SM462 class notes

- Prof. W D Joyner[1]

Contents

[1]wdj@usna.edu. Last updated 2016-04-23.

These are notes for a course based on

- Judson [Ju15], starting with the chapter on rings,
- chapter 2 of Klein [Kl13],
- selected sections of MacWilliams and Sloane [MS77],
- chapter 6 of Hoffstein, Piper and Silverman [HPS].

1 Introduction to rings

This part contains a basic introduction to rings, with lots of examples.

1.1 Examples of rings using Sagemath

Before we formally define a ring (see the next section), the examples below will hopefully convince you that rings are things are are familiar with already. This section shows you how to construct some rings in Sagemath . Sagemath [Sa] is a free computer algebra system which allows you to construct the algebraic structures discussed in this class.

1. $\mathbb{R}$ - the real numbers, using ordinary addition and multiplication.

   and:

Sagemath

```
sage: RR(pi)
3.14159265358979
sage: RR2 = RealField(prec=100)
sage: RR2(pi)
3.1415926535897932384626433833
sage: RR(pi+2*e)
8.57815631050788
sage: RR(pi+2*e-sqrt(2)/3)
8.10675178971685
```

This tells us, for example, that

$$\pi = 3.14159265358979\ldots,$$

and

$$\pi + 2e - \frac{\sqrt{2}}{3} = 8.10675178971685\ldots\ .$$

`RR` is shorthand for `RealField(prec=53)`, the real field with 53 bits of precision. Using the optional argument `prec`, it is easy to change the precision displayed by Sagemath.

2. $\mathbb{C}$ - the complex numbers, using ordinary addition and multiplication.

Sagemath

```
sage: CC(2*pi*i)
6.28318530717959*I
sage: CC2 = ComplexField(prec=100)
sage: CC2(2*pi*i)
6.2831853071795864769252867666*I
```

This tells us, for example, that

$$2\pi i = (6.2831853071795864769252867666\ldots i).$$

`CC` is shorthand for `ComplexField(prec=53)`, the complex number field with 53 bits of precision. As with `RR`, using the optional argument `prec`, it is easy to change the precision displayed by Sagemath.

3. Finite fields $GF(q)$ (defined formally later), where q is a prime power, can constructed in Sagemath .

Sagemath

```
sage: GF(3)
Finite Field of size 3
sage: GF(9,"a")
Finite Field in a of size 3^2
sage: next_prime(2015)
2017
sage: F = GF(2017)
sage: 2^100
1267650600228229401496703205376
sage: F(2)^100
1264
```

Finite fields will be constructed later. For now, note that $GF(p)$, p prime, does not require a variable name, but $GF(q)$, q a prime power but not a prime, does require a variable name.

4. $\mathbb{Z}$ - the integers, using ordinary addition and multiplication.

These are built in and don't need to be constructed but here is the syntax.

Sagemath

```
sage: ZZ
Integer Ring
sage: R = ZZ
sage: 2015 in R
True
```

5. $M_{n\times n}(R)$ - the $n\times n$ matrices with coefficients in a ring F (e.g., $R=\mathbb{Z}$), using matrix addition and multiplication.

Sagemath

```
sage: R = MatrixSpace(ZZ, 2, 2)
sage: a = matrix(ZZ, [[1,2],[3,4]])
sage: b = matrix(ZZ, [[1,-2],[3,-4]])
sage: a in R
True
sage: b in R
```

```
True
sage: a+b
[2 0]
[6 0]
sage: a*b
[  7 -10]
[ 15 -22]
sage: b*a
[ -5  -6]
[ -9 -10]
```

`MatrixSpace(ZZ, m, n)` is the space of all $m \times n$ matrices over $\mathbb{Z}$ or `ZZ`. It is only a ring when $m = n$.

6. $\mathbb{Z}[\sqrt{d}] = \{a + \sqrt{d}b \mid a, b \in \mathbb{Z}\}$ - the extension of $\mathbb{Z}$ by $\sqrt{d}$, where $d \in \mathbb{Z}$. Directly constructing $\mathbb{Z}[\sqrt{d}]$ is possible in Sagemath in a few ways.

and:

Sagemath

```
sage: R = ZZ.extension(x^2 - 5, 'c')
sage: R.basis()
[1, c]
sage: c = R.basis()[1]
sage: c^2
5
sage: R.is_ring()
True
sage: R.is_integral_domain()
True
sage: R.is_field()
False
sage: 2*c in R
True
sage: R = ZZ.extension(x^2 + 3, 'd')
sage: R
Order in Number Field in d with defining polynomial x^2 + 3
sage: R.basis()
[1, d]
sage: d = R.basis()[1]
sage: d^2
-3
sage: R.is_ring()
True
sage: R.is_integral_domain()
True
sage: R.is_field()
False
```

The above constructs $\mathbb{Z}[\sqrt{5}]$ and $\mathbb{Z}[\sqrt{-3}]$. The method below is a more indirect method of computing $\mathbb{Z}[\sqrt{d}]$, and only works when $d > 1$.

and:

Sagemath

```
sage: K.<a> = NumberField(x^2 - 2)
sage: a^2
2
sage: R = K.maximal_order()
sage: R
Maximal Order in Number Field in a with defining polynomial x^2 - 2
sage: R.is_integral_domain()
True
sage: R.basis()
[1, a]
sage: R.is_field()
False
sage: 1/2 in R
False
sage:
sage: L.<b> = NumberField(x^2 + 3)
sage: b^2
-3
sage: S = L.maximal_order()
sage: S.is_field()
False
sage: S.basis()
[1/2*b + 1/2, b]
sage: 1 in S
True
sage: 1/2 in S
False
sage: S.is_integral_domain()
True
```

The above constructs $\mathbb{Z}[\sqrt{2}]$ and $\mathbb{Z}[\frac{1+\sqrt{-3}}{2}]$ (not $\mathbb{Z}[\sqrt{-3}]$).

7. $\mathbb{Z}/n\mathbb{Z}$ - the integers $\pmod{n}$, with addition and multiplication modulo n.

   Directly constructing $\mathbb{Z}/n\mathbb{Z}$ is easy to do in Sagemath (in fact, I can think of three ways to do this). Here is one construction:

and:

Sagemath

```
sage: Z12 = IntegerModRing(12)
sage: Z12.is_integral_domain()
```

```
False
sage: Z12.is_ring()
True
sage: Z12(12) # coerse 12 into Z12
0
sage: Z12(15) # coerse 15 into Z12
3
sage: Z12(6)+Z12(4)
10
sage: Z12(6)+Z12(14)
8
sage: Z12(6)*Z12(4)
0
sage:
sage: Z13 = IntegerModRing(13)
sage: Z12.is_integral_domain()
False
sage: Z13.is_integral_domain()
True
sage: Z13 = IntegerModRing(13)
sage: Z13.is_integral_domain()
True
sage: Z13(6)*Z13(4)
11
sage: Z13(6)+Z13(4)
10
```

For example, `Z12(6)+Z12(14) = 8` means

$$6 + 14 \equiv 8 \pmod{12}.$$

8. $R[x]$ - polynomials with coefficients in a ring R (e.g., $R = \mathbb{Z}$), under ordinary polynomial multiplication.

and:

Sagemath

```
sage: R.<y> = PolynomialRing(Z13, "y")
sage: 1+y in R
True
sage: 14+y in R
True
sage: 14+y
y + 1
sage: (14+y)*(y+12)
y^2 + 12
sage:
sage: S.<z> = PolynomialRing(Z12, "z")
sage: (14+z)*(z+12)
```

```
z^2 + 2*z
sage: 14+z in S
True
sage: 14+z
z + 2
```

9. $\mathbb{H} = \mathbb{H}_{\mathbb{R}}$ - the real quaternions, where

$$\mathbb{H} = \{a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} \mid a, b, c, d \in \mathbb{R}\},$$

where $\mathbf{i} \cdot \mathbf{j} = \mathbf{k}$, $\mathbf{j} \cdot \mathbf{k} = \mathbf{i}$, $\mathbf{k} \cdot \mathbf{i} = \mathbf{j}$, and $\mathbf{i} \cdot \mathbf{j} = -\mathbf{j} \cdot \mathbf{i}$, $\mathbf{i} \cdot \mathbf{k} = -\mathbf{k} \cdot \mathbf{i}$, $\mathbf{k} \cdot \mathbf{j} = -\mathbf{j} \cdot \mathbf{k}$. Note that

$$(a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k})\frac{a - b\mathbf{i} - c\mathbf{j} - d\mathbf{k}}{a^2 + b^2 + c^2 + d^2} = 1. \tag{1}$$

and:

Sagemath

```
sage: H.<i,j,k> = QuaternionAlgebra(QQ,-1,-1)
sage: a = 3*i - j + 2; b = -i + 5*j +7*k
sage: a; b; a*b; b*a
2 + 3*i - j
-i + 5*j + 7*k
8 - 9*i - 11*j + 28*k
8 + 5*i + 31*j
sage: i.matrix()
[ 0  1  0  0]
[-1  0  0  0]
[ 0  0  0 -1]
[ 0  0  1  0]
sage: j.matrix()
[ 0  0  1  0]
[ 0  0  0  1]
[-1  0  0  0]
[ 0 -1  0  0]
sage: k.matrix()
[ 0  0  0  1]
[ 0  0 -1  0]
[ 0  1  0  0]
[-1  0  0  0]
sage: b1,b2,b3,b4 = H.basis()
sage: 2*b2+3*b3+4*b4 in H
True
```

The last Sagemath command simply checks that $2\mathbf{i} + 3\mathbf{j} + 4\mathbf{k} \in \mathbb{H}$.

Note that $\mathbb{H}$ is a 4-dimensional vector space over $\mathbb{R}$ with basis $\{1, \mathbf{i}, \mathbf{j}, \mathbf{k}\}$.

In general, the Sagemath command QuaternionAlgebra(a, b) returns the quaternion algebra over the smallest field containing the nonzero elements a, b with generators $\mathbf{i}, \mathbf{j}, \mathbf{k}$ with $\mathbf{i}^2 = a$, $\mathbf{j}^2 = b$ and $\mathbf{j} \cdot \mathbf{i} = -\mathbf{i} \cdot \mathbf{j}$.

Exercise: Find the matrix representations of a, b, ab and ba.

10. $\mathbb{H}_{\mathbb{Z}}$ - the integral quaternions, where

$$\mathbb{H}_{\mathbb{Z}} = \{a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} \mid a, b, c, d \in \mathbb{Z}\}.$$

1.2 Definition of a ring

The previous section shows that rings are mathematical structures you are already familiar with. Here is the formal definition.

Definition 1. A nonempty set R is a *ring* if it has two closed binary operations, addition $+$ and multiplication $\cdot$ (or juxtaposition), satisfying the following conditions.

- $a + b = b + a$ for $a, b \in R$ ("$+$ is commutative").
- $(a + b) + c = a + (b + c)$, for $a, b, c \in R$ ("$+$ is associative").
- There is an element 0 in R such that $a + 0 = a$ for all $a \in R$ ("R has an additive identity element").
- For every element $a \in R$, there exists an element $-a$ in R such that $a + (-a) = 0$ ("each element of R has an additive inverse").
- $(ab)c = a(bc)$, for $a, b, c \in R$ ("$\cdot$ is associative").
- For $a, b, c \in R$, $a(b+c) = ab+ac$ and $(a+b)c = ac+bc$ ("the distributive laws hold").

Note that if $(R, +, \cdot)$ is a ring then $(R, +)$ is an abelian group.

Remark 1. If $(ab)c = a(bc)$, does *not* hold in general then we call R a *non-associative ring*. Such rings are important but not discussed in this course.

A few basic properties of rings are collected in the following result.

Lemma 2. *(Proposition 16.8 in the book [Ju15].)*

(a) For any $a \in R$, $0 \cdot a = a \cdot 0 = 0$.

(b) We have $(-a)b = a(-b) = -ab$, for all $a, b \in R$.

(c) We have $(-a)(-b) = ab$, for all $a, b \in R$.

Proof. (a) Use the distributive law to expand $a(a+0)$: $a^2 = a \cdot a = a(a+0) = a^2 + a \cdot 0$. The cancellation law implies $a \cdot 0 = 0$. Likewise, $a^2 = a \cdot a = (a+0)a = a^2 + 0 \cdot a$. The cancellation law implies $0 \cdot a = 0$.

(b) Use the distributive law to expand $a(b + (-b))$: Using (a), we have $0 = a(b + (-b)) = ab + a(-b)$. Now add $-ab$ to both sides. Similarly, $0 = (a + (-a))b = ab + (-a)b$. Now add $-ab$ to both sides.

(c) Note $x + (-x) = 0$ and $-(-x) + (-x) = 0$, so $x == (-x)$, for all $x \in R$. Therefore, using (b), we have $(-a)(-b) = -a(-b) = -(-ab) = ab$. □

You might wonder if the property $0 \cdot a = a \cdot 0 = 0$, for all $a \in R$, uniquely specifies 0. In other words, is 0 the only element of R which has this property? Is there a ring R for which there is a non-zero $\zeta \in R$ such that $\zeta \cdot a = a \cdot \zeta = 0$, for all $a \in R$? Yes, as the following bizarre example shows.

Example 3. *Let*

$$R = \{ \begin{pmatrix} 0 & x \\ 0 & 0 \end{pmatrix} \mid x \in \mathbb{Z} \}.$$

This is a commutative ring with the property that $ab = 0$ for any two $a, b \in R$.

If there is an element $1 \in R$ such that $1 \neq 0$ and $1a = a1 = a$, for each $a \in R$, we say that R is a *ring with identity* (or sometimes a *ring with unit*), and 1 is called the *identity element* of R. A ring R for which $ab = ba$, for all $a, b \in R$ is called a *commutative ring*.

An element $a \in R$ is with $a \neq 0$ is called a *unit in* R if there exists a unique element $a^{-1} \in R$ such that $a^{-1}1a = aa^{-1} = 1$. We say $a, b \in R$ are *associates* if there exists a unit u in R such that $a = ub$. The set of units in R is denoted

$$R^{\times}.$$

It is a group. Since the terminology "commutative ring with unit" is, while common, a bit confusing, we shall use the phrase "commutative ring with identity" when appropriate.

Remark 2. *You might think that if $a, b \in R$ and $(a) = (b)$ then a, b are associate. However, this is not true! See the paper [SBGJKMW].*

For any $a, b \in \mathbb{Z}$ with $a > 0$ and $b > 0$, let $gcd(a, b)$ denote the greatest integer whcih divides both a and b. This is called the *greatest common divisor* of a, b. If $n > 1$ is an integer then

$$\phi(n) = |\{m \in \mathbb{Z} \mid 1 \leq m \leq n, gcd(m, n) = 1\}.|$$

This is called the *Euler totient function* or the *Euler ϕ-function*.

Lemma 4. *(Bezout's Lemma) For any integers $a > 0$ and $b > 0$, there are integers x and y satisfying*

$$ax + by = \gcd(a, b).$$

Proof. Consider the set

$$(a, b) = \{ra + sb \mid r \in \mathbb{Z},\ s \in \mathbb{Z}\}.$$

Since $d = \gcd(a, b)$ divides a and b, this set (a, b) must be contained in the set

$$(d) = \{td \mid t \in \mathbb{Z}\},$$

i.e., $(a, b) \subset (d)$.

Suppose now $(d) \neq (a, b)$. Let $n > 0$ be the smallest integer such that

$$n \in (a, b),$$

written $n = ax + by$. By the integer "long division" algorithm, there is a remainder $r < d$ and a quotient q such that $n = qd + r$. But $r = n - qd \in (d)$, so either $r = 0$ (so $(d) \neq (a, b)$ is false) or r is a multiple of d (so $r < d$ is false). This is a contradiction. Therefore, $(d) = (a, b)$. □

Extended Euclidean Algorithm (xgcd):

- Initial table: $(a < b)$

i	q	r	u	v
-1		b	1	0
0		a	0	1

- As you increment i, apply the recursive equations: $q_i = [r_{i-2}/r_{i-1}]$, $r_i = r_{i-2} - r_{i-1}q_i$, $u_i = u_{i-2} - u_{i-1}q_i$, $v_i = v_{i-2} - v_{i-1}q_i$.
- Stop when $r_k = 0$ and then let

$$x = v_{k-1}, \quad y = u_{k-1}.$$

Example 5. *Let $a = x^4 + x + 1$ and let $b = x^7 + 1$.*
Since $x^7 + 1 = (x^3 + 1)(x^4 + x + 1) + (x^3 + x)$, we have

i	q	r	u	v
-1		$x^7 + 1$	1	0
0		$x^4 + x + 1$	0	1
1	$x^3 + 1$	$x^3 + x$	1	$x^3 + 1$

Using the recursive equations, we get

i	q	r	u	v
-1		$x^7 + 1$	1	0
0		$x^4 + x + 1$	0	1
1	$x^3 + 1$	$x^3 + x$	1	$x^3 + 1$
2	x	$x^2 + x + 1$	x	$x^4 + x + 1$
3	$x + 1$	$x + 1$	$x^2 + x + 1$	$x^5 + x^4 + x^3 + x^2$
4	x	1	$x^3 + x^2$	$x^6 + x^5 + x^3 + x + 1$
5	$x + 1$	0		

therefore

$$(x^4 + x + 1)(x^6 + x^5 + x^3 + x + 1) + (x^7 - 1)(x^3 + x^2) = 1.$$

Algorithm to compute the inverse of $c \pmod m$**:** Assume $gcd(c, m) = 1$ and compute x, y such that $cx + my = 1$ via the xgcd. We have $x \pmod m = c^{-1} \pmod m$.

Lemma 6. *Let $R = \mathbb{Z}/n\mathbb{Z}$, where $n > 1$ is a given integer. The group of units of R is given by*

$$(\mathbb{Z}/n\mathbb{Z})^\times = \{m \in \mathbb{Z} \mid 1 \leq m \leq n, gcd(m, n) = 1\}.$$

Moreover,

$$a^{\phi(n)} \equiv 1 \pmod{n},$$

for $a \in (\mathbb{Z}/n\mathbb{Z})^\times$.

Proof. The proof, which uses group theory, is simple short and clever.

Let

$$\Omega = \prod_{x \in (\mathbb{Z}/n\mathbb{Z})^\times} x.$$

Note $\Omega \neq 0$ since all the elements in the product are units. Note also that, for each $a \in (\mathbb{Z}/n\mathbb{Z})^\times$, the map

$$m_a : (\mathbb{Z}/n\mathbb{Z})^\times \to (\mathbb{Z}/n\mathbb{Z})^\times$$

given by $m_a(x) = ax$, is a one-to-one and onto map. Therefore, the sets

$$\{x \in (\mathbb{Z}/n\mathbb{Z})^\times\}, \qquad \{m_a(x) \in (\mathbb{Z}/n\mathbb{Z})^\times\},$$

describe the same set, which implies

$$\prod_{x \in (\mathbb{Z}/n\mathbb{Z})^\times} ax = \Omega.$$

This implies $a^\phi(n) = 1$ (in $(\mathbb{Z}/n\mathbb{Z})^\times$), so

$$a^{\phi(n)} \equiv 1 \pmod{n}.$$

□

Just because $a^{\phi(n)} = 1$, for all $a \in (\mathbb{Z}/n\mathbb{Z})^\times$, does not mean that $(\mathbb{Z}/n\mathbb{Z})^\times$ is cyclic. In other words, it is not true in general that $(\mathbb{Z}/n\mathbb{Z})^\times$ has an element of order $\phi(n)$ (in which case $1, a, a^2, \ldots, a^{\phi(n)-1}$ would exhaust all the elements of $(\mathbb{Z}/n\mathbb{Z})^\times$). However, when $n = p$ is a prime number then $(\mathbb{Z}/p\mathbb{Z})^\times$ is cyclic. In that case, a generator of $(\mathbb{Z}/p\mathbb{Z})^\times$ is called a *primitive root mod p.*

Exercise: Find the set of all units in $\mathbb{H}_\mathbb{Z}$ and show it is a group. What is it's order?

Example 7. *Let* $GF(2) = \mathbb{Z}/2\mathbb{Z}$*, with addition and multiplication* $\pmod 2$*, and let* $R = GF(2)^n$*. Let* $+$ *stand for component-wise addition* $\pmod 2$*, and let* $\cdot$ *stand for component-wise multiplication. The tuple* $(R, +, \cdot)$ *forms a ring.*

Exercise*: (a) Check this. (b) Is it a communtative ring with unit?*

Example 8. *Let* M *be the set of all midshipmen and let* $R = \mathcal{P}(M)$ *denote the collection of all subsets of* M*.*

Let $+$ *stand for the union operation* $\cup$ *and let* $\cdot$ *stand for* $\cap$*. The tuple* $(\mathcal{P}(M), +, \cdot)$ *forms a ring.*

Exercise*: (a) Check this. (b) Is it a communtative ring with unit?*

Question: Do you see a connection between these last two examples?

A commutative ring R with identity is called an *integral domain* if, for every $a, b \in R$ such that $ab = 0$, either $a = 0$ or $b = 0$. A *division ring* is a ring R with identity in which every nonzero element in R is a unit (while older texts call this a *skew field*). A commutative division ring is called a *field*.

Example 9. *The ring* $R = \mathbb{Z}/12\mathbb{Z}$ *is not an integral domain, since*

$$3 \cdot 4 \equiv 0 \pmod{12}.$$

However, the ring $R = \mathbb{Z}/13\mathbb{Z}$ *is an integral domain.*

Example 10. *The set* $\mathbb{Z}[i] = \{m + ni \mid m, n \in \mathbb{Z}\}$ *forms a ring sometimes called the Gaussian integers.*

Exercise*: Find the group of units of* $\mathbb{Z}[i]$*.*

Example 11. *The ring* $R = \mathbb{H}$ *of quaterions is a division ring because of the identity (1).*

Example 12. *The ring* $R = C^0(\mathbb{R})$ *of continuous functions on the real line is not a division ring.*

Exercise*: Construct continuous functions* $f(x)$ *and* $g(x)$*, each not identically zero, such that* $f(x)g(x) = 0$ *for all* $x \in \mathbb{R}$*.*

Definition 13. For any integer $n > 0$ and any element r in a ring R we write $nr = r + \ldots + r$ (n times). The *characteristic* of R is defined to be the least integer $n > 0$ such that $nr = 0$ for all $r \in R$. If no such integer exists, then the characteristic of R is defined to be 0. We will denote the characteristic of R by $char(R)$.

Example 14. *If $r = \mathbb{Z}/n\mathbb{Z}$ then $char(R) = n$.*
If $R = \mathbb{Z}$ then $char(R) = 0$.

Lemma 15. *If R is a ring with identity and if the order of 1 is n (regarding 1 as belonging to the abelian group $(R, +)$) then $char(R) = n$.*

Theorem 16. *The characteristic of an integral domain is either prime or zero.*

1.3 Subrings

In this section we lok at subset of a ring which preserve the operations of addition and multiplication.

Definition 17. A *subring* S of a ring R is a subset $S \subset R$ such that S is also a ring under the inherited operations from R.

Example 18.
- *The integers $R = \mathbb{Z}$ form a ring and the subset of multiples of 6, $S = 6\mathbb{Z}$, forms a subring.*
- *The polynomials having integral coefficients, $R = \mathbb{Z}[x]$, form a ring and the subset of integers, $S = \mathbb{Z}$, forms a subring.*
- *The real quaternions $\mathbb{H}$ form a ring, and the integral quaternions $\mathbb{H}_{\mathbb{Z}}$ forms a subring.*
- *The ring $S = \mathbb{Z}/12\mathbb{Z}$ is not a subring of $R = \mathbb{Z}$ because it is not a subset.*
-

Exercise: Classify all subrings of $\mathbb{Z}$.

Question: For which $n \geq m > 1$ is $\mathbb{Z}/m\mathbb{Z}$ a subring of $\mathbb{Z}/n\mathbb{Z}$?

Exercise: (a) Let $R = GF(2)^2$, with componentwise addition+ and multiplication $\cdot$:

$$(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2), \qquad (x_1, y_1) \cdot (x_2, y_2) = (x_1 \cdot x_2, y_1 \cdot y_2),$$

for $x_i, y_j \in GF(2)$. Show this is a ring.

(b) For $x \in GF(2)$ define $\overline{x} = x + 1$ (the "bit flip"). Let $S = GF(2)^2$, with addition $\oplus$ and componentwise multiplication $\cdot$:

$$(x_1, \overline{y_1}) \oplus (x_2, \overline{y_2}) = (x_1 + x_2, \overline{y_1 + y_2}), \qquad (x_1, y_1) \cdot (x_2, y_2) = (x_1 \cdot x_2, y_1 \cdot y_2),$$

for $x_i, y_j \in GF(2)$. Show this is a ring. However, note S is not a subring of R because the operations are not inherited.

Proposition 19. *Let R be a ring and S a subset of R. Then S is a subring of R if and only if the following conditions are satisfied.*

- $S \neq \emptyset$.
- $rs \in S$ *for all* $r, s \in S$.
- $r - s \in S$ *for all* $r, s \in S$.

Exercise: Prove Proposition 19.

Exercise: Do # 2, 3, 17, 18, 24-25, 29-30, 32-35, 38 from the book [Ju15], page 202-206.

1.4 Integral domains and fields

In this section, we define a very special type of ring called a field.

Definition 20. An *integral domain* is an commutative ring with identity which has not zero divisors. A subring of an integral domain is an integral domain.

If every non-zero element in a ring R with identity is a unit (i.e., is invertible in R) is called[2] a *division ring*. A subring of a division ring is not necessarily a division ring.

A commutative division ring is called a *field*.

[2]In older texts, the term *skew field* is used.

Roughly speaking, an integral domain R is the smallest type of ring for which the quotient field construction works. This is discussed in another chapter, but the simplest example is the integral domain $R = \mathbb{Z}$ and its quotient field

$$\mathbb{Q} = \{\frac{a}{b} \mid a, b \in \mathbb{Z}, b \neq 0\}.$$

Example 21.

- *The ring of rational quaternions, $R = \mathbb{H}_{\mathbb{Q}}$, is a division ring. (This was shown in an exercise above.) The subring $S = \mathbb{Q} \subset R$ is also a division ring. Since § is also commutative, it is a field.*
- *The subring $S = \mathbb{H}_{\mathbb{Z}} \subset R = \mathbb{H}_{\mathbb{Q}}$ is not a division ring.*
- *For which integers $n > 1$ is $R = \mathbb{Z}/n\mathbb{Z}$ a field?*

Proposition 22. Let R be a commutative ring with identity. TFAE:

- R is an integral domain.
- For all nonzero elements $a \in R$, $a \neq 0$, $ab = ac$, implies $b = c$. (The *cancellation law.*)

Proof. (2) $\Rightarrow$ (1):
...
(1) $\Rightarrow$ (2)
...
□

Theorem 23. *(Wedderburn's Theorem)* Every finite integral domain is a field.

Proof. ...
□

Example 24. *The ring $R = \mathbb{Z}/p\mathbb{Z}$ is a field, for each prime p.*

If R is a ring and $x \in R$ then the *order of* x, denoted $ord_R(x)$, is the least integer $n = ord_R(x) > 0$ such that

$$nx = x + \ldots + x = 0 \quad (n \text{ times})$$

if it exists. If no such n exists then define $ord_R(x) = \infty$. If R is a ring and if there exists a least integer $n > 0$ such that

$$nx = x + \ldots + x = 0 \quad (n \text{ times}),$$

for all $x \in R$, then n is called the *characteristic* of R, denoted $char(R) = n$. If no such n exists then define $char(R) = 0$.

It follows immediately from the definition that if R is finite then $char(R) \neq 0$.

Example 25.

- *If* $R = \mathbb{Z}/n\mathbb{Z}$ *then* $char(R) = n$.
- *If* $R = \mathbb{Z}$ *or* $R = \mathbb{Q}$ *then* $char(R) = 0$.
- *If* $R = \mathbb{Z}/3\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$ *then* $char(R) = 6$ *but if*

$$S = \{(x, 0) \mid x \in \mathbb{Z}/3\mathbb{Z}\} \subset R,$$

then $char(S) = 3$.

Lemma 26. *If* R *is a ring with identity* 1 *and if* $ord_R(1)$ *is finite then*

$$char(R) = ord_R(1).$$

Proof. If $n = ord_R(1)$ then

$$n \cdot r = n \cdot (1 \cdot r) = r \cdot (n \cdot 1) = 0,$$

for all $r \in R$. Therefore, $char(R)$ divides n If, on the contrary, we assume $char(R) \neq ord_R(1)$ then we must have $char(R) = m < n$, for some $m|n$. By definition of characteristic, we have $m \cdot 1 = 0$ (since $1 \in R$), which implies $ord_R(1) \neq n$. This is a contradiction. □

Theorem 27. *If* R *is an integral domain then either* $char(R) = 0$ *or* $char(R)$ *is a prime number.*

Proof. Suppose $char(R) = n \neq 0$. Suppose n is not a prime, so $n = ab$, for some $1 < a \leq b < n$. Since R is a commutative ring with identity without zero divisors, $0 = n \cdot 1 = (a \cot 1)(b \cdot 1)$, and therefore $a \cdot 1 = 0$ or $b \cdot 1 = 0$. In either case, we obtain $ord_R(1) < n$. This contradicts Lemma 26. □

We have repeatedly used the division algorithm when proving results about either Z or F [x], where F is a field. We should now ask when a division algorithm is available for an integral domain.

Definition 28. A *Euclidean domain* is an integral domain D such that, the following conditions hold.

(a) There is a *Euclidean valuation* $\nu : D \to \mathbb{R}$ satisfying (b) and (c) below.

(b) If $a, b \in D - \{0\}$, then $\nu(a) \leq \nu(ab)$.

(c) Let $a, b \in D$ with $b \neq 0$. Then there exist elements $q, r \in D$ such that

$$a = bq + r$$

and either $r = 0$ or $\nu(r) < \nu(b)$.

The integers $\mathbb{Z}$ and any polynomial ring in one variable over a field F, $F[x]$, are familiar examples of Euclidean domains.

Exercise: Do # 1, 4-5, 9, 11-12, 27, 39 from the book [Ju15], page 202-206.

1.5 Ring homomorphisms and ideals

Let R be a ring.

1.5.1 Ideals

An ideal of R is a special type of subring. The following definition spells this out more precisely.

Definition 29. An *ideal* of R is a subset $I \subset R$ for which

(a) I is a subring of R,

(b) I is closed under multiplication by elements of R.

Note that if R contains the identity then I cannot, unless of course $I = R$. (Indeed, if $1 \in I$ then $r = r \cdot 1 \in I$ for each $r \in R$ by condition (b) above.)

Example 30. *There are a few special cases where (a) implies (b). In other words, there are rings where every subring is actually an ideal.*

For example, if $R = \mathbb{Z}$ then any subring is closed under addition and therefore (since $n \cdot r = r + \ldots + r$) also closed under integer multiplication. For $R = \mathbb{Z}$ every ideal is of the form $I = a\mathbb{Z}$, for some $a \in \mathbb{Z}$. For example,

$$(3) = \{\ldots, -9, -3, 0, 3, 9, \ldots\},$$

is an ideal in $\mathbb{Z}$.

In general, an ideal of R of the form

$$I = aR = \{ar \mid r \in R\},$$

for some fixed $a \in R$ is called a *principal ideal* of R. When the ring R is clear from the context, it is denoted using the shorthand

$$I = (a),$$

and a is called a *generator* of R.

A ring R is called a *principal ideal ring* if every ideal is principal, and a *principal ideal domain* (PID) if every ideal is principal and R is an integral domain.

Lemma 31. *$R = \mathbb{Z}$ is a PID.*

Proof. Too easy. □

Note that if I is a principal ideal generated by a then ua is also a generator for any unit $u \in R$.

More generally, an ideal of R of the form

$$(a_1, \ldots, a_k) = \{a_1 r_1 + \ldots + a_k r_k \mid r_i \in R\},$$

for some fixed $a_1, \ldots, a_k \in R$, is called the ideal *generated by* $a_1, \ldots, a_k$. An ideal of this form is called *finitely generated*.

Any ideal in a polynomial ring (even in several variables) over a field is finitely generated. This is a very special case of what is called the Hilbert Basis Theorem. While the proof of this goes way beyond the scope of this course, it goes to show that finitely generated ideals are very common. Finding "nice" generators of an ideal in a polynomial ring in several variables is an active area of research called Gröbner basis theory.

Example 32. *Let $R = \mathbb{Z}[x, y]$ and let*

$$I = (x^2, y^2).$$

What is this ideal? It is the set of poynomials in x and y of the form

$$I = \{\sum_{i,j} a_{ij} x^i y^j \mid a_{ij} \in \mathbb{Z}, a_{0,0} = a_{1,0} = a_{0,1} = a_{1,1} = 0\}.$$

Exercise*: Explicitly describe $I = (x^2, y^3)$.*

Bottom line: ideals are a certain type of subring of a ring, and in many cases can be very explicitly described using generators.

Ideals arise naturally in a number of ways. One of the ways they arise is via ring homomorphisms.

1.5.2 Quotient rings

Let R be a ring and $I \subset I$ be an ideal.

Define the *equivalence class* of the element $a \in R$ by

$$\overline{a} = a + I = \{a + r : rinI\}.$$

This equivalence class is also sometimes written as $a \pmod I$ and called the *residue class of a modulo I*. The set of all such equivalence classes is denoted by R/I, called the quotient ring of R modulo I. It becomes a ring, if one defines

$$(a + I) + (b + I) = a + b + I; \qquad (a + I)(b + I) = ab + I.$$

Example 33. *Consider $R = \mathbb{Z}[x]$ and $I = (x^8 - 1$. The quotient ring*

$$\mathbb{Z}[x]/(x^8 - 1)$$

is the polynomial ring with integer coefficients mod $x^8 - 1$. Addition is the usual addition of polynomials. However, for polynomials $f, g \in \mathbb{Z}[x]/(x^8 - 1)$ represented by polynomilas of degree ≤ 7, define the $\cdot$ operation (multiplication) by

$$f(x) \cdot g(x) = \sum_{i=0}^{7} c_i x^i,$$

where

$$c_k = \sum_{i=0}^{7} a_i b_{k-i} \pmod 8, \qquad 0 \leq k \leq 7.$$

1.5.3 Ring homs

Let R and S be rings.

Definition 34. A *ring homomorphism* from R to S is a function

$$phi : R \to S,$$

which respects addition and multiplication:

- $\phi(a + b) = \phi(a) + \phi(b)$, for all $a, b \in R$,
- $\phi(a \cdot b) = \phi(a) \cdot \phi(b)$, for all $a, b \in R$.

Example 35. *Let $R = \mathbb{Z}[x]$ and $S = \mathbb{Z}$ and define $\phi : R \to S$ by*

$$\phi : a_0 + a_1 x + \ldots + a_n x^n \mapsto a_0.$$

Exercise*: Check this is a ring hom.*

Example 36. *Let $R = \mathbb{Z}[x]$ and $S = \mathbb{Z}$ and define $\phi : R \to S$ by*

$$\phi : a_0 + a_1 x + \ldots + a_n x^n \mapsto a_0 + a_1 + \ldots + a_n.$$

It's not immediately obvious, but this is a ring homomomorphism. The reason why is that ϕ can be regarded as a special case of the evaluation homomorphism. Let S be a ring and let R be any ring of functions f from S to itself. For a fixed $a \in S$, define

$$\phi_a : R \to R$$

by

$$\phi_a : f \mapsto f(a).$$

Using only the definitions, it is easy to show ϕ_a is a ring homomorphism. The map ϕ above is the special case $R = \mathbb{Z}[x]$, $S = \mathbb{Z}$, and $a = 1$.

Example 37. *Let $R = \mathbb{Z}$, let $m > 1$ be an integer, and let $S = \mathbb{Z}/m\mathbb{Z}$. Define $\phi : R \to S$ by*

$$\phi : a \mapsto a \pmod{m}.$$

Exercise*: Check this is a ring hom.*

Example 38. *Let $R = \mathbb{Z}^4$ and $S = M_{2\times 2}(\mathbb{Z})$ and define $\phi : R \to S$ by*

$$\phi : (a, b, c, d) \mapsto \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

Exercise*: Check this is not a ring hom.*

So now you know what rings homs are. What connection do they have with ideals?

The connection between them is described using the kernel. The *kernel* of a ring hom $\phi : R \to S$ is defined by

$$ker(\phi) = \{r \in R \mid \phi(r) = 0\}.$$

Example 39. *The kernel of the ring hom $\phi : \mathbb{Z} \to \mathbb{Z}/m\mathbb{Z}$, $\phi : a \mapsto a \pmod{m}$, is*

$$ker(\phi) = (m) = m\mathbb{Z}.$$

The kernel of the ring hom $\phi : \mathbb{Z}[x] \to \mathbb{Z}$, $\phi : a_0 + a_1x + \ldots + a_nx^n \mapsto a_0$, is

$$ker(\phi) = (x) = x\mathbb{Z}[x].$$

The kernel of the ring hom $\phi : \mathbb{Z}[x] \to \mathbb{Z}$, $\phi :: a_0 + a_1x + \ldots + a_nx^n \mapsto a_0 + a - 1 + \ldots + a_n$, is

$$ker(\phi) = (x - 1) = (x - 1)\mathbb{Z}[x].$$

1.5.4 UFDs

Let a, b be elements of a commutative ring with identity, R. We say that a divides b, and write $a|b$, if there exists an element $c \in R$ such that $b = ac$.

Definition 40. Let R be an integral domain. A nonzero element $p \in R - R^\times$ is said to be *irreducible* if $p = ab$ implies either $a \in R^\times$ or $b \in R^\times$. Furthermore, p is *prime* if whenever $p|ab$ either $p|a$ or $p|b$.

Remark 3. *It is important to notice that prime and irreducible elements do not always coincide. See [Ju15], page 282.*

Lemma 41. *Let R be an integral domain. If $p \in R$ is a prime element then p is an irreducible element.*

Proof. Suppose p is prime and $p = ab$, so $p|a$ or $p|b$. Assume WLOG that $p|a$, so that $a = dp$, for some $d \in R$. We have $p = ab = dbp$, so by the cancellation law (Prop. 22), $1 = db$. This implies $b \in R^\times$ so p is irreducible. □

Lemma 42. *Let R be either $\mathbb{Z}$ or $F[x]$ where F is a field. If $p \in R$ is an irreducible element then p is a prime element.*

Proof. Suppose p is irreducible and $p|ab$, so $dp = ab$, for some $d \in R$. Consider

$$I = (p, a) = \{xp + ya \mid x, y \in R\}.$$

This is an ideal in R. For $R = \mathbb{Z}$ or $R = F[x]$ this is a principal ideal. Here's the proof: Let $c \in I$ be a "smallest" non-zero element. In the case $R = \mathbb{Z}$, pick $c > 0$ to be the smallest positive integer in I. In the case $R = F[x]$, pick c to be a non-zero monic polynomial of least degree in I. In either case,

$$(c) = cR \subset I,$$

but we claim that in fact $(c) = I$. If not, pick $z \in I - (c)$. Now, let $r \in R$ be the remainder upon dividing c into z. Since $z = qc + r$, we have $r = z - qc \in I$. If $r \neq 0$ then r is a "smaller" element in I than c, a contradiction. Therefore $I = (c)$. Since $p \in I$, this implies $p = rc$, for some $r \in R$. But p is irreducible, so either r or c must be a unit. If $c \in R^\times$ then $I = R$ and $1 = xp + ya$, for some $x, y \in R$. Multiply by b to get $b = xbp + yab = xbp + ydp = (xb + yd)p$. This implies $p|b$. On the other hand, if $r \in R^\times$ then $I = (p)$. Since $a \in I$, this implies $a = pr'$, for some $r' \in R$. This implies $p|a$. □

Definition 43. We call R a *unique factorization domain* (UFD) if R satisfies the following criteria.

- Each $a \in R - R^\times$ can be written as the product of irreducible elements in R.
- The factorization is unique up to ordering of the factors.

For example, the ring of integers $\mathbb{Z}$ is a UFD.

1.6 Polynomial rings

Assume R is a commutative ring with identity. Any expression of the form

$$f(x) = a_0 + a_1 x + \dots a_n x^n$$

where $a_i \in R$ and $a_n \neq 0$, is called a *polynomial over* R with indeterminate x of *degree* n and write $\deg f(x) = n$. The set of all such polynomials is denoted

$$R[x].$$

The elements $a_0, a_1, \dots, a_n$ are called the *coefficients* of f. The (non-zero) coefficient a_n is called the *leading coefficient.* A polynomial is called *monic* if the leading coefficient is 1: $a_n = 1$.

Suppose

$$f(x) = \sum_{i=0}^{m} a_i x^i, \qquad g(x) = \sum_{i=0}^{n} b_i x^i$$

belong to $R[x]$. The set $R[x]$ has a + operation (addition) given by

$$f(x) + g(x) = \sum_{i=0}^{n} (a_i + b_i) x^i,$$

where $n > m$ and we set $a_{m+1} = \dots = a_n = 0$. It also has a $\cdot$ operation (multiplication)

$$f(x) \cdot g(x) = \sum_{i=0}^{n+m} c_i x^i,$$

where

$$c_k = a_0 b_k + a_1 b_{k-1} + \cdots + a_{k-1} b_1 + a_k b_0, \qquad 0 \leq k \leq m+n.$$

Again, we set $a_{m+1} = \ldots = a_{n+m} = 0$ and $b_{n+1} = \ldots = b_{n+m} = 0$. These operations give $R[x]$ are ring structure.

1.6.1 Application: Shamir's Secret Sharing Scheme

There are t committee members who must make an important unanimous decision[3]. These members are far away and must communicate their decision electronically. Moreover, we want some sort of proof that it is they who made the decision, not some hacker pretending to be them.

More formally, a *secret sharing scheme* consists of one *dealer* and u players. The dealer gives each player a *share* of a key K in such a way that any group of t or more players can together recover K but no group of less than t players can recover K. Such a system is called a (t, u)-*threshold scheme*.

Shamir's secret sharing scheme: Fix integers t and u with $1 < t < u$. Label the players $1, 2, \ldots, u$. Fix a large prime power $q.u$, and let $K \in GF(q)$ be the secret key. The dealer *key share generation and distribution* is follows. For each i with $1 \leq i \leq u$, the dealer generates distinct $x_i \in GFq)^{\times}$. For each i with $1 \leq i \leq t-1$, the dealer randomly generates (not necessarily distinct) $a_i \in GF(q)$. The a_i are kept secret from everyone, but the x_i are publically known. Define $a_0 = K$ and $p \in GF(q)[x]$ by

$$p(x) = p_K(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1}. \tag{2}$$

The dealer computes $y_i = p(x_i)$ and distributes the share

$$(x_i, y_i) \tag{3}$$

to player i.

[3]For example, they might all need to agree to open a safe and each member only has a portion of the combination, so without unanimous agreement, the safe cannot be opened. As another example, you want a committee of doctors in different locations to vote on whether a patient needs an operation. Each doctor who votes yes enters their password into a software program. Unless all t passwords are entered, the program will not return to you a yes vote.

The *key recovery* is this: When any t of the players join their shares together, the polynomial $p(x)$ can be recovered, so in particular $a_0 = p(0) = K$ can be recovered.

Before we prove why this is true, we give an example.

Example 44. *Suppose we have a $u = 10$ person committee, and $t = 6$ people must agree to reveal the key. We call the committee memebers player 1, ..., player 10. They all know $q = 11$ and the field $F = GF(q)$. Suppose the secret they share is $K = 7$. The dealer generates $p(x) = 2x^5 + 5x^4 + x^3 + 3x + 7$, where $p(0) = K = 7$, and computes*

x	1	2	3	4	5	6	7	8	9	10
$p(x)$	7	0	10	1	7	9	10	0	9	6

Give player i, the pair $(i, p(i))$, for $i = 1, \ldots, 10$. Each player knows the threshold number $t = 6$, so they can assume that the dealer's polynomial is degree 5, say

$$p(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5,$$

and they know $a_0 = K$ is the key that they are trying to solve for. If players $1, 2, 3, 4, 5, 10$ pool together their data, they know

$$a_0 + a_1 + a_2 + a_3 + a_4 + a_5 = p(1) = 7,$$

$$a_0 + 2a_1 + 4a_2 + 8a_3 + 5a_4 + 10a_5 = p(2) = 0,$$

$$a_0 + 3a_1 + 9a_2 + 5a_3 + 4a_4 + a_5 = p(3) = 10,$$

$$a_0 + 4a_1 + 5a_2 + 9a_3 + 3a_4 + a_5 = p(4) = 1,$$

$$a_0 + 5a_1 + 3a_2 + 4a_3 + 9a_4 + a_5 = p(5) = 7,$$

and

$$a_0 + 10a_1 + a_2 + 10a_3 + a_4 + 10a_5 = p(10) = 6.$$

This is a matrix equation $A\vec{a} = \vec{b}$, where

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 5 & 10 \\ 1 & 3 & 9 & 5 & 4 & 1 \\ 1 & 4 & 5 & 9 & 3 & 1 \\ 1 & 5 & 3 & 4 & 9 & 1 \\ 1 & 10 & 1 & 10 & 1 & 10 \end{pmatrix},$$

$\vec{a} = (a_0, \ldots, a_5)$ *is the vector of coefficients, and* $\vec{b} = (7, 0, 10, 1, 7, 6)$*. This matrix has non-zero determinant thanks to Lemma 109 on van der Monde matrices. Solving this matrix equation gives* $p(x)$*, and hence* K*. In more detail, the row-reduced echelon form of the augmented matrix* $(A \,|\, \vec{b})$ *is*

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 7 \\ 0 & 1 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 \end{pmatrix},$$

so $\vec{a} = (7, 3, 0, 1, 5, 2)$*. This allows us to recover the polynomial that the dealer generated,* $2x^5 + 5x^4 + x^3 + 3x + 7$*.*

We want to verify the following claim: When any t of the players join their shares (3) together, the polynomial $p(x)$ in (2), and in particular the key K, can be recovered.

Actually, we will show how to verify this in two ways.

First method of key recovery: Assume that

$$a_0 + a_1 x_i + a_2 x_i^2 + \ldots + a_{t-1} x_i^{t-1} = y_i,$$

are known for $1 \leq i \leq t$. This corresponds to a matrix equation $A\vec{a} = \vec{b}$, where A is a van der Monde matrix $A = (x_i^j)$, $\vec{a} = (a_0, \ldots, a_{t-1})$ is the vector of coefficients, and $\vec{b} = (y_1, \ldots, y_t)$. As in the example above, this matrix has non-zero determinant thanks to Lemma 109. Solving this matrix equation gives all the coefficients of $p(x)$, in particular the value of $a_0 = K$.

Second method of key recovery: Again, assume that

$$a_0 + a_1 x_i + a_2 x_i^2 + \ldots + a_{t-1} x_i^{t-1} = y_i,$$

are known for $1 \leq i \leq t$.

Lemma 45. *(Lagrange interpolation polynomial) Given* $k+1$ *points* (t_0, u_0)*,* $\ldots, (t_j, u_j), \ldots, (t_k, u_k)$*, where no two* t_j *are the same, the polynomial*

$$L(t) = \sum_{j=0}^{k} u_j \ell_j(t) \tag{4}$$

where

$$\ell_j(t) = \prod_{\substack{0 \le m \le k \\ m \ne j}} \frac{t - t_m}{t_j - t_m}, \qquad 0 \le j \le k,$$

satisfies $L(t_i) = s_i$. It is the unique polynomial of degree k having this property.

Proof. The fact L satisfies $L(t_i) = s_i$ for all i is true by inspection. To see that it is unique, suppose that $L(t)$ and $M(t)$ are two such polynomials. Then $M(t) - L(t)$ is a polynomial of degree k having $k+1$ zeros. Therefore, it must be the 0 polynomial.

□

To recover the key using Lagrange interpolation (Lemma 45), simply plug the x_is in for the t_is and the y_js in for the u_js. This gives $p(x)$, in particular the value of $a_0 = K$.

Example 46. *We continue with Example 44.*

Using Lagrange interpolation, we compute

$$\ell_1(x) = \prod_{a \in \{2,3,4,5,10\}} (x-a)/(1-a) = 3x^5 + 5x^4 + 6x^3 + 4x^2 + 8x + 8,$$

$$\ell_2(x) = \prod_{a \in \{1,3,4,5,10\}} (x-a)/(2-a) = 3x^5 + 8x^4 + 6x^3 + 10x^2 + 2x + 4,$$

$$\ell_3(x) = \prod_{a \in \{1,2,4,5,10\}} (x-a)/(3-a) = 9x^5 + 3x^3 + 3x^2 + 10x + 8,$$

$$\ell_4(x) = \prod_{a \in \{1,2,3,5,10\}} (x-a)/(4-a) = 4x^5 + 4x^4 + 10x^3 + 8x^2 + 8x + 10,$$

$$\ell_5(x) = \prod_{a \in \{1,2,3,4,10\}} (x-a)/(5-a) = x^5 + 2x^4 + 3x^3 + 7x^2 + 7x + 2,$$

$$\ell_{10}(x) = \prod_{a \in \{1,2,3,4,5\}} (x-a)/(10-a) = 2x^5 + 3x^4 + 5x^3 + x^2 + 9x + 2.$$

The polynomial generated by the dealer can be recovered using the Lagrange interpolation formula (Lemma 45):

$$L(x) = 7\ell_1(x) + 0\ell_2(x) + 10\ell_3(x) + 1\ell_4(x) + 7\ell_5(x) + 6\ell_{10}(x) = 2x^5 + 5x^4 + x^3 + 3x + 7.$$

1.6.2 Application: NTRU

The NTRU encryption algorithm is a cryptosystem whose security relies on the presumed difficulty of factoring certain polynomials in a truncated polynomial ring into a quotient of two polynomials having very small coefficients.

Let $N > 1$ be an integer and let $p > 2$, $q > 2$ be integers (often $p = 3$ and q is a large integer, such as $q = 2048$). Let

$$H = \mathbb{Z}[x]/(x^N - 1),$$

and let

$$H_p = (\mathbb{Z}/p\mathbb{Z})[x]/(x^N - 1), \qquad H_q = (\mathbb{Z}/q\mathbb{Z})[x]/(x^N - 1).$$

as a *set*, we may regard each of these as polynomials of degree $N - 1$ or less with coefficients in the appropriate ground ring. For each modulus $m > 1$, there is a natural ring homomorphism

$$\mathrm{mod}_m : H \to H_m$$

$$a_0 + a_1 x + \ldots + a_{N-1} x^{N-1} \to \overline{a_0} + \overline{a_1} x + \ldots + \overline{a_{N-1}} x^{N-1},$$

where $\overline{a_i} = a_i \pmod{m}$. (Of course, we will take $m = p$ or $m = q$.) When we represent $\overline{a_i}$ in $\{0, 1, \ldots, m-1\}$ then we say $\overline{a_i}$ has the (default) *standard lifting*. When we represent $\overline{a_i}$ in $(-m/2, m/2] \cap \mathbb{Z}$ then we say $\overline{a_i}$ has the 0-*centered lifting*.

Using mod_p, we sometimes abuse terminology and think of an element of H as belonging to H_p, when in fact we are really referring to its image under this map. Conversely, using one of these liftings, we may regard an element of H_p as an element of H.

Remark 4. *One must be very careful in compining these, for different p, q: the diagram*

$$\begin{array}{ccc} \mathbb{Z} & \xrightarrow{id} & \mathbb{Z} \\ {\scriptstyle \mathrm{mod}_p}\downarrow & & {\scriptstyle \mathrm{mod}_q}\downarrow \\ \mathbb{Z}/p\mathbb{Z} & \xrightarrow{???} & \mathbb{Z}/q\mathbb{Z} \end{array}$$

does not commute, at least if the maps are all homomorphisms of abelian groups.

For example, take $2016 \in \mathbb{Z}$, $p = 3$, $q = 5$. *We have* $\mathrm{mod}_p(2016) = 0$, *so the downward map on the left sends* $2016 \mapsto 0$. *The identity map on the top sends* $2016 \mapsto 2016$. *We have* $\mathrm{mod}_q(2016) = 1$, *so the downward map on the right sends* $2016 \mapsto 1$. *If the map on the bottom did exist then it would have to send* $0 \mapsto 1$. *However, no homomorphism of abelian groups sends the identity element to a non-identity element.*

Let $f \in H$ be an element which is a unit in both H_p and H_q. Pick another $g \in H$. In some implementations, the coefficients of both f and g are taken from $\{-1, 0, 1\}$. Let f_p^{-1} be the inverse of f in H_p and let f_q^{-1} be its inverse in H_q. The *private key* is the pair (f, g), known only to Bob. The *public key* is the element

$$h = h_{N,p,q,f,g} = g f_q^{-1} \in H_q.$$

For later use, we observe that

$$f(x)h(x) \equiv g(x) \pmod{q}.$$

Suppose Alice wants to send a message to Bob. Once Bob computes the public and private keys, he sends Alice the public key. Alice converts her message into a polynomial $m \in H$ and then secretly picks a random polynomial b (the "blinding value", known only to Alice). She then computes

$$c = p \cdot b \cdot h + m \in H_q. \tag{5}$$

This is the ciphertext she sends Bob.

Note that the encryption simply adds to the message an element of the ideal $(h(x))$, which is supposed to look like noise. This is not a problem is $h(x)$ is relatively prime to $x^N - 1$, since then this ideal is all of H. If $gcd(h(x), x^N - 1) \neq 1$ then $(h(x)) \neq H$ and $m(x)$ is not well-disguised.

To decrypt, compute $f(x)c(x)$ in H_q. Next, lift $f(x)c(x)$ to H using the 0-centered representation, then compose with mod_p to place the result in H_p. Now multiply this by $f_p^{-1}(x)$:

$$f_p^{-1}(x) f(x) c(x)$$

in H_p. For suitably choosen p and q and N, this agrees with $m(x) \pmod{p}$ with a high degree of probability.

Example 47. *Let*

$$N = 3, \quad p = 3, \quad q = 5, \quad f(x) = x^2 + 1, \quad g(x) = x^2 + 2x + 1.$$

Based on (7), to compute $f_p^{-1}(x)$*, we must solve*

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

Since

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 2 & 1 & 2 \\ 2 & 2 & 1 \\ 1 & 2 & 2 \end{pmatrix},$$

the coefficients of $f_p^{-1}(x)$ *are*

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 2 \\ 2 & 2 & 1 \\ 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}.$$

Therefore, we have

$$f_p^{-1}(x) = x^2 + 2x + 2 = x^2 - x - 1 \in H_p,$$

$$f_q^{-1}(x) = 2x^2 + 3x + 3 = 2(x^2 - x - 1) \in H_q,$$

and

$$h(x) = x^2 + x \in H_q.$$

If

$$b(x) = x^4 + 7x^2 - 2, \quad m(x) = 2x^2 - x + 1,$$

then $pb(x)h(x) = 2x^2 + 4 = 2x^2 - 1 \in H_q$*. Therefore,*

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

is the ciphertext.

Example 48. *Let*

$$N = 3, p = 3, q = 101.$$

In

$$H_p = (\mathbb{Z}/p\mathbb{Z})[x]/(x^N - 1),$$

we let $f(x) = x^2 + 1$. *Its inverse is denoted*

$$f_p^{-1}(x) = x^2 + 2x + 2.$$

In

$$H_q = (\mathbb{Z}/q\mathbb{Z})[x]/(x^N - 1),$$

its inverse is denoted

$$f_q^{-1}(x) = 50x^2 + 51x + 51.$$

Let

$$m(x) = x^2 - 2x - 2,$$

$$b(x) = 1 - x,$$

and let

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

We compute

$$h(x) = g(x)f_q^{-1}(x) = 51x^2 + 51x + 51$$

in H_q. *Let*

$$c(x) = p \cdot b(x)h(x) + m(x) = x^2 - 2x - 2$$

in R_q. *This is the ciphertext*[4]

To decrypt, compute fc *in* H_q*:*

$$f(x)c(x) = -x^2 - x - 4.$$

[4]It is too similar to $m(x)$, presumably indicating a poor choice of the "blinding value" $b(x)$.

Now multiply this by $f_p^{-1}(x)$*:*

$$f_p^{-1}(x)f(x)c(x) = x^2 + x + 1$$

in H_p*. This agrees with* $m \pmod p$*.*

Another example, with a different blinding value helps. Same N*,* p*,* q*,* $f(x)$ *and* $g(x)$*. Therefore* $h(x)$ *is also unchanged.*

Take

$$m(x) = x^2 - x + 1 \in H_q,$$

for the plaintext, and pick the blinding value $b(x) = 1 + x + x^2 \in H_q$*. The ciphertext is*

$$c(x) = b(x)h(x)+m(x) = 49x^6+49x^5+55x^3+7x^2+54x+53 = 56x^2+54x+56$$

in H_q*. To decrypt, compute* fc *in* H_q*:*

$$f(x)c(x) = 11x^2 + 9x + 9.$$

Now multiply this by $f_p^{-1}(x)$*:*

$$f_p^{-1}(x)f(x)c(x) = x^2 + 2x + 1.$$

We have recovered the plaintext $m(x)$*.*

If

$$h(x) = h_0 + h_1x + \ldots + h_kx^{N-1},$$

we associate to h the circulant matrix

$$Mat_N(h) = \begin{pmatrix} h_0 & h_{N-1} & \ldots & h_1 \\ h_1 & h_0 & \ldots & h_2 \\ \vdots & & & \vdots \\ h_{N-1} & h_{N-2} & \ldots & h_0 \end{pmatrix} \tag{6}$$

and define $vec_N(h)$ to be the last row of $Mat_N(h)$:

$$vec_N(h) = (h_0, \ldots, h_{N-1}).$$

In the ring $H = \mathbb{Z}[x]/(x^N - 1)$, the notation of (5), we have

$$vec_N(bh) = Mat_N(h)vec_N(b). \tag{7}$$

1.6.3 Application: Modified NTRU

Next, we give a modified version of NTRU, following Damien Stehlé and Ron Steinfeld, *Making NTRUEncrypt and NTRUSign as Secure as Worst-Case Problems over Ideal Lattices*, Eurocrypt2011 proceedings. This slight variation is provable hard and (so far) is quantum resistant.

Fix integers N, p, q where $N > 1$ is a power of 2, $p > 1$ (typically $p = 3$), $q > p$ and $gcd(p, q) = 1$. Let

$$H = \mathbb{Z}[x]/(x^N + 1),$$

$H_p = (\mathbb{Z}/p\mathbb{Z})[x]/(x^N + 1)$, and $H_q = (\mathbb{Z}/q\mathbb{Z})[x]/(x^N + 1)$. As before, we may regard each of these, regarded as a *set*, as polynomials of degree $N-1$ or less with coefficients in the appropriate ground ring. For each modulus $m > 1$, there is a natural ring homomorphism

$$\mathrm{mod}_m : H \to H_m$$

$$a_0 + a_1 x + \ldots + a_{N-1} x^{N-1} \to \overline{a_0} + \overline{a_1} x + \ldots + \overline{a_{N-1}} x^{N-1},$$

where $\overline{a_i} = a_i \pmod{m}$. (Of course, we will take $m = p$ or $m = q$.) Using this, we sometimes abouse terminology and think of an element of H as belonging to H_p, when in fact we are really referring to its image under this map.

Let $f \in H$ be an element which is a unit in both H_p and H_q. Let f_p^{-1} be its inverse in H_p and f_q^{-1} be its inverse in H_q. Pick another $g \in H$. The *private key* is the pair (f, g), known only to Bob. The *public key* is the element

$$h = h_{N,p,q,f,g} = g f_q^{-1} \in H_q.$$

Suppose Alice wants to send a message to Bob. Once Bob computes the public and private keys, he sends Alice the public key. Alice converts her message into a polynomial $m \in H$ and then secretly picks a random polynomial b (the "blinding value", known only to Alice). She then computes

$$c = p \cdot b \cdot h + m \in H_q. \tag{8}$$

This is the ciphertext she sends Bob.

Note that the encryption simply adds to the message an element of the ideal $(h(x))$, which is supposed to look like noise. This is not a problem

is $h(x)$ is relatively prime to $x^N + 1$, since then this ideal is all of H. If $gcd(h(x), x^N + 1) \neq 1$ then $(h(x)) \neq H$ and $m(x)$ is not well-disguised.

Decryption is the same as in the unmodified case: compute $f(x)c(x)$ in H_q, then lift fc to H using the 0-centered representation, Now multiply this by $f_p^{-1}(x)$:

$$f_p^{-1}(x)f(x)c(x)$$

in H_p. For suitably choosen p and q and N, this agrees with $m(x) \pmod{p}$ with a high degree of probability.

Example 49. *Let*

$$N = 4, \quad p = 3, \quad q = 5, \quad f(x) = x^2 + 1, \quad g(x) = x^2 + 2x + 1.$$

Based on (9), to compute $f_p^{-1}(x)$, we must solve

$$\begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 2 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

over $\mathbb{Z}/3\mathbb{Z}$. Since

$$\begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 2 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 2 & 0 & 2 & 0 \\ 0 & 2 & 0 & 2 \\ 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 2 \end{pmatrix},$$

we have

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 1 \\ 0 \end{pmatrix},$$

so we have

$$f_p^{-1}(x) = x^2 - 1 \in H_p.$$

Likewise, over $\mathbb{Z}/5\mathbb{Z}$, we have

$$\begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 3 & 0 & 3 & 0 \\ 0 & 3 & 0 & 3 \\ 2 & 0 & 3 & 0 \\ 0 & 2 & 0 & 3 \end{pmatrix},$$

so we have

$$f_q^{-1}(x) = 2(x^2 - 1) \in H_q,$$

and

$$h(x) = -x^3 + x + 1 \in H_q.$$

If

$$b(x) = x^4 + 7x^2 - 2, \quad m(x) = 2x^2 - x + 1,$$

then $pb(x)h(x) = x^2 + 2x + 1 \in H_q$. *Therefore,*

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

is the ciphertext.

Some Sagemath *code verifying this is below:*

Sagemath

```
sage: p = 3; q = 5
sage: PRq.<xq> = PolynomialRing(IntegerModRing(q), "xq")
sage: PRp.<xp> = PolynomialRing(IntegerModRing(p), "xp")
sage: R2.<x> = PolynomialRing(ZZ, "x")
sage: QR = QuotientRing(R2, R2.ideal(x^4+1))
sage: fq = 2*(xq^2-1)
sage: g = xq^2 + 2*xq + 1
sage: fq*g       # this is h, the public key
2*xq^4 + 4*xq^3 + xq + 3
sage: QR(2*x^4 + 4*x^3 + x + 3)
4*xbar^3 + xbar + 1
sage: b = xq^4+7*xq^2-2   # this is the blinding value
sage: p*b*(4*xq^3 + xq + 1)
2*xq^7 + 2*xq^5 + 3*xq^4 + 2*xq^3 + xq^2 + 4*xq + 4
sage: QR(2*x^7 + 2*x^5 + 3*x^4 + 2*x^3 + x^2 + 4*x + 4)
xbar^2 + 2*xbar + 1
```

More generally, let $H = \mathbb{Z}[x]/(x^N - c)$, so taking $c = -1$ give the modified case discussed above. If

$$h(x) = h_0 + h_1 x + \ldots + h_k x^{N-1},$$

we associate to h the quasi-circulant[5] matrix

$$Mat_N(h) = \begin{pmatrix} h_0 & ch_{N-1} & \ldots & ch_1 \\ h_1 & h_0 & \ldots & ch_2 \\ \vdots & & & \vdots \\ h_{N-1} & h_{N-2} & \ldots & h_0 \end{pmatrix}$$

and define $vec_N(h)$ to be the first column of $Mat_N(h)$:

$$vec_N(h) = (h_0, \ldots, h_{N-1}).$$

In the ring $H = \mathbb{Z}[x]/(x^N + c)$, we have

$$vec_N(bh) = Mat_N(h) vec_N(b). \tag{9}$$

In general,

$$(\sum_{i=0}^{N-1} b_i x^i)(\sum_{j=0}^{N-1} h_j x^j) = (\sum_{j=0}^{N-1} p_j x^j,$$

where

$$p_k = \sum_{i,j \;\; i+j \equiv k \pmod N} b_i h_j \epsilon_{i,j},$$

where

$$\epsilon_{i,j} = \begin{cases} 1, & i+j \leq N-1, \\ c, & i+j > N-1. \end{cases}$$

[5]All the upper diagonal entries have been multiplied by c but otherwise, the ith row is the right cyclic shift of the $i-1$st row.

1.6.4 Application to LFSRs

For some of the material below, we follow Chapter 2 in Klein [Kl13].

Stream ciphers

Pseudo-random number generators have been used for a long time as a source of stream ciphers.

S. Golomb gives a list of three statistical properties a sequence of numbers $\mathbf{a} = \{a_n\}_{n=0}^{\infty}$, $a_n \in \{0, 1\}$, should display to be considered "random". Define the *autocorrelation* of $\mathbf{a}$ to be

$$C(s) = C(s, \mathbf{a}) = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N} (-1)^{a_n + a_{n+s}}.$$

In the case where $\mathbf{a}$ is periodic with period P then this reduces to

$$C(s) = \frac{1}{P} \sum_{n=0}^{P-1} (-1)^{a_n + a_{n+s}}.$$

Assume $\mathbf{a}$ is periodic with period P.

balance: $|\sum_{n=0}^{P-1} (-1)^{a_n}| \leq 1$.

low autocorrelation:

$$C(k) = \begin{cases} 1, & k = 0, \\ \epsilon, & k \neq 0. \end{cases}$$

(For sequences satisfying these first two properties, it is known that $\epsilon = -1/P$ must hold.)

proportional runs property: In each period, half the runs have length 1, one-fourth have length 2, etc. Moveover, there are as many runs of 1's as there are of 0's.

Definition 50. A *general feedback shift register* is a map $f : GF(q)^d \to GF(q)^d$ of the form

$$f(x_0, ..., x_{n-1}) = (x_1, x_2, ..., x_n),$$
$$x_n = F(x_0, ..., x_{n-1}),$$

where $C : FF(q)^d \to GF(q)$ is a given function. When F is of the form

$$F(x_0, ..., x_{n-1}) = a_0 x_0 + ... + a_{n-1} x_{n-1},$$

for some given constants $a_i \in GF(q)$, the map is called a *linear feedback shift register (LFSR).*

Example 51. *Let*

$$f(x) = a_0 + a_1 x + ... + a_n x^n + ...,$$

$$g(x) = b_0 + b_1 x + ... + b_n x^n + ...,$$

be given polynomials in $GF(2)[x]$ *and let*

$$h(x) = \frac{f(x)}{g(x)} = c_0 + c_1 x + ... + c_n x^n + ... \ .$$

We can compute a recursion formula which allows us to rapidly compute the coefficients of $h(x)$ *(take* $f(x) = 1$*):*

$$c_n = \sum_{i=1}^{n} \frac{-b_i}{b_0} c_{n-i}.$$

This is a linear feedback shift register sequence.

For instance, if

$$f(x) = 1, \qquad g(x) = x^4 + x + 1,$$

then

$$h(x) = 1 + x + x^2 + x^3 + x^5 + x^7 + x^8 + ... \ .$$

The coefficients of h *are*

$$1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1,$$
$$1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, ... \ .$$

The sequence of $0, 1$*'s is periodic with period* $P = 2^4 - 1 = 15$ *and satisfies Golomb's three randomness conditions.*

More generally, the situation is described by the following fact.

Theorem 52. *If* $\mathbf{c} = \{c_n\}_{n=1}^{\infty}$ *are the coefficients of* $f(x)/g(x)$*, where* $f, g \in GF(q)[x]$ *and* $g(x)$ *is an irreducible polynomial with* x *primitive (mod* $g(x)$*) (i.e.,* x *generates* $(GF(q)[x]/g(x)GF(q)[x])^{\times}$*). Then* $\mathbf{c}$ *is periodic with period* $P = q^d - 1$ *(where* d *is the degree of* $g(x)$*). If* $q = 2$ *then the sequence satisfies Golomb's randomness conditions.*

We shall define primiative later (see Definition 60).

Write an LFSR $\{a_k\}_{k=0}^{\infty}$ as

$$a_{d+i} = \sum_{j=0}^{d-1} c_j a_{i+j}, \quad i = 0, 1, \ldots . \tag{10}$$

Let

$$A(z) = \sum_{k=0}^{\infty} a_k z^k,$$

so

$$\sum_{i=0}^{\infty} a_{d+i} z^i = \sum_{j=0}^{d-1} c_j \sum_{i=0}^{\infty} a_{i+j} z^i,$$

and therefore,

$$\sum_{i=0}^{\infty} a_{d+i} z^{i+d} = \sum_{j=0}^{d-1} c_j z^{d-j} \sum_{i=0}^{\infty} a_{i+j} z^{i+j}.$$

Note that $\sum_{i=0}^{\infty} a_{d+i} z^{i+d}$ and $\sum_{i=0}^{\infty} a_{i+j} z^{i+j}$ each differ from $A(z)$ by a polynomial. Therefore,

$$A(z) - \sum_{j=0}^{d-1} c_j z^{d-j} A(z) = f(z),$$

for some polynomial $f(z)$ of degree at most $d-1$. This proves the following fact.

Lemma 53. *The generating function* $A(z)$ *for the LFSR* $\{a_k\}_{k=0}^{\infty}$ *satisfies*

$$A(z) = \frac{f(z)}{g(z)},$$

where

$$g(z) = 1 - \sum_{j=0}^{d-1} c_j z^{d-j}.$$

The polynomial $g(z)$ in the Lemma above is called the *connection* polynomial. If

$$p(z) = b_0 + b_1 z + \ldots + b_n z^n$$

is any polynomial, we call

$$p^*(z) = z^n p(1/z) = b_0 z^n + b_1 z^{n-1} + \ldots + b_n,$$

the *reciprocal* polynomial. The reciprocal of the connection polynomial is called the *feedback* polynomial:

$$g^*(z) = z^n - \sum_{j=0}^{d-1} c_j z^j.$$

Assume the feedback polynomial has no multiple roots (more on this assumption later) and let $\xi_1, \ldots, \xi_n$ be the different zeros of $g^*(z)$. These all belong to some finite extension of $GF(q)$, namely $GF(q^n)$. Then we get a partial fraction decomposition

$$A(z) = \frac{f(z)}{g(z)} = \sum_{j=1}^{n} \frac{d_j}{1 - \xi_j z},$$

for some numbers d_j belonging to $GF(q^n)$. Using the geometric series expansion for $\frac{1}{1-\xi_j z}$, we have

$$A(z) = \sum_{j=1}^{n} \frac{d_j}{1 - \xi_j z} = \sum_{j=1}^{n} \sum_{k=0}^{\infty} d_j \xi_j^k z^k = \sum_{k=0}^{\infty} (\sum_{j=1}^{n} d_j \xi_j^k) z^k.$$

This gives us the formula for the kth term in the LFSR sequence:

$$a_k = \sum_{j=1}^{n} d_j \xi_j^k. \tag{11}$$

Let's go back and discuss the assumption that the feedback polynomial has no multiple roots. Over finite fields (and over fields of characteristic 0, such as $\mathbb{Q}$), it turns out that every irreducible polynomial has distinct roots. Therefore, we have the following fact.

Lemma 54. *If the LFSR $\{a_k\}_{k=0}^{\infty}$, $a_i \in GF(q)$, has an irreducible feedback polynomial with roots $\xi_1, \ldots, \xi_d$ then (11) holds for some d_j belonging to the finite extension $GF(q^d)$.*

In fact, a bit more is true. We can show that, moreover, the d_j are uniquely determined by the a_k. This is true because the matrix $\Xi = (\xi_j^k)_{i \le j, k \le d}$ is a Van der Monde matrix, hence is invertible. Therefore, the equations (11), $1 \le k \le d$, can be converted into an $d \times d$ system of linear equations for which the d_j are uniquely determined by the a_k.

We need more information on finite fields to prove more.

2 Structure of finite fields

We'd like to prove that there is a simple formula for the terms in a LFSR, such as in the following result.

Theorem 55. *Let (a_k) be a LFSR sequence as in (10) with an irreducible feedback polynomial of degree d. If ξ is some zero of the feedback polynomial then*

$$a_k = Tr_{GF(q^n)/GF(q)}(\alpha \xi^k),$$

for some $\alpha \in GF(q^d)$.

2.1 Cyclic multiplicative group

Before we can prove Theorem 55, we need to know more facts about finite fields (such as the definition of $Tr_{GF(q^n)/GF(q)}$).

Lemma 56. *Let $R = GF(q)$, where $q > 1$ is a given prime power. We have*

$$a^{q-1} = 1,$$

for all non-zero $a \in GF(q)$. Consequently,

$$\prod_{a \in GF(q)^\times} (x - a) = x^{q-1} - 1.$$

Proof. We use the method in the proof of Lemma 6.... □

For a (multiplicative) group G and an element $g \in G$, let $ord_G(g)$ denote the smallest integer $d > 0$ for which $g^d = 1$, if it exists.

Lemma 57. *If $y \in G$ has order d, a power y^k has order d if and only if k is relatively prime to d.*

Proof. Let $y \in G$ have order d and y^k have order d. If $gcd(k, d) = h > 1$ then $(y^k)^{d/h} = 1$, a contradiction.

Conversely, let $gcd(k, d) = 1$ and y have order d. If $(y^k)^m = 1$, for some $0 < m < d$, then let $km = qd + r$, for some $0 < r < d$. (Note $r = 0$ is impossible since $gcd(k, d) = 1$.) Then $1 = (y^k)^m = y^{km} = (y^d)^q y^r = y^r$. This contradicts $ord_G(y) = d$. □

Lemma 58. *Let G a finite group with n elements. If for every $d|n$ we have*

$$|\{g \in G \mid g^d = 1\}| \leq d,$$

then G is cyclic.

This lemma is well-known but the proof below is very short and clever. It was found in a math.stackexchange.com thread authored by Andrea Petracci.

Proof. Fix $d|n$ and consider the set

$$G_d = \{x \in G \mid org_G(x) = d\},$$

made up of elements of G with order d. Suppose that $G_d \neq \emptyset$, so there exists $y \in G_d$. It's clear that

$$\langle y \rangle = \{1, y, y^2, \ldots, y^{d-1}\} \subset \{g \in G \mid g^d = 1\}.$$

Since y has order d, the subgroup $\langle y \rangle$ has cardinality d so

$$d = |\langle y \rangle| \subset |\{g \in G \mid g^d = 1\}| \leq d,$$

by hypothesis. Therefore, $\langle y \rangle = \{g \in G \mid g^d = 1\}$. Since this holds for each $y \in G_d$, G_d is the set of generators of the cyclic group $\{g \in G \mid g^d = 1\}$ of order d. By the Lemma above, a power y^k has order d if and only if k is relatively prime to d. Therefore, G_d has $\phi(d)$ elements[6], namely those y^k with $gcn(k, n) = 1$.

[6]Recall, ϕ is the Euler phi-function, which counts integers relatively prime to a given integer.

We have proved that G_d is empty or has cardinality $\phi(d)$, for every $d|n$. So we have:

$$n = |G| = \sum_{d|n} |G_d| \leq \sum_{d|n} \phi(d) = n.$$

Therefore $|G_d| = \phi(d)$ for every $d|n$. In particular $G_n \neq \emptyset$.

This proves that G is cyclic. □

While the group of units of $\mathbb{Z}/n\mathbb{Z}$, i.e., $(\mathbb{Z}/n\mathbb{Z})^\times$, is not cyclic in general, the group of units of $GF(q)$, i.e., $GF(q)^\times$, is always cyclic.

Proposition 59. *$GF(q)^\times$ is cyclic.*

Definition 60. A generator of the cyclic group $GF(q)^\times$ is called a *primitive element.*

In particular, if $f(x) \in GF(q)[x]$ is an irreducible polynomial of degree d then the representative $\overline{x} = x + (f(x)) = x + f(x)GF(q)[x]$ is primitive in $GF(q^d) = GF(q)[x]/f(x)GF(q)[x]$ if and only if $x^j \pmod{f(x)}$ is non-zero for all j with $1 \leq j \leq q^d - 2$.

Proof. Let $G = GF(q)^\times$ and we ask for a simple bound on the subset

$$\{g \in G \mid g^d = 1\}.$$

For any integer $d \geq 1$, the polynomial $x^d - 1$ can have at most d roots (in an extension field of $GF(q)$, and hence in G), so

$$|\{x \in G \mid x^d = 1\}| \leq d.$$

Therefore, Lemma 58 applies and so G is cyclic.

□

2.2 Extension fields

Suppose that we have two finite fields, say $GF(q_1)$ and $GF(q_2)$. If one is contained in the other, is there a relationship between q_1 and q_2? Suppose

$$GF(q_1) \subset GF(q_2),$$

that is, the field operations ($+$ and $\cdot$) on $GF(q_2)$ restricted to $GF(q_1)$ give the field operations ($+$ and $\cdot$) on $GF(q_1)$. This implies that $V = GF(q_2)$ is a vector space over $F = GF(q_1)$, i.e., satisfies the following definition.

Definition 61. Let F be a field and V be a set with operations $+ : V \times V \to V$, written $+ : (\mathbf{u}, \mathbf{v}) \longmapsto \mathbf{u} + \mathbf{v}$, and $\cdot : F \times V \to V$, $+ : (a, \mathbf{v}) \longmapsto a \cdot \mathbf{v}$. V is called a *vector space over* F (or an F*-vector space*) if the following properties hold. For all $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$ and $a, b \in F$,

- $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$ (commutativity)
- $(\mathbf{u} + \mathbf{v}) + \mathbf{w} = \mathbf{u} + (\mathbf{v} + \mathbf{w})$ (associativity)
- the vector $\mathbf{0} = (0, 0, ..., 0) \in V$ satisfies $\mathbf{u} + \mathbf{0} = \mathbf{u}$ (the zero vector $\mathbf{0}$ is the additive identity),
- for each $\mathbf{v} \in V$ the element $(-1)\mathbf{v} = -\mathbf{v} \in V$ satisfies $\mathbf{v} + (-\mathbf{v}) = \mathbf{0}$ (each element $\mathbf{v}$ has an additive inverse $-\mathbf{v}$)
- $(a + b)\mathbf{v} = a\mathbf{v} + b\mathbf{v}$ and $a(\mathbf{v} + \mathbf{w}) = a\mathbf{v} + a\mathbf{w}$ (distributive laws)
- $(ab)\mathbf{v} = a(b\mathbf{v})$
- $1 \cdot \mathbf{v} = \mathbf{v}$.

Suppose V is has dimension d, as an F-vector space. Then, as a vector space, $V \cong F^d$, which implies $|V| = q^d$. This proves the following fact.

Lemma 62. *If $GF(q_1)$ and $GF(q_2)$ are finite fields and if*

$$GF(q_1) \subset GF(q_2),$$

then there is a $d \geq 1$ such that $q_2 = q_1^d$.

If F_1 and F_2 are fields and if the field operations ($+$ and $\cdot$) on F_2 restricted to F_1 give the field operations ($+$ and $\cdot$) on F_1, then we write

$$F_1 \subset F_2 \quad \text{or} \quad F_2/F_1 \quad \text{or} \quad F_1 \backslash F_2,$$

and we call F_2 a *field extension* of F_1.

Example 63. *For example, define*

$$GF(9) = \{0, 1, 2, x, 2x, x + 1, x + 2, 2x + 1, 2x + 2\},$$

with addition and multiplication $\pmod{x^2 + 1}$ *in* $GF(3)[x]$. *In this case,* $GF(3)$ *and* $GF(9)$ *are finite fields, each of characteristic* 3, *and*

$$GF(3) \subset GF(9).$$

However, there is no way to define $GF(27)$ in such a way that

$$GF(9) \subset GF(27).$$

To prove Theorem 65 given later, we need the following analog of Bezuot's lemma.

Lemma 64. *(Bezout's Lemma for polynomials) Let F be a field. For any polynomials $a(x)$ and $b(x)$ in $F[x]$, there are polynomials $u(x)$ and $v(x)$ satisfying*

$$a(x)u(x) + b(x)v(x) = \gcd(a(x), b(x)).$$

The proof of this lemma is the same as that of Lemma 4, but is included for completeness.

Proof. Consider the ideal

$$(a(x), b(x)) = \{r(x)a(x) + s(x)b(x) \mid r \in F[x],\ s \in F[x]\}.$$

Since $d(x) = \gcd(a(x), b(x))$ divides $a(x)$ and $b(x)$, this ideal $(a(x), b(x))$ must be contained in the ideal

$$(d(x)) = \{t(x)d(x) \mid t(x) \in F[x]\},$$

i.e., $(a(x), b(x)) \subset (d(x))$.

Suppose now $(d(x)) \neq (a(x), b(x))$. Let $n(x)$ be the polynomial of smallest non-zero degree such that

$$n(x) \in (a(x), b(x)),$$

written $n(x) = a(x)u(x)+b(x)v(x)$. By the integer "long division" algorithm, there is a remainder $r(x)$ (of smaller degree than $d(x)$) and a quotient $q(x)$ such that $n(x) = q(x)d(x) + r(x)$. But $r(x) = n(x) - q(x)d(x) \in (d(x))$, so either $r(x) = 0$ (so $(d(x)) \neq (a(x), b(x))$ is false) or $r(x)$ is a multiple of $d(x)$ (so $r(x)$ is smaller degree than $d(x)$ is false). This is a contradiction. Therefore, $(d(x)) = (a(x), b(x))$. $\square$

Of more immediate concern for us is the *algorithm to compute the inverse of* $c(x) \pmod{p}(x)$: Assume $gcd(c(x), p(x)) = 1$ and compute $u(x), v(x)$ such that $c(x)u(x) + p(x)v(x) = 1$ via Bezuot's Lemma for polynomials. We have $u(x) \pmod{p(x)} = c(x)^{-1} \pmod{p(x)}$.

Next, we prove the following theorem.

Theorem 65. *The quotient ring* $K = GF(q)[x]/(p(x))$ *is an extension field of* $GF(q)$ *with* q^d *elements if and only if* $p(x)$ *is an irreducible polynomial over* $GF(q)$ *of degree* d.

Proof. First, if $p(x)$ is a polynomial of degree d then $GF(q)[x]/(p(x))$ can be represented by all the polynomials of degree $d-1$ or less:

$$GF(q)[x]/(p(x)) = \{a_0 + a_1 x + \ldots a_k x^k \mid k < d, a_i \in GF(q)\},$$

as sets. There are q^d elements in this set.

Second, if $p(x)$ is not irreducible over $GF(q)$, say $p(x) = f(x)g(x)$, then $GF(q)[x]/(p(x))$ has zero divisors (namely, $f(x)g(x) = 0$ in $GF(q)[x]/(p(x))$), so it cannot be a field.

These last two paragraphs prove that "$K = GF(q)[x]/(p(x))$ is an extension field of $GF(q)$ with q^d elements only if $p(x)$ is an irreducible polynomial over $GF(q)$ of degree d" is true.

To prove the other direction, assume $p(x)$ is an irreducible polynomial over $GF(q)$ of degree d. Let $c(x)$ be any polynomial of degree k, $0 < k < d$. We shall show that $c(x)$ is invertible in $GF(q)[x]/(p(x))$. Since $p(x)$ is irreducible, $c(x)$ and $p(x)$ are relatively prime. By Bezout's Lemma for polynomials, we can compute $u(x), v(x)$ such that $c(x)u(x) + p(x)v(x) = 1$. Therefore, we have $u(x) \pmod{p(x)} = c(x)^{-1} \pmod{p(x)}$. Since every non-zero element of $GF(q)[x]/(p(x))$ is invertible, it must be a field. □

Let $f(x) \in GF(q)[x]$ be an irreducible polynomial of degree d and let $GF(q^d) = GF(q)[x]/f(x)GF(q)[x]$.

Note that $x^q - 1 = 0$ has all its roots in $GF(q)$ (see Lemma 56). More generally, let k be an integer with $1 < k < d$. Again, by Lemma 56, $x^{q^k} - 1 = 0$ has all its roots in $GF(q^k)$. In particular, if $a \in GF(q^d)$ is a root of $x^{q^k} - 1 = 0$ then $a \in GF(q^k)$.

Lemma 66. *The roots of* $f(x) \in GF(q)[x]$ *can be represented by* $\overline{x^{q^j}} = x^{q^j} + (f(x)) = x^{q^j} + f(x)GF(q)[x]$ *in* $GF(q^d) = GF(q)[x]/f(x)GF(q)[x]$, *for* $0 \leq j \leq d-1$.

Proof. Let $\mathcal{F}(a) = a^q$, for $a \in GF(q^d)$. First, we claim that this map (called a *Frobenius map*) defines a field automorphism

$$\mathcal{F} : GF(q^d) \to GF(q^d),$$

for which $\mathcal{F}(a) = a$ if and only if $a \in GF(q)$. To prove this, we must show that, for all $a, b \in GF(q^d)$,

- $\mathcal{F}(a + b) = \mathcal{F}(a) + \mathcal{F}(b)$, and
- $\mathcal{F}(a \cdot b) = \mathcal{F}(a) \cdot \mathcal{F}(b)$.

The second property is obvious. To verify the first property, note q is a power of some prime number p. Now, expand out

$$\mathcal{F}(a + b) = (a + b)^q = \sum_{i=0}^{q} \left(\begin{array}{c} q \\ i \end{array} \right) a^i b^{q-i},$$

which is equal to $\mathcal{F}(a) \cdot \mathcal{F}(b)$ plus a bunch of terms having a binomial coefficient $\left(\begin{array}{c} q \\ i \end{array} \right) \neq 1$. Writing out each binomial coefficient in terms of factorials, it's easy to see that each of these binomial coefficients is divisible by p. But any multiple of p is 0 in $GF(q^d)$. This proves the first property.

Therefore, the Frobenius map defines a field automorphism. Moreover, by the discussion preceeding the statement of the lemma, we see that, for any $a \in GF(q^d)$, we have $\mathcal{F}(a) = a$ if and only if $a \in GF(q)$. More generally, if $1 \leq k \leq d$, for any $a \in GF(q^d)$, we have $\mathcal{F}^k(a) = a$ if and only if $a \in GF(q^k)$.

Consider now the roots of

$$f(x) = a_0 + a_1 x + \ldots + a_d x^d.$$

Since the coefficients a_i are in $GF(q)$, we have

$$\mathcal{F}(f(x)) = \mathcal{F}(a_0 + a_1 x + \ldots + a_d x^d) = a_0 + a_1 x^q + \ldots + a_d x^{dq}$$

$$= a_0 + a_1 \mathcal{F}(x) + \ldots + a_d \mathcal{F}(x^d) = f(\mathcal{F}(x)).$$

for any $x \in GF(q^q)$. In particular, $\mathcal{F}$ sends each root of $f(x)$ to another one. There are d roots of $f(x) = 0$ and there are d distinct elements in the list x, $\mathcal{F}(x) = x^q$, $\mathcal{F}^2(x) = x^{q^2}$, $\ldots$, $\mathcal{F}^{d-1}(x) = x^{q^{d-1}}$. $\square$

Example 67. *Consider the field extension*

$$GF(8) = GF(2)][x]/(x^3 + x + 1)GF(2)[x],$$

defined by the irreducible primitive polynomial $x^3 + x + 1$.

If

$$a = 0, b = x, c = x^2, d = x^3 = x + 1, e = x^4 = x^2 + x,$$
$$f = x^5 = x^2 + x + 1, g = x^6 = x^2 + 1, h = x^7 = 1,$$

then the multiplication table is given by

$*$	a	b	c	d	e	f	g	h
a	a	a	a	a	a	a	a	a
b	a	c	d	e	f	g	h	b
c	a	d	e	f	g	h	b	c
d	a	e	f	g	h	b	c	d
e	a	f	g	h	b	c	d	e
f	a	g	h	b	c	d	e	f
g	a	h	b	c	d	e	f	g
h	a	b	c	d	e	f	g	h

and the addition table by

$+$	a	b	c	d	e	f	g	h
a	a	b	c	d	e	f	g	h
b	b	a	e	h	c	g	f	d
c	c	e	a	f	b	d	h	g
d	d	h	f	a	g	c	e	b
e	e	c	b	g	a	h	d	f
f	f	g	d	c	h	a	b	e
g	g	f	h	e	d	b	a	c
h	h	d	g	b	f	e	c	a

It is not hard to check that $a, a^2, a^4 = a^2 + a$ *are roots of* $x^3 + x + 1 = 0$ *in* $GF(8)$.

2.3 Back to the LFSR

We finish the proof of Theorem 55.

Recall, (a_k) is a LFSR sequence as in (10) with an irreducible feedback polynomial, $f(x)$, of degree d. We must verify that, for each root ξ of f, there is an $\alpha \in GF(q^d)$ such that

$$a_k = Tr_{GF(q^d)/GF(q)}(\alpha \xi^k). \tag{12}$$

We've already established that (11) is true, i.e., that $a_k = \sum_{j=1}^{n} d_j \xi_j^k$, for some *unique* $d_j \in GF(q^d)$.

Since the feedback polynomial is irreducible, its zeros have the form $Fr^i(\xi)$, $0 \leq i \leq d-1$, where $\xi = \xi_1$ and Fr is the Frobenius automorphism of $GF(q^d)$ ($Fr : a \to a^q$). Since $a_j \in GF(q)$, we have $Fr(a_j) = a_j$, for all j. Therefore, for each i,

$$a_k = \sum_{j=1}^{n} d_j Fr^j(\xi)^k = \sum_{j=1}^{n} Fr^i(d_j) Fr^{i+j}(\xi^k).$$

Since the sum is unique, $d_2 = Fr(d_1)$, $d_3 = Fr(d_2) = Fr^2(d_1)$, ... This implies (12), which completes the proof of Theorem 55.

A binary LFSR sequence which has key length d has period at most 2^2-1. Those of maximal period are also called *m-sequences*. We say $b \in GF(2)^\ell$ *occurs* in (a_k) if there is a sequence of consecutive terms $a_m, a_{m+1}, \ldots, a_{m+\ell-1}$ which agrees with b. A sequence of consecutive ones is called a *block* and a sequence of consecutive zeros is called a *gap*. A *run* is either a block or a gap. A sequence of consecutive terms of length d, $a_m, a_{m+1}, \ldots, a_{m+d-1}$ is called a *state* of (a_k).

Assume $q = 2$ and that $P = 2^d - 1$ is the period of the LFSR sequence (a_k). For each $b \in GF(2)^\ell$, let $N(b)$ denote the number of times that b occurs in a period of (a_k).

In §1.6.4, we stated some statistical properties that Solomon Golomb formulated to quantify a pseudo-random sequence of 0s and 1s. These are reformulated below, following Klein [Kl13].

Definition 68. (G1) In every period, the sequence is well-balanced, i.e., the number of ones is nearly equal to the number of zeros. More precisely, we have

$|\sum_{k=0}^{P-1}(-1)^{a_k}| \leq 1$.

This is the *balanced property* or the *distribution test*.

(G2) Then for any k with $1 \leq k \leq d-1$, we have

$$|N(b) - N(b')| \leq 1,$$

for any $b, b' \in GF(2)^k$.

This is the *serial test*.

(G2') In each period, half the runs have length 1, one-fourth have length 2, and so on for all lengths $\leq d-1$. Moveover, there are nearly as many blocks of length k as there are gaps of length k.

This is the *proportional runs property*.

(G3)

$$C(k) = \begin{cases} 1, & k = 0, \\ \epsilon, & k \neq 0, \end{cases}$$

where $\epsilon = -1/P$.

This is the *low autocorrelation property* or *auto-correlation test*.

In the remainder of this section, We prove that there are lots of LFSR sequence which satisfy the properties above.

Lemma 69. *The states of any binary LFSR (a_k) with maximal period $P = 2^d - 1$ run through all elements of $GF(2)^d - \{\vec{0}\}$.*

Proof. Suppose not.

The states $(a_m, a_{m+1}, \ldots, a_{m+d-1})$ are contained in $GF(2)^d$. Moreover, no state can be $\vec{0}$, for otherwise the recursive equation defining (a_k) would force all the terms to be 0 from that point on. Therefore, $(a_m, a_{m+1}, \ldots, a_{m+d-1}) \in GF(2)^d - \{\vec{0}\}$, and so there are at most $2^d - 1$ possible states.

If some d-tuple does not occur as a state then the $2^d - 1$ subsequences $(a_m, a_{m+1}, \ldots, a_{m+d-1})$, $0 \leq m \leq P-1$, must contain a repetition. However, if a state is repeated, say $(a_\ell, a_{\ell+1}, \ldots, a_{\ell+d-1}) = (a_m, a_{m+1}, \ldots, a_{m+d-1})$, then the recursive equation defining (a_k) would force the period to be $m - \ell < P$. This is a contradiction.

□

Example 70. *Consider the LFSR sequence with key* $k = (1, 0, 0, 1)$ *defined by*

$$a_{n+4} = a_{n+3} + a_n,$$

with $a_0 = 1$, $a_1 = 1$, $a_2 = 0$, $a_3 = 1$. *The sequence* (a_k) *is*

$$1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, \ldots,$$

and has period $P = 15$. *The associated primitive polunomial is* $x^4 + x + 1$. *The first* 15 *states are*

$$(1, 1, 0, 1), (1, 0, 1, 0), (0, 1, 0, 1), (1, 0, 1, 1), (0, 1, 1, 0),$$

$$(1, 1, 0, 0), (1, 0, 0, 1), (0, 0, 1, 0), (0, 1, 0, 0), (1, 0, 0, 0),$$

$$(0, 0, 0, 1), (0, 0, 1, 1), (0, 1, 1, 1), (1, 1, 1, 1), (1, 1, 1, 0).$$

As Lemma 69 predicts, all possible non-zero 4*-tuples of* 0*s and* 1*s are achieved.*

We mention a few immediate consequences of Lemma 69. First, we *claim* that $N(b) = 2^{d-k}$ for all $b \in GF(2)^k$ with $b \neq \vec{0}$ and $k \leq d - 1$. Indeed, counting the times b has occured in a state is, because of Lemma 69, the same as counting the number of ways to complete b into an element of $GF(2)^d - \{\vec{0}\}$. This is 2^{d-k}, which proves the claim. Second, we *claim* that $N(b) = 2^{d-k} - 1$ for $b\vec{0} \in GF(2)^k$ and $k \leq d - 1$. Indeed, counting chithe number of ways to complete $b = \vec{0} \in GF(2)^k$ into an element of $GF(2)^d - \{\vec{0}\}$ gives $2^{d-k} - 1$ since the all-0 vector in $GF(2)^d$ is not allowed. This proves the second claim and establishes the following fact.

Lemma 71. *For all* $b \in GF(2)^k$ *with* $k \leq d - 1$, *we have*

$$N(b) = \begin{cases} 2^{d-k}, & b \neq \vec{0}, \\ 2^{d-k} - 1, & b = \vec{0}. \end{cases}$$

For example, this Lemma tells us (a) if $b = 1 \in GF(2)^1$ then the number of 1s in a period is $N(1) = 2^{d-1}$, (b) if $b = 0 \in GF(2)^1$ then the number of 0s in a period is $N(0) = 2^{d-1} - 1$. Therefore,

$$\sum_{k=0}^{P-1} (-1)^{a_k} = N(0) - N(1) = -1.$$

This establishes (G1) in Definition 68.

Lemma 71 also tells us, for any two $b, b' \in GF(2)^k$ that $N(b) = N(b')$ is either $2^{d-k} - 2^{d-k} = 0$, $2^{d-k} - (2^{d-k} - 1) = 1$ or $(2^{d-k} - 1) - 2^{d-k} = -1$. This establishes (G2) in Definition 68.

Furthermore, Lemma 71 gives us a quantitative sense for which the following is true: the number of runs have length $k-1$ is nearly half the number of runs of length k, for all lengths $k \leq d-1$. Moreover, there are nearly as many blocks of length k as there are gaps of length k. This establishes (G2') in Definition 68.

By Theorem 55, we have

$$a_k = Tr_{GF(2^d)/GF(2)}(\alpha_1 \xi^k),$$

for a primitive element $\xi \in GF(2^d)$ and some $\alpha_1 \in GF(2^d)$, and a

$$a_{k+s} = Tr_{GF(2^d)/GF(2)}(\alpha_2 \xi^k),$$

for some $\alpha_2 \in GF(2^d)$. Thus the sequence (a'_k) defined by

$$a'_k = a_k + a_{k+s} = Tr_{GF(2^d)/GF(2)}((\alpha_1 + \alpha_2)\xi^k),$$

is either all 0s (if $\alpha_1 + \alpha_2 = 0$) or else is a binary sequence of maximal period $2^d - 1$. It can only be all 0s if s is an integer multiple of the period $P = 2^d - 1$, or else the period of (a_k) would not be maximal. Since (G3) clearly holds when $s = 0$, we may assume s is not an integer multiple of the period. In this case, we can apply the argument above to show that the sequence (a'_k) satisfies (G1). But then we have

$$\sum_{k=0}^{P-1} (-1)^{a'_k} = -1.$$

This implies $C(s) = -1/P$ and therefore we have established (G3) in Definition 68.

Example 72. *If the key is $k = (1, 0, 1, 1)$ then the recursion equation defining the LFSR is*

$$a_{n+1} = k_3 a_n + k_2 a_{n-1} + k_1 a_{n-2} + k_0 a_{n-3} = a_n + a_{n-1} + a_{n-3}.$$

Taking as the initial values $a_0 = 0$, $a_1 = 1$, $a_2 = 1$, $a_3 = 1$, *we get for the sequence*

$$0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, \ldots \ .$$

Let $GF(8)$ *be the field extension of* $GF(2)$ *defined by the primitive polynomial* $f(x) = x^3 + x + 1$. *Recall*

$$trace_{GF(8)/GF(2)}(a) = a + Fr(a) + Fr^2(a) = a + a^2 + a^4,$$

for $a \in GF(8)$. *It can be shown that*

$$a_k = trace_{GF(8)/GF(2)}(\alpha \xi^k),$$

where $\alpha = \xi + \xi^2$ *and* ξ *is a root of* $f(x) = 0$.

3 Error-correcting codes

Roughly speaking a *code* is a system for converting a message into another form for the purpose of communicating the message more efficiently or reliably. A few examples are listed below.

- Semaphore, where a message is converted into a sequence of flag movements for communication across a distance.
- Morse code, where a message is converted into a sequence of dots and dashes for communication using telegraph. For example, LEG is

  $$\cdot - \cdot \cdot \quad \cdot \quad - - \cdot$$

  and RUN is

  $$\cdot - \cdot \quad \cdot \cdot - \quad - \cdot$$

  (They share the same "bit" pattern in Morse code, as do EARN and URN.)
- Marconi Telegraph Code, where a commonly used phrase is converted into a more compact 5-letter sequence.

A code could be used as a cipher, but most codes are not created with security in mind. For example, during the Prohibition Era, rumrunners used slightly modified telegraph codes to transmit shipment information and meeting places for ship-loads of alchohol. Such ciphers were routinely broken by Coast Guard cryptographers.

Some codes are designed for compression - to store digital data more compactly. Some codes are designed for reliability - to communicate information over a noisy channel, yet to correct the errors which arise.

3.1 The communication model

Consider a source sending messages through a noisy channel. The message sent will be regarded as a vector of length n whose entries are taken from a given finite field F (typically, $F = GF(2)$).

For simplicity, assume that the message being sent is a sequence of 0's and 1's. Assume that, due to noise, when a 0 is sent, the probability that a 1 is (incorrectly) received is p and the probability that a 0 is (correctly) received is $1-p$. The *error rate* p is a small positive number (such as $1/10000$) which represents the "noisiness" of the channel. Assume also that the error rate (and channel noise) is not dependent on the symbol sent: when a 1 is sent, the probability that a 1 is (correctly) received is $1-p$ and the probability that a 0 is (incorrectly) received is p.

3.2 Basic definitions

It was long believed that the theory of error-correcting codes was originated by Richard Hamming in the late 1940's, a mathematician who worked for Bell Telephone. However, recent work by mathematician and historian Chris Christensen and others, it is now known that the theory was developed by Lester Hill about 20 years earlier [CJT12]. Some specific examples of his codes actually arose earlier in various isolated connections - for example, statistical design theory and in soccer betting(!). Hamming's motivation was to program a computer to correct "bugs" which arose in punch-card programs. The overall goal behind the theory of error-correcting codes is to reliably enable digital communication.

Let $\mathbb{F} = GF(q)$ be any finite field.

A *(linear error-correcting) code C of length n over* $\mathbb{F}$ is a vector subspace

of $\mathbb{F}^n$ (provided with the standard basis[7]) and its elements are called *codewords*. When $\mathbb{F} = GF(2)$ it is called a *binary* code. These are the most important codes from the practical point of view. Think of the following scenario: You are sending an n-vector of 0's and 1's (the codeword) across a noisy channel to your friend. Your friend gets a corrupted version (the received word differs from the codeword in a certain number of error positions). Depending on how the code C was constructed and the number of errors made, it is possible that the original codeword can be recovered. This raises the natural question: given C, how many errors can be corrected? Stay tuned...

A code of length n and dimension k (as a vector space over $\mathbb{F}$) is called an $[n, k]$-*code*. In abstract terms, an $[n, k]$-code is given by a short exact sequence[8]

$$0 \to \mathbb{F}^k \xrightarrow{G} \mathbb{F}^n \xrightarrow{H} \mathbb{F}^{n-k} \to 0. \tag{13}$$

We identify C with the image of G.

Example 73. The matrix $G = (1, 1, 1)$ defines a map $G : GF(2) \to GF(2)^3$. The image is

$$C = Im(G) = \{(0, 0, 0), (1, 1, 1)\}.$$

The matrix

$$H = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

defines a map $H : GF(2)^3 \to GF(2)^2$. It is not hard to check $G \cdot H = 0$.

The function

$$G : \mathbb{F}^k \to C,$$

$$\vec{m} \longmapsto \vec{m}G,$$

[7]It is important that the code be provided with a fixed basis which never changes. This is because the minimum distance function is not invariant under a change of basis. However, the minimum distance is one quantity used to measure how "good" a code is, from the practical point of view.

[8]"Short exact" is a compact way of specifying the following three conditions at once: (1) the first map G is injective, i.e., G is a full-rank $k \times n$ matrix, (2) the second map H is surjective, and (3) image(G) = kernel(H).

is called the *encoder*. Since the sequence (13) is exact, a vector $\vec{v} \in \mathbb{F}^n$ is a codeword if and only if $H(\vec{v}) = 0$. If $\mathbb{F}^n$ is given the usual standard vector space basis then the matrix of G is a *generating matrix* of C and the matrix of H is a *check matrix* of C. In other words,

$$\begin{aligned} C &= \{\vec{c} \mid \vec{c} = \vec{m}G, \text{ some } \vec{m} \in \mathbb{F}^k\} \\ &= \{\vec{c} \in \mathbb{F}^n \mid H\vec{c} = \vec{0}\}. \end{aligned}$$

When G has the block matrix form

$$G = (I_k \mid A),$$

where I_k denotes the $k \times k$ identity matrix and A is some $k \times (n-k)$ matrix, then we say G is in *standard form*. By abuse of terminology, if this is the case then we say C is in *standard form* (or in *systemic form*).

Lemma 74. *Suppose C is a linear $[n, k, d]$ code over $GF(q)$ with generator matrix $G = (I_k, A)$, for some $k \times (n-k)$ matrix A. The matrix $H = (-A^T, I_{n-k})$ is a check matrix for C.*

Proof. The rank of H is obviously $n - k$. Therefore, it suffices to prove that the rows of G are all orthogonal to the rows of H, i.e., that $HG^T = 0$. Note the ith row of G is $(\vec{e}_i, \vec{A}_i)$, where $\vec{M}_i$ denotes the ith row of a matrix M. Likewise, the jth row of H is $(-\vec{A^T}_j, \vec{e}_j)$. We have

$$(\vec{e}_i, \vec{A}_i) \cdot (-\vec{A^T}_j, \vec{e}_j) = i^{th} \text{ coord of } -\vec{A^T}_j + j^{th} \text{ coord of } \vec{A}_i = -A_{ij} + A_{ij} = 0,$$

for $1 \leq i \leq k$, $1 \leq j \leq n - k$. □

Example 75. The matrix

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

is a generating matrix for a code in standard form.

The matrix G has rank k, so the row-reduced echelon form of G, call it G', has no rows equal to the zero vector. In fact, the standard basis vectors $\vec{e}_1$, ..., $\vec{e}_k$ of the column space $\mathbb{F}^k$ occur amongst k columns of those of G'.

The corresponding coordinates of C are called the *information coordinates* (or information bits, if C is binary) of C.

Aside: For a "random" $k \times k$ matrix with **real** entries, the "probability" that its rank is k is of course 1. This is because "generically" a square matrix with real entries is invertible. In the case of finite fields, this is not the case. For example, the probability that a "large random" $k \times k$ matrix with entries in $GF(2)$ is invertible is

$$\lim_{k\to\infty} \frac{(2^k-1)(2^k-2)...(2^k-2^{k-1})}{2^{k^2}} = \prod_{i=1}^{\infty}(1-2^{-i}) = 0.288... \ .$$

The *Hamming metric* is the function

$$d : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{R},$$

$$d(\vec{v},\vec{w}) = |\{i \mid v_i \neq w_i\}| = d(\vec{v}-\vec{w},\vec{0}).$$

The *Hamming weight* of a vector is simply its distance from the origin:

$$\mathbf{wt}\,(\vec{v}) = d(\vec{v},\vec{0}).$$

Question: How many vectors belong to the "shell' of radius r about the origin $\vec{0} \in GF(q)^r$?

Answer: $\begin{pmatrix} n \\ r \end{pmatrix}(q-1)^r$. Think about it! (Hint: "distance r" means that there are exactly r non-zero coordinates. The binomial coefficient describes the number of ways to choose these r coordinates.)

The *minimum distance of* C is defined to be the number

$$d(C) = \min_{\vec{c}\neq\vec{0}} d(\vec{c},\vec{0}).$$

(It is not hard to see that this is equal to the closest distance between any two distinct codewords in C.) An $[n,k]$-code with minimum distance d is called an $[n,k,d]$-*code*.

Lemma 76. *(Singleton bound) Every linear* $[n,k,d]$ *code* C *satisfies*

$$k + d \leq n + 1.$$

Note: this bound does not depend on the size of $\mathbb{F}$. A code C whose parameters satisfy $k + d = n + 1$ is called *maximum distance separable* or *MDS*. Such codes, when they exist, are in some sense best possible.

proof: Fix a basis of $\mathbb{F}_q^n$ and write all the codewords in this basis. Delete the first $d-1$ coordinates in each code word. Call this new code C'. Since C has minimum distance d, these codewords of C' are still distinct. There are therefore q^k of them. But there cannot be more than $q^{n-d+1} = |\mathbb{F}_q^{n-d+1}|$ of them. This gives the inequality. $\square$

The *rate* of the code is $R = k/n$ - this measures how much information the code can transmit. The *relative minimum distance* of the code is $\delta = d/n$ - this is directly related to how many errors can be corrected.

Lemma 77. *Let $C \subset \mathbb{F}^n$ be an $[n, k, d]$-code.*

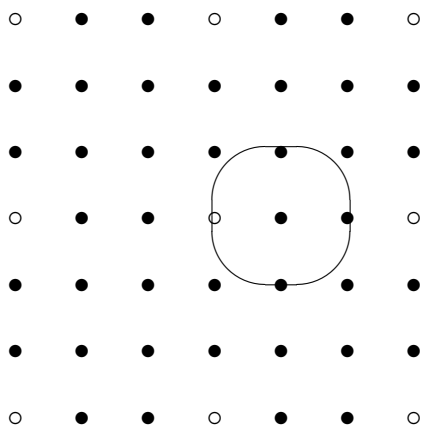
(a) If $\vec{v} \in \mathbb{F}^n$ is arbitrary and $0 < r \leq [\frac{d-1}{2}]$ then the "ball" about $\vec{v}$ with radius r,

$$B_r(\vec{v}) = \{\vec{w} \in \mathbb{F}^n \mid d(\vec{v}, \vec{w}) \leq r\}$$

contains at most one codeword in C.

(b) If $\vec{v} \in \mathbb{F}^n$ is a received vector then the nearest neighbor algorithm (below) returns a unique codeword $\vec{c} \in C$ closest to $\vec{v}$.

Part (b) will be verified after the statement of nearest neighbor algorithm. Part (a) follows easily from the fact that the Hamming metric is, in fact, a metric. Here is a picture of the idea.



Lemma 78. *(sphere-packing bound) For any code $C \subset \mathbb{F}^n$, we have*

$$|C| \sum_{i=0}^{t} \left(\begin{array}{c} n \\ i \end{array} \right) (q-1)^i \leq q^n,$$

where $t = [(d-1)/2]$.

proof: For each codeword of C, construct a ball of radius t about it. These are non-intersecting, by definition of d and the previous lemma. Each such ball has

$$\sum_{i=0}^{t} \left(\begin{array}{c} n \\ i \end{array} \right) (q-1)^i$$

elements. The result follows from the fact that $\cup_{\vec{c} \in C} B_t(\vec{c}) \subset \mathbb{F}^n$ and $|\mathbb{F}^n| = q^n$. □

Suppose (a) you sent $\vec{c} \in C$, (b) your friend received $\vec{v} \in \mathbb{F}^n$, (c) you know (or are very confident) that the number t of errors made is less than or equal to $[\frac{d-1}{2}]$. By Lemma 78 above, the "ball" about $\vec{v}$ of radius t contains a unique codeword. It must be $\vec{c}$, so your friend can recover what you sent (by searching though all the vectors in the ball and checking which one is in C) even though she/he only knows C and $\vec{v}$. This is called the *nearest neighbor decoding algorithm*:

1. Input: A received vector $\vec{v} \in \mathbb{F}^n$.

   Output: A codeword $\vec{c} \in C$ closest to $\vec{v}$.

2. Enumerate the elements of the ball $B_t(\vec{v})$ about the received word. Set $\vec{c}$ ="fail".

3. For each $\vec{w} \in B_t(\vec{v})$, check if $\vec{w} \in C$. If so, put $\vec{c} = \vec{w}$ and break to the next step; otherwise, discard $\vec{w}$ and move to the next element.

4. Return $\vec{c}$.

Note "fail" is not returned unless $t > [\frac{d-1}{2}]$, by the above lemma.

Definition 79. *We say that a linear* C *is* t*-error correcting if* $|B_t(\vec{w}) \cap C| \leq 1$.

Note that $t \leq [\frac{d-1}{2}]$ if and only if $d \geq 2t+1$.
The general goal in the theory is to optimize the following properties:

- the rate, $R = k/n$,
- the relative minimum distance, $\delta = d/n$,
- the speed at which a "good" encoder for the code can be implemented,
- the speed at which a "good" decoder for the code can be implemented.

There are (sometimes very technical) constraints on which these can be achieved, as we have seen with the Singleton bound and the sphere-packing bounds.

3.3 Binary hamming codes

This material can be found in many standard textbooks.

A Hamming code is a member of a family of binary error-correcting codes defined by Richard Hamming, a Bell telephone mathematician, in the 1940s.

Definition 80. *Let $r > 1$. The Hamming $[n, k, 3]$-code C is the linear code with*

$$n = 2^r - 1, \qquad k = 2^r - r - 1,$$

and parity check matrix H defined to be the matrix whose columns are all the (distinct) non-zero vectors in $GF(2)^r$. By Lemma 81, this code has minimum distance $d = 3$.

Lemma 81. *Every binary Hamming code C has minimum distance 3.*

Proof. Indeed, if C has a code wode of weight 1 then the parity check matrix H of C would have to have a column which consists of the zero vector, contradicting the definition of H. Likewise, if C has a code wode of weight 2 then the parity check matrix H of C would have to have two identical columns, contradicting the definition of H. Thus $d \geq 3$.

Since

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \text{ and } \begin{pmatrix} 1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

form three columns of the parity check matrix H of C - say the 1^{st}, 2^{nd}, and 3^{rd} columns - the vector $(1, 1, 1, 0, ..., 0)$ must be a code word. Thus $d \leq 3$. □

Example 82. We consider only two cases of the binary Hamming code construction.

(a) $r = 2$: The Hamming $[3, 1]$-code has parity check matrix

$$H = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

The matrix $G = (1, 1, 1)$ is a generating matrix.

(b) $r = 3$: The Hamming $[7, 4]$-code has parity check matrix

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

The matrix

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

is a generating matrix.

Example 83. Consider the Hamming $[7, 4]$ code in Example 82(b) above. The meaning of the statement that G is a generator matrix is that a vector

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}$$

is a codeword if and only if $\vec{x}$ is a linear combination of the rows of G. The meaning of the statement that H is a check matrix is $H\vec{x} = \vec{0}$, ie

$$x_1 + x_4 + x_6 + x_7 = 0, x_2 + x_4 + x_5 + x_7 = 0, x_3 + x_5 + x_6 + x_7 = 0.$$

This may be visualized via a Venn diagram (see Figure 1).
Decoding algorithm for the Hamming $[7, 4]$*-code*

Denote the received word by
$\vec{w} = (w_1, w_2, w_3, w_4, w_5, w_6, w_7)$.

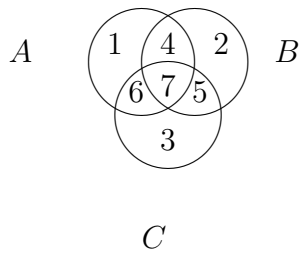


Figure 1: Venn diagram for the Hamming $[7, 4, 3]$ code

1. Put w_i in region i of the Venn diagram above, $i = 1, 2, ..., 7$.

2. Do parity checks on each of the circles A, B, and C.

parity failure region(s)	error position
none	none
A, B, and C	7
B and C	5
A and C	6
A and B	4
A	1
B	2
C	3

Here is some Sage code to illustrate this:

Sagemath

```
sage: C = codes.HammingCode(3,GF(2)); C
Linear code of length 7, dimension 4 over Finite Field of size 2
sage: C.minimum_distance()
3
sage: H = matrix(GF(2), 3, 7, [[1, 0, 0, 1, 0, 1, 1], [0, 1, 0, 1, 1, 0, 1],
  [0, 0, 1, 0, 1, 1, 1]])
sage: H
[1 0 0 1 0 1 1]
[0 1 0 1 1 0 1]
[0 0 1 0 1 1 1]
sage: C = codes.LinearCodeFromCheckMatrix(H)
sage: C.check_mat()
[1 0 0 1 0 1 1]
[0 1 0 1 1 0 1]
[0 0 1 0 1 1 1]
sage: C.minimum_distance()
3
sage: C.list()
[(0, 0, 0, 0, 0, 0, 0),
 (1, 0, 0, 0, 1, 0, 1),
 (0, 1, 0, 0, 0, 1, 1),
 (1, 1, 0, 0, 1, 1, 0),
 (0, 0, 1, 0, 1, 1, 1),
 (1, 0, 1, 0, 0, 1, 0),
 (0, 1, 1, 0, 1, 0, 0),
 (1, 1, 1, 0, 0, 0, 1),
 (0, 0, 0, 1, 1, 1, 0),
```

```
(1, 0, 0, 1, 0, 1, 1),
(0, 1, 0, 1, 1, 0, 1),
(1, 1, 0, 1, 0, 0, 0),
(0, 0, 1, 1, 0, 0, 1),
(1, 0, 1, 1, 1, 0, 0),
(0, 1, 1, 1, 0, 1, 0),
(1, 1, 1, 1, 1, 1, 1)]
```

3.4 Coset leaders and the covering radius

Let $C \subset GF(q)^n$ be a linear block code with generator matrix G and check matrix H.
Question: What is the largest radius r such that the balls of radius r centered about all the codewords,

$$B(c, r) = \{v \in GF(q)^n \mid d(c, v) \leq r\}$$

are disjoint?

Answer: $[(d-1)/2]$. By the above proof, we see that the triangle inequality will not allow two balls centered at neighboring codewords are disjoint if and only if they have radius $\leq [(d-1)/2]$.

The union of all these disjoint balls of radius $[(d-1)/2]$ centered at the codewords in C usually does not equal the entire space $V = GF(q)^n$. (When it does, C is called *perfect*)

How much larger do we have to make the radius so that the union of these balls does cover all of V? In other words, we want to answer the following question:
Question: What is the smallest radius ρ such that

$$\cup_{c \in C} B(c, \rho) = V?$$

Answer: At the present time, there are no simple general formulas for ρ and, in general, it is hard to even find good upper bounds on ρ. However, there is a sharp lower bound:

$$\rho \geq [(d-1)/2].$$

This radius ρ is called the *covering radius*.

If you think about it for a moment, you'll realize that the covering radius is the maximum value of

$$\text{dist}(v, C) = \min_{c \in C} \mathbf{wt}\,(v - c) = \min_{c \in C} \mathbf{wt}\,(v + c),$$

over all $v \in GF(q)^n$.

A *coset* is a subset of $GF(q)^n$ of the form $C + v$ for some $v \in GF(q)^n$. Equivalently, a coset is a pre-image of some y in $GF(q)^{n-k}$ under the check matrix $H : GF(q)^n \to GF(q)^{n-k}$. Let S be a coset of C. A *coset leader* of S is an element of S having smallest weight. The covering radius is, evidently, the highest weight of all the coset leaders of C.

Theorem 84. *The coset leaders of a Hamming code are those vectors of* $\mathbf{wt} \leq 1$.

Proof. Let the Hamming code be defined as a $[n, k, d]$ code as above where for some integer r, $n = 2^r - 1$, $k = 2^r - 1 - r$, and $d = r$. In the binary case, the size of the ambient space is $q^n = 2^n = |GF(q)^n|$ and the size of the code is $q^k = 2^k = |C|$. Thus, the size of any coset S of C is

$$|S| = |GF(q)^n| \diagup |C| = 2^{n-k} = 2^r = n + 1.$$

Claim: Each coset contains a coset leader of $\mathbf{wt} \leq 1$ and no coset contains more than one vector of $\mathbf{wt} \leq 1$. *Proof of claim:* Assume that $v_1 + C$ is one such coset with two distinct vectors w_1, w_2 of $\mathbf{wt} \leq 1$. Then,

$$w_1 = v_1 + c_1, w_2 = v_1 + c_2.$$

So,

$$w_1 - w_2 = c_1 - c_2 \in C.$$

And, since $\mathbf{wt}\,(w_1 - w_2) = 2$ and $d(C) = 3$ for a Hamming code, we have a contradiction. Also, by the Pigeonhole Principle, each coset contains exactly one vector v satisfying $\mathbf{wt}\,(v) = 1$. Thus, the claim holds, and this also proves the theorem. □

Example 85. For the Hamming $[7, 4, 3]$ code above, the cosets are

$$\{\{(0,0,0,0,0,0,0),(0,1,0,1,0,0,0),(1,1,1,0,1,0,0),(1,0,1,1,1,0,0),(0,0,1,0,0,1,0),(0,1,1,1,0,1,0),$$
$$(1,1,0,0,1,1,0),(1,0,0,1,1,1,0),(1,1,1,0,0,0,1),(1,0,1,1,0,0,1),(0,0,0,0,1,0,1),(0,1,0,1,1,0,1),$$
$$(1,1,0,0,0,1,1),(1,0,0,1,0,1,1),(0,0,1,0,1,1,1),(0,1,1,1,1,1,1)\},$$

$\{(1,0,0,0,0,0,0),(1,1,0,1,0,0,0),(0,1,1,0,1,0,0),(0,0,1,1,1,0,0),(1,0,1,0,0,1,0),(1,1,1,1,0,1,0),$
$(0,1,0,0,1,1,0),(0,0,0,1,1,1,0),(0,1,1,0,0,0,1),(0,0,1,1,0,0,1),(1,0,0,0,1,0,1),(1,1,0,1,1,0,1),$
$(0,1,0,0,0,1,1),(0,0,0,1,0,1,1),(1,0,1,0,1,1,1),(1,1,1,1,1,1,1)\},$
$\{(0,1,0,0,0,0,0),(0,0,0,1,0,0,0),(1,0,1,0,1,0,0),(1,1,1,1,1,0,0),(0,1,1,0,0,1,0),(0,0,1,1,0,1,0),$
$(1,0,0,0,1,1,0),(1,1,0,1,1,1,0),(1,0,1,0,0,0,1),(1,1,1,1,0,0,1),(0,1,0,0,1,0,1),(0,0,0,1,1,0,1),$
$(1,0,0,0,0,1,1),(1,1,0,1,0,1,1),(0,1,1,0,1,1,1),(0,0,1,1,1,1,1)\},$
$\{(1,1,0,0,0,0,0),(1,0,0,1,0,0,0),(0,0,1,0,1,0,0),(0,1,1,1,1,0,0),(1,1,1,0,0,1,0),(1,0,1,1,0,1,0),$
$(0,0,0,0,1,1,0),(0,1,0,1,1,1,0),(0,0,1,0,0,0,1),(0,1,1,1,0,0,1),(1,1,0,0,1,0,1),(1,0,0,1,1,0,1),$
$(0,0,0,0,0,1,1),(0,1,0,1,0,1,1),(1,1,1,0,1,1,1),(1,0,1,1,1,1,1)\},$
$\{(0,0,1,0,0,0,0),(0,1,1,1,0,0,0),(1,1,0,0,1,0,0),(1,0,0,1,1,0,0),(0,0,0,0,0,1,0),(0,1,0,1,0,1,0),$
$(1,1,1,0,1,1,0),(1,0,1,1,1,1,0),(1,1,0,0,0,0,1),(1,0,0,1,0,0,1),(0,0,1,0,1,0,1),(0,1,1,1,1,0,1),$
$(1,1,1,0,0,1,1),(1,0,1,1,0,1,1),(0,0,0,0,1,1,1),(0,1,0,1,1,1,1)\},$
$\{(1,0,1,0,0,0,0),(1,1,1,1,0,0,0),(0,1,0,0,1,0,0),(0,0,0,1,1,0,0),(1,0,0,0,0,1,0),(1,1,0,1,0,1,0),$
$(0,1,1,0,1,1,0),(0,0,1,1,1,1,0),(0,1,0,0,0,0,1),(0,0,0,1,0,0,1),(1,0,1,0,1,0,1),(1,1,1,1,1,0,1),$
$(0,1,1,0,0,1,1),(0,0,1,1,0,1,1),(1,0,0,0,1,1,1),(1,1,0,1,1,1,1)\},$
$\{(0,1,1,0,0,0,0),(0,0,1,1,0,0,0),(1,0,0,0,1,0,0),(1,1,0,1,1,0,0),(0,1,0,0,0,1,0),(0,0,0,1,0,1,0),$
$(1,0,1,0,1,1,0),(1,1,1,1,1,1,0),(1,0,0,0,0,0,1),(1,1,0,1,0,0,1),(0,1,1,0,1,0,1),(0,0,1,1,1,0,1),$
$(1,0,1,0,0,1,1),(1,1,1,1,0,1,1),(0,1,0,0,1,1,1),(0,0,0,1,1,1,1)\},$
$\{(1,1,1,0,0,0,0),(1,0,1,1,0,0,0),(0,0,0,0,1,0,0),(0,1,0,1,1,0,0),(1,1,0,0,0,1,0),(1,0,0,1,0,1,0),$
$(0,0,1,0,1,1,0),(0,1,1,1,1,1,0),(0,0,0,0,0,0,1),(0,1,0,1,0,0,1),(1,1,1,0,1,0,1),(1,0,1,1,1,0,1),$
$(0,0,1,0,0,1,1),(0,1,1,1,0,1,1),(1,1,0,0,1,1,1),(1,0,0,1,1,1,1)\}\}$

The coset leaders are:

$\{(0,0,0,0,0,0,0),(1,0,0,0,0,0,0),(0,1,0,0,0,0,0),(1,1,0,0,0,0,0),$

$(0,0,1,0,0,0,0),(1,0,1,0,0,0,0),(0,1,1,0,0,0,0),(1,1,1,0,0,0,0)\}$

Note that the largest weight of these coset leaders is 1, as the theorem above predicts.

Theorem 86. *Hamming codes are perfect.*

Proof. Since $d(C) = 3$ for Hamming codes, we desire to show that equality holds in

$$\rho = [(d-1)/2] = 1.$$

To attain a contradiction, assume

$$\rho = \max_{x \in GF(q)^n} d(x, C) > 1;$$

then for some $x \in GF(q)^n$, $d(x, C) > 1$. But by the previous theorem, the coset $x+C$ must contain a coset leader v satisfying $\mathbf{wt}\,(v) \leq 1$, a contradiction to the assumption that $d(x, C) > 1$. Thus, $\rho = 1$. $\square$

A final remark on coset leaders. These were introduced by David Slepian in the 1950s. Slepian also developed the following general decoding algorithm:

1. Input: A received vector $\vec{v} \in \mathbb{F}^n$.

   Output: A codeword $\vec{c} \in C$ closest to $\vec{v}$.

2. Compute the coset $S = \vec{v} + C$ of $\vec{v}$, the received word. Compute the coset leader of S, call it $\vec{e}$.

   Slepian's way to do this:

   - Precompute all the coset leaders $\vec{u}$ of C and tabulate all the values $(\vec{u}, H\vec{u})$.
   - Compute the *syndrome* of $\vec{v}$: $\vec{s} = H\vec{v}$. Search the 2nd coordinate of the tabulated pairs $(\vec{u}, H\vec{u})$ for this syndrome. Select the 1st coordinate from that pair, $\vec{u}$.
   - Let $\vec{e} = \vec{u}$.

3. Put $\vec{c} = \vec{v} - \vec{e}$.

4. Return $\vec{c}$.

3.5 Reed-Solomon codes as polynomial codes

Let $F = GF(q)$, where q is a prime power. Let $x_1, \ldots, x_n \in F$ be distinct elements of F (this forces $n \geq q$) and let $F[x]_k$ denote the k-dimensional vector space over F of all polynomials of degree less than k.

Formally, the vector space $C \subset F^n$ of codewords of the *Reed-Solomon code* is defined as follows:

$$C = \{(p(x_1), \ldots, p(x_n)) \mid p \in F[x]_k\}.$$

Example 87. *Let $k = 2$, $n = 4$, $q = 5$, with points $x_i \in \{0, 1, 2, 3\}$. Then C is a $[4, 2, 3]$ code over $GF(5)$, with generator matrix*

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{pmatrix}$$

and check matrix

$$\begin{pmatrix} 1 & 0 & 2 & 2 \\ 0 & 1 & 3 & 1 \end{pmatrix}.$$

The codewords of C are

$$(0,0,0,0), (1,1,1,1), (2,2,2,2), (3,3,3,3), (4,4,4,4),$$

$$(0,1,2,3), (1,2,3,4), (2,3,4,0), (3,4,0,1), (4,0,1,2),$$

$$(0,2,4,1), (1,3,0,2), (2,4,1,3), (3,0,2,4), (4,1,3,0),$$

$$(0,3,1,4), (1,4,2,0), (2,0,3,1), (3,1,4,2), (4,2,0,3),$$

$$(0,4,3,2), (1,0,4,3), (2,1,0,4), (3,2,1,0), (4,3,2,1).$$

Since any two distinct polynomials of degree less than k agree in at most $k-1$ points[9], this means that any two codewords of the Reed-Solomon code disagree in at least $n - (k-1) = n - k + 1$ coordinates. In particular, the minimum distance d of C satisfies $d \geq n - k + 1$. There are distinct polynomials in $F[x]_k$ that do agree in $k-1$ points[10], so the minimum distance of the Reed-Solomon code is exactly $n - k + 1$. This proves the following result.

Lemma 88. *The Reed-Solomon code defined above (where $n \leq q$) is an $[n, k, d]$-code over $GF(q)$.*

[9] This follows, for example, from the Lagrange interpolation formula, Lemma 45.

[10] Again, by the Lagrange interpolation formula.

Example 89. *Let $F = GF(11)$ and consider the vector space $F[x]_2$ of polynomials of degree ≤ 1. Pick $x_i = i \in F$, $i = 1, 2, 3, 4$. The associated Reed-Solomon code is the vector space*

$$C = \{a_0 + a_1, a_0 + 2a_1, a_0 + 3a_1, a_0 + 4a_1) \mid a_i \in F\}$$

$$= Span\{(1, 1, 1, 1), (1, 2, 3, 4)\} \subset F^4.$$

Therefore C has generator matrix

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \end{pmatrix}.$$

3.6 Cyclic codes as polynomial codes

In this section, we discuss a class of linear error-correcting block codes called cyclic codes. We shall see that these may be regarded as ideals in the quotient ring $GF(q)[x]/(x^n - 1)$.

Let

$$R_n = GF(q)[x]/(x^n - 1) = \{a_0 + a_1 x + \ldots + a_{n-1}x^{n-1} \mid a_i \in GF(q)\},$$

and let

$$\phi : GF(q)^n \to R_n$$

$$(a_0, a_1, \ldots, a_{n-1}) \mapsto a_0 + a_1 x + \ldots + a_{n-1}x^{n-1}$$

denote the correspondence between polynomials and their coefficients.

Lemma 90. *The map ϕ is a one-to-one onto isomorphism of $GF(q)$-vector spaces.*

Proof. It is clearly one-to-one and onto. It remains to show it preserves the $GF(q)$-vector space operations. If

$$\vec{a} = (a_0, a_1, \ldots, a_{n-1}), \quad \vec{b} = (b_0, b_1, \ldots, b_{n-1}),$$

then

$$\phi(\vec{a} + \vec{b}) = (a_0 + b_0) + (a_1 + b_1)x + \ldots + (a_{n-1} + b_{n-1})x^{n-1}$$

$$= (a_0 + a_1 x + \ldots + a_{n-1}x^{n-1}) + (b_0 + b_1 x + \ldots + b_{n-1}x^{n-1}) = \phi(\vec{a}) + \phi(\vec{b}),$$

so it preserves vector addition. Similarly,

$$\phi(c\vec{a}) = ca_0 + ca_1 x + \ldots + ca_{n-1}x^{n-1} = c(a_0 + a_1 x + \ldots + a_{n-1}x^{n-1}) = c\phi(\vec{a}),$$

so it preserves scalar multiplication. □

Next, define

$$\sigma : GF(q)^n \to GF(q)^n,$$

$$(a_0, a_1, \ldots, a_{n-1}) \mapsto (a_{n-1}, a_0, \ldots, a_{n-2}).$$

Note that σ preserves the $GF(q)$-vector space operations, i.e., is a linear transformation. Indeed, in the standard basis it can be represented by a permutation matrix[11]. This is the *cyclic shift map*.

Proposition 91. *Via the correspondence ϕ between coefficient vectors and the associated polynomial, the cyclic shift map on $GF(q)^n$ corresponds to the multiplication by x map on R_n. In other words,*

$$\phi(\sigma(\vec{a})) = x\phi(\vec{a}),$$

for each $\vec{a} \in GF(q)^n$. Equivalently, the diagram

$$\begin{array}{ccc} GF(q)^n & \xrightarrow{\sigma} & GF(q)^n \\ \phi \downarrow & & \phi \downarrow \\ R_n & \xrightarrow[x]{\text{mult. by}} & R_n \end{array}$$

commutes.

Note, an ideal of R_n is simply a subset $I \subset R_n$ which is closed under

- vector addition,
- scalar multiplication,

[11] A *permutation matrix* is a square matrix of 0s and 1s which has exactly one 1 in each row and column.

- multiplication by arbitrary $r(x) \in R_n$.

This implies the following result.

Lemma 92. *A subset $I \subset R_n$ which is closed under*

- *vector addition,*
- *scalar multiplication,*
- *multiplication by x,*

is an ideal of R_n.

The following result classifies all the cyclic codes.

Theorem 93. *Let $C \subset GF(q)^n$ be a linear code. The following are equivalent:*

- *C is a cyclic code.*
- *$\phi(C)$ is an ideal in R_n.*

Proof. Assume C is a cyclic code.

We want to show that $\phi(C)$ is an ideal. By the lemma above, it sufficies to show that $\phi(C)$ is closed under multiplication by x.

Pick $\vec{c} \in C$ arbitrarily. We must show $x\phi(\vec{c}) \in \phi(C)$. By hypothesis, $\sigma(c) \in C$. By the proposition above $x\phi(\vec{a}) = \phi(\sigma(\vec{c})) \in \phi(C)$, as desired.

Conversely, assume $\phi(C)$ is an ideal in R_n. In particular, it is closed under multiplication by x. Pick $\phi(\vec{c}) \in \phi(C)$ arbitrarily. By the proposition above $\phi(\sigma(\vec{c})) = x\phi(\vec{a}) \in \phi(C)$. Since ϕ is one-to-one and onto, this implies $\sigma(\vec{c}) \in C$. Since $\vec{c} \in C$ was arbitrary, C is cyclic. □

Lemma 94. *Every ideal I in R_n is principal.*

Proof. Let $f(x) \in I$ be a non-zero element of smallest degree and note $(f(x)) \subset I$. If there is a $k(x) \in I - (f(x))$ then the division algorithm gives

$$k(x) = f(x)q(x) + r(x),$$

where $\deg(r(x)) < \deg(f(x))$. But then $r(x) = k(x) - f(x)q(x) \in I$ is an element of lower degree than $f(x)$. This contradicts the fact that $f(x)$ has lowest degree. □

Suppose that $C = (g(x)) = g(x)R_n$ is a cyclic code regarded as an ideal in R_n. There is nothing previenting $C = R_n$, which would not be an interesting code to study.

The condition $g(x)|(x^n - 1)$ guarantees that $C \neq R_n$. When is this true?

Lemma 95. *Let $g(x)$ be a polynomial with $g(0) \neq 0$ and without repeated roots. There is an $n > 1$ such that $g(x)|(x^n - 1)$.*

Proof. There is a $d > 0$ such that all the roots of $g(x) = 0$ are in $GF(q^d)$. Since $GF(q^d)^\times$ is a cyclic group of order $q^d - 1$, each root r of $g(x) = 0$ satisfies $r^{q^d-1} = 1$. Since $g(x)$ has no repeated factors, $g(x)|(x^{q^d-1} - 1)$. □

Example 96. *Let $n = 7$ and $R_7 = GF(2)[x]/(x^7 - 1)$. Note $x^7 = 1 = (x + 1)(x^3+x+1)(x^3+x^2+1)$. Let $g(x) = x^3+x+1$ and let $C = (g(x)) = g(x)R_7$. As a set,*

$$C = \{0, 1+x+x^3, x+x^2+x^4, x^2+x^3+x^5, x^3+x^4+x^6, 1+x^4+x^6, 1+x+x^5, x+x^2+x^6, \ldots\}.$$

The corresponding code $C' = \phi^{-1}(C) \subset GF(2)^7$ is, therefore, as a set

$$C' = \{(0,0,0,0,0,0,0), (1,1,0,1,0,0,0), (0,1,1,0,1,0,0), (0,0,1,1,0,1,0),$$

$$(0,0,0,1,1,0,1), (1,0,0,0,1,1,0), (0,1,0,0,0,1,1), \ldots .$$

In particular, $(0,0,1,0,1,1,1) = (0,0,1,1,0,1,0) + (0,0,0,1,1,0,1) \in C'$, since C and therefore C' is a vector space.

We claim:

$$C' = Span\{(1,0,0,0,1,1,0), (0,1,0,0,0,1,1), (0,0,1,0,1,1,1), (0,0,0,1,1,0,1)\}.$$

Suppose not. Since C' clearly contains the space, there must be a $c = (c_1, \ldots, c_7) \in C'$ such that c is not in the span. But then

$$c' = c - c_1(1,0,0,0,1,1,0) - c_2(0,1,0,0,0,1,1) - c_3(0,0,1,0,1,1,1) - c_4(0,0,0,1,1,0,1)$$

belongs to C' but not the span. This means that C contains a non-zero polynomial of the form $a_1x^4 + a_2x^5 + a_3x^6 = x^4(a_1 + a_2x + a_3x^2)$, which must be a multiple of $g(x) = 1 + x + x^3$. Clearly, this is impossible, which proves the claim.

Because of the claim, we know

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

is a generator matrix for C'.

In general, a binary linear $[n, k, d]$ code with a generator matrix in the form $G = (I_k \ A)$ is said to be systemic and the generator matrix is said to be in standard form.

One can verify directly that

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

is a check matrix for C', that is

$$C' = ker(H).$$

Indeed, $HG^T = 0$ (the 3×4 zero matrix).

Correspondingly, it can be verified that

$$C = \{f \in R_n \mid f(x)h(x) = 0\},$$

where $h(x) = x^4 + x^2 + x + 1$. This polynomial $h(x)$ is called the check polynomial of the code.

More generally, we have the following result.

Theorem 97. *Let $x^n - 1 = g(x)h(x)$, for some*

$$g(x) = g_0 + g_1x + \ldots + g_{n-k}x^{n-k}, \quad g_{n-k} \neq 0,$$

and

$$h(x) = h_0 + h_1x + \ldots + h_kx^k, \quad h_k \neq 0.$$

Define $C = (g(x)) = g(x)R_n \subset R_n$. The preimage $C = \phi^{-1}(C') \subset GF(2)^n$ is a cyclic code of length n and dimension k, with generator matrix

$$G = \begin{pmatrix} g_0 & g_1 & g_2 & \dots & g_{n-k} & 0 & \dots & 0 \\ 0 & g_0 & g_1 & g_2 & \dots & g_{n-k} & \dots & 0 \\ \vdots & & & & & & & \vdots \\ 0 & \dots & 0 & g_0 & g_1 & g_2 & \dots & g_{n-k} \end{pmatrix}$$

and check matrix

$$H = \begin{pmatrix} h_k & h_{k-1} & \dots & h_1 & h_0 & 0 & \dots & 0 \\ 0 & h_k & h_{k-1} & \dots & h_1 & h_0 & \dots & 0 \\ \vdots & & & & & & & \vdots \\ 0 & \dots & 0 & h_k & h_{k-1} & \dots & h_1 & h_0 \end{pmatrix}.$$

The polynomial $g(x)$ is called a *generator polynomial* of C and $h(x)$ the corresponding *check polynomial*.

Example 98. *We return to the example with* $n = 7$, $g(x) = x^3 + x + 1$, $h(x) = x^4 + x^2 + x + 1$, $R_7 = GF(2)[x]/(x^7 - 1)$, *and* $C = (g(x)) = g(x)R_7$. *In this case,*

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

and check matrix

$$H = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

Note that

$$HG^T = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

The codewords are

$$(0,0,0,0,0,0,0), (1,1,0,1,0,0,0), (0,1,1,0,1,0,0), (1,0,1,1,1,0,0),$$
$$(0,0,1,1,0,1,0), (1,1,1,0,0,1,0), (0,1,0,1,1,1,0), (1,0,0,0,1,1,0),$$
$$(0,0,0,1,1,0,1), (1,1,0,0,1,0,1), (0,1,1,1,0,0,1), (1,0,1,0,0,0,1),$$
$$(0,0,1,0,1,1,1), (1,1,1,1,1,1,1), (0,1,0,0,0,1,1), (1,0,0,1,0,1,1).$$

Example 99. *The* $[23, 12, 7]$ *Golay code. This code is named in honor of Marcel J. E. Golay whose 1949 paper introduced it for the first time.*

Over $GF(2)$*, we have*

$$x^{23}-1 = (x+1)(x^{11}+x^9+x^7+x^6+x^5+x+1)(x^{11}+x^{10}+x^6+x^5+x^4+x^2+1).$$

Let $g(x) = x^{11}+x^{10}+x^6+x^5+x^4+x^2+1$ *and* $C = (g(x)) \subset R_{23} \cong GF(2)^{23}$*. This is the* $[23, 12, 7]$ *Golay code.*

This code has generator matrix

$$G = \begin{pmatrix}
1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1
\end{pmatrix}$$

and check matrix

$$H = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1
\end{pmatrix}.$$

By Lemma 74, the generator matrix in standard form associated to H *is*

$$G' = \begin{pmatrix}
1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}.$$

3.6.1 Reed-Solomon codes as cyclic codes

Reed-Solomon codes were defined above as "evaluation codes", i.e., as the image $C = eval_P(F[x]_k)$ under the evaluation map

$$eval_P : F[x]_k \to F^n,$$

where $P = (p_0, p_1, \ldots, p_{n-1})$ is a list of distinct points in $F = GF(q)$ and

$$eval_P(f(x)) = (f(p_0), f(p_1), \ldots, f(p_{n-1})).$$

We proved in §3.5 that this is a linear $[n, k]$-code over F.

In some special cases, this is a cyclic code. We shall discuss this situation in this section.

Lemma 100. *Assume* $q \equiv 1 \pmod{n}$ *and let* $\alpha \in F$ *be a primitive* n*th root of unity. If* $p_0, p_1, \ldots, p_{n-1}$ *are given by* $p_i = \alpha^i$ *then* C *is a cyclic code.*

In this case, note that the map

$$F[x]_k \to F[x]_k$$

defined by $f(x) \mapsto f(\alpha x)$, induces the cyclic shift map on codewords, so C is cyclic.

Since C is a cyclic code, there is a generating polynomial $g(x) \in F[x]$ of degree $n-k$ which divides $x^n - 1$ such that, for any codeword $\vec{c} = (c_0, \ldots, c_{n-1})$ of the Reed-Solomon code, the associated polynomial

$$c(x) = c_0 + c_1 x + c_2 x^2 + \ldots + c_{n-1} x^{n-1}.$$

is a multiple of $g(x)$. Let r_1, r_2, ..., r_{n-k} denote the roots of $g(x) = 0$.

Decoding algorithm: Write the received word $\vec{f}$ as a polynomial: Let $f(x) = c(x) + e(x)$, where $e(x)$ is the error polynomial. We compute the "syndrome" values

$$(r_i, f(r_i)) = (r_i, c(r_i) + e(r_i)) = (r_i, e(r_i)), \qquad 1 \leq i \leq n-k.$$

If the number of errors is $\leq n-k$ and if the errors occur in the first $n-k$ coordinates of $\vec{f}$ then we may use Lagrange interpolation to recover $e(x)$, and hence solve for $c(x)$.

3.6.2 Quadratic residue codes

Let $\ell > 2$ be a prime and let $Q = Q_\ell$ be its set of *quadratic residues*,

$$Q = \{k \mid 0 < k < \ell, x^2 \equiv k \pmod{\ell} \text{ is solvable for } x\}.$$

Similarly, let $N = N_\ell = GF(\ell)^\times - Q$ denote the set of *quadratic non-residues*, Assume from now on that[12] $2 \in Q$.

Lemma 101. *The subgroup Q of $GF(\ell)^\times$ has order $(\ell-1)/2$.*

Proof. Consider the map $s : GF(\ell)^\times \to Q$ given by $s(x) = x^2$. This map is onto (by definition of Q). It is either 1-1 or it's not. It it were 1-1 then $\pm 1 \in GF(\ell)$ both go to 1, which implies $\ell = 2$. But that contradicts the hypothesis that $\ell > 2$. Therefore, s is not 1-1. This means that $|Q| < \ell - 1$, so there is some $n \in N \subset GF(\ell)^\times$.

We claim that the map

$$m_n : GF(\ell)^\times \to GF(\ell)^\times$$

$$x \mapsto nx$$

[12]It is known that this is equivalent to assuming that $p \equiv 1 \pmod 8$ or $p \equiv -1 \pmod 8$.

is 1-1 and onto and satisfies $m_n(Q) = N$. This simply renders into mathematical notation that fact that if you multiply a square by a non-square, you get a non-square. Therefore, $|Q| = |N|$ and we have $\ell - 1 = |GF(\ell)^\times| = |q \cup N| = |Q| + |N| = 2|Q|$. The lemma follows. $\square$

If $S \subset GF(\ell)^\times$, define

$$r_S(x) = \sum_{j \in S} x^j,$$

as a polynomial in $GF(2)[x]$.

Lemma 102. *If* $2S = \{2x \mid x \in S\}$ *then*

$$r_S(x)^2 = r_{2S}(x).$$

In particular, $r_Q^2 = r_Q$ *if and only if* $2 \in Q$ *and* $2 \in N$ *if and only if* $r_Q^2 = r_N$.

For any ring R, an element $r \in R$ with $r \neq 0, 1$ is called an *idempotent* if $r^2 = r$. The above lemma shows that $r_Q(x)$ is an idempotent of R_ℓ when $2 \in Q$.

Proof. This lemma follows from the fact that $(a+b)^2 = a^2 + b^2$ over $GF(2)$, and the fact that Q is a subgroup of $GF(\ell)^\times$. $\square$

If $S = Q$, define $g_Q(x) = c + r_Q(x)$, where we choose c so that $g_Q(1) = 1$. The ideal $C = (g_Q(x)) \subset R_\ell = GF(2)[x]/(x^\ell - 1)$ generated by $g_Q(x)$ is the *quadratic residue code* associated to p. The code C is a $[n, k, d]$ code where $n = \ell$, $k = (\ell + 1)/2$, and d is known to satisfy $d > \sqrt{\ell}$ (this is the *square root bound* for quadratic residue codes).

Alternatively, let $\alpha \in GF(2^m)$ be a primitive ℓth root of unity, where m denotes the order of $2 \pmod{\ell}$ (i.e., the smallest $m > 0$ such that $2^m \equiv 1 \pmod{\ell}$). Let

$$G_\alpha(x) = \prod_{k \in Q} (x - \alpha^k),$$

where Q is the set of $(\ell - 1)/2$ quadratic residues in $GF(\ell)$. The cyclic code $C_\alpha = (G_\alpha(x)) \subset R_\ell$ has dimension

$$k = \ell - \deg(G_\alpha) = \ell - |Q| = \frac{\ell + 1}{2}.$$

Ultimately, we want to show that

$$(g_Q(x)) = (G_\alpha(x)),$$

i.e., that these codes are the same.

Example 103. *Let* $\ell = 17$ *and note* $6^2 \equiv 2 \pmod{17}$*, so* 2 *is a quadratic residue* $\pmod{17}$*,* $2 \in Q_{17}$*. Indeed,*

$$Q_{17} = \{1, 2, 4, 8, 9, 13, 15, 16\},$$

so

$$g_Q(x) = 1+x+x^2+x^4+x^8+x^9+x^{13}+x^{15}+x^{16} = (x^4+x^3+1)^2(x^8+x^7+x^6+x^4+x^2+x+1).$$

Since $x^4 + x^3 + 1$ *is invertible in the ring* R_{17}*, we have* $(g_Q(x)) = (x^8 + x^7 + x^6 + x^4 + x^2 + x + 1)$*. Let*

$$g(x) = x^8 + x^7 + x^6 + x^4 + x^2 + x + 1.$$

The Sagemath *code below checks that* $g(x) = \prod_{k\in Q}(x - \alpha^k)$*, where we pick* $\alpha = a^{15} = a^5+a^2+a$*, where* $a \in GF(2^8)$ *is a primitive element, i.e., any fixed root of* $x^8 + x^4 + x^3 + x^2 + 1 = 0$*. The element* $\alpha' = a^{7\cdot 15} = a^{105} = a^4 + a^3 + a$ *also could have been used, but that would yield*

$$\prod_{k\in Q}(x - (\alpha')^k) = x^8 + x^5 + x^4 + x^3 + 1,$$

which generates a different code.

Sagemath

```
sage: F = GF(2)
sage: multiplicative_order(mod(2,17))
8
sage: F8.<a> = GF(2^8,"a")
sage: R.<x> = PolynomialRing(F, "x")
sage: R8.<xx> = PolynomialRing(F8, "xx")
sage: quadratic_residues(17)
[0, 1, 2, 4, 8, 9, 13, 15, 16]
sage: (2^8-1)/17
15
sage: alpha = a^15; alpha
a^5 + a^2 + a
sage: G_alpha = prod([xx-alpha^k for k in Q]); G_alpha
xx^8 + xx^7 + xx^6 + xx^4 + xx^2 + xx + 1
```

```
sage: H_alpha = prod([xx-alpha^k for k in N])
sage: G_alpha*H_alpha*(xx-1)
xx^17 + 1
sage: g_Q = 1+sum([xx^k for k in Q]); g_Q
xx^16 + xx^15 + xx^13 + xx^9 + xx^8 + xx^4 + xx^2 + xx + 1
sage: g_Q.roots()
[(a^5 + a^2, 1),
 (a^5 + a^2 + a, 1),
 (a^5 + a^3 + a^2, 1),
 (a^5 + a^4 + a^3 + a + 1, 1),
 (a^6 + a^5, 1),
 (a^6 + a^5 + a^2, 1),
 (a^7 + a^5 + a^3 + 1, 1),
 (a^7 + a^5 + a^4 + a^3 + 1, 1),
 (a^3 + a + 1, 2),
 (a^6 + a^2 + 1, 2),
 (a^7 + a^4 + a + 1, 2),
 (a^7 + a^6 + a^4 + a^3 + a^2, 2)]
sage: G_alpha.roots()
[(a^5 + a^2, 1),
 (a^5 + a^2 + a, 1),
 (a^5 + a^3 + a^2, 1),
 (a^5 + a^4 + a^3 + a + 1, 1),
 (a^6 + a^5, 1),
 (a^6 + a^5 + a^2, 1),
 (a^7 + a^5 + a^3 + 1, 1),
 (a^7 + a^5 + a^4 + a^3 + 1, 1)]
sage: factor(g_Q)
(xx + a^5 + a^2) * (xx + a^5 + a^2 + a) * (xx + a^5 + a^3 + a^2) *
 (xx + a^5 + a^4 + a^3 + a + 1) * (xx + a^6 + a^5) * (xx + a^6 + a^5 + a^2) *
 (xx + a^7 + a^5 + a^3 + 1) * (xx + a^7 + a^5 + a^4 + a^3 + 1) *
 (xx + a^3 + a + 1)^2 * (xx + a^6 + a^2 + 1)^2 *
 (xx + a^7 + a^4 + a + 1)^2 * (xx + a^7 + a^6 + a^4 + a^3 + a^2)^2
```

The last few commands tell us that $G_\alpha(x)$ and $g_Q(x)$ have the same roots. This is actually true more generally, as explained by the next result.

Theorem 104. *Let $g(x)$ be a polynomial of degree $n-k$ which divides x^n-1. Let $\alpha \in GF(2^m)$ be a primitive nth root of unity, where m denotes the order of $2 \pmod n$.*

(1) The cyclic code $C = (g(x))$ in $R_n = GF(2)[x]/(x^n-1)$ contains a unique idempotent $E(x)$ such that $C = (E(x))$. Moreover, $E(x) = p(x)g(x)$, for some polynomial $p(x)$, and

$$E(\alpha^i) = 0 \text{ if and only if } g(\alpha^i) = 0, \quad i > 0.$$

(2) $c(x) \in C$ if and only of $c(x)E(x) = c(x)$.

Proof. For the proof of (1), write $x^n - 1 = g(x)h(x)$, for some $h(x)$. Since the roots of $x^n - 1$ are all distinct, $gcd(g(x), h(x)) = 1$. By the extended Euclidean algorithm (the polynomial form of Bezout's Lemma), there are polynomials $p(x), q(x) \in GF(2)[x]$ such that

$$p(x)g(x) + q(x)h(x) = 1. \tag{14}$$

Let $E(x) = p(x)g(x)$. Regarded as an element of R_n, $g(x)h(x) = 0$, so we have

$$E(x)^2 = p(x)g(x) \cdot p(x)g(x) = p(x)g(x)(p(x)g(x) + 0)$$

$$= p(x)g(x)(p(x)g(x) + q(x)h(x)) = p(x)g(x) = E(x),$$

using (14). Therefore, $E(x)$ is an idempotent. By (14), we have $gcd(p(x), h(x)) = 1$. Therefore, any root α^i of $g(x)$ must also be a root of $p(x)$. This implies $E(\alpha^i) = 0$ if and only if $g(\alpha^i) = 0$.

Next, we claim $C = (g(x)) = (E(x))$. One direction is clear: since $E(x) = p(x)g(x)$ is a multiple of $g(x)$, we have $(E(x)) \subset (g(x))$. In the other direction, multiply both sides of (14) by $g(x)$, as an equation in R_n:

$$g(x) = p(x)g(x)^2 + q(x)h(x)g(x) = p(x)g(x)^2 = E(x)g(x).$$

Therefore, $g(x)$ is a multiple of $E(x)$, so $(g(x)) \subset (E(x))$. This establishes the claim.

Moving onto (2), suppose $c(x) = c(x)E(x)$. Then $c(x) \in (E(x)) = C$. Conversely, if $c(x) \in C$ then $c(x) = f(x)E(x)$, so $c(x)E(x) = f(x)E(x)^2 = f(x)E(x) = c(x)$. This proves (2).

It remains, in the proof of (1), to show that the generating idempotent is unique. Suppose $E_1(x)$ and $E_2(x)$ are idempotent generating C. Taking $c(x) = E_1(x)$ and $E_2(x)$ to be the generating idempotent in (2), we have $E_1(x) = E_1(x)E_2(x)$. Taking $c(x) = E_2(x)$ and $E_1(x)$ to be the generating idempotent in (2), we have $E_2(x) = E_2(x)E_1(x)$. Together, these give $E_1(x) = E_2(x)$. □

Assume ℓ has been selected so that $2 \in Q$. Next, we show that the unique idempotent which generates $C = (G_\alpha(x))$ is $(g_Q(x))$.

Proposition 105. *There is a primitive ℓth root of unity $\beta \in GF(2^m)$ such that, for all $s \in Q$, $g_Q(\beta^s) = 0$.*

Proof. Since $g_Q(x)$ is an idempotent (this is a consequence of the proof of Lemma 102), $g_Q(\alpha^i) = g_Q(\alpha^i)^2$. Therefore, $g_Q(\alpha^i)$ is invariant under the Frobenius map $Frob \in Gal(GF(2^m)/GF(2))$. This implies, for every i, we have $g_Q(\alpha^i) \in GF(2)$. For any $r \in Q$, we have

$$g_Q(\alpha^r) = c + \sum_{k \in Q} (\alpha^r)^k = c + \sum_{k \in Q} \alpha^{rk} = c + \sum_{s \in Q} \alpha^s = c + r_Q(\alpha) = g_Q(\alpha)$$

and, for any $r \in N$, we have

$$g_Q(\alpha^n) = c + \sum_{k \in Q} (\alpha^n)^k = c + \sum_{k \in Q} \alpha^{nk} = c + \sum_{s \in N} \alpha^s = c + r_N(\alpha).$$

Moreover, since

$$1 + r_Q(x) + r_N(x) = 1 + x + x^2 + \ldots + x^{\ell-1} = \frac{x^\ell - 1}{x - 1},$$

we have $r_Q(\alpha^i) = 1$ holds if and only if $r_N(\alpha^i) = 0$. Therefore, $r_Q(\alpha^r) = 1$ holds, for all $r \in Q$, if and only if, for any $n \in N$, $r_Q(\alpha^{nr}) = 0$, for all $r \in Q$ (since $r_Q(\alpha^{nr}) = r_N(\alpha^r)$).

Therefore, for every $r \in Q$, $g_Q(\alpha^r) = 0$ or, for every $r \in Q$, $g_Q(\alpha^r) = 1$. If for every $r \in Q$, $g_Q(\alpha^r) = 0$ holds, then we are done (take $\beta = \alpha$ and $s = r$). If for every $r \in Q$, $g_Q(\alpha^r) = 1$ holds, then then by replacing α by α^n for some $n \in N$, we have, for every $r \in Q$, $g_Q((\alpha^n)^r) = 0$ holds. We are done in this case as well, taking $\beta = \alpha^n$ and $s = r$. $\square$

Example 106. *Let's go through the proof of this proposition using some examples.*

If $\ell = 17$ then $m = 8$ is the multiplicative order of $2 \pmod{\ell}$. Define $GF(2^8) = GF(2)/(x^8 + x^4 + x^3 + x^2 + 1)$. If $a \in GF(2^8)$ is a (primitive) root of $x^8 + x^4 + x^3 + x^2 + 1 = 0$ then a has order $2^8 - 1 = 255$. We need an element α of order 17. Since $255 = 15 \cdot 17$, the element $\alpha = a^{15}$ has order 17.

Since $Q = \{1, 2, 4, 8, 9, 13, 15, 16\}$ has an even number of elements,

$$g_Q(x) = 1 + r_Q(x) = 1 + x + x^2 + x^4 + x^8 + x^9 + x^{13} + x^{15} + x^{16}.$$

Note, as an element of R_{17}, we have

$$g_Q(x)^2 = x^{32} + x^{30} + x^{26} + x^{18} + x^{16} + x^8 + x^4 + x^2 + 1 =$$

$$x^{15} + x^{13} + x^9 + x + x^{16} + x^8 + x^4 + x^2 + 1 = g_Q(x).$$

Let $N = \{3, 5, 6, 7, 10, 11, 12, 14\}$ *denote the non-residues and write*

$$r_N(x) = x^3 + x^5 + x^6 + x^7 + x^{10} + x^{11} + x^{12} + x^{14}.$$

Note, as an element of R_{17}*, we have*

$$r_N(x)^2 = x^{28} + x^{24} + x^{22} + x^{20} + x^{14} + x^{12} + x^{10} + x^6$$

$$= x^{11} + x^7 + x^5 + x^3 + x^{14} + x^{12} + x^{10} + x^6 = r_N(x).$$

Note

$$1 + r_Q(x) + r_N(x) = 1 + x + x^2 + x^4 + x^8 + x^9 + x^{13} + x^{15} + x^{16} +$$

$$x^3 + x^5 + x^6 + x^7 + x^{10} + x^{11} + x^{12} + x^{14} = 1 + x + \ldots + x^{16}.$$

We have[13]

$$g_Q(x) = (x - \alpha^{15})(x - \alpha)(x - \alpha^{16})(x - \alpha^8)(x - \alpha^2)(x - \alpha^{13})(x - \alpha^9)(x - \alpha^4)\cdot$$

$$\cdot(x - a^{238})^2(x - a^{221})^2(x - a^{119})^2(x - a^{187})^2 = G_\alpha(x)(x^8 + x^6 + 1).$$

The proof of the proposition above tells us that either (1) for every $r \in Q$*,* $g_Q(\alpha^r) = 0$ *or, (2) for every* $r \in Q$*,* $g_Q(\alpha^r) = 1$*. Clearly, in this case, it is case (1) that holds.*

Sagemath

```
sage: F = GF(2)
sage: m = multiplicative_order(mod(2,17)); m
8
sage: F8.<a> = GF(2^m,"a")
sage: F8.polynomial()
a^8 + a^4 + a^3 + a^2 + 1
sage: R.<x> = PolynomialRing(F, "x")
sage: R8.<xx> = PolynomialRing(F8, "xx")
sage: alpha = a^15
sage: alpha^17
1
sage: Q = [x for x in quadratic_residues(17) if x<>0]; Q
[1, 2, 4, 8, 9, 13, 15, 16]
sage: N = [x for x in GF(17) if x<>0 and not(x in Q)]; N
[3, 5, 6, 7, 10, 11, 12, 14]
sage: g_Q = 1+sum([xx^k for k in Q]); g_Q
xx^16 + xx^15 + xx^13 + xx^9 + xx^8 + xx^4 + xx^2 + xx + 1
```

[13]Note that the product contains αs and as.

```
sage: g_Q(1)
1
sage: [g_Q(alpha^k) for k in Q]
[0, 0, 0, 0, 0, 0, 0, 0]
sage: [g_Q(alpha^k) for k in N]
[1, 1, 1, 1, 1, 1, 1, 1]
```

Next, take $\ell = 23$. *In this case,* $m = 11$ *is the multiplicative order of* $2 \pmod{\ell}$. *Define* $GF(2^{11}) = GF(2)/(x^{11} + x^2 + 1)$. *If* $a \in GF(2^{11})$ *is a (primitive) root of* $x^{11} + x^2 + 1 = 0$ *then* a *has order* $2^{11} - 1 = 2047 = 23 \cdot 89$. *We need an element* α *of order* 23, *so we take* $\alpha = a^{89}$.

The quadratic residues mod 23 *are*

$$Q = \{1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18\}$$

and the non-residues are

$$N = \{5, 7, 10, 11, 14, 15, 17, 19, 20, 21, 22\}.$$

We have

$$\begin{aligned} g_Q(x) = r_Q(x) &= x^{18} + x^{16} + x^{13} + x^{12} + x^9 + x^8 + x^6 + x^4 + x^3 + x^2 + x \\ &= x(x^3 + x + 1)^2(x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1) \\ &= x(x^3 + x + 1)^2 \prod_{k \in Q} (x - \alpha^k). \end{aligned}$$

Since

$$x^{23} - 1 = (x+1)(x^{11}+x^9+x^7+x^6+x^5+x+1)(x^{11}+x^{10}+x^6+x^5+x^4+x^2+1),$$

it follows that $x(x^3 + x + 1)^2$ *is a unit in* R_{23}. *Therefore,*

$$(g_Q(x)) = (x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1) = (\prod_{k \in Q} (x - \alpha^k)).$$

Sagemath

```
sage: Q = [x for x in quadratic_residues(23) if x<>0]; Q
[1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18]
sage: N = [x for x in GF(23) if x<>0 and not(x in Q)]; N
[5, 7, 10, 11, 14, 15, 17, 19, 20, 21, 22]
sage: m = multiplicative_order(mod(2,23)); m
```

```
11
sage: F11.<a> = GF(2^11,"a")
sage: factor(2^11-1)
23 * 89
sage: alpha = a^(89)
sage: alpha^23
1
sage: g_Q = 1+sum([xx^k for k in Q]); g_Q
xx^18 + xx^16 + xx^13 + xx^12 + xx^9 + xx^8 + xx^6 + xx^4 + xx^3 + xx^2 + xx + 1
sage: g_Q(1)
0
sage: g_Q = sum([xx^k for k in Q]); g_Q
xx^18 + xx^16 + xx^13 + xx^12 + xx^9 + xx^8 + xx^6 + xx^4 + xx^3 + xx^2 + xx
sage: g_Q(1)
1
```

Corollary 107. *There is a primitive ℓth root of unity $\beta \in GF(2^m)$ such that $(g_Q(x)) = (G_\beta(x))$.*

Define $H_\beta(x) \in GF(2)[x]$ by $G_\beta(x)H_\beta(x) = x^\ell - 1$, where β is as in the previous proposition.

Proof. We know that $g_Q(x)$ and $G_\beta(x)$ have the same set of roots of the form $x = \alpha^i$. This implies $g_Q(x)$ is a multiple of $G_\beta(x)$, so $(g_Q(x)) \subset (G_\beta(x))$. This also implies that $g_Q(x)$ is relatively prime to $H_\beta(x)$. Therefore, by the extended Euclidean algorithms, $a(x)g_Q(x) + b(x)H_\beta(x) = 1$, for some $a(x), b(x) \in GF(2)[x]$. Multiply both sides by $G_\beta(x)$ to get

$$G_\beta(x) = G_\beta(x)a(x)g_Q(x) + G_\beta(x)b(x)H_\beta(x) = G_\beta(x)a(x)g_Q(x),$$

in R_ℓ. This implies $G_\beta(x)$ is a multiple of $g_Q(x)$, so $(g_Q(x)) \subset (G_\beta(x))$. □

Example 108. *Let C be the same code as in Example 103. This time, we try to decode a received word.*

The code $C = (g(x)) \subset R_{17}$, regarded as a subspace of $GF(2)^{17}$ has generator matrix

$$\begin{pmatrix}
1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1
\end{pmatrix}$$

and check matrix in standard form

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

The generator matrix in standard form associated to H is

$$G' = \left(\begin{array}{cccccccc|ccccccccc} 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right).$$

While the first of these generator matrices has rows of weight at least 7, the second has some rows of weight 5. Indeed, it is known that this code has minimum distance 5. Therefore, it is capable to correcting 2 errors.

Suppose that the received vector is $\vec{v} = (0,1,1,0,1,0,1,1,1,0,0,0,0,0,0,0,0)$, and suppose you know one error was made. Can you decode it?

3.6.3 BCH bound for cyclic codes

From Lemma 77, we know that the minimum distance d if a linear code determines how many errors can be corrected. In other words, it is a measurement of how "good" a code is.

In this section, we establish a lower bound on cyclic codes. For certain ("designed") cyclic codes, this bound can be close to the actual minimum distance. Fist, we need a fact about Vandermonde matrices.

A *Vandermonde matrix* is a square matrix each of whose column is a geometric series:

$$V(a_1, \ldots, a_n) = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ a_1 & a_2 & \ldots & a_n \\ a_1^2 & a_2^2 & \ldots & a_n^2 \\ \vdots & & & \vdots \\ a_1^{n-1} & a_2^{n-1} & \ldots & a_n^{n-1} \end{pmatrix},$$

for any $a_1, \ldots, a_n$ in a field (e.g., our finite $GF(q)$).

Lemma 109. $\det V(a_1, \ldots, a_n) = \prod_{i<j\leq n}(a_j - a_i)$.

Proof. We prove this by induction.

When $n = 2$, we have

$$\det V(a_1, a_2) = \det \begin{pmatrix} 1 & 1 \\ a_1 & a_2 \end{pmatrix} = a_2 - a_1.$$

Suppose $n > 2$. Let

$$A(x) = \det \begin{pmatrix} 1 & 1 & \ldots & 1 \\ a_1 & a_2 & \ldots & x \\ a_1^2 & a_2^2 & \ldots & x^2 \\ \vdots & & & \vdots \\ a_1^{n-1} & a_2^{n-1} & \ldots & x^{n-1} \end{pmatrix},$$

which is a polynomial of degree $n-1$. Since a matrix with identical columns is singular, we know $A(x)$ has roots at $x = a_1, \ldots, a_{n-1}$. Therefore, there is a constant $c \neq 0$ suvh that

$$A(x) = c \cdot \prod_{i=1}^{n-1}(x - a_i).$$

What is c? We can compute it by plugging in $x - 0$. On one hand,

$$A(0) = (-1)^{n-1} c a_1 \ldots a_{n-1}.$$

On the other hand,

$$A(0) = \det \begin{pmatrix} 1 & 1 & \dots & 1 \\ a_1 & a_2 & \dots & 0 \\ a_1^2 & a_2^2 & \dots & 0 \\ \vdots & & & \vdots \\ a_1^{n-1} & a_2^{n-1} & \dots & 0 \end{pmatrix} = (-1)^{n-1} \det \begin{pmatrix} 1 & 1 & \dots & 1 \\ a_1 & a_2 & \dots & a_{n-1} \\ a_1^2 & a_2^2 & \dots & a_{n-1}^2 \\ \vdots & & & \vdots \\ a_1^{n-1} & a_2^{n-1} & \dots & a_{n-1}^{n-1} \end{pmatrix}.$$

By the induction hypothesis, this is

$$= (-1)^{n-1} \prod_{i<j\leq n-1} (a_j - a_i).$$

These imply

$$c = \prod_{i<j\leq n-1} (a_j - a_i).$$

Plugging in this value of c into $A(x)$ and taking $x = a_n$ gives the formula in the statement of the lemma. □

Theorem 110. *Let $C = (g(x)) \subset R_n$ be a cyclic code of length n with generating polynomial*

$$g(x) = g_0 + g_1 x + \ldots + g_{n-k} x^{n-k}, \quad g_0 \neq 0, g_{n-k} \neq 0.$$

If there is an $\alpha \in GF(q^{n-k})$ such that

$$g(\alpha^r) = \ldots = g(\alpha^{r+s}) = 0,$$

then C is an $[n, k, d]$ code with $d \geq s + 2$.

Proof. Let $c(x) \in C$ be a non-zero codeword,

$$c(x) = c_0 + c_1 x + \ldots + c_{n-1} x^{n-1}.$$

By hypothesis $c(x)$ is a multiple of $g(x)$ and

$$c(\alpha^r) = \ldots = c(\alpha^{r+s}) = 0.$$

Suppose that $c(x)$ has weight t, so there are coefficients $c_{i_j} \neq 0$ such that

$$c(x) = c_{i_1}x^{i_1} + c_{i_1}x^{i_2} + \ldots + c_{i_t}x^{i_t},$$

and

$$\begin{gathered} c(\alpha^r) = c_{i_1}(\alpha^r)^{i_i} + \ldots + c_{i_t}(\alpha^r)^{i_t} = 0, \\ c(\alpha^{r+1}) = c_{i_1}(\alpha^{r+1})^{i_i} + \ldots + c_{i_t}(\alpha^{r+1})^{i_t} = 0, \\ \vdots \\ c(\alpha^{r+s}) = c_{i_1}(\alpha^{r+s})^{i_i} + \ldots + c_{i_t}(\alpha^{r+s})^{i_t} = 0. \end{gathered}$$

This can be written as a matrix equation $A\vec{v} = \vec{0}$, where $\vec{v} = (c_{i_1}, \ldots, c_{i_t})$ and

$$A = \begin{pmatrix} \alpha^{ri_i} & \alpha^{ri_2} & \ldots & \alpha^{ri_t} \\ \alpha^{(r+1)i_i} & \alpha^{(r+1)i_2} & \ldots & \alpha^{(r+1)i_t} \\ \alpha^{(r+2)i_i} & \alpha^{(r+2)i_2} & \ldots & \alpha^{(r+2)i_t} \\ \vdots & & & \vdots \\ \alpha^{(r+s)i_i} & a\alpha^{(r+s)i_2} & \ldots & \alpha^{(r+s)i_t} \end{pmatrix} = \alpha^{ri_i}\alpha^{ri_2}\cdots\alpha^{ri_t} \begin{pmatrix} 1 & 1 & \ldots & 1 \\ \alpha^{i_i} & \alpha^{i_2} & \ldots & \alpha^{i_t} \\ \alpha^{2i_i} & \alpha^{2i_2} & \ldots & \alpha^{2i_t} \\ \vdots & & & \vdots \\ \alpha^{si_i} & a\alpha^{si_2} & \ldots & \alpha^{si_t} \end{pmatrix}.$$

Suppose $t \leq s+1$. By the lemma on Vandermonde matrices above, the submatrix M obtained by taking the first t rows of A has non-zero determinant. Moreover, $M\vec{v} = \vec{0}$. Therefore the rank of A is at least t. Since $t \leq s+1$, the rank is at most t, so $\text{rank}(M) = t$. This implies M is 1-1, so $\vec{v} = \vec{0}$, and therefore $c(x) = 0$. This is a contradiction. $\square$

Example 111. *We return to the example with $n = 7$, $g(x) = x^3 + x + 1$, $R_7 = GF(2)[x]/(x^7 - 1)$, and $C = (g(x)) = g(x)R_7$. There is an $\alpha \in GF(8)$ for which*

$$g(x) = (x - \alpha)(x - \alpha^2)(x - \alpha^4),$$

so the hypotheses to the above theorem holds with $s = 1$. The BCH bound in the theorem above implies that $d \geq 3$. Indeed, C is known to be a $[7, 4, 3]$ code, so the BCH bound is sharp in this case.

Example 112. *Let $g(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1 \in GF(2)[x]$ and $C = (g(x)) \subset R_{23} \cong GF(2)^{23}$ be the $[n, 12, 7]$ Golay code. where $n = 23$, $k = 12$, $d = 7$.*

We have

$$g(x) = (x-\alpha)(x-\alpha^2)(x-\alpha^3)(x-\alpha^4)(x-\alpha^5)(x-\alpha^8)(x-\alpha^{12})(x-\alpha^{13})(x-\alpha^{16})(x-\alpha^{18}),$$

so the BCH bound implies that $d \geq 5$.

3.6.4 Decoding cyclic codes

We explain the method of decoding known as *error trapping*. For simplicity, we restrict to the case of binary cyclic codes.

Let C be a cyclic $[n, k, d]$ code over $GF(2)$, $C \subset GF(2)^n$. Let G be a $k \times n$ generator matrix and H be a $(n-k) \times n$ check matrix.

To implement the error-trapping algorithm, we must assume that we know the k information bits of the code C. In other words, if $\vec{m} \in GF(2)^k$ is the sender's original message, then the codeword sent is $\vec{c} = \vec{m}G$. We assume that all the coordinates of $\vec{m}$ occur in the coordinates of $\vec{c}$. The example below illustrates this.

Example 113. *Let $C \subset GF(2)^7$ be the cyclic code associated to the generator polynomial $g(x) = x^3 + x + 1$. By Theorem 97, we may take*

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

However, we use instead

$$G' = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

which has coordinates $1, 2, 6, 7$ as the information bits. Indeed, the cordewords are of the form

$$\vec{c} = \vec{m}G' = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

$$= (m_1, m_2, m_1 + m_2 + m_3, m_1 + m_3 + m_4, m_1 + m_2 + m_4, m_3, m_4)$$

and we can recover $\vec{m}$ from c_1, c_2, c_6, c_7.

The set-up: suppose $\vec{c} = (c_1, \ldots, c_n)$ is the codeword sent and an error vector $\vec{e} = (e_1, \ldots, e_n)$ occurs with weight t, so the received vector has the

form $\vec{v} = \vec{c} + \vec{e} = (v_1, \ldots, v_n)$, where $dist(\vec{v}, \vec{c}) = t$. Note the we require $t \leq (d-1)/2$ in order to hope to decode it, because of Lemma 77.

The rough idea behind error-trapping is that, using a suitable cyclic shift of the received vector, we want to "trap" the error-coordinates in the non-information bits. That is, we hope to find a cyclic shift σ such that the non-zero coordinates of $\vec{e}^{\sigma} = (e_{\sigma(1)}, \ldots, v_{\sigma(n)})$ all occur outside the information bits of C. This will help us decode $\vec{v}$.

Example 114. *Let $C \subset GF(2)^7$ be the cyclic code associated to the generator polynomial $g(x) = x^3 + x + 1$, with generator matrix*

$$G' = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

Assume that $\vec{v} = (0, 0, 1, 1, 0, 0, 1)$ was received and assume at most 1 error was made. We keep shifting $\vec{v}$ to the right until we find a vector whose information bits yields a nearby codeword.

If the information bits are correct then $\vec{m} = (0, 0, 0, 1)$ was the original message, so $\vec{c} = \vec{m}G' = (0, 0, 0, 1, 1, 0, 1)$ was the codeword sent. Note $dist(\vec{v}, \vec{c}) = 2$, so this can't be the correct codeword (since we assume at most 1 error was made).

Shift by one to the right: $\sigma\vec{v} = (1, 0, 0, 1, 1, 0, 0)$. If the information bits are correct then $\vec{m} = (1, 0, 0, 0)$ was the original message, so $\vec{c} = \vec{m}G' = (1, 0, 1, 1, 1, 0, 0)$ was the codeword sent. Note $dist(\sigma\vec{v}, \vec{c}) = 1$, so

$$decode[(1, 0, 0, 1, 1, 0, 0)] = (1, 0, 1, 1, 1, 0, 0).$$

Left-shifting to our original receiced word, we have

$$decode[(0, 0, 1, 1, 0, 0, 1)] = decode[\sigma^{-1}(1, 0, 0, 1, 1, 0, 0)]$$

$$= \sigma^{-1}(1, 0, 1, 1, 1, 0, 0) = (0, 1, 1, 1, 0, 0, 1).$$

It is simpler to mathematically prove error-trapping works using the language of syndromes. For this purpose, we introduce some new notation.

Set-up: Again, assume $C \subset GF(2)^n$ is a cyclic code. After possibly permuting the coordinates of C (i.e., the columns the generator matrix), we can assume that the generator matrix in standard form, $G = (I, A)$, for some

$k \times (n-k)$ matrix A. This equivalent code might not be cyclic. None-theless, for the remainder of this section, we assume (for simplicity) that C is a cyclic code with generator matrix in standard form, $G = (I_k, A)$. By Lemma 74, we may take $H = (-A^T, I_{n-k}) = (A^T, I_{n-k})$ as the check matrix.

We need the following fact.

Proposition 115. *Suppose $\vec{c} = (c_1, \ldots, c_n)$ is the codeword sent and the received vector has the form $\vec{v} = \vec{c} + \vec{e} = (v_1, \ldots, v_n)$, where the error vector is $\vec{e} = (e_1, \ldots, e_n)$ and $dist(\vec{v}, \vec{c}) = t \leq (d-1)/2$. If $\vec{s} = H\vec{v}$ has weight $\leq t$ then the information bits of $\vec{v}$ are correct. In this case, the errors are contained in the non-zero coordinates of the syndrome $H\vec{v}$.*

Proof. Suppose $wt(H\vec{v}) \leq t$. Let $\vec{e} = (\vec{e}^{(1)}, \vec{e}^{(2)})$, where $\vec{e}^{(1)} = (e_1, \ldots, e_k)$ and $\vec{e}^{(2)} = (e_{k+1}, \ldots, e_n)$.

Suppose that that the information bits are not all zero: $\vec{e}^{(1)} \neq \vec{0}$. We derive a contradiction.

Let

$$\vec{c'} = \vec{e}^{(1)}G = \vec{e}^{(1)}(I_k, A) = (\vec{e}^{(1)}, \vec{e}^{(1)}A),$$

Note

$$H\vec{v} = H\vec{c} + H\vec{e} = H\vec{e} = H(\vec{e}^{(1)}, \vec{e}^{(2)}) = (A^T, I_{n-k})(\vec{e}^{(1)}, \vec{e}^{(2)}) = (A^T\vec{e}^{(1)}, \vec{e}^{(2)}),$$

so $wt(A^T\vec{e}^{(1)}) = wt(H\vec{v}) - wt(\vec{e}^{(2)})$. On one hand, we have $wt(\vec{c'}) \geq d \geq 2t+1$. On the other hand,

$$wt(\vec{c'}) = wt(\vec{e}^{(1)}) + wt(\vec{e}^{(1)}A) = wt(\vec{e}^{(1)}) + wt(H\vec{v}) - wt(\vec{e}^{(2)}).$$

This implies

$$wt(H\vec{v}) \geq 2t + 1 - wt(\vec{e}^{(1)}) + wt(\vec{e}^{(2)}) \geq t + 1.$$

This is a contradiction.

For the second part, since $\vec{e}^{(1)} = \vec{0}$, we have $H\vec{v} = \vec{e}^{(2)}$. □

Example 116. *Let $C \subset GF(2)^9$ be the cyclic code associated to the generator polynomial $g(x) = x^6 + x^3 + 1$, with generator matrix*

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

and check polynomial

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

This is a $[9, 3, 3]$ *code (and it may be verified that the BCH bound is sharp in this example). Furthermore, the information bits are in the last three coordinates.*

Suppose that $\vec{v} = (1, 0, 1, 1, 1, 1, 1, 0, 1)$ *is a received word. The syndrome is* $H\vec{v} = (0, 0, 0, 0, 1, 0)$*. The above proposition applies since the weight of this syndrome is* $\leq t = [(d-1)/2] = 1$*. Therefore, the error occured outside the information bits, so* $\vec{m} = (1, 0, 1)$ *was the original message. This tells us that* $\vec{c} = \vec{m}G = (1, 0, 1, 1, 0, 1, 1, 0, 1)$ *is the codeword sent:*

$$decode[(1, 0, 1, 1, 1, 1, 1, 0, 1)] = (1, 0, 1, 1, 0, 1, 1, 0, 1).$$

Suppose that $\vec{v} = (1, 1, 1, 1, 0, 1, 1, 0, 1)$ *is a received word. The syndrome is* $H\vec{v} = (0, 1, 0, 0, 0, 0)$*. Therefore, the error occurred outside the information bits, so* $\vec{m} = (1, 0, 1)$ *was the original message. This tells us that* $\vec{c} = \vec{m}G = (1, 0, 1, 1, 0, 1, 1, 0, 1)$ *is the codeword sent, as above. Alternatively, since the weight of* $\vec{e}$ *is assumed to be* ≤ 1 *(because* $d = 3$*),* $H\vec{e}$ *is the column of* H *associated with the non-zero coordinate of* $\vec{e}$*. The fact that* $H\vec{v} = (0, 1, 0, 0, 0, 0)$ *tells us the error is in the* $2nd$ *position, so*

$$decode[(1, 1, 1, 1, 0, 1, 1, 0, 1)] = (1, 0, 1, 1, 0, 1, 1, 0, 1).$$

Suppose that $\vec{v} = (1, 0, 1, 1, 0, 1, 1, 1, 1)$ *is a received word. The syndrome is* $H\vec{v} = (0, 1, 0, 0, 1, 0)$*. This has weight* > 1*, so the information bits of* $\vec{v}$ *are wrong. We shift to the right, and check again:*

$$\sigma\vec{v} = (1, 1, 0, 1, 1, 0, 1, 1, 1), H\sigma\vec{v} = (0, 0, 1, 0, 0, 1).$$

This too has weight > 1*, so the information bits of* $\sigma\vec{v}$ *are wrong. We shift again to the right, and check again:*

$$\sigma\vec{v} = (1, 1, 1, 0, 1, 1, 0, 1, 1), H\sigma\vec{v} = (1, 0, 0, 0, 0, 0).$$

Therefore, the error occurred outside the information bits, so $\vec{m} = (0, 1, 1)$. *Therefore,*

$$decode[(1, 1, 1, 0, 1, 1, 0, 1, 1)] = \vec{m}G = (0, 1, 1, 0, 1, 1, 0, 1, 1).$$

We left-shift twice to get to our original received vector:

$$decode[(1, 0, 1, 1, 0, 1, 1, 1, 1)] = decode[\sigma^{-2}(1, 1, 1, 0, 1, 1, 0, 1, 1)]$$
$$= \sigma^{-2}(0, 1, 1, 0, 1, 1, 0, 1, 1) = (1, 0, 1, 1, 0, 1, 1, 0, 1).$$

Example 117. *Let* $C \subset GF(2)^{17}$ *be the quadratic residue cde in Example 103.*

Suppose that the received vector is $\vec{v} = (0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1)$ *(the codeword* $c = (1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0)$ *was the originally transmitted vector). Suppose you know two errors were made. Can you decode it using the error-trapping method?*

Suppose that the received vector is $\vec{v} = (0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ *(the codeword* $c = (1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0)$ *was the originally transmitted vector). Suppose you know two errors were made. Can you decode it using the error-trapping method?*

3.6.5 Cyclic codes and LFSRs

Suppose

$$x^n - 1 = g(x)h(x),$$

for some $g(x) \in GF(q)[x]$ of degree $n - k$ and $h(x) \in GF(q)[x]$ of degree k. Write

$$g(x) = g_0 + g_1 x + \ldots + g_{n-k}x^{n-k}, \quad g_0 \neq 0, \quad g_{n-k} \neq 0,$$

and

$$h(x) = h_0 + h_1 x + \ldots + h_k x^k, \quad h_0 \neq 0, \quad h_k \neq 0.$$

Consider the LFSR sequence

$$a_{n+1} = \sum_{j=1}^{k} c_j a_{n+1-j}, \tag{15}$$

of length k. It turns out there is a close connection between these.

Proposition 118. *Let $g(x), h(x) \in R_n$ be as above (their product is $x^n - 1$). The LFSR sequence*

$$a_{n+1} = \sum_{j=1}^{k} -h_{k+1-j} a_{n+1-j},$$

has period n and connection polynomial $-h^*(x)$, where h^* is the reverse polynomial of $h(x)$. Moreover, the states of this LFSR are all codewords in $\phi^{-1}(C)$, where $C = (g(x))$.

Remark 5. *In fact, if $\vec{a}_0 = (a_0, \ldots, a_{k-1})$ is any initial fill, then the initial state $\vec{a}_0 G$ is a codeword in $\phi^{-1}(C)$.*

Proof. Let $C' = \phi^{-1}(C)$ be the corresponding cyclic $[n, k]$ code over $GF(q)$. By a theorem above, the code C has an $(n-k) \times n$ check matrix of the form

$$H = \begin{pmatrix} h_k & h_{k-1} & ... & h_0 & 0 & ... & 0 \\ 0 & h_k & ... & h_0 & 0 & ... & 0 \\ . & & & & & & \\ . & & & & & & \\ . & & & & & & \\ 0 & ... & 0 & h_k & h_{k-1} & ... & h_0 \end{pmatrix}.$$

Since $h(x)$ divides $x^n - 1$, we must have $h_0 = 1$. If $\vec{c} = (c_1, ..., c_n) \in C$ then $H \cdot \vec{c} = \vec{0}$. So $\vec{c} \cdot \vec{H}_1 = 0$ ($\vec{H}_1 =$ top row of H). This means

$$c_1 \cdot h_k + c_2 \cdot h_{k-1} + ... + c_{k+1} \cdot h_0 + c_{k+2} \cdot 0 + ... + c_n \cdot 0 = 0,$$

so

$$c_{k+1} = -c_1 \cdot h_k - c_2 \cdot h_{k-1} - ... - c_k \cdot h_1 \tag{16}$$

Likewise, for the second row,

$$c_{k+2} = -c_2 \cdot h_k - c_3 \cdot h_{k-1} - ... - c_{k+1} \cdot h_1, \tag{17}$$

and so on for the other rows. This recursive relation defines a LFSR sequence. From the definitions, it follows that the connection polynomial is $-h^*(x)$, where h is the above check polynomial. Since we are over $GF(2)$, this is the same as $h^*(x)$, as claimed. □

Example 119. *Consider the binary LFSR with key* $(1,0,1,1)$ *and fill* $(1,1,0,1)$*:*

$$c_{k+1} = c_k + c_{k-1} + c_{k-3}, \tag{18}$$

and $c_0 = 1, c_1 = 1, c_2 = 0, c_3 = 1$*. The first several terms are*

$$1,1,0,1,0,0,0,1,1,0,1,0,0,0,1,1,0,1,0,0,0,1,\ldots\ .$$

Note that the states

$$(1,1,0,1,0,0,0),(1,0,1,0,0,0,1),(0,1,0,0,0,1,1),\ldots,$$

of this LFSR sequence are all codewords of the cyclic code in Example 98.

Now, let's build different LFSR sequences, one for each possible initial fill, but all using the same key (18). For each, we only compute the entries in one full period:

$(0,0,0,0,0,0,0)$ with initial fill $(0,0,0,0)$,

$(1,0,0,0,1,1,0)$ with initial fill $(1,0,0,0)$,

$(0,1,0,0,0,1,1)$ with initial fill $(0,1,0,0)$,

$(1,1,0,0,1,0,1)$ with initial fill $(1,1,0,0)$,

$(0,0,1,0,1,1,1)$ with initial fill $(0,0,1,0)$,

$(1,0,1,0,0,0,1)$ with initial fill $(1,0,1,0)$,

$(0,1,1,0,1,0,0)$ with initial fill $(0,1,1,0)$,

$(1,1,1,0,0,1,0)$ with initial fill $(1,1,1,0)$,

$(0,0,0,1,1,0,1)$ with initial fill $(0,0,0,1)$,

$(1,0,0,1,0,1,1)$ with initial fill $(1,0,0,1)$,

$(0,1,0,1,1,1,0)$ with initial fill $(0,1,0,1)$,

$(1,1,0,1,0,0,0)$ with initial fill $(1,1,0,1)$,

$(0,0,1,1,0,1,0)$ with initial fill $(0,0,1,1)$,

$(1,0,1,1,1,0,0)$ with initial fill $(1,0,1,1)$,

$(0,1,1,1,0,0,1)$ with initial fill $(0,1,1,1)$,

$(1,1,1,1,1,1,1)$ with initial fill $(1,1,1,1)$.

Note that these are all codewords of the cyclic code in Example 98.

4 Lattices

In mathematics, the term lattice is used for several completely different things. A few of them are listed below.

(a) It can refer to a partially ordered set in which any pair of elements has a least upper bound and a unique greatest lower bound.

(b) It can refer to a discrete co-compact subgroup of a (possibly non-abelian) topological group.

(c) It can refer to a finitely generated free abelian group.

(d) It can refer to a subgroup of $\mathbb{R}^n$ which is isomorphic (as an abelian group) to $\mathbb{Z}^k$, for some $k \leq n$.

In this class, we only use a special case of (d).

4.1 Basic definitions

Before defining (d) more carefully, we discuss the connection between (b), (c) and (d). In either case, a lattice will be a finitely generated abelian group, written additively.

If G is any finitely generated abelian group then let

$$G_{tor} = \{g \in G \mid mg = 0\}$$

denote its *torsion subgroup*. For example, if

$$G = \mathbb{Z} \times \mathbb{Z}/2\mathbb{Z} = \{(x, y) \mid x \in \mathbb{Z}, y \in \mathbb{Z}/2\mathbb{Z}\}$$

then

$$G_{tor} = \{(0, y) \mid y \in \mathbb{Z}/2\mathbb{Z}\} \cong \mathbb{Z}/2\mathbb{Z}.$$

The following result will not be proven[14] here, but will help place groups arising in (c) in context of groups arising in (d).

Theorem 120. *(Theorem 2.4.1 in [C93]) Let G be a finitely generated abelian group.*

[14] A proof is given, for example, in chapter 1 of Lang's book **Algebra**.

(1) There is an $r > 0$ such that

$$G \cong G_{tor} \times \mathbb{Z}^r.$$

(2) If $r = 0$ in (a) then there is an $n > 0$ and a subgroup L of $\mathbb{Z}^n$ such that $G \cong \mathbb{Z}^n/L$.

The integer r is called the *rank* of G.

Example 121. *Let A be any $m \times n$ matrix having entries in $\mathbb{Z}$. Let L_1 denote the $\mathbb{Z}$-linear row span of A and let L_2 denote the $\mathbb{Z}$-linear column span of A. The above theorem tells us that*

$$\mathbb{Z}^n/L_1 \cong H_1 \times \mathbb{Z}^{r_1},$$

for some integer $r_1 \geq 0$ and some finite group H_1, and

$$\mathbb{Z}^m/L_2 \cong H_2 \times \mathbb{Z}^{r_2},$$

for some integer $r_2 \geq 0$ and some finite group H_2. In general, $r_1 = r_2 = rank(A)$ and the finite groups H_1 and H_2 can be determined from the Smith Normal Form of A.

With regard to (b) and (d), we say that an infinite subset $L \subset \mathbb{R}^n$ is *discrete* if there is an $\epsilon > 0$ such that, for all $v_1, v_2 \in L$, $||v_1 - v_2|| > \epsilon$.

Theorem 122. *If $L \subset \mathbb{R}^n$ is a discrete subgroup (with respect to the usual vector addition and subtraction operations) then, for some k with $0 \leq k \leq n$, there are $v_1, \ldots, v_k \in \mathbb{R}^n$ such that*

$$L = Span_{\mathbb{Z}}\{v_1, \ldots, v_k\}.$$

This is not obvious and the proof is omitted. However, it is the result which connects (b) to (d).

For us, a *lattice* in $\mathbb{R}^n$ is a subgroup of $\mathbb{R}^n$ which is isomorphic (as an abelian group) to $\mathbb{Z}^n$, and whose $\mathbb{R}$-linear combinations spans the real vector space $\mathbb{R}^n$. Others call this a full-rank lattice.

A typical lattice L in $\mathbb{R}^n$ has the form

$$L = Span_{\mathbb{Z}}\{v_1, \ldots, v_n\} = \{c_1v_1 + \ldots + c_nv_n \mid c_i \in \mathbb{Z}\},$$

where $\{v_1, \ldots, v_n\}$ is a basis for $\mathbb{R}^n$ (as a vector space over $\mathbb{R}$). We call L an *integral lattice* if it has the above form, where $\{v_1, \ldots, v_n\}$ is a subset of $\mathbb{Z}^n$.

A *basis* for L is any set of independent vectors whose $\mathbb{Z}$-linear combinations spans L. The dimention or *rank* of L is the number of elements in a basis.

Lemma 123. *Let L be an integral lattice in $\mathbb{R}^n$¿ Suppose that $\{v_1, \ldots, v_n\}$ and $\{w_1, \ldots, w_n\}$ each are a basis for L. Then there is an $a \in GL(n, \mathbb{Z})$ such that $w_i = Av_i$, for $i = 1, 2, \ldots, n$.*

Proof. Express the v_is as a linear combination of the w_is. This relation may be described as a matrix equation, $v_i = Bw_i$, where B is an $n \times n$ matrix with integer entries. It is unique by linear independence. Express the w_is as a linear combination of the v_is. This relation may be described as a matrix equation, $w_i = Cv_i$, where C is an $n \times n$ matrix with integer entries. It is unique by linear independence. This forces $B = C^{-1}$. Take $A = C$. □

Example 124. *Consider the basis $v_1 = (0, -1, -1)$, $v_2 = (1, 2, 2)$, $v_3 = (0, 1, 2)$ of $L = \mathbb{Z}^3$. Another basis is $w_1 = (1, -4, -1)$, $w_2 = (-1, -5, 0)$, $w_3 = (1, 4, 0)$. Let*

$$A = \begin{pmatrix} 1 & -3 & 2 \\ -13 & 4 & 0 \\ -2 & 2 & -1 \end{pmatrix}.$$

It is easy to check that $Av_i = w_i$, for $i = 1, 2, 3$.

If $L \subset \mathbb{R}^n$ is a rank n lattice generated by vectors $v_1, \ldots, v_n$ then the *fundamental domain* of L is defined to be

$$F = \{a_1 v_1 + \ldots + a_n v_n \mid 0 \leq a_i < 1,\ 1 \leq i \leq n\}.$$

This is a parallelopiped generated by $v_1, \ldots, v_n$.

Lemma 125. *(a) Every element $v \in \mathbb{R}^n$ can be written as $\ell + f$, for a unique $\ell \in L$ and $f \in F$.*

(b) The quotient map of abelian groups

$$\mathbb{R}^n \to \mathbb{R}^n / L,$$

$$x \mapsto x + L,$$

restricts to an isomophism $F \to \mathbb{R}^n / L$.

Proof. ... □

From matrix theory, we know

$$vol(F) = \det(v_1, \ldots v_n),$$

where each v_i is regarded as a column vector. This quantity is called the *determinant* of L. The determinant of L occurs in several important results on lattices.

Theorem 126. *(Minkowski's Theorem) Let $L \subset \mathbb{R}^n$ denote a lattice and let $S \subset \mathbb{R}^n$ denote a symmetric convex subset. If*

$$vol(S) > 2^n \det(L),$$

then S contains a non-zero element of L.

Theorem 127. *(Hermite's Theorem) Let $L \subset \mathbb{R}^n$ denote a lattice. There is a non-zero $v \in L$ such that*

$$||v|| \leq \sqrt{n} \det(L).$$

The *Gauss expected shortest length* of a "random" lattice $L \subset \mathbb{R}^n$ is

$$\sigma(L) = \sqrt{\frac{n}{2\pi e}} (\det L)^{1/n}.$$

Example 128. *For a "random" 2×2 integral matrix A, the expected shortest length of $RowSpan_{\mathbb{Z}}(A)$ is*

$$|\det(A)|^{1/2} (\frac{1}{\pi e})^{1/2}.$$

In the case of

$$A = \begin{pmatrix} 20 & 16 \\ 20 & 17 \end{pmatrix},$$

a matrix with determinant 20 and shortest vector $(0, 1)$, we have

$$|\det(A)|^{1/2} (\frac{1}{\pi e})^{1/2} = 1.53 \ldots \ .$$

4.2 The shortest vector problem

There are a number of computationally difficult problems associated with a lattice. This section lists some of them.

The Shortest Vector Problem (SVP): Find a shortest nonzero vector in an integer lattice $L \subset \mathbb{R}^n$, i.e., find a nonzero vector $v \in L$ that minimizes the Euclidean norm $||v||$.

The best known at this time is Kannan's HKZ basis reduction algorithm, which solves the problem in $n^{\frac{n}{2e}+o(n)}$. Ajtai showed that the SVP problem was NP-hard.

The Closest Vector Problem (CVP): Given a vector $w \in \mathbb{R}^n - L$, find a vector $v \in L$ that is closest to w, i.e., that minimizes the Euclidean norm $||w - v||$.

It is known that any hardness of SVP implies the same hardness for CVP.

Shortest Basis Problem (SBP): Find a basis $v_1, \ldots, v_n$ for a lattice that is shortest in some sense.

To this end, we present *Gauss' algorithm* for computing a shortest basis of a two-dimensional lattice.
`Input`: a basis v_1, v_2 for a lattice $L \subset \mathbb{Z}^2$
`Output`: a non-zero shortest vector in L

(1) If $||v_2|| < ||v_1||$, swap v_1 and v_2.

(2) Compute $m = \text{round}(v_1 \cdot v_2 / ||v_1||^2)$, where round : $\mathbb{R} \to \mathbb{Z}$ rounds to the nearest integer (and away from 0 in case of a half-integer).

(3) If $m = 0$, return the basis vectors v_1 and v_2.

(4) Replace v_2 with $v_2 - mv_1$.

(5) Go to (1)

Remark 6. *The example of* $L = Span_{\mathbb{Z}}\{(20, 16), (16, 20)\}$ *shows that one cannot in general replace* `nearest integer` *by* `floor` *in the above algorithm.*

Alternatively, we could describe Gauss' algorithm as follows:

(1) If $||v_2|| < ||v_1||$, swap v_1 and v_2.

(2) Replace v_2 with $v_2 - mv_1$, where $m \in \mathbb{Z}$ is computed so that $v_2 - mv_1$ is as short as possible.

(3) If $||v_2|| \geq ||v_1||$ then return v_1, v_2.

(4) Go to (1)

This last description makes it clearer that we are running through the following loop: (a) make sure v_1 is shorter than v_2, (b) try to replace v_2 by its difference with the vector projection of v_2 onto v_1 (or at least the vector in the lattice closest to this).

Example 129. Let $L = Span_{\mathbb{Z}}\{(20, 16), (20, 17)\}$. Find a shortest basis for L.

and:

Sagemath

```
sage: A = matrix(ZZ, [[20,16],[20,17]])
sage: A.LLL()
[ 0  1]
[20  0]
sage: A.BKZ()
[ 0  1]
[20  0]
```

In other words, Sagemath tells us that the answer is $\{(0, 1), (20, 0)\}$, whether we use the LLL algorithm or the BKZ variant.

Let us use Gauss' algorithm.

(1) Note $||v_1|| = 25.61... < ||v_2|| = 26.24...$.

(2) Compute $m = [v_1 \cdot v_2/||v_1||^2] = [\frac{20^2+16\cdot 17}{20^2+16^2}] = 1$.

(4) Replace v_2 with $v_2 - mv_1 = (20, 17) - (20, 16) = (0, 1)$.

(1) Note $||v_2|| = 1 < ||v_1|| = 25.61...$, so let $v_1 = (0, 1)$ and $v_2 = (20, 16)$.

(2) Compute $m = [v_1 \cdot v_2/||v_1||^2] = [\frac{16}{0^2+1^2}] = 16$.

(4) Replace v_2 with $v_2 - mv_1 = (20, 16) - (0, 16) = (20, 0)$.

(1) Note $||v_2|| = 20 > ||v_1|| = 1$.

(2) Compute $m = [v_1 \cdot v_2/||v_1||^2] = 0$. Return $v_1 = (0,1)$ and $v_2 = (20,0)$.

The Sagemath commands for this are below.

Sagemath

```
sage: v1 = vector(ZZ,[20,16]); v2 = vector(ZZ,[20,17])
sage: Nv1 = v1.norm(); Nv2 = v2.norm()
sage: RR(Nv1); RR(Nv2)
25.6124969497314
26.2488094968134
sage: m = int(v1.dot_product(v2)/Nv1^2); m
1
sage: v2 = v2-m*v1; v2
(0, 1)
sage: Nv1 = v1.norm(); Nv2 = v2.norm()
sage: RR(Nv1); RR(Nv2)
25.6124969497314
1.00000000000000
sage: v1,v2=v2,v1
sage: Nv1 = v1.norm(); Nv2 = v2.norm()
sage: RR(Nv1); RR(Nv2)
1.00000000000000
25.6124969497314
sage: m = int(v1.dot_product(v2)/Nv1^2); m
16
sage: v2 = v2-m*v1; v2
(20, 0)
sage: Nv1 = v1.norm(); Nv2 = v2.norm()
sage: RR(Nv1); RR(Nv2)
1.00000000000000
20.0000000000000
sage: v1; v2
(0, 1)
(20, 0)
```

In either case, the result is that the Gauss algorithm, too, tells us that the answer is $\{(0,1),(20,0)\}$,

Example 130. Let $L = Span_{\mathbb{Z}}\{(201,217),(120,123)\}$. Find a shortest basis for L. The steps above give

$$(201,217),(120,123),$$

$$(120,123),(201,217),$$

$$(120,123),(81,94),\theta = 3.6^o,$$

$$(81,94),(120,123),$$

$$(81, 94), (39, 29), \theta = 13^o,$$
$$(39, 29), (81, 94),$$
$$(39, 29), (3, 36), \theta = 49^o,$$
$$(3, 36), (39, 29).$$

This is depicted in Figure 2.

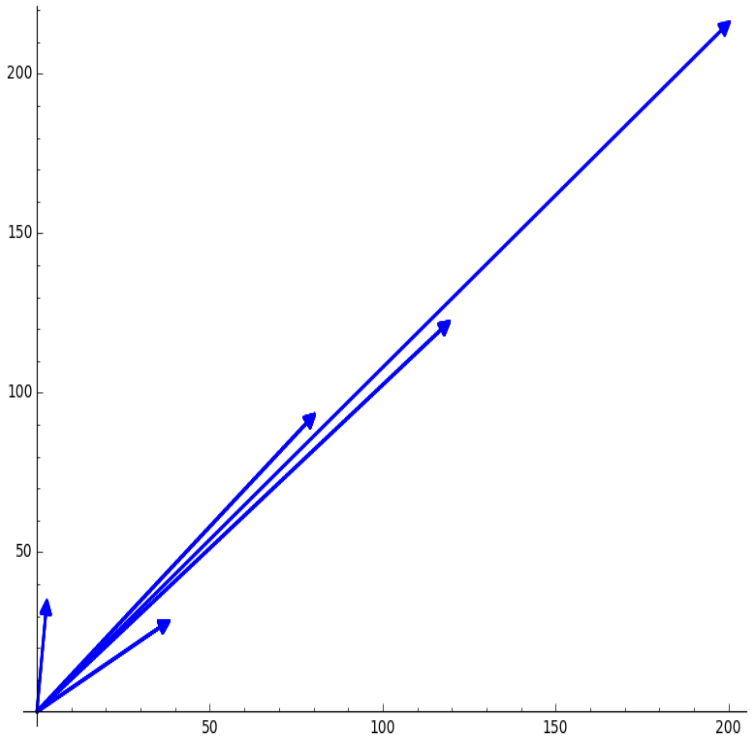


Figure 2: Gauss' algorithm

Lemma 131. *Gauss' algorithm above terminates in a finite number of steps and produces a shortest vector in L.*

Proof. Suppose we have a basis v_1, v_2 for a lattice $L \subset \mathbb{Z}^2$, where $||v_1|| \leq ||v_2||$.

The algorithm must terminate in a finite number of steps since the vectors v_1, v_2 in stage k decreased in size from those v_1, v_2 in stage $k-1$.

Suppose the algorithm has terminated and returned v_1, v_2. Since nearest integer to $v_1 \cdot v_2/||v_1||^2$ is 0, the length of $proj_{v_1}(v_2)$ satisfies

$$|proj_{v_1}(v_2)| = \frac{|v_1 \cdot v_2|}{||v_1||^2} \leq 1/2.$$

Let $v = a_1 v_1 + a_2 v_2$, for $a_1, a_2 \in \mathbb{Z}$. We have

$$||v||^2 = a_1^2||v_1||^2 + 2a_1a_2(v_1 \cdot v_2) + a_2^2||v_2||^2 \geq (a_1^2 - |a_1a_2| + a_2^2)||v_1||^2 \geq ||v_1||^2.$$

Therefore, v_1 is a shortest vector in L. $\square$

4.2.1 Application to a congruential PKC

This application follows §6.1 in the excellent book [HPS].

Alice wants to talk to Bob. Bob tells Alice that he has to generate keys first. He picks a large integer q and secret positive integers f, g with

$$\sqrt{q}/2 < f, g < \sqrt{q/2}, \quad gcd(f, q) = 1.$$

He then computes

$$h = f^{-1}g \pmod{q}, \tag{19}$$

where f^{-1} is computed in $(\mathbb{Z}/q\mathbb{Z})^\times$.

The *public key* is (h, q) and the *private key* is (f, g).

Encryption: Alice converts her plaintext into an integer m, satisfying $0 < m < \sqrt{q}/2$. She also picks a random integer r satisfying $0 < r < \sqrt{q/2}$. She computes the ciphertext

$$c = rh + m \pmod{q}.$$

and sends it to Bob.

Decryption: Bob decrypts by first computing

$$a = fc \pmod{q},$$

and then computing

$$m = f^{-1}a \pmod{g},$$

where f^{-1} is computed in $(\mathbb{Z}/g\mathbb{Z})^{\times}$.

Break: Note that (19) is equivalent to

$$g = hf + jq,$$

for some integer j. Therefore,

$$f(1, h) - j(0, q) = (f, g),$$

and in particular, the private key belongs to the lattice $L = Span_{\mathbb{Z}}\{(1, h), (0, q)\}$.

Here is an example using Sagemath :

Sagemath

```
sage: q = 100000
sage: RR(sqrt(q/2))  ## upper bound for f,g
223.606797749979
sage: f = 211; g = 213
sage: Zq = IntegerModRing(q)
sage: Zq(f)^(-1)
92891
sage: h = Zq(f)^(-1)*g
sage: h
85783
sage: RR(sqrt(q/4)) ## lower bound for g, upper bound for m
158.113883008419
sage: m = 125
sage: r = 199
sage: c = r*h+m; c
70942
sage: a = f*c; a
68762
sage: Zg = IntegerModRing(g)
sage: b = Zg(ZZ(f))^(-1)*ZZ(a); b
125
sage: A = matrix(ZZ, [[1,h],[0,q]]); A
[     1  85783]
[     0 100000]
sage: A.LLL()
[-211 -213]
[-204  268]
sage: f; g
211
213
```

4.3 LLL and a reduced lattice basis

The LenstraLenstraLovász (LLL) algorithm is a generalization of Gauss' algorithm. However, in higher dimensions it does not always produce a shortest vector.

Let $\mathcal{B} = \{v_1, \ldots, v_n\}$ denote an integral basis of the lattice $L \subset \mathbb{R}^n$. The GramSchmidt algorithm produced an orthogonal basis of $\mathbb{R}^n$ as follows. Start with $w_1 = v_1$, and then for $i > 1$ let

$$w_i = v_i - \sum_{j=1}^{i-1} \mu_{i,j} v_j \tag{20}$$

where $\mu_{i,j} = (v_i \cdot w_j)/||w_j||^2$. Note: The collection of vectors $\mathcal{B}^* = \{w_1, \ldots, w_n\}$ is an orthogonal basis for $\mathbb{R}^n$ but is (typically) not a basis for the lattice L spanned by $\mathcal{B}$.

Definition 132. We say that the basis $\mathcal{B}$ of L is *reduced* if

$$|\mu_{i,j}| \leq 1/2,$$

and

$$|||w_i|| \geq \sqrt{3/4 - \mu_{i,i-1}^2}||w_{i-1}||, \quad i > 1.$$

This definition is makes precise the idea of a "nearly orthogonal" basis of a lattice. The last condition in Definition 132 basically says that the projection of v_i onto $Span_{\mathbb{Z}}(v_1, \ldots, v_{i-2})^{\perp}$ is not too short relative to the projection of v_{i-1} onto the same space.

Example 133. Let $L = RowSpan_{\mathbb{Z}}(A)$ be generated by the rows of

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & -1 & 0 \\ 3 & 2 & 1 \end{pmatrix}.$$

A reduced basis for L is given by

$$\{(2,\, 0,\, -2)\,, \quad (-2,\, 1,\, -2)\,, \quad (-1,\, 3,\, 1)\}.$$

Let the ith row of A be denoted by v_i. Now, sort the $\{v_i\}$ by length, shortest to longest. By analogy with (20), the basic idea behind LLL is, for each j, to repeatedly apply

$$v_j = v_j - \text{round}(\mu_{j,i}) v_i, \quad i < j,$$

provided it shortens v_j.

4.4 Hermite normal form

While the LLL algorithm is, in some very rough sense, an analog of Gram-Schmidt, the Hermite normal form algorithm is, in some very rough sense, an analog of row reduction. While the Hermite normal form (HNF) is more reduced in some sense, and while rows of $HNF(A)$ and A do generate the same integral lattice, the rows of $HNF(A)$ are often longer than the rows of A itself. Therefore, the HNF is not a substitute for LLL.

An $m \times n$ matrix A with integer entries is in Hermite normal form (HNF) if

- All nonzero rows are above any rows of all zeroes,
- The leading coefficient (the pivot) of a nonzero row is always strictly to the right of the leading coefficient of the row above it; moreover, it is positive.
- All entries in a row above a leading entry are nonnegative and strictly smaller than the leading entry.
- All entries in a column below a leading entry are zeroes.

In particular, a nonsingular square matrix with integer entries will be in Hermite normal form if (a) it is upper triangular, (b) its diagonal entries are positive, (c) in every column, the entries above the diagonal are non-negative and smaller than the entry on the diagonal.

Theorem 134. *For each $n \times n$ matrix A with integer entries, there is a (non-unique) matrix U having integer entries and determinant ± 1 such that $UA = H$, where H is the HNF of A.*

Example 135. *Let*

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & -1 & 0 \\ 3 & 2 & 1 \end{pmatrix}.$$

Take -4 times row 1 and add it to row 2:

$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & -9 & -12 \\ 3 & 2 & 1 \end{pmatrix}.$$

Take -3 *times row 1 and add it to row 3:*

$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & -9 & -12 \\ 0 & -4 & -8 \end{pmatrix}.$$

Take -2 *times row 3 and add it to row 2:*

$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & -1 & 4 \\ 0 & -4 & -8 \end{pmatrix}.$$

Take -4 *times row 2 and add it to row 3:*

$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & -1 & 4 \\ 0 & 0 & -24 \end{pmatrix}.$$

Take 2 *times row 2 and add it to row 1:*

$$\begin{pmatrix} 1 & 0 & 11 \\ 0 & -1 & 4 \\ 0 & 0 & -24 \end{pmatrix}.$$

Rescale rows 2 and 3:

$$\begin{pmatrix} 1 & 0 & 11 \\ 0 & 1 & -4 \\ 0 & 0 & 24 \end{pmatrix}.$$

Finally, to get the Hermite normal form of A*, add row 3 to row 2:*

$$HNF(A) = \begin{pmatrix} 1 & 0 & 11 \\ 0 & 1 & 20 \\ 0 & 0 & 24 \end{pmatrix}.$$

Also, note that if we started with the augmented matrix (A, I_3) *instead of* A *and performed exactly the same operations, we would obtain*

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 11 & 5 & 2 & -4 \\ 0 & 1 & 20 & 9 & 3 & -7 \\ 0 & 0 & 24 & 11 & 4 & -9 \end{array}\right)$$

as our last step. In this case, it is not hard to verify that the block

$$U = \begin{pmatrix} 5 & 2 & -4 \\ 9 & 3 & -7 \\ 11 & 4 & -9 \end{pmatrix}$$

satisfies the criterion in the Theorem above: $UA = HNF(A)$.

4.5 NTRU as a lattice cryptosystem

This section continues using the notation in the previous section on NTRU, §1.6.2.

One attempt at an interpretation is to compute the NTRU public key

$$h(x) = g(x)f_q^{-1}(x) \in H_q,$$

lift this to $h^*(x) \in H$, then regard the collection of multiples $b(x)h^*(x)$ as belonging to the lattice L spanned by the columns of $Mat_N(h^*)$, where Mat_N is defined in (6). While h is the public key, L depends on h^*. The decryption problem boils down to this: given an element of a coset $vec_N(m) + L$, can one (efficiently) recover $m(x)$ independent of the lifting choosen? Maybe one can, but I don't see how. There is another approach, discussed next, which is much better.

The following interpretation is found in chapter 6 of the book by Hoffstein, Piper and Silverman [HPS].

Consider the upper-triangular $2N \times 2N$ matrix

$$M_h = \begin{pmatrix} I_N & Mat_N(h)^T \\ 0 & qI_N \end{pmatrix}, \tag{21}$$

where I_N denotes the $N \times N$ identity matrix, 0 denotes the $N \times N$ matrix of all zeros. The $\mathbb{Z}$-span of the rows of M_h defines the *NTRU lattice*,

$$L_h = Row_{\mathbb{Z}}(M_h).$$

The following result is Prop. 6.59 in [HPS].

Proposition 136. *We have*

$$f(x)h(x) = g(x) + qj(x)$$

in H if and only if

$$(\vec{f}, -\vec{j})M_h = (\vec{f}, \vec{g}).$$

In particular, $(\vec{f}, \vec{g}) \in L_h$.

Before proving this, let's look at an example.

Example 137. *Let*

$$N = 3, p = 3, q = 101.$$

Let $f(x) = x^2 + 1$, so $f_p^{-1}(x) = x^2 + 2x + 2 \in H_p$, and $f_q^{-1}(x) = 50x^2 + 51x + 51 \in H_q$. Let $g(x) = x^2 + x + 1$ and compute

$$h(x) = g(x)f_q^{-1}(x) = 51x^2 + 51x + 51$$

in H_q. In H, we have

$$f(x)h(x) - g(x) = 51x^4 + 51x^3 + 101x^2 + 50x + 50 = 101x^2 + 101x + 101 = qj(x),$$

where $j(x) = x^2 + x + 1$.

We have

$$M_h = \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 51 & 51 & 51 \\ 0 & 1 & 0 & 51 & 51 & 51 \\ 0 & 0 & 1 & 51 & 51 & 51 \\ \hline 0 & 0 & 0 & 101 & 0 & 0 \\ 0 & 0 & 0 & 0 & 101 & 0 \\ 0 & 0 & 0 & 0 & 0 & 101 \end{array}\right).$$

Since $(\vec{f}, -\vec{j}) = (1, 0, 1, -1, -1, -1)$, we have $(\vec{f}, -\vec{j})M_h = (1, 0, 1, 1, 1, 1)$, as desired.

To see how to recover the private key $(f(x), g(x))$ from the NTRU lattice, first look at the following computation using Sagemath *:*

Sagemath

```
sage: r1 = [1 , 0 , 0 , 51 , 51 , 51]
sage: r2 = [0 , 1 , 0 , 51 , 51 , 51]
sage: r3 = [0 , 0 , 1 , 51 , 51 , 51]
sage: r4 = [0 , 0 , 0 , 101 , 0 , 0]
```

```
sage: r5 = [0 , 0 , 0 , 0 , 101 , 0]
sage: r6 = [0 , 0 , 0 , 0 , 0 , 101]
sage: M_h = matrix(ZZ,[r1,r2,r3,r4,r5,r6])
sage: M_h.LLL()
[ -1   1   0   0   0   0]
[ -1   0   1   0   0   0]
[ -1   0  -1  -1  -1  -1]
[-23 -23 -23  16  16  16]
[ -7  -8  -8  39  39 -62]
[  7   8   8  62 -39 -39]
```

Notice that one of the shortest vectors returned by LLL is $(-1, 0, -1, -1, -1, -1)$*, or rescaled, is*

$$(1, 0, 1, 1, 1, 1).$$

This is precisely the vector representation of the private $(f(x), g(x)) = (x^2 + 1, x^2 + x + 1)$*.*

References

[CJT12] Chris Christensen, David Joyner, Jenna Torres, *Lester Hill's error-detecting codes*, Cryptologia 36(2012)88-103.

[C93] H. Cohen, **A course in computational algebraic number theory**, Springer, 1993.

[HPS] Jeffery Hoffstein, Jill Pipher, Joseph H. Silverman, **An Introduction to Mathematical Cryptography**, Springer, 2008.

[Kl13] Andreas Klein, **Stream Ciphers**, Springer-Verlag, 2013.

[Ju15] Thomas W. Judson, **Abstract Algebra**, `http://abstract.ups.edu/`.

[MS77] F. MacWilliams and N. Sloane, **The theory of error-correcting codes**, North-Holland, 1977.

[Sa] Sagemath, (free web-based collaboration platform) `http://cloud.sagemath.com` and online information (such as the online documentation) `http://www.sagemath.org`.

[SBGJKMW] Dennis Spellman, Georgia M. Benkart, Anthony M. Gaglione, W. David Joyner, Mark E. Kidwell, Mark D. Meyerson, William P. Wardlaw, *Principal Ideals and Associate Rings*, preprint, 2001. `http://wdjoyner.com/papers/ring3.pdf`

[SS] Damien Stehlé and Ron Steinfeld, *Making NTRUEncrypt and NTRUSign as Secure as Worst-Case Problems over Ideal Lattices*, Eurocrypt2011 proceedings.