5. Lecture

Freitag, 22. September 2023 10:16

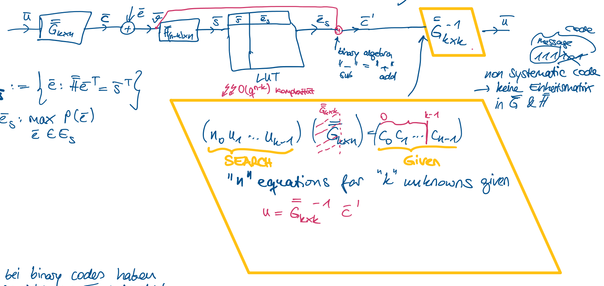
RS codes over GF(q)

Given number of errors "t" to be corrected

→ n-k = 2t , n = q-1 "q" - prime → GF(q) ∋ α (primitive element)

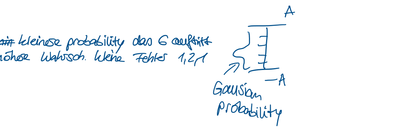
$$\bar{G}_{k\times n} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \alpha^1 & & \alpha^{n-1} \\ \vdots & & & \vdots \\ 1 & \alpha^{k-1} & & \alpha^{(n-1)(k-1)} \end{pmatrix} \qquad \bar{H}_{(n-k)\times n} = \begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{(n-1)} \\ 1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(n-1)} \\ \vdots & & & & \vdots \\ 1 & \alpha^{n-k} & \alpha^{2(n-k)} & \cdots & \alpha^{(n-k)(n-1)} \end{pmatrix}$$

parity check matrix



$E_s := \{ \bar{e} : \bar{H}\bar{e}^T = \bar{s}^T \}$

$\bar{e}_s : \max P(\bar{e})$, $\bar{e} \in E_s$



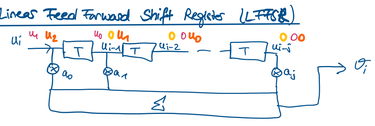
bei binary codes haben Gewicht von ē betrachtet → Hamming Kodes

$\bar{e}' = (0\,0_1 \cdots 0\,6\,0 \cdots 0)$ → mit kleinste probability dass 6 auftritt

$\bar{e}' = (0\,0 \cdots 1\,2\,1 \cdots 0)$ → höhere Wahrsch. kleine Fehler 1,2,1

OPERATIONS BY SHIFT REGISTERS ⇒ manipulate polynoms, real-time, fast

Linear Feed Forward Shift Register (LFFSR)



$i = 0 \rightarrow v_0 = a_0 u_0$

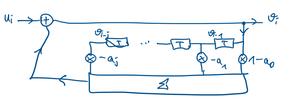
$i = 1 \rightarrow v_1 = a_0 u_1 + a_1 u_0$

$i = 2 \rightarrow v_2 = a_0 u_2 + a_1 u_1 + a_2 u_0$

$$i \rightarrow v_i = \sum_j a_j u_{i-j} \Rightarrow v(x) = a(x)\,u(x)$$

⇓ LFFSR can multiply polynoms

Linear FeedBack Shift Register (LFBSR)



$v_i = u_i + (1 - a_0)\,v_i - a_1 v_{i-1} \cdots - a_j v_{i-j}$

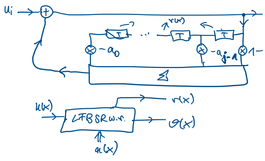
$a_0 v_i + a_1 v_{i-1} + \cdots + a_j v_{i-j} = u_i$

$$\sum_{j=0}^{i} a_j v_{i-j} = u_i \quad \rightarrow \quad a(x)\,v(x) = u(x)$$

$$v(x) = \frac{u(x)}{a(x)}$$

⇒ division of polynoms

Linear FeedBack Shift Registers with remainder (LFBSR w.r.)



$\deg(u(x)) = \deg(a(x))$

$u(x) = v(x)\,a(x) + r(x)$

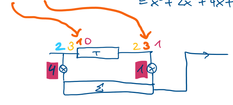
Challenge: How to implement the whole coding scheme on SRs?

↓

How to define the operations of the whole coding scheme as operations on polynoms?

$(x^2+3x+2) \cdot (x+3) = x^3 + 3x^2 + 2x + 4x^2 + 2x + 3 \mod 5$

$= x^3 + 2x^2 + 4x + 3$ GF(5)



$v_0 = 3 \quad (2 \cdot 4)$

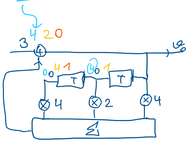
$v_1 = 3 \cdot 4 + 2 \cdot 1 = 4$

$v_2 = 2$

$v_3 = 1$

$x^3 + 2x^2 + 4x + 3$

$(x^3 + 2x^2 + 4x + 3) : (x^2 + 3x + 2) = x + 4$



$1 - a_0 = 1 - 2 \equiv 4$

$-a_1 = -3 \equiv 2$

$-a_2 = -1 \equiv 4$

$v_0 = 4 v_0 + 3 \rightarrow -3 v_0 = 3 \rightarrow 2 v_0 = 3 \rightarrow v_0 = 2^{-1}_{m} \cdot 3 \equiv 3 \cdot 3 \equiv 4$

$3 \cdot 2 = 1 \cdot 5 + 1$

$v_1 = 4 v_1 + 4 + 4 \cdot 2 \overset{\text{mod } 3}{\equiv} v_1 = 4 v_1 + 4 + 2$

$v_2 = 2 v_2 + 2 + 3$

$v_2 = 0$

$v_3 = 4 v_3 + 1 + 4$

$-3 v_3 = \emptyset \rightarrow v_3 = 0$

$-3 v_1 = 2$

$2 v_1 = 2$

$v_1 = 1$

Cyclic shift: $\bar{c} = (c_0 c_1 \cdots c_{n-2} c_{n-1})$

$\bar{c}' = \underset{\text{cyclic shift}}{S}\bar{c} = (c_{n-1}\, c_0 c_1 \cdots c_{n-2})$

$c'(x) = x \cdot c(x) \mod (x^n - 1)$

$x \cdot c(x) = c_0 x + c_1 x^2 + \ldots + c_{n-2} x^{n-1} + c_{n-1} x^n$ (modulo x^n)

$\Rightarrow$

$c'(x) = c_{n-1} + c_0 x + c_1 x^2 + \ldots + c_{n-2} x^{n-1}$

$x \cdot c(x) = c_{n-1}(x^n - 1) + c'(x)$

$c'(x)$: remainder

$c'(x) = x \cdot c(x) \mod (x^n - 1)$