MT461/MT5461
Theory of Error Correcting Codes

Mark Wildon, mark.wildon@rhul.ac.uk

The extra content on the syllabus for MT5461 is on Reed–Solomon codes and cyclic codes over finite fields. These codes are examples of the linear codes that will be covered in Part C of the main lectures.

The MSc/MSci lecture on Thursday 17 January will be moved to Friday 18 January, at 2pm in C219.

Fields

Definition 1.1

A *field* is a set of elements $\mathbf{F}$ with two operations, $+$ (addition) and $\times$ (multiplication), and two special elements $0, 1 \in \mathbf{F}$ such that $0 \neq 1$ and

(1) $a + b = b + a$ for all $a, b \in \mathbf{F}$;
(2) $0 + a = a + 0 = a$ for all $a \in \mathbf{F}$;
(3) for all $a \in \mathbf{F}$ there exists $b \in \mathbf{F}$ such that $a + b = 0$;
(4) $a + (b + c) = (a + b) + c$ for all $a, b, c \in \mathbf{F}$;

(5) $a \times b = b \times a$ for all $a, b \in \mathbf{F}$;
(6) $1 \times a = a \times 1 = a$ for all $a \in \mathbf{F}$;
(7) for all non-zero $a \in \mathbf{F}$ there exists $b \in \mathbf{F}$ such that $a \times b = 1$;
(8) $a \times (b \times c) = (a \times b) \times c$ for all $a, b, c \in \mathbf{F}$;

(9) $a \times (b + c) = a \times b + a \times c$ for all $a, b, c \in \mathbf{F}$.

If $\mathbf{F}$ is finite, then we define its *order* to be its number of elements.

Exercise: Show, from the field axioms, that if $x \in \mathbf{F}$, then x has a unique additive inverse, and that if $x \neq 0$ then x has a unique multiplicative inverse. Show also that if $\mathbf{F}$ is a field then $a \times 0 = 0$ for all $a \in \mathbf{F}$.

Exercise: show from the axioms for a field that if $\mathbf{F}$ is a field and a, $b \in \mathbf{F}$ are such that $a \times b = 0$, then either $a = 0$ or $b = 0$.

Theorem 1.2

Let p be a prime. The set $\mathbf{F}_p = \{0, 1, \ldots, p-1\}$ with addition and multiplication defined modulo p is a field.

Other finite fields

Example 1.3

The addition and multiplication tables for the finite field $\mathbf{F}_4 = \{0, 1, \alpha, 1+\alpha\}$ of order 4 are shown below.

$+$	0	1	α	$1+\alpha$
0	0	1	α	$1+\alpha$
1	1	0	$1+\alpha$	α
α	α	$1+\alpha$	0	1
$1+\alpha$	$1+\alpha$	α	1	0

$\times$	1	α	$1+\alpha$
1	1	α	$1+\alpha$
α	α	$1+\alpha$	1
$1+\alpha$	$1+\alpha$	1	α

Probably the most important thing to realise is that $\mathbf{F}_4$ **is not the integers modulo** 4.

Polynomials

Let $\mathbf{F}$ be a field. Let $\mathbf{F}[x]$ denote the set of all polynomials

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_m x^m$$

where $m \in \mathbf{N}_0$ and $a_0, a_1, a_2, \ldots, a_m \in \mathbf{F}$.

Definition 1.4

If $f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_m x^m$ where $a_m \neq 0$, then we say that m is the *degree* of the polynomial $f(x)$, and write $\deg f = m$. We define the degree of the *zero polynomial* $f(x) = 0$ to be -1.

Lemma 1.5 (Division algorithm)

Let $\mathbf{F}$ be a field, let $f(x) \in \mathbf{F}[x]$ be a non-zero polynomial and let $g(x) \in \mathbf{F}[x]$. There exist polynomials $s(x), r(x) \in \mathbf{F}[x]$ such that

$$g(x) = s(x)f(x) + r(x)$$

and either $r(x) = 0$ or $\deg r(x) < \deg f(x)$.

Other Results on Polynomials

For Reed–Solomon codes we shall need the following properties of polynomials.

Lemma 1.6

Let $\mathbf{F}$ *be a field.*

(i) *If* $f(x) \in \mathbf{F}[x]$ *has* $a \in \mathbf{F}$ *as a root, i.e.* $f(a) = 0$, *then there is a polynomial* $g(x) \in \mathbf{F}[x]$ *such that* $f(x) = (x - a)g(x)$.

(ii) *If* $f(x) \in \mathbf{F}[x]$ *has degree* $m \in \mathbf{N}_0$ *then* $f(x)$ *has at most* m *distinct roots in* $\mathbf{F}$.

(iii) *Suppose that* f, $g \in \mathbf{F}[x]$ *are non-zero polynomials such that* $\deg f$, $\deg g < k$. *If there exist distinct* $c_1, \ldots, c_k \in \mathbf{F}$ *such that* $f(c_i) = g(c_i)$ *for each* $i \in \{1, \ldots, k\}$ *then* $f(x) = g(x)$.

Polynomial Interpolation

Lemma 1.7 (Polynomial interpolation)

Let $\mathbf{F}$ *be a field. Let*

$$c_1, c_2, \ldots, c_k \in \mathbf{F}$$

be distinct and let $y_1, y_2, \ldots, y_k \in \mathbf{F}$*. The unique polynomial* $f(x) \in \mathbf{F}[x]$ *of degree* $< k$ *such that* $f(c_i) = y_i$ *for all* i *is*

$$f(x) = \sum_{i=1}^{k} y_i \frac{\prod_{j \neq i}(x - c_j)}{\prod_{j \neq i}(c_i - c_j)}.$$

§2 Definition and Basic Properties of Reed–Solomon Codes

Definition 2.1

Let p be a prime and let k, $n \in \mathbf{N}$ be such that $k \leq n \leq p$. Let

$$a_1, a_2, \ldots, a_n$$

be distinct elements of $\mathbf{F}_p$. For each polynomial $f(x) \in \mathbf{F}_p[x]$ we define a word $u(f) \in \mathbf{F}_p^n$ by

$$u(f) = (f(a_1), f(a_2), \ldots, f(a_n)).$$

The *Reed–Solomon code* associated to the parameters p, n, k and the field elements $a_1, a_2, \ldots, a_n$ is the length n code over $\mathbf{F}_p$ with codewords

$$\{u(f) : f \in \mathbf{F}_p[x],\ \deg f \leq k-1\}.$$

Example 2.2

Let $p = 5$ and let $k = 2$.

(1) If $n = 3$ and we take $a_1 = 0$, $a_2 = 1$ and $a_3 = 2$, then the associated Reed–Solomon code has a codeword

$$(f(0), f(1), f(2))$$

for each $f(x) \in \mathbf{F}_p[x]$ of degree ≤ 1. If $f(x) = bx + c$ then

$$u(f) = (c, b + c, 2b + c)$$

so the full set of codewords is

$$\{(c, b + c, 2b + c) : b, c \in \mathbf{F}_5\}.$$

(2) If $n = 4$ and we take a_1, a_2, a_3 as before, and $a_4 = 3$ then we get an extension of the code in (1).

Instructive Exercise

Exercise: Let $C = \{(c, b+c, 2b+c, 3b+c) : b, c \in \mathbf{F}_5\}$. Show that if $u \in C$ is sent down a noisy channel, and v is received such that $d(u, v) \leq 2$ then either $v = u$ or $v \notin C$. Can the receiver guarantee to detect if three errors occur?

Basic Properties

Lemma 2.3

If $f, g \in \mathbf{F}_p[x]$ *are distinct polynomials of degree* $\leq k-1$ *then*

$$d(u(f), u(g)) \geq n - k + 1.$$

Lemma 2.4

The Reed–Solomon code $RS_{p,n,k}$ *has size* p^k.

By Lemma 2.3, the minimum distance (as defined in Definition 4.1 of the main notes) of $RS_{p,n,k}$ is at least $n - k + 1$.

Minimum Distance of Reed–Solomon Codes

Theorem 2.5

The minimum distance of $RS_{p,n,k}$ is $n - k + 1$.

The Singleton Bound (to be proved in Part B of the main course) states that any p-ary code of length n and minimum distance d has at most p^{n-d+1} codewords. By Theorem 2.5, the Reed–Solomon codes meet this bound, and so have the largest possible size for their length and minimum distance.

Corollary 2.6

Let p be a prime. If k, $e \in \mathbf{N}$ are such that $k + 2e \leq p$ then the Reed–Solomon code $RS_{p,k+2e,k}$ is e-error correcting.

Decoding by Polynomial Interpolation

Example 2.7

Suppose we use the Reed–Solomon code with $p = 5$, $n = 4$ and $k = 2$ evaluating at $a_1 = 0$, $a_2 = 1$, $a_3 = 2$, $a_4 = 3$, as in Example 2.2(2). By Corollary 2.6, this code is 1-error correcting. Suppose we receive $v = (4, 0, 3, 0)$.

Given any two positions i and j, it follows from Lemma 1.7 that there is a unique polynomial g of degree < 2 such that $g(a_i) = v_i$ and $g(a_j) = v_j$.

The table on the next slide shows the interpolating polynomials for each pair of positions and the corresponding codewords. For example, to find $f(x)$ such that $f(0) = 4$ and $f(2) = 3$, we use Lemma 1.7 and get

$$f(x) = 4\frac{x-2}{0-2} + 3\frac{x-0}{2-0} = 3(x-2) - x = 2x + 4.$$

Conditions on f	Solution	Codeword $u(f)$
$f(0) = 4,\ f(1) = 0$	$f(x) = 4 + x$	$(4, 0, 1, 2)$
$f(0) = 4,\ f(2) = 3$	$f(x) = 4 + 2x$	$(4, 1, 3, 0)$
$f(1) = 0,\ f(2) = 3$	$f(x) = 2 + 3x$	$(2, 0, 3, 1)$
$f(0) = 4,\ f(3) = 0$	$f(x) = 4 + 2x$	$(4, 1, 3, 0)$
$f(1) = 0,\ f(3) = 0$	$f(x) = 0$	$(0, 0, 0, 0)$
$f(2) = 3,\ f(3) = 0$	$f(x) = 4 + 2x$	$(4, 1, 3, 0)$

In practice, we would stop as soon as we found the codeword $(4, 1, 3, 0)$ since $d(4130, 4030) = 1$, and by Question 6 on Sheet 2, there is at most one codeword within distance 1 of any received word.

Administration:

- Next week: projects advice (mathematical writing, marking scheme, getting started with LaTeX, . . .).

  As part of this we will discuss Hamming's original paper *Error Detecting and Error Correcting Codes*: see Question 5 on Sheet 4.

§3 Efficient Decoding of Reed–Solomon Codes

In this section we shall see an efficient algorithm for decoding Reed–Solomon codes invented by Berlekamp and Welch in 1983.

As usual we work with the Reed–Solomon code $RS_{p,n,k}$ where p is prime and n, $k \in \mathbf{N}$, and polynomials are evaluated at $a_1, a_2, \ldots, a_n$. Assume that $n = k + 2e$, so by Corollary 2.6 the code is e-error correcting.

Definition 3.1 (Key Equation)

The *Key Equation* for the received word $(v_1, v_2, \ldots, v_n)$ is

$$Q(a_i) = v_i E(a_i)$$

where $Q(x)$, $E(x) \in \mathbf{F}_p[x]$ are polynomials such that

- $\deg Q(x) \leq k + e - 1$
- $\deg E(x) \leq e$.

The polynomial $E(x)$ is called the *error locator polynomial*.

Key Equation: Small Example

Here is a small observation that helps to motivate the Key Equation.

Example 3.2

Suppose that $u(f)$ is sent and that a single error occurs in position j. Then $f(a_i) = v_i$ at all $i \neq j$. If we put in an extra term $x - a_j$ to 'hide' the error in position j, then the equation

$$f(x)(x - a_j) = v_i(x - a_j)$$

holds when we replace x with any a_i. So $Q(x) = f(x)(x - a_j)$, $E(x) = x - a_j$ is a solution to the Key Equation. Note that $f(x) = Q(x)/E(x)$ and the root of $E(x)$ tell us which position of v is in error.

Main Theorem on Key Equation

Recall that we work with the Reed–Solomon code $RS_{p,n,k}$ where p is prime and n, $k \in \mathbf{N}$, $n = k + 2e$ and polynomials are evaluated at $a_1, a_2, \ldots, a_n$.

Theorem 3.3

Suppose that $u(f)$ is sent and that errors occur in positions $j_1, j_2, \ldots, j_t$ where $t \leq e$. Let v be the received word. Then $Q(x)$, $E(x)$ solve the Key Equation if and only if

(i) $E(x) = (x - a_{j_1})(x - a_{j_2}) \ldots (x - a_{j_t})s(x)$ *for some polynomial* $s(x)$ *such that* $\deg s(x) \leq e - t$, *and*

(ii) $Q(x) = E(x)f(x)$.

Lemma 3.4

Suppose that the word $(v_1, \ldots, v_n)$ *is received. The polynomials*

$$Q(x) = Q_0 + Q_1 x + \cdots + Q_{k+e-1} x^{k+e-1}$$
$$E(x) = E_0 + E_1 x + \cdots + E_e x^e$$

in $\mathbf{F}_p[x]$ *satisfy the Key Equation if and only if*

$$Q_0 + a_i Q_1 + a_i^2 Q_2 + \cdots + a_i^{k+e-1} Q_{k+e-1} = v_i (E_0 + a_i E_1 + a_i^2 E_2 + \cdots + a_i^e E_e)$$

for each $i \in \{1, \ldots, n\}$. *An equivalent condition is that:*

$$\begin{pmatrix} 1 & a_1 & a_1^2 & \cdots & a_1^{k+e-1} & -v_1 & -v_1 a_1 & \cdots & -v_1 a_1^e \\ 1 & a_2 & a_2^2 & \cdots & a_2^{k+e-1} & -v_2 & -v_2 a_2 & \cdots & -v_2 a_2^e \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & a_n & a_n^2 & \cdots & a_n^{k+e-1} & -v_n & -v_n a_n & \cdots & -v_n a_n^e \end{pmatrix} \begin{pmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_{k+e-1} \\ E_0 \\ E_1 \\ \vdots \\ E_e \end{pmatrix} = 0$$

Example of Lemma 3.4

The matrix in Lemma 3.4 has n rows and $k + 2e + 1 = n + 1$ columns. So we can solve the Key Equation by solving an $n \times (n + 1)$ system of linear equations.

Example 3.5

We shall use the code of Example 2.2(2) and Example 2.7. Let $p = 5$, let $k = 2$, let $e = 1$ (so $n = 4$) and let $a_1 = 0$, $a_2 = 1$, $a_3 = 2$, $a_4 = 3$. With these parameters, the Key Equation for the polynomials $Q(x) = Q_0 + Q_1 x + Q_2 x^2$ and $E(x) = E_0 + E_1 x$ is

$$\begin{pmatrix} 1 & 0 & 0 & 4v_1 & 0 \\ 1 & 1 & 1 & 4v_2 & 4v_2 \\ 1 & 2 & 4 & 4v_3 & 3v_3 \\ 1 & 3 & 4 & 4v_4 & 2v_4 \end{pmatrix} \begin{pmatrix} Q_0 \\ Q_1 \\ Q_2 \\ E_0 \\ E_1 \end{pmatrix} = 0.$$

Example 3.5 [continued]

(1) Suppose we receive the word 4130. (This is the codeword for $f(x) = 4 + 2x$.) Then $v_1 = 4$, $v_2 = 1$, $v_3 = 3$, $v_4 = 0$ and we must solve

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 4 & 4 \\ 1 & 2 & 4 & 2 & 4 \\ 1 & 3 & 4 & 0 & 0 \end{pmatrix} \begin{pmatrix} Q_0 \\ Q_1 \\ Q_2 \\ E_0 \\ E_1 \end{pmatrix} = 0.$$

The kernel is two dimensional, spanned by the vectors

$$(0, 4, 2, 0, 1)^t, \quad (4, 2, 0, 1, 0)^t.$$

The first vector gives $Q(x) = 4x + 2x^2$ and $E(x) = x$, so we decode using $f(x) = Q(x)/E(x) = 4 + 2x$ to get $u(f) = 4130$. (The second vector gives the same answer even more quickly.)

Example 3.5 [continued]

(2) Suppose we receive the word 4030. Then $v_1 = 4$, $v_2 = 0$, $v_3 = 3$, $v_4 = 0$ and we must solve

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 2 & 4 & 2 & 4 \\ 1 & 3 & 4 & 0 & 0 \end{pmatrix} \begin{pmatrix} Q_0 \\ Q_1 \\ Q_2 \\ E_0 \\ E_1 \end{pmatrix} = 0.$$

The kernel is one dimensional spanned by $(1, 2, 2, 4, 1)^t$. So we take $Q(x) = 1 + 2x + 2x^2$ and $E(x) = 4 + x$. Polynomial division gives

$$Q(x)/E(x) = 2x + 4$$

so we decode using $f(x) = 2x + 4$ to get $u(f) = 4130$.

(3) Receive 4020. The kernel is one dimensional, spanned by $(4, 3, 3, 1, 0)^t$. So we take $Q(x) = 4 + 3x + 3x^2$ and $E(x) = 1$, but $Q(x)/E(x)$ does not have degree ≤ 1, so we are unable to decode. Since the Key Equation method always works when $\leq e$ errors occur, we know that ≥ 2 errors have occurred, but we are unable to correct them.

Final Remarks on Key Equation

When more than e errors occur it can also happen that the received word is decoded incorrectly. For example, this would happen in the setup of Example 3.4 if we received 4000. Another possibility is that $E(x)$ does not divide $Q(x)$: in this case we detect an error but are unable to correct it.

Final remarks:

(1) When preparing this section I used §4 of: `math.berkeley.edu/~mhaiman/math55/reed-solomon.pdf`.

(2) A MATHEMATICA notebook for solving the Key Equation is available on Moodle. Unless you really want to do it by hand, I suggest you use it (or another computer algebra program) to do the computational questions on problem sheets.

Frameworks for "Intellectual Property"

> *It has become fashionable to toss copyright, patents, and trademarks — three separate and different entities involving three separate and different sets of laws — plus a dozen other laws into one pot and call it "intellectual property". . . . The clearest way out of the confusion is to reject the term entirely.*
>
> Richard M. Stallman
>
> `http://www.gnu.org/philosophy/not-ipr.html`

- Government secrets (Confidential works)
- Trade secrets (Confidential works)
- Patents (Published works, decision to license and cost of license usually up to patent holder)
- Copyright (Published works, no obligation on copyright holder to distribute, even for a fee; basis of GPL)
- Trademarks (Published works)
- Academic good practice (Published works)

§4 Cyclic Codes

Definition 4.1

Let p be prime. A code C over $\mathbf{F}_p$ is *linear* if

(i) for all $u \in C$ and $a \in \mathbf{F}_p$ we have $au \in C$;

(ii) for all $u, w \in C$ we have $u + w \in C$.

Exercise: Show that any Reed–Solomon code is linear.

Definition 4.2

Let p be a prime. A code C over $\mathbf{F}_p$ is said to be *cyclic* if C is linear and

$$(u_0, u_1, \ldots, u_{n-1}) \implies (u_{n-1}, u_0, \ldots, u_{n-2}) \in C.$$

We say that $(u_{n-1}, u_0, \ldots, u_{n-2})$ is the *cyclic shift* of $(u_0, u_1, \ldots, u_{n-1})$.

Example 4.3

(1) Let p be prime and let $n \in \mathbf{N}$. The repetition code of length n over $\mathbf{F}_p$ is cyclic.

(2) Let C be the binary parity check code of length n consisting of all binary words of length n with evenly many 1s. We may define C using addition in $\mathbf{F}_2$ by

$$C = \{(u_0, \ldots, u_{n-1}) : u_i \in \mathbf{F}_2, u_0 + \cdots + u_{n-1} = 0\}.$$

Then C is a cyclic code.

(3) Let D be the binary code with codewords

$$\{000000, 110110, 011011, 101101\}.$$

Exercise: check that D is linear. The shift map acts on D by fixing 000000 and cyclically permuting the other three codewords. Hence D is cyclic.

Correspondence with Polynomials: First Attempt

There is a very helpful correspondence between codewords in a cyclic code and polynomials. Consider the code D in Example 4.3(3). If we associate the polynomial

$$u_0 + u_1x + u_2x^2 + u_3x^3 + u_4x^4 + u_5x^5$$

to the codeword $(u_0, u_1, u_2, u_3, u_4, u_5)$ then we have

$$\begin{aligned} 000000 &\longleftrightarrow 0 \\ 110110 &\longleftrightarrow 1 + x + x^3 + x^4 \\ 011011 &\longleftrightarrow x + x^2 + x^4 + x^5 \\ 101101 &\longleftrightarrow 1 + x^2 + x^3 + x^5. \end{aligned}$$

When we multiply $1 + x + x^3 + x^4$ by x we get $x + x^2 + x^4 + x^5$, which is the polynomial corresponding to 011011. So far so good! But when we multiply $x + x^2 + x^4 + x^5$ we get $x^2 + x^3 + x^5 + x^6$, which is not the polynomial we choose to correspond to 101101. Somehow we need to make $x^2 + x^3 + x^5 + x^6$ correspond to 101101, as well.

Correspondence with Polynomials

Definition 4.4

Let $n \in \mathbf{N}$ and let p be prime. Let $f(x) \in \mathbf{F}_p[x]$. We say that $f(x)$ *corresponds* to $(u_0, u_1, \ldots, u_{n-1}) \in \mathbf{F}_p^n$ if, when $f(x)$ is divided by $x^n - 1$, the remainder is

$$u_0 + u_1 x + u_2 x^2 + \ldots + u_{n-1} x^{n-1}.$$

Note, in particular, that if

$$f(x) = u_0 + u_1 x + u_2 x^2 + \cdots + u_{n-1} x^{n-1}$$

then, when we divide $f(x)$ by $x^n - 1$, the quotient is 0 and the remainder is $f(x)$. So $f(x)$ corresponds to the word $(u_0, u_1, \ldots, u_{n-1}) \in \mathbf{F}_p^n$.

Exercise: Check that, when $n = 6$, the polynomial $x^2 + x^3 + x^5 + x^6$ corresponds to $(1, 0, 1, 1, 0, 1)$. Find the word corresponding to the polynomial $x(1 + x^2 + x^3 + x^5)$.

Remark on Quotient Rings

Remark 4.5

Definition 4.4 defines a map from $\mathbf{F}_p[x]$ *to words in* $\mathbf{F}_p^n$. *Two polynomials have the same image if and only if they have the same remainder on division by* $x^n - 1$. *So the polynomials that map to the word* $(u_0, u_1, \ldots, u_{n-1})$ *are exactly the elements of the coset*

$$u_0 + u_1 x + \cdots + u_{n-1} x^{n-1} + \langle x^n - 1 \rangle$$

in the quotient ring $\mathbf{F}_p[x]/\langle x^n - 1 \rangle$. *Thus Definition 4.4 defines a bijection* $\mathbf{F}_p[x]/\langle x^n - 1 \rangle \longleftrightarrow \mathbf{F}_p^n$.

Critical Lemma

Lemma 4.6

Let p be a prime and let C be a cyclic code over $\mathbf{F}_p$. Let $u \in C$.

(i) *Suppose that $f(x) \in \mathbf{F}_p[x]$ corresponds to u. Then $xf(x)$ corresponds to the cyclic shift of u, namely $(u_{n-1}, u_0, \ldots, u_{n-2})$.*

(ii) *If $s(x) \in \mathbf{F}_p[x]$ then $s(x)f(x)$ corresponds to a codeword in C.*

It will often be useful to think of a cyclic code as a set of polynomials of degree $< n$, by choosing the obvious polynomial $u_0 + u_1 x + \cdots + u_{n-1}x^{n-1}$ to correspond to the codeword $(u_0, u_1, \ldots, u_{n-1})$.

Generator Polynomials

It will often be useful to think of a cyclic code as a set of polynomials of degree $< n$, by choosing the obvious polynomial $u_0 + u_1 x + \cdots + u_{n-1} x^{n-1}$ to correspond to the codeword $(u_0, u_1, \ldots, u_{n-1})$.

Definition 4.7

Let p be a prime. Let C be a cyclic code of length n over the finite field $\mathbf{F}_p$. Think of C as a set of polynomials of degree $< n$. A *generator polynomial* for C is a polynomial $g(x) \in \mathbf{F}_p[x]$ of degree $k < n$ such that $g(x)$ divides $x^n - 1$ and

$$C = \left\{ f(x)g(x) : f(x) \in \mathbf{F}_p[x], \deg f(x) \leq n - 1 - k \right\}.$$

Example 4.8

Let

$$C = \{0, 1 + x + x^3 + x^4, x + x^2 + x^4 + x^5, 1 + x^2 + x^3 + x^5$$

be the polynomial version of the code in Example 4.2(3). We shall show that $g(x) = 1 + x + x^3 + x^4$ is a generator polynomial for C.

Theorem 4.9

Let $\mathbf{F}$ *be a finite field and let* $C \subseteq \mathbf{F}_p[x]$ *be a cyclic code of length* n*, represented by polynomials of degree* $< n$*. Then* C *has a generator polynomial.*

From Generator Polynomial to Cyclic Code

Theorem 4.10

Let p be a prime, let $n \in \mathbf{N}$ and let $g(x) \in \mathbf{F}_p[x]$ be a divisor of $x^n - 1$. If $g(x)$ has degree $r < n$ then

$$\{g(x), xg(x), \ldots, x^{n-r-1}g(x)\}.$$

is a basis for a cyclic code C with generator polynomial $g(x)$.

In particular, Theorem 4.10 implies that a cyclic code of length n with a generator polynomial of degree r over the finite field $\mathbf{F}_p$ has dimension $n - r$ and size p^{n-r}.

Binary cyclic Codes of Length 6.

Example 4.11

In $\mathbf{F}_2[x]$ we have

$$x^6 - 1 = (1 + x)^2(1 + x + x^2)^2$$

where the factors $1 + x$ and $1 + x + x^2$ are irreducible, i.e. they cannot be written as products of polynomials of smaller degree. The polynomial divisors of $x^6 - 1$ are therefore 1, $1 + x$, $1 + x + x^2$ and

$$\begin{aligned}
(1 + x)^2 &= 1 + x^2, \\
(1 + x + x^2)^2 &= 1 + x^2 + x^4, \\
(1 + x)(1 + x + x^2) &= 1 + x^3, \\
(1 + x)^2(1 + x + x^2) &= 1 + x + x^3 + x^4, \\
(1 + x)(1 + x + x^2)^2 &= 1 + x + x^2 + x^3 + x^4 + x^5.
\end{aligned}$$

Generator Matrices

Theorem 4.12

Let p be prime and let C be a cyclic code of length n over $\mathbf{F}_p$ with generator polynomial $g(x) \in \mathbf{F}_p[x]$ of degree r. If $g(x) = a_0 + a_1 x + \cdots + a_r x^r$ then the $(n-r) \times n$ matrix

$$G = \begin{pmatrix} a_0 & a_1 & a_2 & \dots & a_r & 0 & \dots & 0 \\ 0 & a_0 & a_1 & \dots & a_{r-1} & a_r & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & a_0 & \dots & \dots & a_{r-1} & a_r \end{pmatrix}$$

is a generator matrix for C.

Let $k = n - r$. Encoding using this generator matrix, we encode the binary word $(b_0, b_1, \dots, b_{k-1})$ of length $k-1$ as $(b_0, b_1, \dots, b_{k-1})G$ As a polynomial, this codeword is

$$b_0 g(x) + b_1 x g(x) + \cdots + b_{k-1} x^{k-1} g(x) = (b_0 + b_1 x + \cdots + b_{k-1} x^{k-1}) g(x).$$

We end this section with a small result showing that most cyclic **binary** codes are 1-error correcting.

Theorem 4.13

Let C be a cyclic binary code of length n with generator polynomial $g(x) = a_0 + a_1x + \cdots + a_rx^r \in \mathbf{F}_2[x]$ of degree $r \geq 1$. Assume that $a_0 \neq 0$. If $x^s - 1$ is not divisible by $g(x)$ for any $i \in \{2, 3, \ldots, n-1\}$ then C has minimum distance at least 3.

In fact it is unnecessary to assume that $a_0 \neq 0$. If $a_0 = 0$ then $g(x)$ is divisible by x. Since $g(x)$ divides $x^n - 1$, it follows that x divides $x^n - 1$, which is impossible.

§5 Reed–Solomon Codes as Cyclic Codes

Lemma 5.1

Let p be a prime. There exists an element $a \in \mathbf{F}_p$ such that the non-zero elements of $\mathbf{F}_p$ are exactly a^j for $0 \leq j \leq p-2$.

For example, in $\mathbf{F}_7$, one can take $a = 3$, since

$$3^0 = 1,\ 3^1 = 3,\ 3^2 = 2,\ 3^3 = 6,\ 3^4 = 4,\ 3^5 = 5.$$

Such an element a is said to be a *primitive root*.

Throughout this section, fix a prime p and let $k < p$. We suppose that $p - 1 - k$ is even and let $p - 1 - k = 2e$. (This agrees with notation in §3.) Let $a \in \mathbf{F}_p$ be a primitive root and let $RS_{p,p-1,k}$ be the Reed–Solomon code where polynomials are evaluated at $a_i = a^{i-1}$ for $i \in \{1, \ldots, p-1\}$.

Lemma 5.2

The code $RS_{p,p-1,k}$ is cyclic.

Parity Check Matrix for Reed–Solomon Codes

Theorem 5.3

A parity check matrix for the cyclic Reed–Solomon code $RS_{p,p-1,k}$ *is*

$$H = \begin{pmatrix} 1 & a & a^2 & \cdots & a^{p-2} \\ 1 & a^2 & a^4 & \cdots & a^{2(p-2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a^{2e} & a^{4e} & \cdots & a^{2e(p-2)} \end{pmatrix}.$$

Corollary 5.4

The cyclic Reed–Solomon code $RS_{p,p-1,k}$ *has generator polynomial*

$$g(x) = (x-a)(x-a^2)\ldots(x-a^{2e}).$$

Parity Check Matrix for Reed–Solomon Codes

Theorem 5.3

A parity check matrix for the cyclic Reed–Solomon code $RS_{p,p-1,k}$ is

$$H = \begin{pmatrix} 1 & a & a^2 & \cdots & a^{p-2} \\ 1 & a^2 & a^4 & \cdots & a^{2(p-2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a^{2e} & a^{4e} & \cdots & a^{2e(p-2)} \end{pmatrix}.$$

Corollary 5.4

The cyclic Reed–Solomon code $RS_{p,p-1,k}$ has generator polynomial

$$g(x) = (x - a)(x - a^2) \ldots (x - a^{2e}).$$

Remark 5.5

Theorem 5.3 can be used, with Theorem 14.7 in the main notes, to give an alternative proof that the minimum distance of the cyclic Reed–Solomon code $RS_{p,p-1,k}$ is at least $p - k$.

Fast Decoding

Theorem 5.6

Let $v \in \mathbf{F}_p^n$ be a received word with syndrome

$$vH^{tr} = (S_1, \ldots, S_{2e}).$$

Suppose that at most e errors occurred. Then given any $A(x), B(x) \in \mathbf{F}_p[x]$ with $\deg A(x) \leq e-1$ and $\deg B(x) \leq e$ such that

$$B(x) = (S_1 + S_2 x + \cdots + S_{2e} x^{2e-1}) A(x)$$

where we work modulo x^{2e}, so all powers x^{2e} and higher in this equation are regarded as 0, *the sent codeword can be determined.*

Provided at most e errors occurred, such polynomials $A(x)$, $B(x)$ always exist. They can be found by linear algebra, in a similar way to Lemma 3.4. (See MATHEMATICA notebook on Moodle for an example.)