Mathematics for the Digital Systems Engineer: Essentials for Modern Cryptography, Computer Security and Communications Technology: *Supplementary Material*

Chris J. Mitchell

21st February 2026

Contents

List of Figures

List of Tables

What is this all about?

This document is intended as a supplement to the book *Mathematics for the Digital Systems Engineer: Essentials for Modern Cryptography, Computer Security and Communications Technology.* It includes a variety of material, including:

- things that I would have liked to include but which I felt were unnecessary to meet the main objectives of the book;
- clarifications of topics which, on reflection, I could have explained better;
- thoughts that have occurred to me since I finished the book, which could have been included had I thought of them in time;
- corrections of mistakes.

I hope you find it useful. If you spot any errors I have not yet noticed, or indeed think of things you feel I should have included, please do contact me at me@chrismitchell.net — I will do my best to acknowledge all contributions.

If Wiley and the IEEE ever decide they would like me to produce a second edition of this book, I hope to include some of this material.

Chris J Mitchell
Wiltshire
February 2026

Corrigenda

Please excuse the following errors in the book.

Printed book

7.4.3 LFSR sequences and polynomials

Halfway down page 161, I define the characteristic polynomial for an LFSR to be the polynomial:

$$a_0 + a_1 x + a + 2x^2 + \cdots + a_{n-1}x^{n-1} + x^n.$$

There are actually *two* errors here. First the full stop at the end is erroneous, and second, and much more importantly, it should be:

$$a_0 + a_1 x + a + 2x^2 + \cdots + a_{n-1}x^{n-1} - x^n,$$

i.e. it should end with '$-x^n$' and not '$+x^n$'. Of course, since $-1 = +1$ in the binary case, it makes no difference there, which is almost certainly why I didn't spot the error before.

e-book

Preface

In note 5, in two places '104' should be '10^4'.

Notes for Chapter 3

The clickable links in notes 5 and 9 don't work. In both cases the link should be to `https://en.wikipedia.org/wiki/Interesting_number_paradox`.

7.4.3 LFSR sequences and polynomials

Midway between Figure 7.6 and Theorem 7.5, I define the characteristic polynomial for an LFSR to be the polynomial:

$$a_0 + a_1 x + a + 2x^2 + \cdots + a_{n-1} x^{n-1} + x^n.$$

There are actually *two* errors here. First the full stop at the end is erroneous, and second, and much more importantly, it should be:

$$a_0 + a_1 x + a + 2x^2 + \cdots + a_{n-1} x^{n-1} - x^n,$$

i.e. it should end with '$-x^n$' and not '$+x^n$'. Of course, since $-1 = +1$ in the binary case, it makes no difference there, which is almost certainly why I didn't spot the error before.

Chapter 1

A Gentle Introduction

1.2 Mathematics as Mathematicians See It

The Four Colour Theorem

At the end of Section 1.2 I briefly mention the Four Colour Theorem (4CT). A very nice and highly accessible article on the history of this theorem, written by mathematician and historian of mathematics, Robin Wilson[1], was published in early 2026[2].

[1]For those of you interested in British politics, Robin Wilson is the son of the former British Prime Minister, Harold Wilson.

[2]R. Wilson, 'Four-Color Theorem 1852–1976', *Notices of the American Mathematical Society*, **73** (2926) 216-228; `https://doi.org/10.1090/noti3305`

Chapter 2

Sets, Functions and Relations

2.2 Sets

Russell's Paradox

In Section 2.2 I very briefly introduced Russell's Paradox. As I described there, the paradox involves formulating a set which is itself contradictory, namely the set S of all sets which are not members of themselves. I also stated that when you ask the question, 'Is this set a member of itself?', the problem becomes clear. However, I didn't explain the problem further. I thought it might be helpful to amplify the difficulties a little here.

So, is this set S a member of itself? If we suppose it is, then it is an example of a set which is a member of itself, and hence it cannot be a member of S. Thus, by contradiction, S cannot be a member of itself. But since S is the set of all sets which are not members of themselves, then S *is* a member of itself, yielding a further contradiction.

A recasting of the paradox, known as the *Barber's Paradox*, was proposed as a more easily understood version. Although Russell pointed out that it is not quite the same as the original paradox, it nonetheless illustrates the idea. The Barber's Paradox involves considering a barber in a small town who shaves all the men who do not shave themselves, and no one else. The paradox arises when one considers who might shave the barber. If he shaves himself then, by the assumption, he cannot shave himself. Thus he cannot shave himself, and so he is someone who does not shave himself, and hence he does shave himself. Of course, this paradox is easier to resolve — perhaps he grows a beard!

2.8 Operations

Relations and operations are functions

In Section 2.4, I introduced the notion of a Relation, and pointed out that a relation R on a set S can be thought of as a subset of $S \times S$. Subsequently, in Section 2.8, I introduced Operations. I want to point out here that both relations and operations are actually special types of function.

If R is a relation defined on a set S, then $a\ R\ b$ for some subset of pairs $(a, b) \in S \times S$. This means that I can define a function $f_R : S \times S \to \{0, 1\}$ where, if $a, b \in S$, $f_R(a, b) = 1$ if and only if $a\ R\ b$. That is, binary relations on S can be thought of as functions from $S \times S$ to the set $\{0, 1\}$. It is not hard to transpose possible properties of a relation to the function setting; for example, a relation R is reflexive if and only if $f_R(a, a) = 1$ for every $a \in S$, and R is symmetric if and only if $f_R(a, b) = f_R(b, a)$ for every $a, b \in S$.

Analogously, if $*$ is an operation defined on a set S, then it is possible to define a function $f_* : S \times S \to S$, where $f_*(a, b) = a * b$ for every $a, b \in S$; equally any function mapping from $S \times S$ to S defines an operation. Again, the properties that an operation might hold can easily be transposed into the function setting; for example, e is an identity element for the operation $*$ if and only if $f_*(a, e) = f_*(e, a) = a$ for every $a \in S$.

So this raises the question of why I bother to introduce the notions of operations and relations if they are just examples of special types of function? The answer is simple. Relations and operations occur very frequently across mathematics, and it is very convenient to treat them as special classes of object. Apart from anything else, convincing the world to write $f_+(1, 1) = 2$ instead of the simpler $1 + 1 = 2$, is clearly a lost cause (and completely pointless).

Chapter 3

Numbers as We Know and Love Them

3.4 Ordering The Integers

Set containment is a partial ordering

In Section 3.4, I claimed without proof that the subset relation, $\subseteq$, when applied to the set of subsets of a set, forms a partially ordered set. To show this is true I need to establish reflexivity, antisymmetry and transitivity. Consider all subsets of a set S.

- *Reflexivity.* If $A \subseteq S$, then by definition $A \subseteq A$, and hence reflexivity holds.
- *Antisymmetry.* If $A \subseteq B$ and $B \subseteq A$, then by Lemma 2.1 it follows that $A = B$, and hence antisymmetry holds.
- *Transitivity.* Finally, suppose $A \subseteq B$ and $B \subseteq C$. Choose an arbitrary element $x \in A$. Then, since $A \subseteq B$, by definition it follows that $x \in B$; also, since $B \subseteq C$, by definition $x \in C$. Hence $A \subseteq C$ and transitivity follows.

Chapter 4

Modular arithmetic on the integers

4.5 Elementary properties of $\mathbb{Z}_m$

4.5.2 The Euler totient function

After the statement of Theorem 4.5 (Fermat's Little Theorem) I mentioned a very simple proof for this theorem due to Solomon Wolf Golomb, details of which are included in the 'Answers to questions for the reader' at the end of the book. Interestingly, Golomb is responsible for popularising a wide range of interesting mathematical ideas, including *Golomb Rulers* and *Polyominoes*, both of which have since received significant attention by mathematicians.

Golomb Rulers

A Golomb Ruler is a subset $\{a_1, a_2, \ldots, a_m\}$ of the non-negative integers with the property that no two differences $a_j - a_i$ $(j \neq i)$ are the same. Conventionally, a_0 is set to zero, and $a_i < a_{i+1}$ for all i $(1 \leq i < m)$; this makes sense since if the same value is added to each element of a Golomb Ruler it remains a Golomb Ruler, and the order of the elements is of no significance.

The number of integers in the ruler is called its *order*; for example, the set $\{0, 1, 3\}$ is a Golomb Ruler of order 3. A Golomb Ruler is in some sense optimal if a_m is as small as possible for a given value of m. Clearly $\{0, 1, 3\}$ is optimal — indeed it is *perfect* in the sense that every possible difference between 1 and 3 arises. Similarly, $\{0, 1, 4, 6\}$ is a perfect and optimal Golomb Ruler of order 4. However, as I next show, there are no

perfect Golomb Rulers with order greater than 4.

First observe that in any Golomb Ruler, $a_m \geq \frac{m(m-1)}{2}$ since there are $\frac{m(m-1)}{2}$ possible differences between distinct elements in a set of size m, and hence the maximum difference must be at least $\frac{m(m-1)}{2}$; also, in a perfect ruler $a_m = \frac{m(m-1)}{2}$. It follows that, since $1 + 2 + \cdots + (m-1) = \frac{m(m-1)}{2}$, this means that

$$\{a_{i+1} - a_i : 1 \leq i < m\} = \{1, 2, \ldots, m-1\}$$

in a perfect Golomb Ruler. Thus, if $m \geq 3$, $a_{i+2} - a_i > m$ for every i, and hence $\{a_2 - a_1, a_m - a_{m-1}\} = \{1, 2\}$. Without loss of generality assume $a_2 - a_1 = 1$ and $a_m - a_{m-1} = 2$. Then, since $a_{i+2} - a_i > m$, if $m > 4$ we must have $a_3 - a_2 = m$ and $a_{m-1} - a_{m-2} = m - 1$; but this means that $a_m - a_{m-2} = m + 1 = a_3 - a_1$, which contradicts the requirement that all possible differences are distinct. Hence $m \leq 4$.

There are two inequivalent optimal rules of order 5, namely $\{0, 1, 4, 9, 11\}$ and $\{0, 2, 7, 8, 11\}$, i.e. with largest element one more than would be the case for a perfect ruler. However, determining the optimal Golomb Ruler for a general n is an unsolved problem, with computer search having established optimal rulers for orders up to 28. This is another example of what seems like a very simple problem but yet which eludes a mathematical solution, at least to date[1].

Golomb Rulers were, in fact, independently studied by several mathematicians during the first half of the 20th century, and Golomb was not the first. Indeed they are also known as Sidon sets or Sidon sequences, after Hungarian mathematician Simon Sidon, who introduced the concept in 1932.

Polyominoes

Whilst polyominoes have been studied in one way or another for well over a hundred years, often in the context of popular puzzles, the name was given to them by Golomb in the early 1950s. Later, in 1965, Golomb published a book on the subject[2]. Whilst the study of polyominoes forms part of what is known as Recreational Mathematics, there do appear to be genuine applications.

A polyomino is simply a set of squares of a particular size arranged in a connected planar configuration where the configuration forms a part of a

[1] A survey of construction methods for Golomb Rulers can be found in: K. Drakakis, 'A Review Of The Available Construction Methods For Golomb Rulers'. *Advances in Mathematics of Communications* **3 (3)** (2009) 235–250, `https://doi.org/10.3934/amc.2009.3.235`.

[2] A revised version of the book is available as S. W. Golomb, *Polyominoes: Puzzles, Patterns, Problems, and Packings — Revised and Expanded Second Edition*, Princeton Science Library, 1994.

square grid. That is, all squares are aligned in the same direction, and are joined via a single touching side. Martin Gardner in the June 1960 edition of his wonderful Mathematical Games column in Scientific American, describes them as 'shapes that cover connected squares on a checkerboard'. The name polyomino is, of course, a back-derivation from domino.

There is just one domino, namely two squares in a line. There are either two or three triominoes (or 3-ominoes), depending on how you count them, namely three in a line and an L shape. There are two versions of the L shape, one a mirror image of the other, and they are the 'same' if you allow the shape to be 'turned over'. That is, there are two *free* triominoes, and three *one-sided* triominoes, one-sided meaning you cannot turn them over. The two free triominoes and the three one-sided triominoes are shown in Figure A4.1 (the free triominoes to the left and the one-sided triominoes to the right).

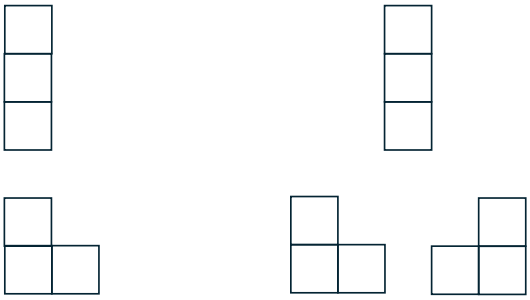

Figure A4.1: Triominoes

In a similar vein, there are five free tetrominoes (4-ominoes), and 7 one-sided tetrominoes. The five free tetrominoes are shown in Figure A4.2 — adding the mirror images of the third and fith tetrominoes gives the full set of seven one-sided tetrominoes.

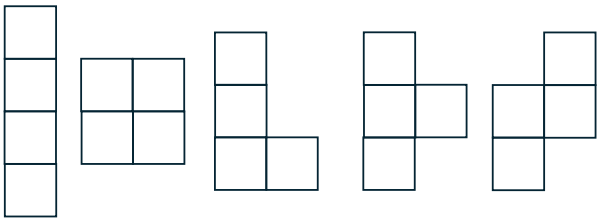

Figure A4.2: Tetrominoes

The five square case (pentominoes) is particularly well-discussed, probably because there enough different shapes to make them interesting, but not so many that working with them is difficult. There are, in fact, 12 free pentominoes and 18 one-sided pentominoes. Much work has been done

to arrange the set of 12 free pentominoes into appealing shapes, including 6×10, 5×12, 4×15 and 3×20 rectangles.

One key mathematical question is the number of free or one-sided polyominoes containing a certain number of squares. In general the number is not known — computer searches have been used to enumerate them for 50 component squares and beyond. Some upper and lower bounds are known; this is another example of a problem which is easy to state yet apparently very difficult to solve.

4.12 Other applications of modular arithmetic

1089 and a magic trick

In Section 4.11 I described how modular properties can be used as the basis of simple ways of testing numbers for divisibility. There are many other properties of this type that crop up in mathematical puzzles and curiosities. I want to describe one mathematical magic trick which nearly made it into the book, but I couldn't find a good excuse for including it. It all relates to the number 1089[3].

First the trick itself, which I remember being shown to me as a young boy — at the time I had no idea how this piece of magic works. The magician first writes down the number 1089 on a piece of paper which is folded over and given to the subject with the instruction not to look at it. The subject is then invited to choose a three-digit number with the proviso that the first and last digits must be different; the magician is not told the number and does not get to see any of the subsequent calculations. The subject is then asked to reverse the order of the digits of their number, and to take the difference between this reversed number and the original number (i.e. to subtract the smaller from the larger value). This new number will contain at most thee digits, and if it has only two the subject is asked to put a zero at the front to make it a three-digit number. To see how this works suppose the subject had chosen 123; its reverse is 321, and the difference between the larger and the smaller of the two values is $321 - 123 = 198$.

The subject is now invited to reverse the order of the digits in this new number, and to add the number and its reverse together. The magician now invites the subject to look at the piece of paper provided at the start and, lo and behold, 1089 is both written on the paper and is the value the subject just computed — so it appears that the magician is able to predict the future! To complete the example, the reverse of 198 is 891, and $198 + 891 = 1089$.

[3]The trick based on the number 1089 has even inspired an excellent and highly recommended book: D. Acheson, '1089 and all that: A journey into mathematics'. Oxford University Press, 2010.

So why does the trick always work? Well, the difference between a three-digit number and its reverse is always congruent to 0 modulo 99. This holds because if the three-digit number is abc, then this represents the value $100a+10b+c$. The reverse, cba, represents $100c+10b+a$, and the difference is either

$$(100a + 10b + c) - (100c + 10b + a) = 99(a - c)$$

or

$$(100c + 10b + a) - (100a + 10b + c) = 99(c - a)$$

depending on whether $a > c$ or $a < c$. This simple calculation also reveals why the first and last digits of the original number must be different, since if $a = c$ then the difference is 0 and the trick fails. Since we ensured that the difference is not zero and is a multiple of 99, then it must be equal to $99x$ for some small positive integer x. If such a multiple of 99 is equal to abc, then it is easy to check that $a = x - 1$, $b = 9$ and $c = 10 - x$. This means that the number added to its reverse must be equal to

$$100(x - 1 + 10 - x) + 10(9 + 9) + 1(10 - x + x - 1) = 900 + 180 + 9 = 1089.$$

So there's no magic involved at all, just some special properties arising from the fact that we write numbers using base 10.

Chapter 5

Groups

5.2 A first example: The integers

Groups containing elements of finite order

In this part of Chapter 5 I discussed how the integers with the operation of addition forms a group of infinite order, in which every element apart from zero has infinite order. Clearly any group $\mathcal{G} = (G, *)$ containing an element g of infinite order must itself have infinite order, since g, $g * g$, $g * g * g$, and so on must all be distinct (or else g will have finite order). This raises an interesting question.

Can a group of infinite order contain only elements of finite order? The simple answer is yes. There are many possible examples. One I find particularly appealing involves the set of all fractions between zero and one; that is, the group operation applies to the set of all fractions $\frac{p}{q}$, where p and q are coprime integers, i.e. $(p, q) = 1$, and $0 \leq p < q$. The group operation is addition 'modulo 1', which we write as $\oplus$. That is:

$$\frac{p}{q} \oplus \frac{p'}{q'} = \begin{cases} \frac{p}{q} + \frac{p'}{q'} & \text{if } \frac{p}{q} + \frac{p'}{q'} < 1 \\ \frac{p}{q} + \frac{p'}{q'} - 1 & \text{if } \frac{p}{q} + \frac{p'}{q'} \geq 1 \end{cases}$$

where the resulting fraction is reduced to ensure that it is in its lowest terms.

Clearly $0 \leq \frac{p}{q} \oplus \frac{p'}{q'} < 1$, i.e. the operation is well-defined. It is not hard to see that this forms a group. It is also clearly a group of infinite order, since there are an infinite number of possibilities for q. However, for any element $\frac{p}{q}$:

$$\underbrace{\frac{p}{q} \oplus \frac{p}{q} \oplus \cdots \oplus \frac{p}{q}}_{q \text{ times}} = 0.$$

That is, every element has finite order.

This group can be expressed more concisely using notation I introduced in Section 9.5. First observe that $(\mathbb{Z}, +)$ is a normal subgroup of $(\mathbb{Q}, +)$, i.e. $(\mathbb{Z}, +) \trianglelefteq (\mathbb{Q}, +)$. This follows because $\mathbb{Z}$ is a subset of $\mathbb{Q}$, and the fact that it is a subgroup is trivial to check; moreover, it is automatically normal since $(\mathbb{Q}, +)$ is abelian. Hence the quotient group $(\mathbb{Q}, +)/(\mathbb{Z}, +)$ is well-defined, and I leave you to check that this is precisely the group I just described, albeit in a less formal way.

5.6 Proving Euler's Theorem

An equivalence relation on p-tuples

In the proof of Theorem 5.3 I defined S to be the set of all vectors of length p (or p-tuples) with elements drawn from a group G with the property that their combination is equal to e, the group identity. That is

$$S = \{(a_1, a_2, \ldots, a_p) \ : \ a_1 * a_2 * \cdots * a_p = e\}.$$

I also defined the *rotation* of any p-tuple to be the p-tuple where all elements move forward one place and the last element goes to the front. So, for the p-tuple $(a_1, a_2, \ldots, a_p)$, its rotation is $(a_p, a_1, a_2, \ldots, a_{p-1})$. I write

$$R(a_1, a_2, \ldots, a_p) = (a_p, a_1, a_2, \ldots, a_{p-1}).$$

I then defined a relation $\sim$ on the set of p-tuples in S by $(a_1, a_2, \ldots, a_p) \sim (b_1, b_2, \ldots, b_p)$ if and only if $(b_1, b_2, \ldots, b_p)$ is a rotation of $(a_1, a_2, \ldots, a_p)$, i.e. if there exists an i such that

$$(b_1, b_2, \ldots, b_p) = R^i(a_1, a_2, \ldots, a_p).$$

I claimed that $\sim$ is an equivalence relation on S, something I will now prove. I need to show reflexivity, symmetry and transitivity.

- *Reflexivity.* This holds immediately since

  $$R^0(a_1, a_2, \ldots, a_p) = (a_1, a_2, \ldots, a_p),$$

  i.e. $(a_1, a_2, \ldots, a_p) \sim (a_1, a_2, \ldots, a_p)$ for any p-tuple $(a_1, a_2, \ldots, a_p)$.

- *Symmetry.* Next suppose $(a_1, a_2, \ldots, a_p) \sim (b_1, b_2, \ldots, b_p)$ for some $(a_1, a_2, \ldots, a_p)$ and $(b_1, b_2, \ldots, b_p)$, i.e.

  $$(b_1, b_2, \ldots, b_p) = R^i(a_1, a_2, \ldots, a_p)$$

  for some i. But, applying R^{p-i} to both sides of the equation gives

  $$R^{p-i}(b_1, b_2, \ldots, b_p) = R^p(a_1, a_2, \ldots, a_p),$$

and hence, since $R^p = R^0$,

$$R^{p-i}(b_1, b_2, \ldots, b_p) = (a_1, a_2, \ldots, a_p);$$

that is $(b_1, b_2, \ldots, b_p) \sim (a_1, a_2, \ldots, a_p)$, as required.

- *Transitivity.* Finally, suppose $(a_1, a_2, \ldots, a_p) \sim (b_1, b_2, \ldots, b_p)$ and $(b_1, b_2, \ldots, b_p) \sim (c_1, c_2, \ldots, c_p)$, and hence

$$(b_1, b_2, \ldots, b_p) = R^i(a_1, a_2, \ldots, a_p)$$

and

$$(c_1, c_2, \ldots, c_p) = R^j(b_1, b_2, \ldots, b_p)$$

for some i and j. This immediately means that

$$(c_1, c_2, \ldots, c_p) = R^{i+j}(a_1, a_2, \ldots, a_p)$$

and so $(a_1, a_2, \ldots, a_p) \sim (c_1, c_2, \ldots, c_p)$, as I needed to show.

5.7 Elementary properties of $\mathbb{Z}_m$

Abelian finite groups

At the start of Section 5.7 I made the claim that the abelian finite groups are relatively straightforward to classify without explaining why, or what the classification is.

I first want to give a simple lemma, which I should probably have included in Section 5.9. As will become clear below, this sheds light on the classification.

Lemma A5.1. *If s, t are coprime positive integers then $C_s \times C_t \simeq C_{st}$ where, as throughout, C_n represents the cyclic group of order n.*

Proof. Suppose g is a generator of C_s and h is a generator of C_t. Then it is simple to verify that (g, h) is a generator of $C_s \times C_t$, and the result follows. □

I will be a little lazy from hereon and simply write $C_s \times C_t = C_{st}$. The classification itself is actually very simple, as captured by the following theorem.

Theorem A5.1. *Suppose $\mathcal{G} = (G, *)$ is an abelian group of order n, where $n = p_1^{a_1} p_2^{a_2} \ldots p_s^{a_s}$ and the values p_i are the distinct prime factors of n. Then*

$$\mathcal{G} = \mathcal{G}_1 \times \mathcal{G}_2 \times \cdots \times \mathcal{G}_s$$

where $\mathcal{G}_i$ is an abelian group of order $p_i^{a_i}$.
Moreover, each group $\mathcal{G}_i$ satisfies

$$\mathcal{G}_i = C_{i,1} \times C_{i,2} \times \cdots \times C_{i,r_i}$$

where $C_{i,j}$ is a cyclic group of order $p_i^{b_{i,j}}$ for every i, and

$$a_i = b_{i,1} + b_{i,2} + \cdots + b_{i,r_i}$$

for every i. That is, each of the component groups $\mathcal{G}_i$ is made up of the product of a series of cyclic groups of order a power of the prime p_i.

For example, since $24 = 2^3 \times 3^1$, we know that any abelian group $\mathcal{G} = (G, *)$ of order 24 is equal to $\mathcal{G}_1 \times \mathcal{G}_2$ for abelian groups $\mathcal{G}_1$ and $\mathcal{G}_2$ of orders $2^3 = 8$ and 3 respectively. Moreover, while there is only one possibility for $\mathcal{G}_2$, namely C_3, the cyclic group of order 3, there are three possibilities for $\mathcal{G}_1$. These are C_8, $C_2 \times C_4$ and $C_2 \times C_2 \times C_2$. Hence there are precisely three abelian groups of order 24, namely $C_8 \times C_3$, $C_2 \times C_4 \times C_3$ and $C_2 \times C_2 \times C_2 \times C_3$.

Of course, from Lemma A5.1, there are other ways of expressing this result. The three distinct abelian groups of order 24 could also be written as:

- $C_{24} = C_8 \times C_3$;
- $C_6 \times C_4 = C_2 \times C_{12} = C_2 \times C_4 \times C_3$; and
- $C_6 \times C_2 \times C_2 = C_2 \times C_2 \times C_2 \times C_3$.

Showing that Theorem A5.1 holds is not difficult but requires quite a bit more work. If you are interested, the classification is a standard result to be found in almost any textbook on abstract algebra.

Every finite group is isomorphic to a subgroup of S_n

At the end of Section 5.7.2 I made the claim that by studying the symmetric group and its subgroups you are studying all finite groups, since it happens that every finite group is isomorphic to a subgroup of S_n for some n. This slightly surprising claim is easy to justify.

The result can be expressed in the following way. The result is named after the English Mathematician Arthur Cayley, whose name also cropped up in Chapter 5 in the context of Cayley Tables for groups. He is credited as being the first to formally define the abstract notion of a group.

Theorem A5.2 (Cayley's Theorem)**.** *Every group of finite order is isomorphic to a subgroup of the symmetric group S_n.*

Proof. Suppose $\mathcal{G} = (G, *)$ is a finite group. For every element of g let the function $f_g : G \to G$ be defined such that

$$f_g(h) = h * g$$

for every $h \in G$. I first need to show that f_g is actually a permutation of the elements of G; to do so I need to show it is one-to-one and onto.

- Suppose $f_g(h) = f_g(h')$; if I can show that this means that $h = h'$ then I have shown that f_g is one-to one. By definition of f_g we know that $h * g = h' * g$, and so $h * g * g^{-1} = h' * g * g^{-1}$, and hence $h = h'$, as required.
- Now suppose $k \in G$; to show f_g is onto I need to find an h such that $f_g(h) = k$. If I set $h = k * g^{-1}$, then $f_g(h) = h * g = k * g^{-1} * g = k$, and I am done.

That is, for every element $g \in G$, I have defined an element f_g of S_n — where, instead of permuting $\{1, 2, \ldots, n\}$, here the elements of S_n permute the n elements of G — but this is just a change of labels. Formally define the function $\alpha : G \to S_n$ to map any group element g to the permutation f_g; that is, $\alpha(g) = f_g$, where f_g is as I have just defined it. I next want to show that the set of images of α form a subgroup of S_n. By Lemma 5.2 I only need to show that if $g, h \in G$, then $\alpha(g) \cdot \alpha(h)^{-1} = \alpha(k)$ for some $k \in G$, where I use $\cdot$ to denote composition of permutations (observing that $\alpha(h)^{-1}$ is well defined since we know that f_h is a permutation). Since, by definition, $f_h(\ell) = \ell * h$ for every $\ell \in G$, it follows that

$$f_h \cdot f_{h^{-1}}(\ell) = f_{h^{-1}}(f_h(\ell)) = f_{h^{-1}}(\ell * h) = \ell * h * h^{-1} = \ell.$$

That is, $f_h \cdot f_{h^{-1}}$ is the identity permutation, and hence $f_h^{-1} = f_{h^{-1}}$.
Thus $\alpha(g) \cdot \alpha(h)^{-1} = f_g \cdot f_h^{-1} = f_g \cdot f_{h^{-1}}$, and hence

$$\alpha(g) \cdot \alpha(h^{-1})(\ell) = \ell * g * h^{-1}$$

i.e. $\alpha(g) \cdot \alpha(h^{-1}) = \alpha(g * h^{-1})$ and we have shown that the set of images of α, which we can write as $\alpha(G)$, forms a subgroup of S_n.

It remains to show that α is one-to-one, and that it satisfies the key property of an isomorphism, namely that $\alpha(g) * \alpha(h) = \alpha(g * h)$. But this latter property follows immediately from the above argument. Thus the only thing left to show is that α is one-to-one. So suppose that $\alpha(g) = \alpha(h)$ for $g, h \in G$, i.e. suppose that $f_g = f_h$. Then, by definition, it follows that $k * g = k * h$ for every $k \in G$, including, of course, the identity element. Thus it immediately follows that $g = h$, and hence α is one-to-one. The theorem follows. □

Chapter 7

Polynomials and polynomial rings

7.4 Shift register sequences

Galois registers

There are actually two types of linear feedback shift register: the structure I described in Section 7.4[1], and a slightly different structure known as a *Galois Shift Register* (GSR), named after Évariste Galois for reasons that I hope will become clear. The general form of a Galois Register is shown in Figure A7.1.

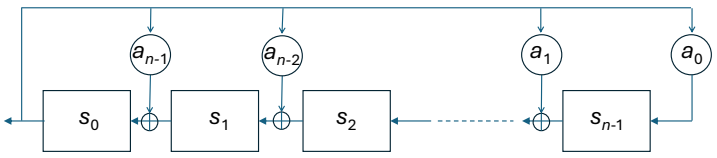

Figure A7.1: A Galois Shift Register

The main difference between this and the 'regular' LFSR is that, instead of feeding back a linear combination of the contents of various stages, a linear multiple of the contents of the leftmost stage is added to every value in the register. Of course, in the binary case, which is what I mainly focus on here, every value a_i is either 0 or 1, and hence the contents of the leftmost stage

[1]The structure I described in Section 7,4 is sometimes known as the Fibonnaci form, named after the 12th/13th century Italian mathematician Leonardo Bonacci, commonly known as Fibonacci (short for 'son of Bonacci'). This naming arises from the Fibonacci Sequence, introduced in a footnote in Section 4.5.1, which, like any shift register sequence, is generated using a linear recurrence.

is added to selected values as they shift left in the register.

Using the same notation as I introduced for LFSRs in Chapter 7, this means that the value fed into stage s_i at time t $(0 \leq i \leq n-1)$ is equal to $s_{i+1}(t) + a_{n-1-i} \cdot s_0(t)$, i.e.

$$s_i(t+1) = s_{i+1}(t) + a_{n-1-i} \cdot s_0(t),$$

where $0 \leq i \leq n-1$ and every value a_i is a field element. Exactly as before I always set a_0 to be non-zero, or else I can simply delete the rightmost stage of the register and get exactly the same effect. Note that the labelling of the feedback coefficients a_i uses the reverse order to that used for the LFSR — as I will show, this will ensure that the sequences output by a GSR are the same as those output by an LFSR for the same values a_i. Note that, as is common practice, in the rest of this discussion I omit the $\cdot$ representing finite field multiplication.

Again as in Chapter 7, in Figure A7.2 I give an example of a binary GSR with $n = 5$ using the same values of a_i as in Figure 7.4. That is, $a_0 = a_2 = 1$ and $a_1 = a_3 = a_4 = 0$.

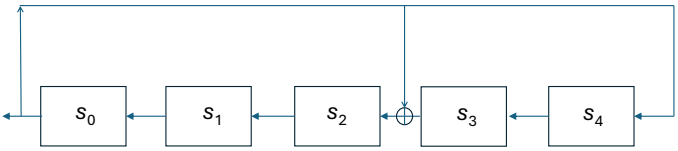

Figure A7.2: A Galois Shift Register for $n = 5$

Suppose the initial state is 00111, i.e.

$$s_0(0) = 0; \quad s_1(0) = 0; \quad s_2(0) = 1; \quad s_3(0) = 1; \quad \text{and} \quad s_4(0) = 1.$$

Then, moving everything left one place, and observing that the contents of stage s_2 will equal the sum of the contents of states s_3 and s_0 and the new contents of stage s_4 will simply equal the current contents of stage s_0, it follows that

$$s_0(1) = 0; \quad s_1(1) = 1; \quad s_2(1) = 1; \quad s_3(1) = 1 \quad \text{and} \quad s_4(1) = 0.$$

Hence the new state of the register (at time $t = 1$) is 01110. By a similar argument, the state for $t = 2$ is 11100. The successive states of the shift register for $t = 0, 1, \ldots, 32$ are shown in Table A7.1.

Just as in Table 7.1, the sequence of states has period exactly 31. Indeed, more than this, the sequence output by the register is identical to that output by the LFSR of Figure 7.4. This is easy to check by comparing the successive states of state s_0 in Tables 7.4 and A7.1. Of course, the sequence of *states* is not the same, although each stage of the GSR contains the same sequence of period 31, albeit at varying shifts.

Table A7.1: Successive states of the Galois Shift Register in Figure A7.2

t	s_0	s_1	s_2	s_3	s_4
0	0	0	1	1	1
1	0	1	1	1	0
2	1	1	1	0	0
3	1	1	1	0	1
4	1	1	1	1	1
5	1	1	0	1	1
6	1	0	0	1	1
7	0	0	0	1	1
8	0	0	1	1	0
9	0	1	1	0	0
10	1	1	0	0	0
11	1	0	1	0	1
12	0	1	1	1	1
13	1	1	1	1	0
14	1	1	0	0	1
15	1	0	1	1	1
16	0	1	0	1	1
17	1	0	1	1	0
18	0	1	0	0	1
19	1	0	0	1	0
20	0	0	0	0	1
21	0	0	0	1	0
22	0	0	1	0	0
23	0	1	0	0	0
24	1	0	0	0	0
25	0	0	1	0	1
26	0	1	0	1	0
27	1	0	1	0	0
28	0	1	1	0	1
29	1	1	0	1	0
30	1	0	0	0	1
31	0	0	1	1	1
32	0	1	1	1	0

To show why the sequence output by a GSR is the same as that output by an LFSR with the same values of a_i, I need the following simple lemma.

Lemma A7.1. *Suppose an n-stage GSR has feedback coefficients* a_i *(*$0 \leq i \leq n-1$*) and initial state* $(s_0(0), s_1(0), \ldots, s_{n-1}(0))$*. Then*

$$s_0(t+1) = a_{n-1}s_0(t)+a_{n-2}s_0(t-1)+\cdots+a_{n-i}s_0(t-i+1)+\cdots+a_0s_0(t-n+1),$$

for every $t \geq n-1$*, i.e. if the sequence output by the GSR is* $u_0, u_1, u_2, \ldots$ *then*

$$u_{i+1} = a_0u_{i-n+1} + a_1u_{i-n+2} + \cdots + a_{n-1}u_i$$

for every $i \geq n-1$*.*

Proof. I observed above that

$$s_i(t+1) = s_{i+1}(t) + a_{n-1-i}s_0(t), \tag{7.1}$$

for any i $(0 \leq i \leq n-1)$ and any $t \geq 0$. Hence, in particular,

$$s_0(t+1) = s_1(t) + a_{n-1}s_0(t). \tag{7.2}$$

But, setting $i = 1$ in (7.1), it also follows that

$$s_1(t) = s_2(t-1) + a_{n-2}s_0(t-1). \tag{7.3}$$

Substituting (7.3) into (7.2) I get

$$s_0(t+1) = s_2(t-1) + a_{n-2}s_0(t-1) + a_{n-1}s_0(t).$$

Repeating this process a further $n-2$ times yields the desired equation:

$$s_0(t+1) = a_{n-1}s_0(t)+a_{n-2}s_0(t-1)+\cdots+a_{n-i}s_0(t-i+1)+\cdots+a_0s_0(t-n+1).$$

Since the output sequence is the same as the successive contents of stage s_0, the recurrence for the output sequence (u_i) follows immediately. □

But the recurrence for the sequence (u_i) in Lemma A7.1 is exactly the same as the recurrence I established for the sequence output by an LFSR in Section 7.4.3, showing why the sequences output by the two types of register will always be identical if the feedback coefficients a_i are the same.

However, this does not explain why this type of register is known as a *Galois* Shift Register. In fact the name arises from the properties of the sequence of states of a GSR, which I next examine. To explain what is going on, it is necessary to think about the characteristic polynomial of the GSR, as well as the polynomial representing the state of the GSR. Analogously to the LFSR case, the characteristic polynomial is

$$a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{n-1} - x^n$$

of degree n. In the example of a 5-stage GSR shown in Figure A7.2, the characteristic polynomial is simply:

$$1 + x^2 + x^5$$

where $a_0 = a_2 = 1$ and $a_1 = a_3 = a_4 = 0$. Observe that this example involves the field with two elements, i.e. $(\mathbb{Z}_2, +, \cdot)$, where $+x^5$ is the same as $-x^5$.

I also need to consider the *state polynomial* $w_t(x)$ of the GSR, namely a polynomial representing the contents of the register at any time t. Let

$$w_t(x) = s_0(t)x^{n-1} + s_1(t)x^{n-2} + \cdots + s_i(t)x^{n-1-i} + \cdots + s_{n-2}(t)x + s_{n-1}(t).$$

That is, the coefficient of x^i in $w_t(x)$ is simply the value stored in stage s_{n-i-1} at time t. So, for the example in Figure A7.2, and referring to Table A7.1,

$$w_0(x) = x^2 + x + 1.$$

I can now give the following key result.

Lemma A7.2. *If a GSR has characteristic polynomial $p(x)$ and the state polynomial at time t is $w_t(x)$, then:*

$$w_{t+1}(x) = xw_t(x) \bmod p(x).$$

Proof. First observe that, ignoring the effect of the feedback for the moment, moving from state $w_t(x)$ to $w_{t+1}(x)$ involves shifting every variable left by one position. This is precisely the same as multiplying the polynomial by x as well as losing the value in stage s_0 at time t, i.e. $s_0(t)$. In polynomial terms this is equal to:

$$xw_t(x) - x^n s_0(t).$$

But the value in $s_0(t)$ is added back into each stage, after multiplication by the appropriate coefficient a_i; this simply means adding the polynomial $p(x) + x^n$ multiplied by $s_0(t)$ to the state polynomial. That is

$$w_{t+1}(x) = (xw_t(x) - x^n s0(t)) + s_0(t)(p(x) + x^n) = xw_t(x) + s_0(t)p(x).$$

It immediately follows that $p(x) \mid w_{t+1}(x) - xw_t(x)$, and the desired result follows. □

That is, to go from one state to the next involves multiplication by x modulo the characteristic polynomial. If $p(x)$ is an irreducible polynomial then, from Theorem 8.2, $\mathcal{F}(x)/p(x)$ is a field, where $\mathcal{F}$ is the field in which additions and multiplications take place. So the set of all register states, regarded as polynomials, correspond to the elements of this finite field. In particular,

if $\mathcal{F}$ is the field with two elements, the 2^n register states correspond to the 2^n elements of GF(2^n). Moreover, since going from one state to the next corresponds to multiplication by x in the field $\mathcal{F}(x)/p(x)$, the list of states of a GSR correspond to the successive powers of x multiplied by the state polynomial of the initial state.

The close connection between the operation of the GSR and multiplication in a finite field explains why they are known as Galois Registers. We can also immediately observe that the sequence of states of a GSR will have maximum period if and only if x is a primitive element in the multiplicative group of non-zero elements in GF(2^n). That is, if and only if $p(x)$ is a primitive polynomial. This gives us an alternative way of establishing Theorem 7.5.

7.4.5 Applications of m-sequences

Stream cipher decryption

I gave a description of how stream cipher encryption operates in Section 7.4.5, but I did not describe stream cipher *decryption*. I must apologise for this omission — I have worked with the idea for so long that it is second nature to me. In fact, in case you haven't worked it out already, stream cipher decryption works in exactly the same way as encryption.

The same secret key is required as is used for encryption, and is used to generate exactly the same keystream sequence as was used for encryption. This is combined with the ciphertext to recover the plaintext in exactly the same way as the keystream is combined with the plaintext to derive the ciphertext. This works because if the same bit is added modulo 2 to a plaintext bit *twice*, once during encryption and once during decryption, then the original plaintext bit will be recovered.

The idea of stream cipher encryption and decryption is analogous to a method of encryption called the *one time pad*, invented in the late 19th century by American cryptographer Franklin Miller; the idea was subsequently patented by American engineer Gilbert Sandford Vernam. In a one time pad system the sequence generator is replaced with a randomly-generated sequence of zeros and ones. The random process employed to generate the sequence must ensure that the ones and zeros are equiprobable. The random sequence needs to be generated in advance and given to both the encrypting and decrypting parties.

The system then works in exactly the same way as a stream cipher, where one random bit is combined with one plaintext bit to yield one ciphertext bit. The key requirement for secure use is that a random sequence is only ever used once for encryption. If this requirement is met, and the sequences used are truly randomly generated, then the system can be proved to be

completely unbreakable. This result is due to American engineer and mathematician Claude Elwood Shannon[2].

[2]The proof is contained in the remarkable paper: C E Shannon, 'Communication theory of secrecy systems', *Bell System Technical Journal* **28 no 4** (1949) 656-715, `https://doi.org/10.1002/j.1538-7305.1949.tb00928.x`, in which Shannon developed a mathematical theory of cryptography.

Chapter 8

Finite Fields

8.4 Prime Power Fields

Ideals

In Section 8.4 I mentioned the notion of an *ideal*, but in order to simplify the presentation I did not define what an ideal is. However, given their importance in abstract algebra, I thought it would be useful to provide a brief introduction here. The notion of an ideal evolved out of ideas of German mathematicians Ernst Kummer and Julius Wilhelm Richard Dedekind, but only took on the modern meaning in the early 20th century, notably in the work of Emmy Noether, mentioned in various places in the book.

Suppose $\mathcal{R} = (R, +, \cdot)$ is a ring, and let I be a subset of R that forms a subgroup of the additive group of $\mathcal{R}$; that is, $(I, +)$ is a subgroup of $(R, +)$. Then I is an ideal in R if, for any $a \in R$ and any $x \in I$, both $a \cdot x$ and $x \cdot a$ are elements of I. That is, the ideal is closed under left and right multiplication by arbitrary elements of R. Another way of expressing this is to say that $xI \subseteq I$ and $Ix \subseteq I$ for every $x \in R$, where xI and Ix simply denote the set of all pairs xa and ax, respectively, where $a \in I$. Of course, in a commutative ring we only need to require that $xI \subseteq I$.

An example of an ideal in the ring of integers is provided by the set of all even integers, since the product of any integer with an even integer is always an even integer and the set of all even integers forms an additive subgroup of the group of integers under addition. Similarly, if I is set to all multiples of any integer, e.g. all multiples of 6, then the result will be an ideal.

In a commutative ring, a *principal ideal* I is one for which there exists an $x \in I$ with the property that every element of I is a multiple of x, i.e. for every $y \in I$, $y = xa$ for some $a \in R$. In such a case it is conventional to write $I = (x)$, where every element of I is a multiple of x; thus the ideal of

all even integers can be written as (2). This notion can be extended to non-commutative rings, but for simplicity I consider only the commutative case. The examples I have given of ideals in the ring of integers are all principal ideals, since the set of even integers consists of all multiples of 2. In fact it can be shown that all ideals in $\mathbb{Z}$ are principal.

An integral domain in which every ideal is principal is called a *Principal Ideal Domain*, and it can be shown that every Principal Ideal Domain is a Unique Factorisation Domain. The integers are an obvious example of a Principle Ideal Domain, as are the ring of polynomials over a field.

I want to briefly mention one other important property of ideals. If I is an ideal in a ring $\mathcal{R}$, define $\sim$ to be the following relation on R: if $x, y \in R$ then $x \sim y$ if and only if $y - x \in I$. It is not hard to see that $\sim$ is an equivalence relation on R, and hence partitions R into distinct equivalence classes.

The set of equivalence classes is denoted by R/I, and if the operations $+$ and $\cdot$ are defined in the natural way from the underlying ring operations, then $(R/I, +, \cdot)$ forms a ring known as the *quotient ring*. This immediately gives an alternative way of defining the finite fields. Looking at Section 8.4, if $p(x)$ is a polynomial in $\mathcal{F}[x]$, then $(p(x))$ is a principal ideal in $\mathcal{F}[x]$, and hence $\mathcal{F}[x]/(p(x))$ is a ring — indeed, it is a field if $p(x)$ is an irreducible polynomial. In fact, this is exactly the same as the definition of Section 8.4, except expressed in the language of ideals.

8.5 Uniqueness and Representation of Finite Fields

I made two claims at the beginning of Section 8.5, namely that (a) the multiplicative group made up of the non-zero elements of GF(q) is cyclic, and (b) GF(q) is unique up to isomorphism. For the sake of completeness I provide a further brief discussion of these two results here.

The multiplicative group of a finite field is cyclic

It turns out that this result is straightforward to prove using other results I have given in this supplement to the book.

Theorem A8.1. *If q is an arbitrary prime power, the multiplicative group $(GF(q)^*, \cdot)$ is cyclic.*

Proof. We know that $(\mathrm{GF}(q)^*, \cdot)$ is a finite abelian group of order $q-1$. So, if $q - 1 = p_1^{a_1} p_2^{a_2} \dots p_s^{a_s}$, where the values p_i are the distinct prime factors of $q-1$, by Theorem A5.1, $(\mathrm{GF}(q)^*, \cdot)$ is equal to

$$\mathcal{G}_1 \times \mathcal{G}_2 \times \cdots \times \mathcal{G}_s$$

where $\mathcal{G}_i$ is an abelian group of order $p_i^{a_i}$. I claim that $\mathcal{G}_i$ is cyclic for every i, and the result will follow since the direct product of two cyclic groups of coprime orders is also cyclic (from Lemma A5.1).

To simplify matters, suppose $i = 1$, and consider all the elements of $\mathcal{G}_1 \times \mathcal{G}_2 \times \cdots \times \mathcal{G}_s$ of the form $(g, e_2, e_3, \ldots, e_s)$, where $g \in \mathcal{G}_1$ and e_i is the identity element of $\mathcal{G}_i$. There are clearly $p_1^{a_1}$ of these values, all with order dividing the order of $\mathcal{G}_1$, namely $p_1^{a_1}$. Now suppose $\mathcal{G}_1$ is not cyclic, and hence no element has order $p_1^{a_1}$, i.e. they must all have order dividing $p_1^{a_1-1} < p_1^{a_1}$.

Hence all the elements $(g, e_2, e_3, \ldots, e_s)$ are roots of the polynomial $x^t - 1$, where $t = p_1^{a_1-1}$. By the Factor Theorem (see page 232), and since $\mathrm{GF}(q)[x]$ is a Unique Factorisation Domain, there are at most t roots to a polynomial of degree t. Hence $t \geq p_1^{a_1}$ since $x^t - 1$ has $p_1^{a_1}$ roots. But $t = p_1^{a_1-1} < p_1^{a_1}$, and we have a contradiction. Thus $\mathcal{G}_1$ is cyclic and I am done. □

Finite fields are unique

I regret to say that I have been unable to find (or devise) an elementary proof that finite fields are unique up to isomorphism, although I can't help feeling that one ought to exist. The 'standard' proof technique involves using the notions of splitting field and algebraic closure, and relies on the uniqueness of splitting fields (itself a non-trivial result). In essence, given any polynomial $p(x) \in \mathcal{F}[x]$ for some field $\mathcal{F}$, a *splitting field* for $p(x)$ is the minimal field extension of $\mathcal{F}$ in which $p(x)$ can be expressed as a product of linear factors.

Of course, this begs the question of what an extension field is. An extension field $\mathcal{K}$ of a field $\mathcal{F}$ is simply a field that contains $\mathcal{F}$ as a subfield so that, for example, the reals are an extension field of the rationals.

Chapter 9

Why stop now?

9.2 Sizes of Infinity

Cardinality equality is an equivalence relation

Early in Section 9.1 I asserted that the notion of two sets having equal cardinality is an equivalence relation, but I did not prove it. Recall that if S and T are sets, then I say that their cardinality is equal, i.e. $|S| = |T|$, if and only if there is a one-to-one and onto function f mapping from S to T. In such a case there is said to be a *one-to-one correspondence* between S and T.

To establish that this is an equivalence relation, as ever I need to establish reflexivity, symmetry and transitivity.

- *Reflexivity.* Suppose $I : S \to S$ is the identity function, i.e. $I(s) = s$ for every $s \in S$. Clearly I is a 1-1 and onto function, and so we have reflexivity.
- *Symmetry.* Suppose $f : S \to T$ is a 1-1 and onto function. Then there exists an inverse $f^{-1} : T \to S$ that is also a 1-1 and onto function, and so we have symmetry.
- *Transitivity.* Suppose $f : S \to T$ and $g : T \to U$ are 1-1 and onto functions. Then the composition $f \circ g : S \to U$ is also 1-1 and onto, giving transitivity.

Recurring 9s in decimal representations of numbers

In the proof that the reals are uncountable, I asserted that $0.09999\ldots$ (with recurring nines) is the 'same' as 0.1. I imagine that many of you are already

familiar with this claim, i.e. that any decimal expansion ending with recurring 9s can be replaced by one where all the 9s are replaced by zeros, and the final digit before the recurring 9s is increased by 1, e.g. so that $23.44999\ldots$ is the same as 23.45.

But why is this true? To justify this I would like to introduce the well-known paradox of Achilles and the Tortoise, one of several paradoxes due to Greek philosopher Zeno of Alea[1], active almost 2500 years ago. The paradox involves a hypothetical race between Achilles (who is fast) and the very slow tortoise. If we suppose the tortoise is given a start, then Achilles will gradually catch up. The paradox is that Achilles can never catch up with the tortoise, because when Achilles reaches the point where the tortoise started, the tortoise will have moved further on. When Achilles reaches this point, the tortoise will have moved even further, and so on. That is, even after an arbitrarily large number of 'catching up' steps, Achilles will still be behind the tortoise.

Of course, in practice we know that Achilles will actually catch up and overtake the tortoise, so how do we resolve this apparent paradox? As I imagine you realise, the solution is that the sum of an infinite number of increasingly small values can be equal to a finite number. For example, I imagine you are happy with the notion that

$$\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots = 1.$$

This is exactly the same as asserting that $0.999\ldots = 1$, since

$$0.999\ldots = \frac{9}{10} + \frac{9}{100} + \frac{9}{1000} + \cdots = 1.$$

More formally, we can consider the *limit* of the sum

$$0.999\ldots = \frac{9}{10} + \frac{9}{100} + \frac{9}{1000} + \cdots + \frac{9}{10^i}$$

as i becomes indefinitely large or, as it is usually phrased, as $i \to \infty$. This can be made formal, although this is well beyond the scope of this book.

9.4 Difference sets, sequences & finite geometry

Barker Sequences

In Section 9.4, I introduced the notion of a *difference set*, i.e. a subset D of the elements of a group $\mathcal{G} = (G, *)$, with the property that every element

[1]For further details see, for example, `https://mathshistory.st-andrews.ac.uk/Biographies/Zeno_of_Elea/`

of G can be expressed as a product $d * e^{-1}$ in a fixed number of ways (written as λ). An important special case is where $\mathcal{G} = (\mathbb{Z}_v, +)$, and in this case I showed that the existence of a (v, k, λ)-difference set is equivalent to a periodic sequence with elements from $\mathbb{Z}_2$ of period v with out-of-phase autocorrelation equal to $v - 4(k - \lambda)$.

I now what to consider a closely related mathematical notion, which arises from an application in digital communications, namely the *Barker sequence*. When sending a sequence of bits through a communications channel, it is important for a receiver to be able to tell when the data starts. This led Irish physicist and digital communications pioneer Ronald Hugh Barker to devise the notion of what has become known as a Barker Sequence (or Barker code), described in a 1953 paper[2].

The idea is to devise a binary sequence, i.e a two-valued sequence, with the property that at any shift there are almost as many agreements as disagreements between the sequence and its shifted value. For mathematical convenience we label the two possible symbols in the sequence as $+1$ and -1. At first sight this is the same notion as that of autocorrelation, as described in Section 9.4; however, there is one important difference; that is, for this application we are interested in what is known as the *aperiodic autocorrelation*. To make this clearer I will next describe the application, before I give a formal definition.

Suppose a sequence $(s_1, s_2, \ldots, s_n)$, of length n say, is sent at the beginning of a data transmission. The idea is that a receiver finds out where the data starts by detecting this sequence, or *preamble*, i.e. by comparing every sequence of n consecutive binary digits with the expected sequence. If we suppose that the receiver looks at every 'window' of n consecutive bits, then as the n-bit sequence starts to be received, the receiver will successively compare the expected sequence with a sequence of n bits made up of:

- $n - 1$ 'noise' bits followed by s_1;
- $n - 2$ 'noise' bits followed by s_1 and s_2;
- ...
- 1 'noise' bit followed by $s_1, s_2, \ldots, s_{n-1}$; and, finally,
- $s_1, s_2, \ldots, s_n$.

By 'noise' bits here I mean meaningless bits extracted from a signal where nothing is actually being transmitted.

[2]R H Barker, 'Group synchronizing of binary digital systems', in *Communication Theory* (W. Jackson, ed.), Academic Press, New York, 1953, pp. 273–287.

One step in the comparison process is shown in Figure A9.1. For the purposes of the example I suppose that the preamble consists of the 11-bit sequence $(+1, +1, +1, -1, -1, -1, +1, -1, -1, +1, -1)$. The top line of the figure represents the incoming signal to the receiver, where the 11-bit preamble is preceded by four 'random' noise bits, denotes by x. We suppose that the receiver is examining the 11-bit sequence shown in a box to see if it is the preamble sequence. This is achieved by comparing this sequence with the expected preamble, shown on the second line of the figure. An agreement or disagreement between the received bit and the corresponding preamble bit is shown by a tick (✓) or a cross (✗), respectively, on the third line of the figure; the result of the comparisons with noise bits are shown as a question mark (?) since we cannot know what the noise bits will be.

In this case there are four agreements, five disagreements, and two unknowns. This means that the receiver will decide that this is not the preamble, and will move the 'comparison window', i.e. the box in the figure, right by one bit.

x	x	x	x	$+1$	$+1$	$+1$	-1	-1	-1	$+1$	-1	-1	$+1$	-1
		$+1$	$+1$	$+1$	-1	-1	-1	$+1$	-1	-1	$+1$	-1		
		?	?	✓	✗	✗	✓	✗	✓	✗	✗	✓		

Figure A9.1: Comparing a received sequence with the expected sequence

Obviously, in the last step there will be n agreements, or hopefully close to n agreements if we allow the possibility of errors occurring during transmission. In all the other cases we would like there to be as few agreements as possible to minimise the risk of a false detection of the sequence, as was the case in our example. False detections cannot be ruled out completely since, in the first case in the above sequence, if the $n-1$ noise bits happen to be the same as $s_1, s_2, \ldots, s_{n-1}$ then there will be at least $n-1$ agreements between the received sequence and the expected sequence, and a 'false synchronisation' will occur. Note that we make the implicit assumption when designing our preamble sequences that, since we cannot control them, the noise bits are equiprobable and randomly distributed.

To achieve this we would like the number of agreements between the first m bits of the preamble and the last m bits of the preamble to be equal to (or at most one different) from the number of agreements for all $m < n$. Put another way, the absolute value of the out-of-phase aperiodic correlation must be at most one, where the aperiodic correlation at a shift of t is defined to equal

$$s_1 s_{t+1} + s_2 s_{t+2} + \cdots + s_{n-t+1} s_n.$$

Since each s_i is either $+1$ or -1, each term in this sum is $+1$ when there is an agrement, and -1 when there is a disagreement, and so having an

aperiodic correlation of absolute value at most one at a shift of t is asking for the number of agreements minus the number of disagreements at a shift of t to be one of -1, 0 or $+1$.

A sequence with this 'ideal' out-of-phase aperiodic autocorrelation property is known as a Barker sequence, after Ronald Hugh Barker who first defined the idea. Indeed, Barker did more than this — he found examples of Barker sequences of length up to $n = 13$, presumably by trial and error. Examples of Barker sequences for lengths up to 13 are given in Table A9.1. It is simple to check that, in a Barker sequence, changing every $+1$ to a -1 (and vice versa), changing every other $+1$ to a -1 (and vice versa), and reversing the order of the elements will result in another (or the same) Barker sequence.

Table A9.1: Barker Sequences

$+1$												
$+1$	$+1$											
$+1$	$+1$	-1										
$+1$	$+1$	$+1$	-1									
$+1$	$+1$	$+1$	-1	$+1$								
$+1$	$+1$	$+1$	-1	-1	$+1$	-1						
$+1$	$+1$	$+1$	-1	-1	-1	$+1$	-1	-1	$+1$	-1		
$+1$	$+1$	$+1$	$+1$	$+1$	-1	-1	$+1$	$+1$	-1	$+1$	-1	$+1$

Table A9.1 contains sequences of length 1, 2, 3, 4, 5, 7, 11 and 13. In fact these are the only known lengths for which there exists a Barker Sequence. It has long been known that there are no more Barker Sequences of odd length, and that any Barker Sequence of even length must correspond to a $(4m^2, 2m^2 - m, m^2 - m)$ difference set. If such a difference set exists with $m > 1$ then m must be large — in fact m must be at least 10^{22}. It is widely conjectured that there are no more Barker Sequences, but it seems we are a long way from establishing this[3].

9.5 Finite simple groups

Permutations and transpositions

In Section 9.5 I defined what it means for a permutation to be even or odd, namely whether it can be expressed as a product of an even or an odd

[3]A nice discussion of the history of the *Barker Sequence Conjecture* in the context of the application domain is given in J. Jedwab, 'What can be used instead of a Barker sequence?', *Contemporary Mathematics*, **461 (2008),** `https://doi.org/10.1090/conm/461`

number of transpositions, i.e. permutations which exchange just two values. This is the *parity* of a permutation. I made two claims about this which I did not substantiate, namely that:

- every permutation can be expressed as a product of transpositions; and
- the notion of parity is well-defined, i.e. it is not possible to express a permutation as both an odd and an even number of transpositions.

These assertions, notably the second, are not trivially easy to prove, which is why I didn't include proofs in the book. However, I will give proofs here.

Theorem A9.1. *Every permutation can be written as a product of transpositions.*

Proof. Suppose f is a permutation on $\{1, 2, \ldots, n\}$, i.e. $f \in S_n$. Suppose j is the smallest integer for which $f(j) \neq j$; say $f(j) = k$, where $k > j$ (note that k cannot be less than j since $f(i) = i$ for every $i < j$). Let $g = t \circ f$, where t is the transposition that exchanges j and k. g now has the property that $g(i) = i$ for every $i < j + 1$. The process can be repeated for g, i.e. expressing g as the product of a transposition and a permutation h for which $h(i) = i$ for every $i < j + 2$; this means that f can be expressed as the product of two transpositions and a permutation h for which $h(i) = i$ for every $i < j + 2$. This process will eventually end where the permutation remaining is the identity permutation, enabling f to be expressed as a product of transpositions, and the result follows. □

Parity of permutations

Whilst it was straightforward to show that every permutation can be expressed as a product of transpositions, showing that the definition of the parity of a permutation is sound is not quite so simple. There are many proofs, but one that seems to me relatively simple (and elementary) can be found in a course handout due to American mathematician Keith Conrad[4]; I present a slight reworking of this proof below.

I first need the following simple set of identities. To simplify matters, and following common practice, I will write a transposition that exchanges a and b as (ab), and omit $\circ$ for composition, i.e. I will write $(ab)(cd)$ instead of $(ab) \circ (cd)$.

[4] See `www.math.uconn.edu/~kconrad/blurbs/grouptheory/sign.pdf`.

Lemma A9.1. *Suppose a, b, c, d are distinct integers from $\{1, 2, \ldots, n\}$. Then:*

$$(ab)(ac) = (bc)(ab);$$
$$(ab)(bc) = (bc)(ac);$$
$$(ab)(cd) = (cd)(ab);$$
$$(ab)(ab) = e;$$

where, as previously, e denotes the identity permutation.

Proof. The proof is by inspection.

- Both $(ab)(ac)$ and $(bc)(ab)$ map $a \to b$, $b \to c$ and $c \to a$.
- Both $(ab)(bc)$ and $(bc)(ac)$ map $a \to c$, $b \to a$ and $c \to b$.
- (ab) and (cd) commute since they move completely different elements of $\{1, 2, \ldots, n\}$.
- Finally $(ab)(ab) = e$ since every transposition is self-inverse.

□

The key result is the following Lemma.

Lemma A9.2. *Suppose*

$$t_1 \circ t_2 \circ \cdots \circ t_k = e,$$

where the t_i are transpositions in S_n, for some n, and e is the identity permutation. Then k is even.

Proof. The proof follows by induction on k. Clearly k cannot be 1 as no single transposition can equal the identity permutation. Obviously $k = 2$ is possible, since $t \circ t = e$ for any transposition t. So suppose $k > 2$ and that every product of fewer than k transpositions that equals the identity contains an even number of transpositions.

Now suppose $t_1 \circ t_2 \circ \cdots \circ t_k = e$ and $t_1 = (ab)$, for some $a \neq b$. Then $t_1 t_2$ will look like one of the four cases in Lemma A9.1. So, using the appropriate identity from this lemma, replace $t_1 t_2$ with either $t_1' t_2'$ where t_1' does not involve a, or e if $t_1 = t_2$. This gives a new product of either k or $k - 2$ transpositions equalling the identity. In the latter case, by the inductive hypothesis, it follows that $k - 2$ is even and hence k is even.

In the former case $t_2 = (ab')$ for some b', and hence I can repeat precisely the same argument with t_2 and t_3. Repeating this argument i times, for

$i \leq k-2$, either the final case of Lemma A9.1 is encountered at some stage, when the result follows by the inductive hypothesis, or the process gives a product of k transpositions where the first i transpositions do not involve a and the $(i+1)$st transposition *does* involve a.

Continuing this process, if the last case of the lemma is never encountered, when $i = k-2$ the process yields a product where the first $k-2$ transpositions do not involve a, and $t_{k-1} = (ab'')$ for some b''. I now claim that $t_k = t_{k-1} = (ab'')$. If not, then one of the first three identities from Lemma A9.1 will apply, yielding a product of k transpositions only the last of which involves a. But such a product cannot equal the identity since a will not map to itself. Hence $t_k = t_{k-1}$ and the last two transpositions cancel, giving a product of $k-2$ transpositions equalling the identity. The result follows from the inductive hypothesis. □

I can now establish that the definition of parity is sound.

Theorem A9.2. *No permutation can be written as a product of both an even and an odd number of transpositions.*

Proof. Suppose

$$t_1 \circ t_2 \circ \cdots \circ t_k = t'_1 \circ t'_2 \circ \cdots \circ t'_{k'}$$

where t_i and i'_j are transpositions for every i and j. Then, since $t \circ t = e$ for any transposition i, it immediately follows that

$$t_1 \circ t_2 \circ \cdots \circ t_k \circ t'_{k'} \circ t'_{k'-1} \circ \cdots \circ t'_1 = e.$$

Hence, by Lemma A9.2, $k + k'$ must be even. The desired result follows. □