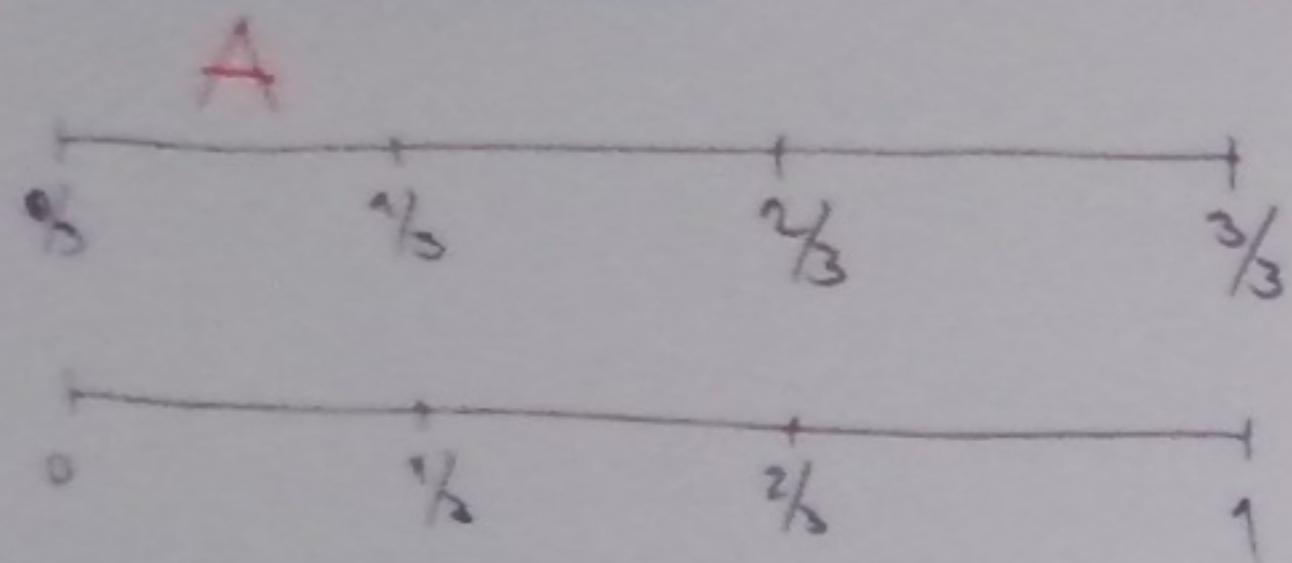


ARITHMETIC CODING

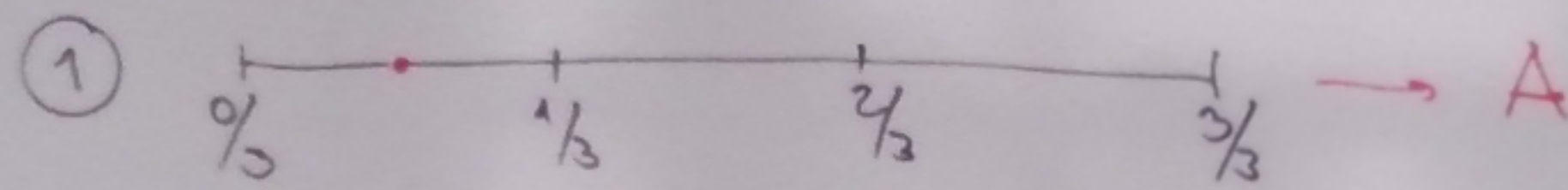
- re-normalisation



we will always use $\langle 0,1 \rangle$ to assign intervals for letters and scale afterwards.

DECODING

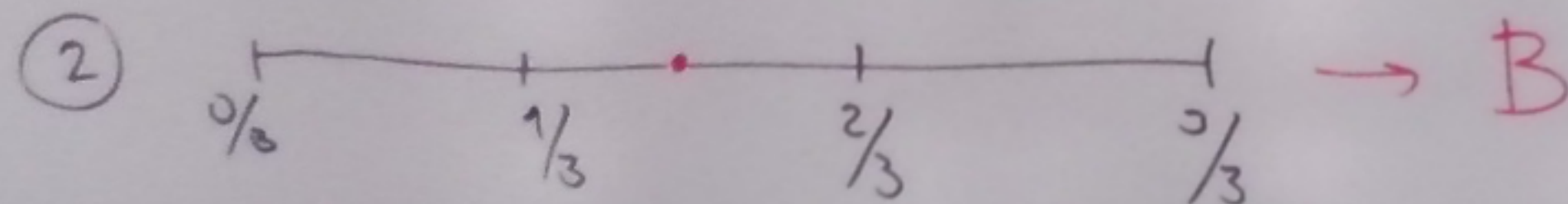
message: $0.001011011 = \frac{91}{512} \quad 0.177\dots$



$(\frac{91}{512} - \text{lower bound of interval}) / \text{length of interval}$

\Rightarrow a number from $\langle 0,1 \rangle = \frac{273}{512} \quad 0.53\dots$

$\frac{91}{512} / \frac{1}{3} = \frac{3 \cdot 91}{512} = \frac{273}{512}$



$(\frac{273}{512} - \frac{1}{3}) \cdot 3 = \frac{3 \cdot 273 - 512}{512} = \frac{307}{512}$

ABBBCBA EOM

ABBBCBAAAAA...

A ... 0

B ... 1/3

C ... 2/3

Ex: Hamming (3,1) ... repetition code

$$G_{\text{sys}} = [1 \ 1 \ 1]$$

$$P = [1 \ 1] \Rightarrow P^T = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$H_{\text{sys}} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$$\vec{w} = (000) \quad \vec{e} = (001) \quad \left. \vphantom{\vec{w}} \right\} \vec{w}' = (001)$$

$$\vec{w}' \cdot H^T = (001) \cdot \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\vec{w}' \cdot H^T = (101) \cdot \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$n = 2^m - 1$$

HAMMING CODES

Construction:

a) from H $(7,4)$

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \left. \vphantom{H} \right\} \begin{array}{l} m \text{ rows} \\ n \text{ columns} \end{array}$$

- columns are distinct
- represent numbers from 1 to n

$$H_{\text{sys}} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

b) using parity bit assignment
non-systematic code
construct a parity bit assignment and copy it to systematic G afterwards

	1	2	3	4	5	6	7
	p_0	p_1	d_0	p_2	d_1	d_2	d_3
p_0	///		X		X		X
p_1		///	X			X	X
p_2				///	X	X	X

position 1 $\Rightarrow \dots 1 \quad (2^1)$
position 2 $\Rightarrow \dots 1 \quad (2^2)$
position 4 $\Rightarrow \dots 1 \quad (2^3)$

- 1 ... $p_2 p_1 p_0$
- 2 ... 010
- 3 ... 011
- 4 ... 100
- 5 ... 101
- 6 ... 110
- 7 ... 111

$$G = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \left. \vphantom{G} \right\} k$$

$$\vec{u} \cdot G = \vec{w}^m$$

$$\Rightarrow \begin{aligned} p_0 &= d_0 \oplus d_1 \oplus d_3 \\ p_1 &= d_0 \oplus d_2 \oplus d_3 \\ p_2 &= d_1 \oplus d_2 \oplus d_3 \end{aligned}$$

Construction:

$$w(x) = u(x) \cdot g(x) \pmod{x^n \oplus 1}$$

n ... code length, k ... number of data bits

m ... number of parity bits

$g(x)$: generator polynomial

$$\deg(g(x)) = m$$

$$g(x) \mid x^n \oplus 1 \quad (\neq \text{ divides } x^n \oplus 1)$$

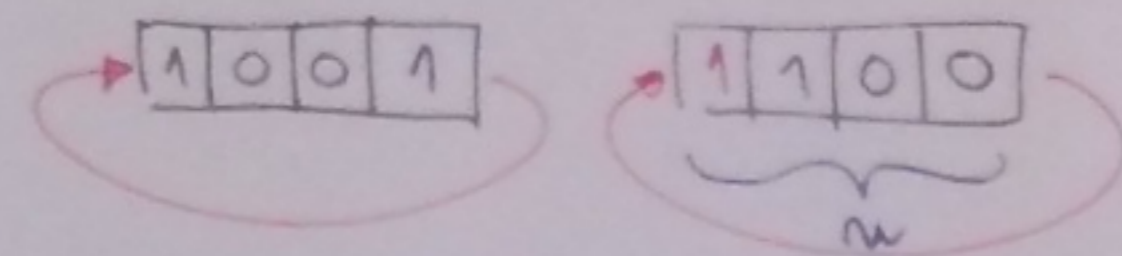
$$\Rightarrow g(x) \cdot h(x) = x^n \oplus 1$$

$h(x)$: parity check polynomial

$$w(x) \cdot h(x) = 0$$

BINARY CYCLIC CODES

- a bit rotation of a code word yields again a code word



Can be represented as polynomials of degree $n-1$

\Rightarrow all operations performed in $\pmod{x^n \oplus 1}$

$$\vec{w} = (1001) \Rightarrow w(x) = 1 \oplus x^3$$

$$\vec{w}' = (0110) \Rightarrow w'(x) = x \oplus x^2$$

$$\text{Ex: } w = (1001) \Rightarrow w(x) = 1 \oplus x^3$$

rotate one bit to the right:

$$w'(x) = x \cdot w(x) = x \oplus x^4 \pmod{x^4 \oplus 1}$$

$$= 1 \oplus x$$

$$\begin{array}{r} x^4 \oplus x : x^4 \oplus 1 = 1 \\ \hline x \oplus 1 \end{array} \leftarrow \text{not interesting for us}$$

but this is

rotate 2 bits:

$$w''(x) = x^2 \cdot w(x) = x^2 \oplus x^5 \pmod{x^4 \oplus 1}$$

$$= x \oplus x^2$$

$$\begin{array}{r} x^5 \oplus x^2 : x^4 \oplus 1 = x \\ x^5 \oplus x \\ \hline x^2 \oplus x \end{array}$$

Ex: $w = (1001)$ is a cyclic code

other code words are for sure:

$$(0011)$$

$$(0110)$$

$$(1100)$$

$$\begin{array}{r} 001001 \\ \oplus 010001 \\ \hline 011000 \end{array}$$