sequence of bits.

## EXERCISES

- 1.1. The following messages have been translated from ‘proper English’ into the alphabet  $\mathbb{A}$ . Write down the original messages and comment upon any ambiguity or loss of meaning that has occurred.

CANINE $\sqcup$ HAS $\sqcup$ SIX $\sqcup$ LETTERS $\sqcup$ AND $\sqcup$ ENDS $\sqcup$ IN $\sqcup$ NINE  
ITS $\sqcup$ HOT $\sqcup$ SAID $\sqcup$ ROBERT $\sqcup$ BROWNING

- 1.2. A 32-bit word is a sequence of 32 symbols from the alphabet  $\mathbb{B}$ . How many different 32-bit words are there? If my printer can print one every second, how many years (approximately) will it take to print them all?

- 1.3. In the period 1967-86 the ASCII alphabet was widely used as a standard for electronic communication. It has 128 symbols, 95 of which were printable. In this book we have already used some symbols that were not in the ASCII alphabet. Which ones? [ASCII is an abbreviation for *American Standard Code for Information Interchange*, and is pronounced ‘askey’. The ASCII alphabet is now part of a much more comprehensive system known as Unicode.]
- 1.4. Not all natural languages use 26 letters. How many letters are there in (i) the modern Greek alphabet and (ii) the Russian Cyrillic alphabet?

## 1.2 Coding

Roughly speaking, coding is a rule for replacing one message by another message. The second message may or may not use the same alphabet as the first.

### Example 1.5

A simple rule for coding messages in the 27-symbol alphabet  $\mathbb{A}$  using the same alphabet is: *write each word backwards*. So the message

SEE□YOU□TOMORROW      becomes      EES□UOY□WORROMOT      .

### Example 1.6

A rule for coding messages in  $\mathbb{A}$  using the binary alphabet  $\mathbb{B}$  is: *replace vowels by 0, replace consonants by 1, and ignore the spaces*. With this rule

SEE□YOU□TOMORROW      becomes      10010010101101      .

These two examples are very artificial, and the rules are of limited value. For greater realism and utility we must look at the purposes for which coding is used, and evaluate proposed coding rules in that context.

There are three major reasons for coding a message.

**ECONOMY**    In many situations it is necessary or desirable to use an alphabet smaller than those that occur in natural languages. It may also be desirable to make the message itself smaller: in recent times this has led to the development of techniques for *Data Compression*.

**RELIABILITY** Messages may be altered by ‘noise’ in the process of transmission. Thus there is a need for codes that allow for *Error Correction*.

**SECURITY** Some messages are sent with the requirement that only the right person can understand them. Historically, secrecy was needed mainly in diplomatic and military communications, but nowadays it plays an important part in everyday commercial transactions. This area of coding is known as *Cryptography*.

## EXERCISES

- 1.5. The following messages are coded versions of meaningful English sentences. Explain the coding rules used and find the original messages.

7 15 15 4 27 12 21 3 11

00111 01111 01111 00100 11011 01100 10101 00011 01011

- 1.6. Explain formally (as if you were writing a computer program) the coding rule *write each word backwards*. [You must explain how to convert a sequence of symbols such as TODAY␣IS␣MONDAY into YADOT␣SI␣YADNOM.]

## 1.3 Basic definitions

We are now ready to make some proper definitions.

### Definition 1.7 (Alphabet)

An *alphabet* is a finite set  $S$ ; we shall refer to the members of  $S$  as *symbols*.

### Definition 1.8 (Message, string, word)

A *message* in the alphabet  $S$  is a finite sequence of members of  $S$ :

$$x_1 x_2 \cdots x_n \quad (x_i \in S, 1 \leq i \leq n).$$

A message is often referred to as a *string* of members of  $S$ , or a *word* in  $S$ . The number  $n$  is called the *length* of the message, string, or word.

The set of all strings of length  $n$  is denoted by  $S^n$ . For example, when  $S = \mathbb{B}$  and  $n = 3$ , the set  $\mathbb{B}^3$  consists of the strings

$$000 \quad 001 \quad 010 \quad 011 \quad 100 \quad 101 \quad 110 \quad 111 \quad .$$

The set of all strings in  $S$  is denoted by  $S^*$ :

$$S^* = S^0 \cup S^1 \cup S^2 \cup \dots \quad .$$

Note that  $S^0$  consists of the string with length zero; in other words, the string with no symbols. We include it in the definition because sometimes it is convenient to use it.

### Definition 1.9 (Code, codeword)

Let  $S$  and  $T$  be alphabets. A *code*  $c$  for  $S$  using  $T$  is an injective function  $c : S \rightarrow T^*$ . For each symbol  $s \in S$  the string  $c(s) \in T^*$  is called the *codeword* for  $s$ . The set of all codewords,

$$C = \{c(s) \mid s \in S\},$$

is also referred to as the code. When  $|T| = 2$  the code is said to be *binary*, when  $|T| = 3$  it is *ternary*, and in general when  $|T| = b$ , it is *b-ary*.

For example, let  $S = \{x, y, z\}$ ,  $T = \mathbb{B}$ , and define

$$c(x) = 0, \quad c(y) = 10, \quad c(z) = 11.$$

This is a binary code, and the set of codewords is  $C = \{0, 10, 11\}$ .

According to the definition, a code  $c$  assigns to each *symbol* in  $S$  a *string of symbols* in  $T$ . The strings may vary in length. For example, suppose we are trying to construct a code for the 27-symbol English alphabet  $\mathbb{A}$  using the binary alphabet  $\mathbb{B}$ . We might begin by choosing codewords of length 4, as follows:

$$\mathbf{A} \mapsto 0000 \quad \mathbf{B} \mapsto 0001 \quad \mathbf{C} \mapsto 0010 \quad \dots \quad .$$

Now, the definition requires  $c$  to be an injective function or (as we usually say) an *injection*. This is the mathematical form of the very reasonable requirement that  $c$  does not assign the same codeword to two different symbols. In other words, if  $c(s) = c(s')$  then  $s = s'$ . Clearly, there are only 16 strings of length 4 in  $\mathbb{B}$ , so the 27 symbols in  $\mathbb{A}$  cannot all be assigned different ones.

Thus far we have considered only the coding of individual symbols. The extension to messages (strings of symbols) is clear.

### Definition 1.10 (Concatenation)

If  $c : S \rightarrow T^*$  is a code, we extend  $c$  to  $S^*$  as follows. Given a string  $x_1x_2 \cdots x_n$  in  $S^*$ , define

$$c(x_1x_2 \cdots x_n) = c(x_1)c(x_2) \cdots c(x_n).$$

This process is known as *concatenation*. Note that we denote the extended function  $S^* \rightarrow T^*$  by the same letter  $c$ .

It is not always possible to recover the original string uniquely from the coded version. For example, let  $S = \{x, y, z\}$ , and define  $c : S \rightarrow \mathbb{B}^*$  by

$$x \mapsto 0, \quad y \mapsto 01, \quad z \mapsto 10.$$

Suppose we are given the string 010100 which, we are told, is the result of coding a string in  $S^*$  using  $c$ . By trial and error we find two possibilities (at least):

$$xzzx \mapsto 010100, \quad yyxx \mapsto 010100.$$

Clearly, this situation is to be avoided, if possible.

### Definition 1.11 (Uniquely decodable)

The code  $c : S \rightarrow T^*$  is *uniquely decodable* (or *UD* for short) if the extended function  $c : S^* \rightarrow T^*$  is an injection. This means that any string in  $T^*$  corresponds to at most one message in  $S^*$ .

In Chapter 2 we shall explain how the UD property can be guaranteed by a simple construction.

## EXERCISES

1.7. A binary code is defined by the rule

$$s_1 \mapsto 10, \quad s_2 \mapsto 010, \quad s_3 \mapsto 100.$$

Show by means of an example that this code is not uniquely decodable.

- 1.8. Suppose the code  $c : S \rightarrow T^*$  is such that every codeword  $c(s)$  has the same length  $n$ . Is this code uniquely decodable?
- 1.9. Express the coding rules used in Exercise 1.5 as functions  $c : S \rightarrow T^*$ , for suitable alphabets  $S$  and  $T$ .

## 1.4 Coding for economy

When the electric telegraph was first introduced, it could transmit only simple electrical impulses. Thus, in order to send messages in a natural language it was necessary to code them into an alphabet with very few symbols. A suitable code was invented by Samuel Morse (1791-1872).

The *Morse Code* uses an alphabet of three symbols:  $\{\bullet, -, \odot\}$ . The  $\bullet$  (*dot*, pronounced *di*) is a short impulse, the  $-$  (*dash*, pronounced *dah*) is a long impulse, and the  $\odot$  is a pause. Every codeword comprises dots and dashes, ending with a pause. (Strictly speaking, there are also symbols for the shorter pause that separates the dots and dashes within a codeword, and for the longer pause at the end of a message word, but we shall ignore them for the sake of simplicity.) Here are the codewords for  $A, B, C, D, E, F, X, Y, Z$ .

$$\begin{array}{lll} A \mapsto \bullet - \odot & B \mapsto - \bullet \bullet \bullet \odot & C \mapsto - \bullet - \bullet \odot \\ D \mapsto - \bullet \bullet \odot & E \mapsto \bullet \odot & F \mapsto \bullet \bullet - \bullet \odot \\ X \mapsto - - \bullet \bullet - \odot & Y \mapsto - \bullet - - \odot & Z \mapsto - - \bullet \bullet \odot \end{array}$$

In Chapters 2, 3, and 4 we shall look at the basic theory of economical coding and explain how it can be applied to the compression of data. This subject has become very important, because huge amounts of data are now being generated and transmitted electronically.

### EXERCISES

- 1.10. Search the internet to find the standard version of Morse Code, known as the International Morse Code. If this code is defined formally as a function  $S \rightarrow T^*$ , what are the alphabets  $S$  and  $T$ ?
- 1.11. Decode the following Morse messages:

$$\begin{array}{l} \bullet \bullet \bullet \odot - - - \odot \bullet \bullet \bullet \odot \quad ; \\ - - \odot \bullet - \odot - \bullet - - \odot - \bullet \bullet \odot \bullet - \odot - \bullet - - \odot \quad . \end{array}$$

- 1.12. Suppose we try to use a version of Morse code without the symbol  $\odot$  that indicates the end of each codeword. What is the code for *BAD*? Find another English word with the same code, showing that this is not a uniquely decodable code.

- 1.13. The *semaphore* code enables messages to be exchanged between people who can see each other. Each person has two flags, each of which can be displayed in one of eight possible positions. The two flags cannot occupy the same position. How many symbols can be encoded in this way, remembering that the coding function must be an injection?

## 1.5 Coding for reliability

It is frequently necessary to send messages through unreliable channels, and in such circumstances we should like to use a method of coding that will reduce the likelihood of a mistake. An obvious technique is simply to repeat the message.

For example, suppose an investor communicates with a broker by sending the symbols  $B$  and  $S$  ( $B = \text{Buy}$  and  $S = \text{Sell}$ ). With this code, if any symbol is received incorrectly, the broker will make a mistake, and perform the wrong action.

However, suppose the investor uses the code  $\text{Buy} \mapsto BB$  and  $\text{Sell} \mapsto SS$ . Now if any one symbol is received incorrectly the broker will know that something is wrong, because  $BS$  and  $SB$  are not codewords, and will be able to ask for further instructions.

If the investor uses more repetitions the broker may be able to make a reasonable decision about the intention, even when it is not possible to ask for further instructions. Suppose the investor uses the codewords  $BBB$  and  $SSS$ . Then, if  $SSB$  is received, it is more likely that the message was  $SSS$ , because that would imply that only one error had occurred, whereas  $BBB$  would imply that two errors had occurred.

In Chapters 6-9 we shall describe more efficient methods of coding messages so that the probability of a mistake due to errors in transmission is reduced.

### EXERCISES

- 1.14. Suppose an investor uses the 5-fold repetition code, that is,  $\text{Buy} \mapsto BBBBB$ ,  $\text{Sell} \mapsto SSSSS$ . If the following messages are received, which instruction is more likely to have been sent in each case?

$BBBSB \quad SBSBS \quad SSSSB$

- 1.15. Suppose we wish to send the numbers 1, 2, 3, 4, 5, 6, representing the outcomes of a throw of a die, using binary codewords, all of the



same length. What is the smallest possible length of the codewords? Suppose it is required that the receiver will notice whenever one bit in any codeword is in error. Find a set of codewords with length four which has this property.

## 1.6 Coding for security

One of the oldest codes is said to have been used by Julius Caesar over two thousand years ago, with the intention of communicating secretly with his army commanders. For a message in the 27-symbol alphabet  $\mathbb{A}$ , the rule is:

*choose a number  $k$  between 1 and 25 and replace each letter by the one that is  $k$  places later, in alphabetical order.*

The rule is extended in an obvious way to the letters at the end of the alphabet, as in the example below. The space  $\square$  is not changed. Thus if  $k = 5$  the symbols are replaced according to the rule:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	$\square$	
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	$\square$	.

For example, the message

SEE $\square$ YOU $\square$ TOMORROW becomes XJJ $\square$ DTZ $\square$ YTRTWWTB .

In mathematical terms the coding rule is a function  $c_k : \mathbb{A} \rightarrow \mathbb{A}$ , which depends on the *key*  $k$ : in the example given above,  $k = 5$ . It is a basic assumption of cryptography that, although the value of  $k$  may be kept secret, the general form of the coding rule cannot. In other words, it will become known that the rule is  $c_k$  (apply a shift of  $k$  to the letters) for some  $k$ .

When a coded message such as

XJJ $\square$ DTZ $\square$ YTRTWWTB

is sent, it is presumed that the intended recipient knows the key – the value  $k = 5$  in our example. In that case it is easy to decode the message. On the other hand, if someone who does not know the key intercepts the message, decoding is not necessarily so easy. In cryptography, decoding by finding the value of the key  $k$  (or otherwise) is said to be *breaking* the system, and any method which may achieve this is an *attack*.

In fact, Caesar's system is not very secure, because there is a simple attack by the method known as *exhaustive search*. The only possible values of  $k$  are

1, 2, 3, ..., 25, and it is easy to try each of them in turn, until a meaningful message is found.

### Example 1.12

Suppose we have intercepted the message

SGZNY□OY□MUUJ□LUX□EUA .

We suspect that Caesar's system is being used. How do we find the key?

*Solution* Trying the possible keys, beginning with  $k = 1$  and  $k = 2$  produces the following possibilities. Remember that if the key is  $k$ , we must go *back*  $k$  places to find the original message.

$k = 1$  :      RFYMX□NX□    ...  
 $k = 2$  :      QEXLW□MW□    ...

Thus the key is not 1 or 2, because if it were, the original message would not make sense. There is no need to 'decode' the whole message in order to establish this fact. So we must continue to work through the keys  $k = 3, 4, \dots, 25$ , until a meaningful message is found.

## EXERCISES

- 1.16. Find the original message in Example 1.12.
- 1.17. Could the following message have been sent by Julius Caesar himself?

ZLJB□LK□BKDIXKA

- 1.18. Caesar's system is an example of a *substitution* code, because each letter in the message is replaced by a substitute letter, according to a fixed rule. Suggest other substitution rules, with a view to defending against the attack by exhaustive search.

## Further reading for Chapter 1

The internet is a treasury of information about Morse code, semaphore, and other historically important coding systems. The pioneering work of Claude Shannon on the theory of information and communication is also well-represented.

Internet sites relating to cryptography are very variable in quality, and it is better to rely on good books such as those by Kahn [1.2] and Singh [1.3]. Older books on cryptography can also provide an important perspective for understanding the modern approach. The books by d’Agapeyeff [1.1] and Sacco [1.4] are recommended.

Books about the so-called ‘Bible Codes’ and similar matters should be regarded as entertainment. They are more entertaining (often unintentionally) when considered from the viewpoint of an informed reader, such as someone who has studied this book.

- 1.1 A. d’Agapeyeff. *Codes and Ciphers*. Oxford University Press, London (1939).
- 1.2 D. Kahn. *The Codebreakers*. Scribner, New York (1996).
- 1.3 S. Singh. *The Code Book*. Fourth Estate, London (2000).
- 1.4 L. Sacco. *Manuel de Cryptographie*. Payot, Paris (1951).

<http://www.springer.com/978-1-84800-272-2>

Codes: An Introduction to Information Communication  
and Cryptography

Biggs, N.L.

2008, X, 274 p. 36 illus., Softcover

ISBN: 978-1-84800-272-2