4. Algebra over GF(q); Reed-Solomon and cyclic linear codes

Coding Technology

Axioms of GF(q)

GF(q) is the Galois field (or finite field) with q elements.

Field axioms

Addition "+"	Multiplication "$*$"
$\alpha, \beta \in GF(q) \to \alpha + \beta \in GF(q)$	$\alpha, \beta \in GF(q) \to \alpha * \beta \in GF(q)$
$\alpha + \beta = \beta + \alpha$	$\alpha * \beta = \beta * \alpha$
$(\alpha + \beta) + \gamma = \alpha + (\beta + \gamma)$	$(\alpha * \beta) * \gamma = \alpha * (\beta * \gamma)$
$\exists\, 0 : \forall \alpha \in GF(q) : \alpha + 0 = \alpha$	$\exists\, 1 : \forall \alpha \in GF(q) : \alpha * 1 = \alpha$
$\forall \alpha \in GF(q)\, \exists \beta : \alpha + \beta = 0;$	$\forall \alpha \in GF(q) \backslash \{0\} : \exists \beta : \alpha * \beta = 1;$
$\beta = \alpha_a^{-1} = -\alpha$	$\beta = \alpha_m^{-1} = \alpha^{-1}$

$$\alpha * (\beta + \gamma) = \alpha * \beta + \alpha * \gamma$$

Liberty to define "+" and "$*$" as long as they satisfy the above axioms.

Examples of GF(q)

q can be either a prime or p^m (with p prime and $m \geq 2$).

We focus on the q prime case first. When q is a prime, GF(q) has the mod q arithmetics:

$$GF(q) = \{0, 1, \ldots, q-1\},$$

and

$$\begin{aligned} \alpha + \beta &= \alpha + \beta \mod q, \\ \alpha * \beta &= \alpha \cdot \beta \mod q. \end{aligned}$$

Examples in GF(7):

$$\begin{array}{ll} 6 + 5 = 4 \mod 7 & (6 + 5 = 11 = 4 \mod 7) \\ 6 * 5 = 2 \mod 7 & (6 \cdot 5 = 30 = 2 \mod 7) \\ -4 = 3 \mod 7 & (4 + 3 = 7 = 0 \mod 7) \\ 4^{-1} = 2 \mod 7 & (4 \cdot 2 = 8 = 1 \mod 7) \end{array}$$

Power table

Basic property: $\forall \alpha \in GF(q) \backslash \{0\} : \alpha^{q-1} = 1$.

The order of α is the minimal m for which $\alpha^m = 1$. If $m = q - 1$, we call α a primitive element.

Power table

Basic property: $\forall \alpha \in GF(q) \backslash \{0\} : \alpha^{q-1} = 1$.

The order of α is the minimal m for which $\alpha^m = 1$. If $m = q - 1$, we call α a primitive element.

element α	powers α^1	α^2	α^3	α^4	α^5	α^6	order m	
1	1						1	
2	2	4	1				3	
3	3	2	6	5	4	1	6	– primitive element
4	4	2	1				3	
5	5	4	6	2	3	1	6	– primitive element
6	6	1					2	

The powers of a primitive element give all nonzero elements in GF(q).

Polynomials over GF(q)

$\alpha(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \cdots + \alpha_m x^m$; $\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_m \in GF(q)$

Roots $x_1, \ldots, x_m$: $\alpha(x_i) = 0$, $i = 1, \ldots, m$

number of roots $\leq \deg(\alpha(x)) = m$

If $\alpha(x)$ has $\deg(\alpha(x)) = m$ roots $x_1, \ldots, x_m$, then

$$\alpha(x) = \alpha_m \prod_{i=1}^{m} (x - x_i).$$

Polynomial division: given $\alpha(x)$ and $d(x)$ with $\deg(\alpha(x)) = m > \deg(d(x)) = k$,

$$\exists q(x), r(x) : \alpha(x) = q(x)d(x) + r(x); \quad \deg(r(x)) < k.$$

$$a(x), d(x) \;\rightarrow\; \underset{m-k \text{ steps}}{\text{Euclidean division algorithm}} \;\rightarrow\; q(x), r(x)$$

Problem 1

What is the additive inverse of 2 in GF(5)?

Problem 1

What is the additive inverse of 2 in GF(5)?

Solution. $2 + 3 = 1 \cdot 5 + 0$, so the additive inverse of 2 in GF(5) is

$$-2 = 2_a^{-1} = 3.$$

Problem 2

What is the multiplicative inverse of 2 in GF(5)?

Problem 2

What is the multiplicative inverse of 2 in GF(5)?

Solution. $2 \cdot 3 = 1 \cdot 5 + 1$, that is,

$$2 * 3 = 1 \mod 5,$$

so the multiplicative inverse of 2 in GF(5) is

$$2^{-1} = 2_m^{-1} = 3.$$

Problem 3

What is the additive inverse of 5 in GF(11)?

Problem 3

What is the additive inverse of 5 in GF(11)?

Solution. $5 \cdot 6 = 1 \cdot 11 + 0$, so the additive inverse of 5 in GF(11) is

$$-5 = 5_a^{-1} = 6.$$

Problem 4

What is the multiplicative inverse of 7 in GF(11)?

Problem 4

What is the multiplicative inverse of 7 in GF(11)?

Solution. $7 \cdot 8 = 5 \cdot 11 + 1$, that is,

$$7 * 8 = 1 \mod 11,$$

so the multiplicative inverse of 7 in GF(11) is

$$7^{-1} = 7_m^{-1} = 8.$$

Problem 5

Solve the equation $6x + 5 = 2$ in GF(7).

Problem 5

Solve the equation $6x + 5 = 2$ in GF(7).

Solution.

$$\begin{aligned} 6x + 5 &= 2 \\ 6x &= 2 - 5 \\ 6x &= -3 \\ 6x &= 4 \\ x &= 6^{-1} * 4 \\ x &= 6 * 4 \\ x &= 24 \\ x &= 3. \end{aligned}$$

Reed-Solomon codes

Let $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$ be distinct nonzero elements of GF(q), where $n = q - 1$.

Then the corresponding $C(n, k)$ Reed-Solomon code over GF(q) is a linear code with generator matrix

$$G = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ \alpha_0 & \alpha_1 & \alpha_2 & \ldots & \alpha_{n-1} \\ \vdots & & & \ddots & \vdots \\ \alpha_0^{k-1} & \alpha_1^{k-1} & \alpha_2^{k-1} & \ldots & \alpha_{n-1}^{k-1} \end{bmatrix}$$

Reed-Solomon codes

Let $\alpha_0, \alpha_1, \dots, \alpha_{n-1}$ be distinct nonzero elements of $\mathrm{GF}(q)$, where $n = q - 1$.

Then the corresponding $C(n, k)$ Reed-Solomon code over $\mathrm{GF}(q)$ is a linear code with generator matrix

$$G = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ \alpha_0 & \alpha_1 & \alpha_2 & \dots & \alpha_{n-1} \\ \vdots & & & \ddots & \vdots \\ \alpha_0^{k-1} & \alpha_1^{k-1} & \alpha_2^{k-1} & \dots & \alpha_{n-1}^{k-1} \end{bmatrix}$$

RS codes have the MDS property:

$$d_{\min} = n - k + 1,$$

so the code can

- detect $n - k$ errors, and
- correct $\left\lfloor \frac{n-k}{2} \right\rfloor$ errors.

Reed-Solomon codes

Special case: RS code generated by a primitive element α. If we choose $\alpha_i = \alpha^i$, then

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

and its parity check matrix is

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \dots & \alpha^{2(n-1)} \\ \vdots & & & \ddots & \vdots \\ 1 & \alpha^{n-k} & \alpha^{2(n-k)} & \dots & \alpha^{(n-k)(n-1)} \end{bmatrix}.$$

Problem 6

Design an RS code over GF(7) that corrects every double error.

Problem 6

Design an RS code over GF(7) that corrects every double error.

Solution. First we want to compute the parameters (n, k). The error correcting capability is

$$t = \left\lfloor \frac{n-k}{2} \right\rfloor = 2 \qquad \rightarrow \qquad n - k = 4.$$

Problem 6

Design an RS code over GF(7) that corrects every double error.

Solution. First we want to compute the parameters (n, k). The error correcting capability is

$$t = \left\lfloor \frac{n-k}{2} \right\rfloor = 2 \qquad \rightarrow \qquad n - k = 4.$$

Next, $n = q - 1 = 6$, so

$$(n, k) = (6, 2).$$

Problem 6

Any C(6,2) RS code over GF(7) is suitable; for example, for the RS code generated by the primitive element 5, we have

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 5 & 4 & 6 & 2 & 3 \end{bmatrix}$$

and

$$H = \begin{bmatrix} 1 & 5 & 4 & 6 & 2 & 3 \\ 1 & 4 & 2 & 1 & 4 & 2 \\ 1 & 6 & 1 & 6 & 1 & 6 \\ 1 & 2 & 4 & 1 & 2 & 4 \end{bmatrix}.$$

Problem 7

Using the previous code, determine the codewords assigned to the message vectors $u = (4, 4)$, $u = (3, 5)$ and $u = (5, 1)$.

Problem 7

Using the previous code, determine the codewords assigned to the message vectors $u = (4, 4)$, $u = (3, 5)$ and $u = (5, 1)$.

Solution.

$$(4\,4) \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 5 & 4 & 6 & 2 & 3 \end{bmatrix} = (1\,3\,6\,0\,5\,2)$$

Problem 7

Using the previous code, determine the codewords assigned to the message vectors $u = (4, 4)$, $u = (3, 5)$ and $u = (5, 1)$.

Solution.

$$(4\,4) \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 5 & 4 & 6 & 2 & 3 \end{bmatrix} = (1\,3\,6\,0\,5\,2)$$

$$(3\,5) \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 5 & 4 & 6 & 2 & 3 \end{bmatrix} = (1\,0\,2\,5\,6\,4)$$

Problem 7

Using the previous code, determine the codewords assigned to the message vectors $u = (4, 4)$, $u = (3, 5)$ and $u = (5, 1)$.

Solution.

$$(4\,4) \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 5 & 4 & 6 & 2 & 3 \end{bmatrix} = (1\,3\,6\,0\,5\,2)$$

$$(3\,5) \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 5 & 4 & 6 & 2 & 3 \end{bmatrix} = (1\,0\,2\,5\,6\,4)$$

$$(5\,1) \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 5 & 4 & 6 & 2 & 3 \end{bmatrix} = (6\,3\,2\,4\,0\,1)$$

Problem 8

Give the generator matrix and parity check matrix of a RS code capable of correcting every single error over GF(5), using the primitive element 2.

Problem 8

Give the generator matrix and parity check matrix of a RS code capable of correcting every single error over GF(5), using the primitive element 2.

Solution. For the error correcting capability, we have

$$t = \left\lfloor \frac{n-k}{2} \right\rfloor = 1 \qquad \rightarrow \qquad n-k = 2.$$

Problem 8

Give the generator matrix and parity check matrix of a RS code capable of correcting every single error over GF(5), using the primitive element 2.

Solution. For the error correcting capability, we have

$$t = \left\lfloor \frac{n-k}{2} \right\rfloor = 1 \qquad \rightarrow \qquad n-k=2.$$

Due to $q=5$, we have $n=q-1=4$, so $(n,k)=(4,2)$, and

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 3 \end{bmatrix} \qquad H = \begin{bmatrix} 1 & 2 & 4 & 3 \\ 1 & 4 & 1 & 4 \end{bmatrix}.$$

Problem 9

A C(10,4) RS code over GF(11) has generator matrix

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 6 & 3 & 7 & 9 & 10 & 5 & 8 & 4 & 2 \\ 1 & 3 & 9 & 5 & 4 & 1 & 3 & 9 & 5 & 4 \\ 1 & 7 & 5 & 2 & 3 & 10 & 4 & 6 & 9 & 8 \end{bmatrix}$$

(a) How many errors can the code correct?

(b) What is the primitive element used?

(c) Calculate the parity check matrix H.

Problem 9

Solution.

(a) This is a RS code, so the code can correct $\left\lfloor \frac{n-k}{2} \right\rfloor = 3$ errors.

Problem 9

Solution.

(a) This is a RS code, so the code can correct $\left\lfloor \frac{n-k}{2} \right\rfloor = 3$ errors.

(b) The primitive element used is 6:

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 6 & 3 & 7 & 9 & 10 & 5 & 8 & 4 & 2 \\ 1 & 3 & 9 & 5 & 4 & 1 & 3 & 9 & 5 & 4 \\ 1 & 7 & 5 & 2 & 3 & 10 & 4 & 6 & 9 & 8 \end{bmatrix}$$

Problem 9

Solution.

(a) This is a RS code, so the code can correct $\left\lfloor \frac{n-k}{2} \right\rfloor = 3$ errors.

(b) The primitive element used is 6:

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 6 & 3 & 7 & 9 & 10 & 5 & 8 & 4 & 2 \\ 1 & 3 & 9 & 5 & 4 & 1 & 3 & 9 & 5 & 4 \\ 1 & 7 & 5 & 2 & 3 & 10 & 4 & 6 & 9 & 8 \end{bmatrix}$$

(c)

$$H = \begin{bmatrix} 1 & 6 & 3 & 7 & 9 & 10 & 5 & 8 & 4 & 2 \\ 1 & 3 & 9 & 5 & 4 & 1 & 3 & 9 & 5 & 4 \\ 1 & 7 & 5 & 2 & 3 & 10 & 4 & 6 & 9 & 8 \\ 1 & 9 & 4 & 3 & 5 & 1 & 9 & 4 & 3 & 5 \\ 1 & 10 & 1 & 10 & 1 & 10 & 1 & 10 & 1 & 10 \\ 1 & 5 & 3 & 4 & 9 & 1 & 5 & 3 & 4 & 9 \end{bmatrix}$$

Problem 10

The parity check matrix of a RS code over GF(7) is

$$H = \begin{bmatrix} 1 & 3 & 2 & 6 & 4 & 5 \\ 1 & 2 & 4 & 1 & 2 & 4 \\ 1 & 6 & 1 & 6 & 1 & 6 \\ 1 & 4 & 2 & 1 & 4 & 2 \end{bmatrix}$$

(a) What is the type of the code (n and k parameters)?

(a) How many errors can the code correct?

(c) Determine the codeword assigned to the message vector which contains only 2's.

Problem 10

Solution.

(a) The parity check matrix H for a $C(n, k)$ RS code has size $(n-k) \times n$. In this case, H is 4×6, so $(n, k) = (6, 2)$.

Problem 10

Solution.

(a) The parity check matrix H for a C(n, k) RS code has size $(n-k) \times n$. In this case, H is 4×6, so $(n, k) = (6, 2)$.

(b) It is a RS code, so the error correcting capability is $\left\lfloor \frac{n-k}{2} \right\rfloor = 2$.

Problem 10

Solution.

(a) The parity check matrix H for a $C(n,k)$ RS code has size $(n-k)\times n$. In this case, H is 4×6, so $(n,k)=(6,2)$.

(b) It is a RS code, so the error correcting capability is $\left\lfloor \frac{n-k}{2} \right\rfloor = 2$.

(c) This code is generated by the primitive element 3, so

$$G=\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 3 & 2 & 6 & 4 & 5 \end{bmatrix},$$

and

$$c = uG = (2\,2)\cdot\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 3 & 2 & 6 & 4 & 5 \end{bmatrix} = (4\,1\,6\,0\,3\,5).$$

Problem 11

A C(6,3) RS code is generated by the largest primitive element belonging to the field.

(a) Give the generator matrix G.

(b) Give the parity check matrix H.

(c) How many errors can be detected using this code? How many errors can be corrected?

Problem 11

Solution.

(a) The value of q is not given directly, but from $n = q - 1$, we can deduce $q = 7$. The largest primitive element in GF(7) is 5, so

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 5 & 4 & 6 & 2 & 3 \\ 1 & 4 & 2 & 1 & 4 & 2 \end{bmatrix}$$

Problem 11

Solution.

(a) The value of q is not given directly, but from $n = q - 1$, we can deduce $q = 7$. The largest primitive element in GF(7) is 5, so

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 5 & 4 & 6 & 2 & 3 \\ 1 & 4 & 2 & 1 & 4 & 2 \end{bmatrix}$$

(b)

$$H = \begin{bmatrix} 1 & 5 & 4 & 6 & 2 & 3 \\ 1 & 4 & 2 & 1 & 4 & 2 \\ 1 & 6 & 1 & 6 & 1 & 6 \end{bmatrix}$$

Problem 11

Solution.

(a) The value of q is not given directly, but from $n = q - 1$, we can deduce $q = 7$. The largest primitive element in GF(7) is 5, so

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 5 & 4 & 6 & 2 & 3 \\ 1 & 4 & 2 & 1 & 4 & 2 \end{bmatrix}$$

(b)

$$H = \begin{bmatrix} 1 & 5 & 4 & 6 & 2 & 3 \\ 1 & 4 & 2 & 1 & 4 & 2 \\ 1 & 6 & 1 & 6 & 1 & 6 \end{bmatrix}$$

(c) The code can

- detect $n - k = 3$ errors, and
- correct $\left\lfloor \frac{n-k}{2} \right\rfloor = 1$ error.

Linear cyclic codes

A code is cyclic if for any codeword

$$c = (c_0\, c_1\, c_2 \ldots c_{n-1}),$$

its cyclically shifted version

$$Sc = (c_{n-1}\, c_0\, c_1 \ldots c_{n-2})$$

is also a codeword. S is the cyclic shift operator.

Linear cyclic codes

A code is cyclic if for any codeword

$$c = (c_0\, c_1\, c_2 \ldots c_{n-1}),$$

its cyclically shifted version

$$Sc = (c_{n-1}\, c_0\, c_1 \ldots c_{n-2})$$

is also a codeword. S is the cyclic shift operator.

The Reed-Solomon code generated by a single primitive element α is a cyclic linear code.

Linear cyclic codes

Example. The C(4,2) RS code over GF(5) that can correct 1 error has the following codewords:

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

Problem 12

A C(6,2) linear cyclic code over GF(7) can correct 2 errors. (6,0,3,5,4,1) is one of the codewords.

(a) Is (5,4,1,6,0,3) a codeword?

(b) Is (1,0,4,2,3,6) a codeword?

(c) Is (1,0,4,3,5,2) a codeword?

Problem 12

A C(6,2) linear cyclic code over GF(7) can correct 2 errors. (6,0,3,5,4,1) is one of the codewords.

(a) Is (5,4,1,6,0,3) a codeword?

(b) Is (1,0,4,2,3,6) a codeword?

(c) Is (1,0,4,3,5,2) a codeword?

Solution.

(a) Yes, because it is the cyclic shifted version of the given codeword (shifted 3 times).

Problem 12

A C(6,2) linear cyclic code over GF(7) can correct 2 errors. (6,0,3,5,4,1) is one of the codewords.

(a) Is (5,4,1,6,0,3) a codeword?

(b) Is (1,0,4,2,3,6) a codeword?

(c) Is (1,0,4,3,5,2) a codeword?

Solution.

(a) Yes, because it is the cyclic shifted version of the given codeword (shifted 3 times).

(b) Yes, because it is equal to the given codeword multiplied by 6.

Problem 12

A C(6,2) linear cyclic code over GF(7) can correct 2 errors. (6,0,3,5,4,1) is one of the codewords.

(a) Is (5,4,1,6,0,3) a codeword?

(b) Is (1,0,4,2,3,6) a codeword?

(c) Is (1,0,4,3,5,2) a codeword?

Solution.

(a) Yes, because it is the cyclic shifted version of the given codeword (shifted 3 times).

(b) Yes, because it is equal to the given codeword multiplied by 6.

(c) No, because the code can correct 2 errors $\rightarrow$ $d_{\min} \geq 5$, but the (b) and (c) vectors have Hamming-distance 3.

Code polynomials

We can assign code polynomials to codewords:

$$c = (c_0\, c_1\, c_2 \ldots c_{n-1}) \quad \to \quad c(x) = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}$$

Then the code polynomial assigned to Sc is

$$c'(x) = [xc(x)] \mod (x^n - 1).$$

Code polynomials

We can assign code polynomials to codewords:

$$c = (c_0\, c_1\, c_2 \ldots c_{n-1}) \quad \to \quad c(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$$

Then the code polynomial assigned to Sc is

$$c'(x) = [xc(x)] \mod (x^n - 1).$$

For any linear cyclic $\mathrm{C}(n, k)$ code, there exists a code polynomial $g(x)$ of degree $n - k$ such that all code polynomials are of the form

$$c(x) = u(x)g(x).$$

$g(x)$ is called the generator polynomial of $\mathrm{C}(n, k)$.

Code polynomials

We can assign code polynomials to codewords:

$$c = (c_0\, c_1\, c_2 \ldots c_{n-1}) \quad \to \quad c(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$$

Then the code polynomial assigned to Sc is

$$c'(x) = [xc(x)] \mod (x^n - 1).$$

For any linear cyclic $\mathrm{C}(n,k)$ code, there exists a code polynomial $g(x)$ of degree $n-k$ such that all code polynomials are of the form

$$c(x) = u(x)g(x).$$

$g(x)$ is called the generator polynomial of $\mathrm{C}(n,k)$.

$g(x) | x^n - 1$ always holds, and any such $g(x)$ is a suitable generator polynomial for a cyclic linear code.

Code polynomials

We similarly assign polynomials to message vectors too:

$$u = (u_0 \ \dots \ u_{k-1}) \quad \to \quad u(x) = u_0 + \cdots + u_{k-1}x^{k-1},$$

and also to error vectors e, received vectors v etc.

One (not the only!) way to make the $u(x) \to c(x)$ assignment is

$$c(x) = u(x)g(x).$$

Note that this is an assignment different from $c = uG$. It is not systematic either, but it can still be computed very efficiently using LFFSR and LFBSR architectures (coming soon).

We will stick to using $c(x) = u(x)g(x)$.

Linear cyclic codes

Example. C(4,2) RS code over GF(5) with the $c(x) = u(x)g(x)$ codeword assignment:

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

Linear cyclic codes

Example. C(4,2) RS code over GF(5) with systematic codeword assignment:

$$
\begin{array}{l|l}
(0\,0) \to (0\,0\,0\,0) & (2\,3) \to (2\,3\,0\,4) \\
(0\,1) \to (0\,1\,3\,2) & (2\,4) \to (2\,4\,3\,1) \\
(0\,2) \to (0\,2\,1\,4) & (3\,0) \to (3\,0\,4\,2) \\
(0\,3) \to (0\,3\,4\,1) & (3\,1) \to (3\,1\,2\,4) \\
(0\,4) \to (0\,4\,2\,3) & (3\,2) \to (3\,2\,0\,1) \\
(1\,0) \to (1\,0\,3\,4) & (3\,3) \to (3\,3\,3\,3) \\
(1\,1) \to (1\,1\,1\,1) & (3\,4) \to (3\,4\,1\,0) \\
(1\,2) \to (1\,2\,4\,3) & (4\,0) \to (4\,0\,2\,1) \\
(1\,3) \to (1\,3\,2\,0) & (4\,1) \to (4\,1\,0\,3) \\
(1\,4) \to (1\,4\,0\,2) & (4\,2) \to (4\,2\,3\,0) \\
(2\,0) \to (2\,0\,1\,3) & (4\,3) \to (4\,3\,1\,2) \\
(2\,1) \to (2\,1\,4\,0) & (4\,4) \to (4\,4\,4\,4) \\
(2\,2) \to (2\,2\,2\,2) &
\end{array}
$$

Code polynomials

The parity check polynomial corresponding to $g(x)$ is

$$h(x) = \frac{x^n - 1}{g(x)}.$$

The syndrome polynomial assigned to a received code polynomial $v(x)$ is

$$s(x) = v(x) \mod g(x) \quad \Longleftrightarrow \quad s(x) = v(x) : g(x)$$

A received polynomial $v(x)$ is a codeword $\iff$ $s(x) = 0$.

Code polynomials

The parity check polynomial corresponding to $g(x)$ is

$$h(x) = \frac{x^n - 1}{g(x)}.$$

The syndrome polynomial assigned to a received code polynomial $v(x)$ is

$$s(x) = v(x) \mod g(x) \quad \Longleftrightarrow \quad s(x) = v(x) : g(x)$$

A received polynomial $v(x)$ is a codeword $\Longleftrightarrow$ $s(x) = 0$.

The Reed-Solomon code generated by a single primitive element α has generator polynomial and parity check polynomial

$$g(x) = \prod_{i=1}^{n-k}(x - \alpha^i), \qquad h(x) = \prod_{i=n-k+1}^{n}(x - \alpha^i).$$

Code polynomials

Example. The C(4,2) RS code over GF(5) that can correct 1 error has generator polynomial

$$g(x) = (x - 2^1)(x - 2^2) = (x - 2)(x - 4).$$

Some examples of code polynomials:

$$\begin{aligned}
(1\,2\,4\,3) &\to 1 + 2x + 4x^2 + 3x^3 = (4 + 3x)(x - 2)(x - 4), \\
(0\,3\,4\,1) &\to 3x + 4x^2 + x^3 = x(x - 2)(x - 4), \\
(4\,4\,4\,4) &\to 4 + 4x + 4x^2 + 4x^3 = (3 + 4x)(x - 2)(x - 4).
\end{aligned}$$

Problem 13

Give the generator polynomial and parity check polynomial of the cyclic C(6,2) RS code over GF(7) generated by the primitive element 3.

Problem 13

Give the generator polynomial and parity check polynomial of the cyclic C(6,2) RS code over GF(7) generated by the primitive element 3.

Solution.

$$g(x) = \prod_{i=1}^{n-k}(x - \alpha^i) = (x-3)(x-3^2)(x-3^3)(x-3^4) =$$

$$(x-3)(x-2)(x-6)(x-4) = (x^2+2x+6)(x^2+4x+3) =$$

$$x^4 + 6x^3 + 3x^2 + 2x + 4.$$

$$h(x) = \prod_{i=n-k+1}^{n}(x - \alpha^i) = (x-3^5)(x-3^6) =$$

$$(x-5)(x-1) = x^2 + x + 5.$$

Problem 14

Using the previous code, calculate the codewords for the message vectors $(1\,1)$ and $(0\,2)$.

Problem 14

Using the previous code, calculate the codewords for the message vectors $(1\,1)$ and $(0\,2)$.

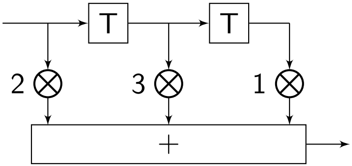
Solution.

$$c_1(x) = u_1(x)g(x) = (1+x)(4+2x+3x^2+6x^3+x^4) =$$
$$4+6x+5x^2+2x^3+0\cdot x^4+x^5 \;\rightarrow\; c_1 = (4\,6\,5\,2\,0\,1)$$
$$c_2(x) = u_2(x)g(x) = (0+2x)(4+2x+3x^2+6x^3+x^4) =$$
$$0+1\cdot x+4x^2+6x^3+5x^4+2x^5 \;\rightarrow\; c_2 = (0\,1\,4\,6\,5\,2)$$

(We also note that $c_2 = S^2 c_1$.)

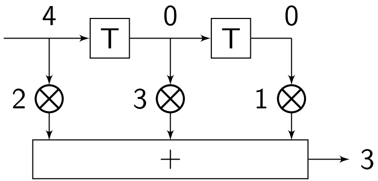
Polynomial multiplication by LFFSR

The Linear FeedForward Shift Register architecture for multiplication by $2 + 3x + x^2$:



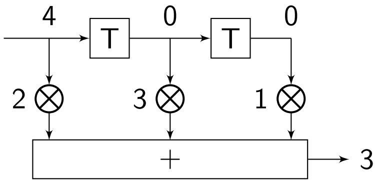
Polynomial multiplication by LFFSR

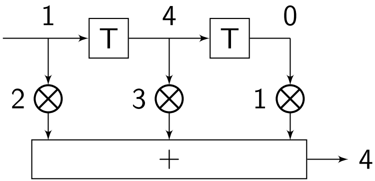
Compute $(2 + 3x + x^2)(4 + x)$ over GF(5):



Polynomial multiplication by LFFSR

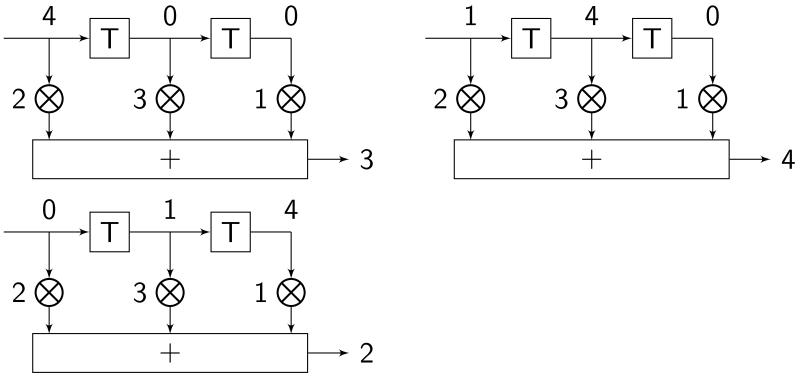
Compute $(2 + 3x + x^2)(4 + x)$ over GF(5):





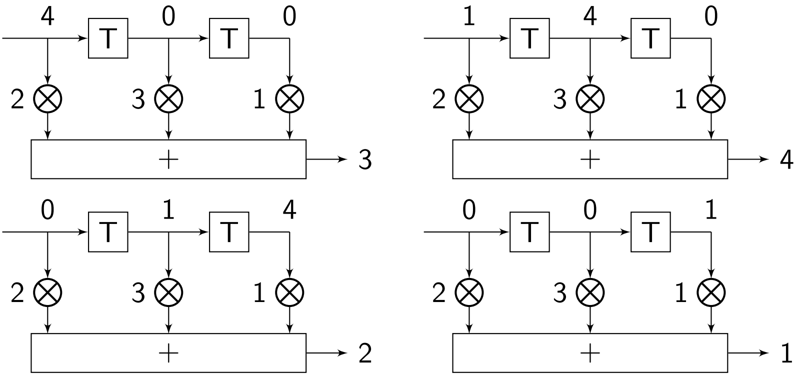
Polynomial multiplication by LFFSR

Compute $(2+3x+x^2)(4+x)$ over GF(5):



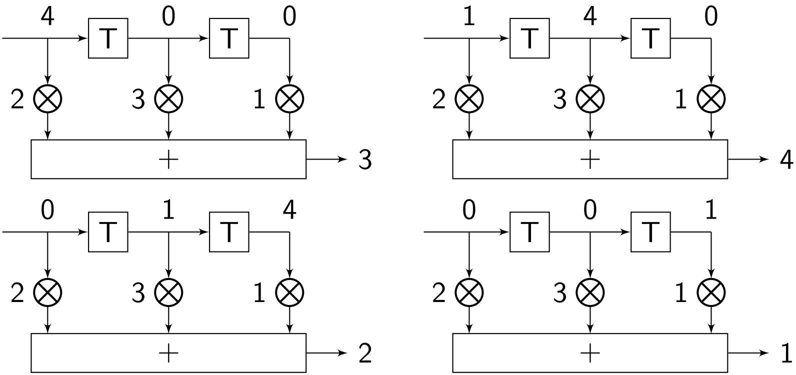
Polynomial multiplication by LFFSR

Compute $(2+3x+x^2)(4+x)$ over GF(5):



Polynomial multiplication by LFFSR

Compute $(2 + 3x + x^2)(4 + x)$ over GF(5):



$$(3, 4, 2, 1) \longrightarrow 3 + 4x + 2x^2 + x^3$$

Polynomial division by LFBSR

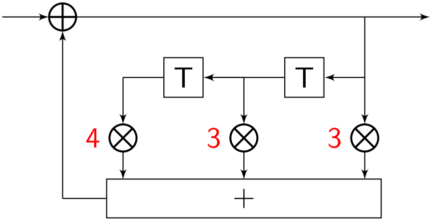
The Linear Feedback Shift Register architecture for division by $3 + 2x + x^2$ over GF(5). Preparation: the coefficients are

$$a_0 = 3, \quad a_1 = 2, \quad a_2 = 1;$$

we put

$$1 - a_0 = 3, \quad -a_1 = 3, \quad -a_2 = 4$$

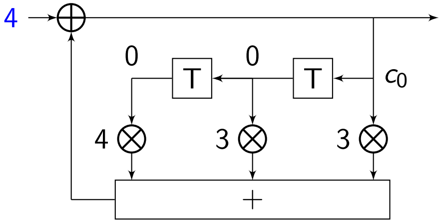
in the registers:



Polynomial division by LFBSR

We want to compute $(4 + 4x + x^3) : (3 + 2x + x^2)$ over GF(5).

An LFBSR works in 2 steps. First, it derives a linear equation, starting from c_0 and completing an entire loop.



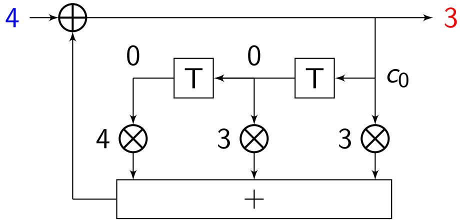
$$4 + 3c_0 = c_0$$

Polynomial division by LFBSR

We want to compute $(4 + 4x + x^3) : (3 + 2x + x^2)$ over GF(5).

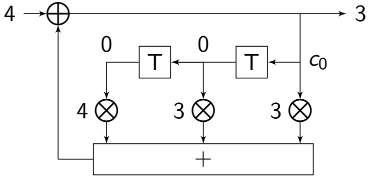
Then that linear equation is solved and the solution is forwarded at the exit.

$$4 + 3c_0 = c_0 \;\rightarrow\; 4 = 3c_0 \;\rightarrow\; c_0 = 3^{-1} * 4 = 2 * 4 = 3.$$



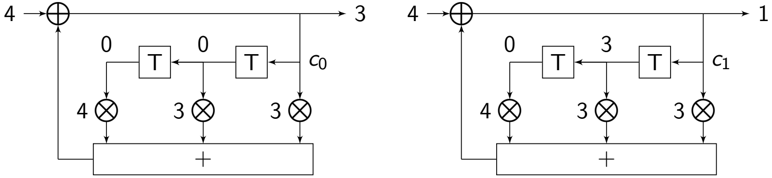
Polynomial division by LFBSR

We want to compute $(4 + 4x + x^3) : (3 + 2x + x^2)$ over GF(5).



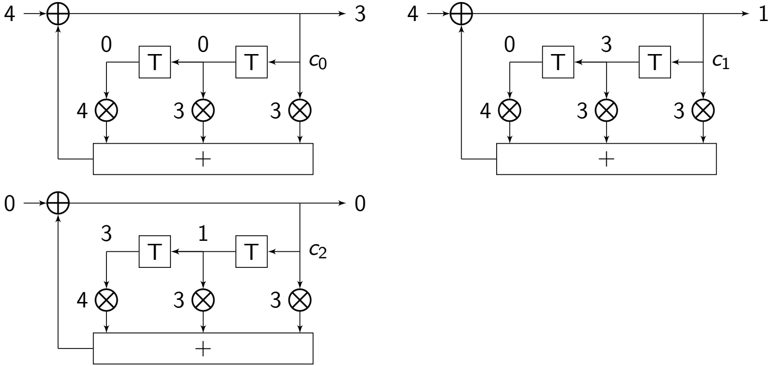
Polynomial division by LFBSR

We want to compute $(4 + 4x + x^3) : (3 + 2x + x^2)$ over GF(5).



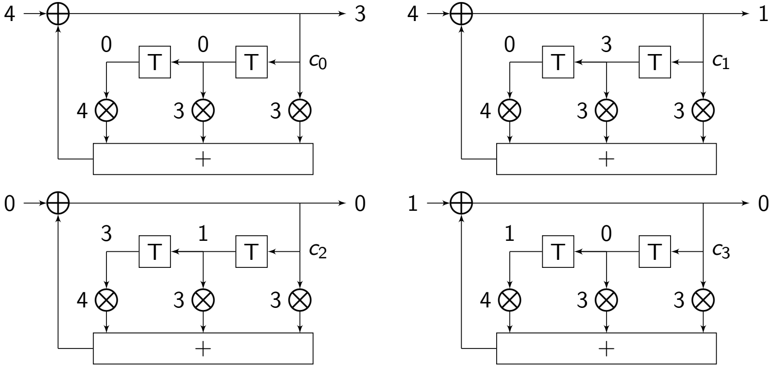
Polynomial division by LFBSR

We want to compute $(4 + 4x + x^3) : (3 + 2x + x^2)$ over GF(5).



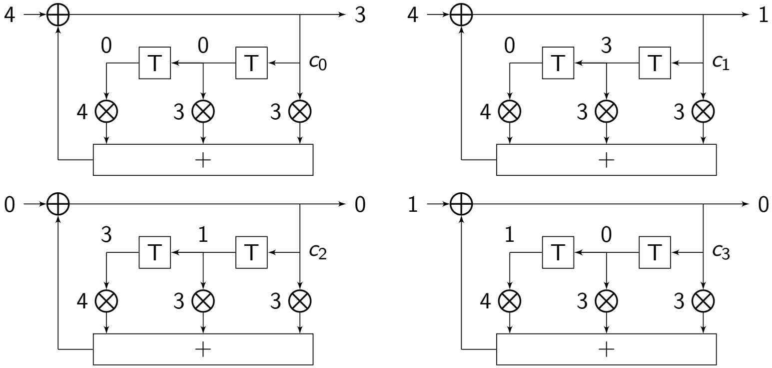
Polynomial division by LFBSR

We want to compute $(4 + 4x + x^3) : (3 + 2x + x^2)$ over GF(5).



Polynomial division by LFBSR

We want to compute $(4 + 4x + x^3) : (3 + 2x + x^2)$ over GF(5).



$$(3, 1, 0, 0) \quad \rightarrow \quad 3 + x$$

Implementing the coding scheme

Depending on the parameters, the syndrome decoding table can be large, but syndrome decoding can be replaced by a fast algorithm called the Error Trapping Algorithm (ETA) that can compute the detected error in real time.