



Chapter 5. Finite Field and Reed-Solomon Codes

§1. Motivation

Linearity in Coding makes encoding and decoding simpler.

In binary case, linearity has limited choices, i.e., limited codes that may not be good in finding good codes

$$((10)(00)) \begin{pmatrix} \uparrow & \downarrow & \uparrow & \uparrow \\ 1 & 1 & 0 & 1 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ \downarrow & \downarrow & \downarrow & \downarrow \end{pmatrix} = (0100)$$

linear

When we block 3 binary bits together into one single vector, if we can define such multiplications and divisions between binary vectors, we may explore more choices

$$([101], [100]) \left(\begin{matrix} [111]^t & [010]^t & [100]^t & [111]^t \\ [001]^t & [101]^t & [110]^t & [010]^t \end{matrix} \right)$$

$$= ([101] * [111] + [100] * [001], [101] * [010] + [100] * [101], [101] * [100] + [100] * [110], [101] * [111] + [100] * [010])$$

Although these operations are linear for vectors but may not be linear for binary bits.



Thus, finite fields generalize binary linear operations into binary non linear operations!

⇒ Binary non linear Codes.

but finite field (or binary vector) linear!

Finite Fields : Define binary vector multiplications and divisions

§2 Definition

Let $\mathbb{F}$ be a set of elements on which two binary operations (for each two elements), called addition "+" and multiplication " $\cdot$ ", are defined. The set $\mathbb{F}$ together with the two binary operations + and $\cdot$ is a field if the following conditions are satisfied: For any $a, b, c \in \mathbb{F}$

i) $a+b \in \mathbb{F}$ & $a \cdot b \in \mathbb{F}$,

ii) $(a+b)+c = a+(b+c)$,

iii) there exists an element $0 \in \mathbb{F}$ such that

$a+0 = 0+a = a$, which is called the zero element,



- (iv) For any $a \in F$, there exists $a' \in F$ such that
 $a + a' = a' + a = 0$ and a' is denoted $-a$,
- (v) $a + b = b + a$, commutativity
- (vi) $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
- (vii) there exists an element $e \in F \setminus \{0\}$ such that
 $e \cdot a = a \cdot e = a$, where e is denoted by 1 ,
- (viii) For any $a \in F \setminus \{0\}$, there exists a' such that
 $a \cdot a' = a' \cdot a = 1$, a' is called the inverse of a ,
 is denoted by a^{-1} ,
- (IX) $a \cdot b = b \cdot a$
- (X) $a \cdot (b + c) = a \cdot b + a \cdot c$ (distributive)

If F has only finite elements, F is called a
 finite field (or Galois field).

Ex. 1: $F = \{0, 1\}$

Modulo 2 addition

+	0	1
0	0	1
1	1	0

Binary field

modulo 2 multiplication

$\cdot$	0	1
0	0	0
1	0	1



Example 2 : $\mathbb{F} = \mathbb{Z}_5 = \{0, 1, 2, 3, 4\}$

$+$ is the modulo 5 addition : $a+b=c \pmod{5}$

$\cdot$ — — — multiplication : $a \cdot b = c \pmod{5}$

$+$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

$\cdot$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

In general $\mathbb{F} = \mathbb{Z}_p = \{0, 1, 2, \dots, p-1\}$ for any prime p is a field.

Why do we need these self-defined rules?

① for linear operations (simple)

② solve linear equations $Y_{1 \times N} = X_{1 \times K} \cdot G_{K \times N}$

$$a \cdot b = 0 \text{ and } a \neq 0 \Rightarrow b = 0$$

$$a \neq 0 \text{ \& } a \cdot b = a \cdot c \Rightarrow b = c$$

$a \cdot b = 0$ & $a \neq 0$, we need a^{-1} such that
 $a^{-1} \cdot (a \cdot b) = (a^{-1} \cdot a) \cdot b = e \cdot b = b = 0$

Galois, one of the earliest people, first studied group, ring, field, for proving that polynomials with order 5 or higher do not have any explicit root formula.



The beginning of the modern mathematics.

Finite fields are also called ~~finite~~ Galois fields.

A field $\mathbb{F}$ with q elements is denoted by $GF(q)$

* Let $a \neq 0 \in GF(q)$. Then

$$a^1 = a, a^2 = a \cdot a, a^3 = a \cdot a \cdot a, \dots$$

* There are two positive integers $m < n$ such that

$$a^m = a^n \Rightarrow a^{n-m} = 1$$

* The smallest positive integer n such that $a^n = 1$ is called the order of the field element a .

* Different elements may have different orders

$$1 = a^0, a^1, a^2, \dots, a^{n-1} \text{ are } \underline{\text{distinct}}.$$

* Let n be the order of an non-zero[?] element $a \in GF(q)$. Then $n | (q-1)$, n divides $q-1$.

* If the order of an element $a \neq 0 \in \mathbb{F}$ is $q-1$, this element a is called primitive.

In this case, $GF(q) = \{0, a^0=1, a^1, \dots, a^{q-2}\}$

power representation



Consider $GF(5)$

$$2^0=1, 2^1=2, 2^2=4, 2^3=3, 2^4=1$$

$\Rightarrow$ the order of 2 is $4=5-1$

$\Rightarrow$ 2 is primitive.

$$3^0=1, 3^1=3, 3^2=4, 3^3=2, 3^4=1$$

$\Rightarrow$ the order of 3 is $4=5-1$

$\Rightarrow$ 3 is primitive

$$4^0=1, 4^1=4, 4^2=1 \Rightarrow \text{the order of 4 is } 2 \neq 5-1$$

$\Rightarrow$ 4 is not primitive.

For a field, all linear algebra operations can be implemented, such as, vector, matrix, determinant etc.

* For a field, we may define a polynomial

$$f(x) = f_0 + f_1 x + f_2 x^2 + \dots + f_n x^n$$

where x is a variable.

All polynomial operations can be done similarly.

If $f(a)=0$, a is called a root of $f(x)$.

If F is the complex field $\mathbb{C}$, any $f(x)$ has roots.

It is not true for other fields.



Field extension: Rational number field $\mathbb{Q}$

$\Downarrow$
Real number field $\mathbb{R}$ $z x^2 = 1$

$\Downarrow$
Complex number field $\mathbb{C} = \mathbb{R}(j)$ $x^2 = -1$

§3 Finite field $GF(2^m)$

Now we consider the most useful finite fields $GF(2)$ and its extensions $GF(2^m)$.

We know how to do operations of 0 & 1 on $GF(2)$

Question: how to do the algebraic operations for vectors of 0 and 1 such as

$$[001] \cdot [110] = ?$$

$$[001] + [110] = [111]$$

$$[001] \div [110] = ?$$

$\Downarrow$

Finite fields $GF(2^m)$ where m is the binary vector length.



* Consider polynomials with coefficients on $GF(2)$, i.e., binary coefficients. Some of them may have binary roots, such as $1+x^2 = (1+x)^2$.

Some of them may not have binary roots, such as $1+x+x^2$.

* A polynomial $p(x)$ over $GF(2)$ (with binary coefficients) of degree m is said to be irreducible over $GF(2)$ if $p(x)$ is not divisible by any polynomial over $GF(2)$ of degree less than m but greater than zero.

Ex: x^2+x+1 does not have any 0 or 1 roots

$\Rightarrow x^2+x+1$ is an irreducible polynomial over $GF(2)$.

Theorem: Any irreducible polynomial over $GF(2)$ of degree m divides $x^{2^m-1}+1$.

Any irreducible polynomial $p(x)$ of degree m is said to be primitive if the smallest positive integer n for which $p(x)$ divides x^n+1 is $n=2^m-1$.



List of Primitive Polynomials

m	
3	$1+x+x^3$
4	$1+x+x^4$
5	$1+x^2+x^5$
6	$1+x+x^6$
7	$1+x^3+x^7$
8	$1+x^2+x^3+x^4+x^8$

For a given m, there may be more than one primitive polynomials of degree m

There books listing all primitive polynomials.

* Construction of Galois Field $GF(2^m)$

$GF(2^m)$ is extended from $GF(2)$ by introducing a new symbol α with operations

$$0 \cdot 0 = 0$$

$$0 \cdot 1 = 1 \cdot 0 = 0$$

$$1 \cdot 1 = 1$$

$$0 \cdot \alpha = \alpha \cdot 0 = 0$$

$$1 \cdot \alpha = \alpha \cdot 1 = \alpha$$

$$\alpha^2 = \alpha \cdot \alpha$$

$$\alpha^3 = \alpha \cdot \alpha \cdot \alpha$$

$$\vdots$$

$$\alpha^j = \alpha \cdot \alpha \cdot \dots \cdot \alpha \quad (j \text{ times})$$

$$\vdots$$

$$F = \{0, 1, \underset{\uparrow \alpha^0}{\alpha}, \alpha^2, \dots, \alpha^j, \dots\}$$



We next impose a condition on the element α so that $\mathbb{F}$ has only 2^m elements.

Let $p(x)$ be a primitive polynomial of degree m over $\mathbb{GF}(2)$. Assume $p(\alpha) = 0$.

By Theorem, $p(x) \mid (x^{2^m-1} + 1)$, i.e.,

$$x^{2^m-1} + 1 = q(x) p(x)$$

$$\Rightarrow \alpha^{2^m-1} + 1 = q(\alpha) p(\alpha) = 0$$

$$\Rightarrow \alpha^{2^m-1} = 1 \quad (-1 = 1)$$

$$\Rightarrow \mathbb{F} = \{0, 1, \alpha, \dots, \alpha^{2^m-2}\} \quad \leftarrow \begin{array}{l} \text{I} \\ \text{power representation} \\ \text{of } \mathbb{GF}(2^m) \end{array}$$

Since the polynomial $p(x)$ is primitive, all the elements in $\mathbb{F}$ are distinct. $\mathbb{F} \Leftrightarrow \alpha \Leftrightarrow p(x)$

α is a primitive element

* Vector Representation and Polynomial Representation

For $0 \leq i \leq 2^m - 1$,

$$x^i = q_i(x) p(x) + a_i(x),$$

$$\text{where } a_i(x) = a_{i,0} + a_{i,1}x + \dots + a_{i,m-1}x^{m-1}$$

$$= x^i \bmod p(x),$$

all coefficients $a_{i,k}$ are binary in $\mathbb{GF}(2)$

polynomial $\Leftrightarrow$ vector
(coefficients form vector)



$$\Rightarrow \alpha^i = f_i(\alpha) p(\alpha) + a_i(\alpha)$$

$$\alpha^i = a_i(\alpha) = a_{i,0} + a_{i,1}\alpha + \dots + a_{i,m-1}\alpha^{m-1}$$

$$0 \leq i < 2^m - 1$$

$$0 = 0$$

↑ polynomial representation

Also

$$\begin{aligned} \alpha^i &\leftrightarrow [a_{i,0}, a_{i,1}, \dots, a_{i,m-1}] \\ 0 &\leftrightarrow [0, 0, \dots, 0] \end{aligned}$$

II

binary vectors

$a_{i,k} \in \mathbb{GF}(2)$, binary

↑ vector representations

III

$$\alpha^i + \alpha^j = [a_{i,0}, a_{i,1}, \dots, a_{i,m-1}] + [a_{j,0}, a_{j,1}, \dots, a_{j,m-1}]$$

$$= [a_{i,0} + a_{j,0}, a_{i,1} + a_{j,1}, \dots, a_{i,m-1} + a_{j,m-1}]$$

$$\alpha^i \cdot \alpha^j = \alpha^{i+j} = [a_{i,0}, a_{i,1}, \dots, a_{i,m-1}] \cdot [a_{j,0}, a_{j,1}, \dots, a_{j,m-1}]$$

$$= [a_{i+j,0}, a_{i+j,1}, \dots, a_{i+j,m-1}]$$

$$\uparrow i+j = i+j \bmod (2^m - 1)$$

↑ This way defines a field for m dimensional binary vectors.

Binary vector divisions $\leftrightarrow$ power division

$$\begin{aligned} \alpha^{-j} &= [a_{j,0}, a_{j,1}, \dots, a_{j,m-1}]^{-1} = \alpha^{-j \bmod (2^m - 1)} \\ &= [a_{2^m-1-j,0}, \dots, a_{2^m-1-j,m-1}] = \alpha^{2^m-1-j} \end{aligned}$$



Example: Three representations for $GF(2^4)$ with primitive polynomial $p(x) = 1 + x + x^4$

<u>power representation</u>	<u>polynomial representation</u>	<u>vector representation</u>
0	0	[0 0 0 0]
1	1	[1 0 0 0]
α	α	[0 1 0 0]
α^2	α^2	[0 0 1 0]
α^3	α^3	[0 0 0 1]
$\alpha^4 (= 1 + \alpha)$	$1 + \alpha$	[1 1 0 0]
$\alpha^5 (= \alpha(\alpha^4))$	$\alpha + \alpha^2$	[0 1 1 0]
α^6	$\alpha^2 + \alpha^3$	[0 0 1 1]
$\alpha^7 = \alpha(\alpha^4 + \alpha^3)$ $= \alpha^5 + \alpha^4$	$1 + \alpha + \alpha^3$	[1 1 0 1]
α^8	$1 + \alpha^2$	[1 0 1 0]
α^9	$\alpha + \alpha^3$	[0 1 0 1]
α^{10}	$1 + \alpha + \alpha^2$	[1 1 1 0]
α^{11}	$\alpha^2 + \alpha^2 + \alpha^3$	[0 1 1 1]
α^{12}	$1 + \alpha + \alpha^2 + \alpha^3$	[1 1 1 1]
α^{13}	$1 + \alpha^2 + \alpha^3$	[1 0 1 1]
α^{14}	$1 + \alpha^3$	[1 0 0 1]



How to do binary vector multiplications divisions?

For example:

$$[0110] \cdot [1010]$$

$$= \alpha^5 \cdot \alpha^8 = \alpha^{13} = [1011]$$

$$[0110] \div [1010]$$

$$= \alpha^{-3 \bmod (2^4-1)}$$

$$= \alpha^5 \div \alpha^8 = \alpha^{5-8} = \alpha^{-3} = \alpha^{-3+15} = \alpha^{12}$$

$$= [1111]$$

$$[1110] \cdot [1011]$$

$$= \alpha^{10} \cdot \alpha^{13} = \alpha^{23} = \alpha^{23 \bmod 15} = \alpha^8$$

$$= [1010]$$

$$[1110] \div [1011]$$

$$= \alpha^{10} \div \alpha^{13} = \alpha^{10-13} = \alpha^{-3} = \alpha^{12}$$

$$= [1111]$$

So you may do 4x1 binary vector additions, subtractions, multiplications, and divisions, all operation of linear processes.



§4. Reed-Solomon (RS) Codes

RS Codes: A special class of non-binary BCH codes
Bose, Chaudhuri & Hocquenghem

Cyclic codes over a field $\mathbb{F}$ of rate k/n

generator polynomial $g(x)$ of degree $(n-k)-1$

information polynomial $p(x)$ of degree $k-1$

code polynomial $q(x)$ of degree $n-1$

Encoding

$$q(x) = g(x) \cdot p(x)$$

$$G = \begin{bmatrix} g_0 & g_1 & \dots & g_{n-k-1} & & \\ & g_0 & \dots & g_{n-k-1} & & \\ & & \dots & & & \\ & & & g_0 & \dots & g_{n-k-1} \end{bmatrix}$$

General Cyclic Codes

RS Codes

Consider the Galois field $GF(2^m)$. Let α be a primitive element, a root of a primitive polynomial $p(x)$, in $GF(2^m)$.

The generator polynomial of a primitive t -error-correcting Reed-Solomon code of length 2^m-1 is

$$\begin{aligned} g(x) &= (x+\alpha)(x+\alpha^2) \dots (x+\alpha^{2t}) \\ &= g_0 + g_1x + \dots + g_{2t}x^{2t-1} + x^{2t} \end{aligned}$$



Parameters

Block length : $n = q - 1 = 2^m - 1$ (symbol length)
non-binary

number of parity-check symbols: $n - k = 2t$

minimum distance : $d_{\min} = 2t + 1$

The parity-check matrix

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^{n-1} \\ 1 & \alpha^2 & (\alpha^2)^2 & \dots & (\alpha^2)^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{2t} & (\alpha^{2t})^2 & \dots & (\alpha^{2t})^{n-1} \end{bmatrix}$$

any $2t$ columns vectors
are linearly independent
 $\Rightarrow d_{\min} = 2t + 1$
Vandermonde matrix

Why? Let $r(x) = r_0 + r_1 x + \dots + r_{n-1} x^{n-1}$ be a received polynomial (received vector $(r_0, r_1, \dots, r_{n-1}) = \underline{r}$).

If $r(x)$ can be represented by $r(x) = g(x) p(x)$

for some degree $n-k-1$ polynomial $p(x)$, then $\underline{r}$ is a codeword, i.e., $r(x)$ has all roots of $g(x)$, or $r(x)$ can be divided by $g(x)$.

Since $g(x)$ has all roots $\alpha, \alpha^2, \dots, \alpha^{2t}$

$\Rightarrow$ We need to check whether

$$r(\alpha) = 0?$$

$$r(\alpha^2) = 0?$$

$$r(\alpha^{2t}) = 0?$$



or $r_0 + r_1 \alpha^l + r_2 (\alpha^l)^2 + \dots + r_{n-1} (\alpha^l)^{n-1} = 0?$
for $l=1, 2, \dots, 2t$.

or $\underline{r} H' = 0?$

↑ parity-check matrix

* Fourier transform interpretation of the RS codes

For any $1 \times n$ vector $\underline{r} = (r_0, r_1, \dots, r_{n-1})$ on $GF(2^m)$,
it is a codeword in ^{the} t -error correcting RS code iff

$$\sum_{m=0}^{n-1} r_m (\alpha^l)^m = 0 \quad \text{for } l=1, 2, \dots, 2t$$

or $\sum_{m=0}^{n-1} r_m 2^{lm} = 0, \quad l=1, 2, \dots, 2t.$

Total $l: 1, 2, \dots, n$

DFT $S(l) = \sum_{m=0}^{n-1} r_m W_N^{lm}$, where $W_N = e^{-\frac{2\pi j}{N}}$

Set $S(l)=0$ for $l=1, 3 \dots 2t$
(set low frequency = 0)



§5. Reed-Solomon Code Decoding

Consider t -error-correction RC code on $GF(2^m)$

$$n = 2^m - 1$$

$$n - k = 2t$$

Let $\underline{r} = (r_0, r_1, \dots, r_{n-1})$ be a received vector

$\underline{e} = (e_0, e_1, \dots, e_{n-1})$ be a noise vector

$$\underline{r} = \underline{c} + \underline{e}, \quad \text{where } \underline{c} \text{ is a codeword}$$

The goal of a decoder is to recover $\underline{c}$ (or equivalently $\underline{e}$) from $\underline{r}$.

Let τ be the weight of the noise vector $\underline{e}$, i.e., the number of errors in the received vector.

To decode $\underline{c}$, it must $\tau \leq t$.

Let $0 \leq i_1 < i_2 < \dots < i_\tau \leq n-1$ be the indices of the locations where $e_{i_j} \neq 0$ (error locations).

For convenience, we recast the error vector $\underline{e}$ as a set of error locations $\underline{x}_j = \alpha^{-i_j}, 1 \leq j \leq \tau$

and a set of error values

$$y_j = e_{i_j}, \quad 1 \leq j \leq \tau.$$



Define the error locator polynomial

$$G(x) = \prod_{j=1}^2 (x - \alpha_j) = 1 + \sum_{j=1}^2 \alpha_j x$$

and the error evaluator polynomial

$$W(x) = \prod_{i=1}^2 Y_i \prod_{l \neq i} (x - \alpha_l) = \sum w_j x^j$$

interpolating

$$W(\alpha_j) = Y_j, \quad j=1, \dots, 2.$$

The decoding process for RS codes is divided into three main stages

1. Syndrome Computation:

$$r(x) = \sum_{i=0}^{n-1} r_i x^i \quad \text{received vector polynomial}$$

$$S_j = r(\alpha^j) = \sum_{i=0}^{n-1} r_i (\alpha^j)^i, \quad 1 \leq j \leq 2t$$

$$= \sum_{i=0}^{n-1} e_i (\alpha^j)^i$$

$$= \sum_{l=1}^2 e_{lj} (\alpha^j)^{l_i}$$

2. Finding the error locator and error evaluator polynomials:



The error locator and error evaluator polynomials satisfy the key equation $\sigma(x)S(x) = W(x) \mod x^{2t+1}$

Where $S(x) = \sum_{j=1}^{2t} S_j x^j$ in $\mathbb{Z}$ -domain

To solve the above key equations for $\sigma(x)$ and $W(x)$, there are many algorithms for the solutions.

The key equation can be rewritten (in the time domain)

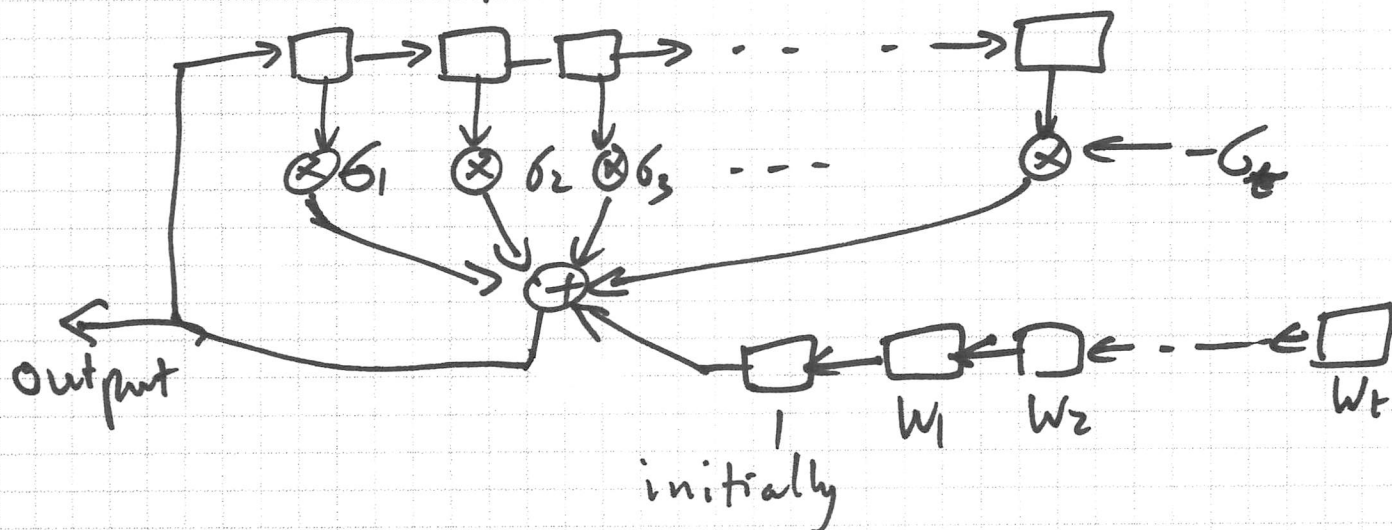
$$S_k + \sum_{i=1}^{k-1} \sigma_i S_{k-i} + \sigma_k = W_k$$

Massey introduced the linear feedback shift register interpretation.

Berlekamp - Massey Algorithm

Continued Fraction Algorithms

Berlekamp - Welch Algorithm



Similar to Toeplitz equation solving in signal processing
Levinson recursive algorithm



6-20

Extended Euclidean Algorithm

(1) Initialize $\delta_{(-1)}(x) = 0, \delta_{(0)}(x) = 1$

$$w_{(-1)}(x) = x^{2t+1}, w_{(0)}(x) = S(x)$$

(2) For $i = 0, 1, 2, \dots$; Compute $Q_{(i)}(x)$ and $w_{(i+1)}(x)$ satisfying
 $w_{(i+1)}(x) = w_{(i-1)}(x) - Q_{(i)}(x)w_{(i)}(x), \deg w_{(i+1)} < \deg w_{(i)}$

and Compute $\delta_{(i+1)}(x) = \delta_{(i-1)}(x) - Q_{(i)}(x)\delta_{(i)}(x)$

Stop at $i+1 = j$, where j is the smallest integer such that $\deg w_{(j)} \leq t$

(3) Set $\delta(x) = \delta_{(j)}(x), w(x) = w_{(j)}(x)$

3. Search for error locations and Computations of error values:

Evaluate $\delta(x)$ at $x = \alpha^{-i}$ for $0 \leq i \leq n-1$
 (Chien search)

Values of x for which $\delta(x) = 0$ corresponds to the error location $\mathcal{E}_j$.

Once an error location $\mathcal{E}_j$ is found, the corresponding error value is either 1 or

$$\gamma_j = \frac{w(\mathcal{E}_j)}{\delta'(\mathcal{E}_j)}$$

HWK: Calculate $(1, 0, 1) \times (0, 1, 1) = ?$
 $(1, 0, 1) \div (0, 1, 1) = ?$