Information Theory

→ Multimedia data is always stored in compressed form.

eg) Low quality $\frac{1}{2}$ hr video may be of 150 GB size.

Hence, we require video encoder (decode compressed data).

→ Information of an event $\propto \frac{1}{\text{occurrence of an event}}$

Entropy is one of the fundamental way to used here.

⇒ By constructing a code book, we can drastically reduce the # bits required to represent the code.

eg)

A	00
B	01
C	10
D	11

A A B A B C D A

each

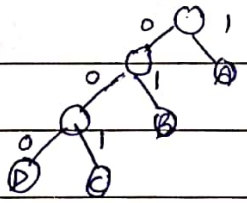
$9 \times 2 = 18$ bits (now)

$9 \times 8 = 72$ " (earlier)

$P(A) = 4/9$ $P(B) = 2/9$ $P(C) = 1/9$ $P(D) = 2/9$

A particular event occur more frequently while using less no. of bits.

→ Here,



A − 1

B − 01

C − 001

D − 000

Avg. code word length = $1 \times 4 + 2 \times 2 + 3 \times 1 + 3 \times 3$

$= 17$

$\frac{17}{9}$ ← length of code

$\vec{R} = \sum P_i L_i$

↙ Avg. length

Data Compression :=

→ Source coding

a	0
b	01
c	110
d	11

To decode: 1100 11

Possible results : c a d

d a a d

Result must be unique.
Hence this is not a desirable code.

∴ we design prefix code.

Prefix code: variable length code in which no codeword is a prefix of another word.

Fano codes:

a	b	c	d	e	f
0.3	0.3	0.1	0.1	0.2	0.2

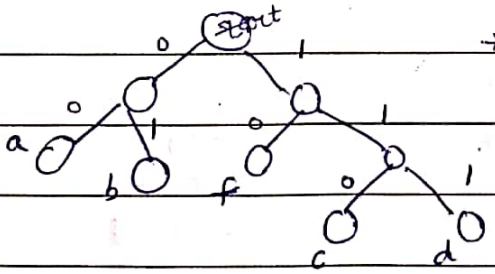
divide set in 2 subparts, where one subpart's sum is almost equal to 2^{nd}.

a	0.3	00
b	0.3	01
f	0.2	10
c	0.1	110
d	0.1	111

c a b a b f

110 00 01 00 01 10

For decoding, we require a tree.



→ leaf node position indicates a valid codeword.

⊛ A codeword is valid if it terminates in a particular leaf position.

∴ for 1100001 00 01 10

c a b a b f

Kraft Inequality :=

A necessary & suffivent cond^n for a code to be instantaneous

$\sum r^{-l_i} \leq 1$ ~~$\sum 2^{l_i} \geq 1$~~, for binary data: $\sum 2^{-l_i} \leq 1$

$S_1 \quad S_2 \quad \dots \quad S_n$

$L_1 \leq L_2 \leq \dots \leq L_n$

no. of codewords $\leftarrow NC(S_1) = 2^{L_1} \geq 1$

codewords where L_1 is not prefix of $L_2 = 2^{L_2 - L_1}$

$\therefore NC(S_2) = 2^{L_2} - 2^{L_2 - L_1} \geq 1$

Similarly, $NC(S_3) = 2^{L_3} - 2^{L_3 - L_1} - 2^{L_3 - L_2} \geq 1$

$\therefore NC(S_n) = 2^{L_n} - 2^{L_n - L_{n-1}} - 2^{L_n - L_{n-2}} - \dots - 2^{L_n - L_2} - 2^{L_n - L_1} \geq 1$

multiplying both side by 2^{-L_n}

$\Rightarrow 2^{-L_n}(2^{L_n} - 2^{L_n - L_{n-1}} - 2^{L_n - L_{n-2}} - \dots - 2^{L_n - L_2} - 2^{L_n - L_1}) \geq 2^{-L_n}$

$1 - 2^{-L_{n-1}} - 2^{-L_{n-2}} - \dots - 2^{-L_2} - 2^{-L_1} \geq 2^{-L_n}$

$\Rightarrow 2^{-L_1} + 2^{-L_2} + \dots + 2^{-L_{n-1}} + 2^{-L_n} \leq 1$

$\Rightarrow \sum_{i=1}^{n} 2^{-L_i} \leq 1$

eg)

a	0
b	0 1
c	0 1 1
d	0 1 1 1

This code is uniquely decodable but it is not valid Prefix code.

Here 0 is working as separator.

Compact code : If its avg. length is less than or equal to the avg. length of all other uniquely decodable codes for the same source & code alphabet.

Source	P_i	Code A	Code B
S_1	0.5	00	1
S_2	0.1	01	000
S_3	0.2	10	001
S_4	0.2	11	01

$L_A = 0.5 \times 2 + 0.1 \times 2 + 0.2 \times 2 + 0.2 \times 2 = 2$ bits per symbol

$L_B = 0.5 \times 1 + 0.1 \times 3 + 0.2 \times 3 + 0.2 \times 2 = 1.8$ " "

⇒ Code B is more compact than code A.

(*) Fano code is not a compact code, hence it can be further improved.

Efficiency of Code: $\eta = \frac{H(S)}{\bar{R}} \times 100\%$.

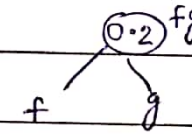
($\bar{R}$ ← avg. length)

Huffman code: a b c d e (f having least Prob.) (g)

0.2 0.2 0.15 0.15 0.1 0.1 0.1

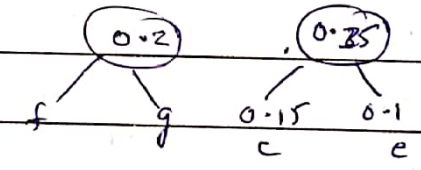
⇒ a b c d e fg

0.2 0.2 0.15 0.15 0.1 0.2



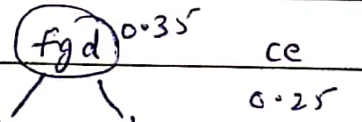
⇒ a b fg d ce

0.2 0.2 0.2 0.15 0.25

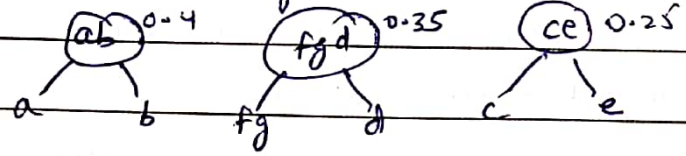


⇒ a b fgd ce

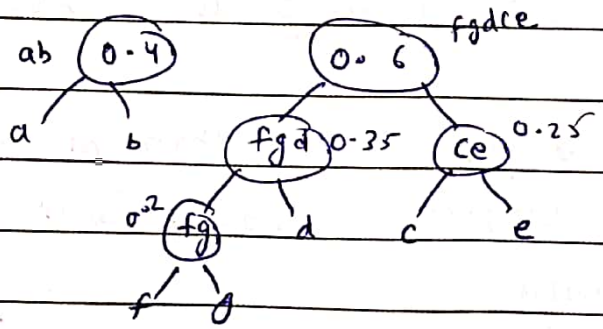
0.2 0.2 0.35 0.25



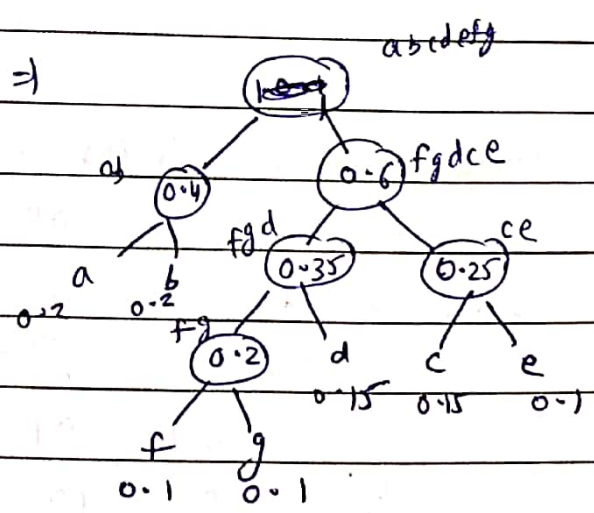
⇒



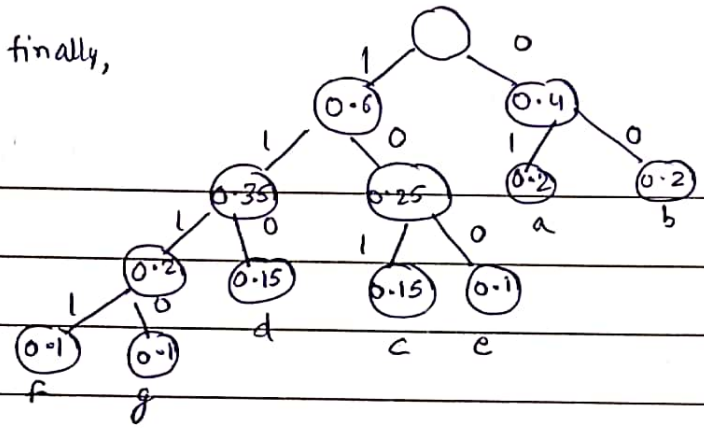
⇒



⇒



finally,



Min. variance Huffman codes:

Giving more Priority to uncombined source than combined source, we get less σ^2 value. where $\sigma^2 = \Sigma P_i (L_i - L)^2$

Case A: L_A or $\bar{R}_A = 0.2 \times 2 + 0.4 \times 1 + 0.1 \times 4 + 0.1 \times 4 + 0.2 \times 3$

$= 0.4 + 0.4 + 0.4 + 0.4 + 0.6 = 2.2$ bits per symbols.

Case B: L_B or Avg. length $\bar{R}_B = 0.2 \times 2 + 0.4 \times 2 + 0.1 \times 3 + 0.1 \times 3 + 0.2 \times 2$

$= 0.4 + 0.8 + 0.3 + 0.3 + 0.4 = 2.2$ bits per symbol

In a second, a communication channel can transmit 3000 bits. In every second, source produced 1000 symbols.

In code A	In code B
For S_3, 0.4×1000 bits needs to transfer but 3000 will be transfer via channel & 1000 will store in buffer.	For S_3, 3×1000 bits needs to be transfer which will directly get transfer without storing some bits in buffer.

Hence, code B is much more efficient than code A in communication aspect bcoz channel bandwidth is properly utilized.

Canonical Huffman Code: (well structured code). 100 1110

→ 1) Assign 0 to left & 1 to right branch.

2) Build the tree from left to right in rising order of depth.

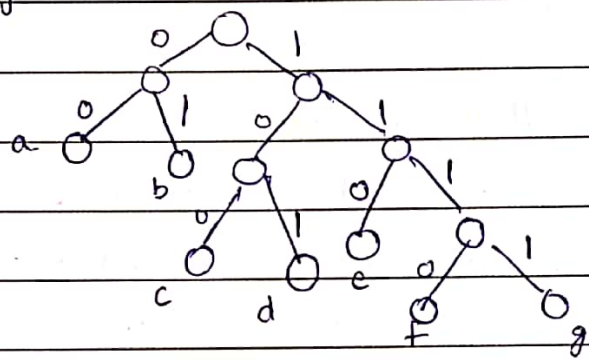
3) Each leaf is placed at the 1st available position.

→ It reduces the Codebook size without compromizing in efficiency.

eg) new assigned codes

a	01	(2)	00
b	00	(2)	01
c	101	(3)	100
d	110	(3)	101
e	100	(3)	110
f	1111	(4)	1110
g	1110	(4)	1111

Partition codes based on length



1st symbols in each class: a – 01 (2), c – 101 (3), f – 1110 (4)

+ .abcdefg

	binary	decimal Lyn +
0	a	01
1	b	10
2	c	101
3	d	110
4	e	111
5	f	1110
6	g	1111

This ~~produce~~ procedure reduce no. of symbols to transfer to the recieving end.

Entropy: Information of an event is inversly proportional to occurence of event. i.e, $I(s) \propto \frac{1}{P(s)}$

→ $I(s) = \log_2 \frac{1}{P(s)}$ bits

Instead of introducing Const. fano introduced $\log_2$ bcoz :

1) Information is always additive in nature. i.e, $I(ab) = I(a) + I(b)$

Acc. to fano's theorem, $I(ab) = \log_2 \frac{1}{P(ab)} = \log_2 \frac{1}{P(a)} + \log_2 \frac{1}{P(b)}$

$\Rightarrow$ $I(ab) = I(a) + I(b)$

(*) For a particular (certain) event, information is zero.

bcoz $P(S) = 1$

and $I(S) = \log \frac{1}{1} = 0$

Entropy: let say n events : $S_1, S_2, S_3, \cdots S_n$

$n_i \rightarrow$ # times event S_i occurs

$\therefore$ Total Information $= n_1 I(S_1) + n_2 I(S_2) + \cdots + n_n I(S_n)$

let say $\sum n_i = n$

$\therefore$ Avg. Information $= \frac{n_1}{n} I(S_1) + \frac{n_2}{n} I(S_2) + \cdots + \frac{n_n}{n} I(S_n)$

$\Rightarrow H(S) = P(S_1) \log \frac{1}{P(S_1)} + P(S_2) \log \frac{1}{P(S_2)} + \cdots + P(S_n) \log \frac{1}{(S_n)}$

$\Rightarrow$ Entropy $\rightarrow$ $\boxed{H(S) = \sum_{i=1}^{n} P(S_i) \log_2 \frac{1}{P(S_i)}}$ bits/symbol

Limits of Entropy : 1) $H(S) \geq 0$ (lowerbound)

Proof : $\because P(S_i) \leq 1$

$\therefore \frac{1}{P(S_i)} \geq 1 \Rightarrow \log_2 \frac{1}{P(S_i)} \geq 0$

After multiplying both side by $P(S_i)$ & taking Summission.

$\sum P(S_i) \log_2 \frac{1}{P(S_i)} \geq 0$

i.e, $\boxed{H(S) \geq 0}$

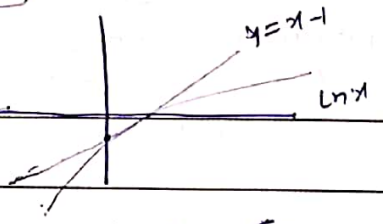
2) Upperbound.

Lemma : If $P_1, P_2, \cdots P_N$ and $q_1, q_2, \cdots, q_n$ are all non-ve numbers that satisfy the conditions

$\sum_{P=1}^{N} P_i = 1$ & $\sum_{i=1}^{N} q_i = 1$ then

$$\boxed{\sum p_i \log_2 \frac{1}{p_i} \leq \sum p_i \log_2 \frac{1}{q_i}}$$

Proof → we know, $\ln x \leq x-1$



Put $x = \frac{q_i}{p_i}$ $\therefore \ln\left(\frac{q_i}{p_i}\right) \leq \left(\frac{q_i}{p_i} - 1\right)$

$\Rightarrow \sum p_i \ln \frac{q_i}{p_i} \leq \sum p_i \left(\frac{q_i}{p_i} - 1\right)$ $\left(\because \sum q_i = 1 \ \& \ \sum p_i = 1\right)$

$= 0$

$\Rightarrow \sum p_i \ln \frac{1}{p_i} - \sum p_i \ln \frac{1}{q_i} \leq 0$

$\Rightarrow \sum p_i \ln \frac{1}{p_i} \leq \sum p_i \ln \frac{1}{q_i}$

$\Rightarrow \ln 2 \sum p_i \log_2 \frac{1}{p_i} \leq \ln 2 \sum p_i \log_2 \frac{1}{q_i}$

$\Rightarrow \boxed{\sum p_i \log_2 \frac{1}{p_i} \leq \sum p_i \log_2 \frac{1}{q_i}}$

$\Rightarrow H(S) \leq \sum p_i \log_2 \frac{1}{q_i}$

$\therefore$ Put $q_i = \frac{1}{N}$

$\therefore H(S) \leq \sum p_i \log_2 N$

$\Rightarrow H(S) \leq \log_2 N \sum p_i$ $(=1)$

$\Rightarrow \boxed{H(S) \leq \log_2 N}$

q_i is a source.
∴ Its min. value will be $\frac{1}{N}$

(*) max. Info is obtain when each symbol/event is equiprobable.

(*) when only a particular event occur then $p_i = 0$ & rest $p_i = 0$

So $H(S) = \sum p_i \log \frac{1}{p_i(S)}$

$H(S) = 0$

(P is actually p)

Source coding Theorem: The efficiency of data compression is based on the efficient representation of the symbols. Source coding theorem establish the limit to possible data compression.

It provide the lower & upper bound condition of data compression.

& the condⁿ is $\boxed{H(S) \le \bar{R} \le H(S)+1}$

$H(S)$ ↓ Entropy of source; $\bar{R}$ ↓ avg length of code

(*) $H(S)$ is fixed for a source

$$\eta = \frac{H(S)}{\bar{R}} \times 100\%$$

Proof: i) we know, $H(S) - \bar{R} = \sum p_i \log_2 \frac{1}{p_i} - \sum p_i R_i$

$$= \frac{1}{\ln 2} \sum p_i \ln \frac{1}{p_i} - \frac{1}{\ln 2} \sum p_i \ln 2^{R_i}$$

$$= \frac{1}{\ln 2} \sum p_i \ln \frac{1}{p_i 2^{R_i}}$$

$$\le \frac{1}{\ln 2} \sum p_i \left(\frac{1}{p_i 2^{R_i}} - 1 \right) \quad \because (\ln x \le x-1)$$

$$\le \frac{1}{\ln 2} \left(\sum 2^{-R_i} - \sum p_i \right)$$

$$\le \frac{1}{\ln 2} (1-1)$$

$$\le 0$$

$$\Rightarrow H(S) - \bar{R} \le 0$$

$$\rightarrow \boxed{H(S) \le \bar{R}}$$

ii) now we have to Proof: $\bar{R} < H(S)+1$

we know, $\bar{R} = \sum p_i R_i$

$$= \sum p_i \left\lceil \log \frac{1}{p_i} \right\rceil$$

$$< \sum p_i \left(\log \frac{1}{p_i} + 1 \right)$$

(*) Shanon's length criteria

$R_i = \left\lceil \log \frac{1}{p_i} \right\rceil$

$\because \lceil x \rceil < x+1$

$< \sum P_i \log \frac{1}{P_i} + \sum P_i$

$< H(S) + 1$

$\Rightarrow \boxed{\bar{R} < H(S)+1}$

eg) $a = 0.8$ 0 $\quad \therefore \bar{R} = \sum P_i l_i = 0.8 \times 1 + 0.2 \times 1$

$b = 0.2$ 1 $\quad = 1$

now, $H(S) = 0.8 \log \frac{1}{0.8} + 0.2 \log \frac{1}{0.2}$

$= 0.72$ bits/symbol

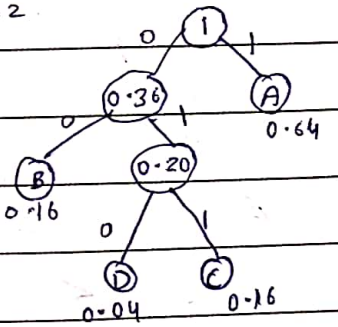
$\therefore \eta = \frac{0.72}{1} \times 100 = 72\%$

* Extensions of source :-

$\sigma_i = (S_i\ S_j)$

2nd order extension of source ES $\Rightarrow S^2$

aa	(A)	0.64	1
ab	(B)	0.16	00
ba	(C)	0.16	011
bb	(D)	0.04	010



(*) $\bar{R_2}$ = avg. length ~~cost~~ consists of 2 characters

$\therefore \bar{R_2} = 0.64 \times 1 + 0.16 \times 2 + 0.16 \times 3 + 0.04 \times 3 = 1.56$

$H(S^2) = 0.64 \log \frac{1}{0.64} + 0.16 \log \frac{1}{0.16} + 0.16 \log \frac{1}{0.16} + 0.04 \log \frac{1}{0.04}$

$= 1.44$ bits/symbol

$\therefore \eta = \frac{1.44}{1.56} \times 100 = 92\%$

$\Rightarrow$ 2nd order extension is more efficient than 1st order source code.

eg)

a a	p^2
a b	$p(1-p)$
b a	$p(1-p)$
b b	$(1-p)^2$

$a = p$
$b = 1-p$

$$H(S^2) = p^2 \log_2 \frac{1}{p^2} + 2p(1-p)\log_2 \frac{1}{p(1-p)} + (1-p)^2 \log_2 \frac{1}{(1-p)^2}$$

$$\therefore H(S^2) = 2p^2 \log_2 \frac{1}{p} + 2p(1-p)\log_2 \frac{1}{1-p} + 2p(1-p)\log_2 \frac{1}{p} + 2(1-p)^2 \log_2 \frac{1}{1-p}$$

$$= 2p\left[p \log_2 \frac{1}{p} + (1-p)\log \frac{1}{1-p}\right] + 2(1-p)\left[p \log_2 \frac{1}{p} + (1-p)\log \frac{1}{1-p}\right]$$

$\therefore$ $H(S^n)$

$S_1, S_2, S_3, \ldots, S_q$

$\sigma_i = (S_{i_1}, S_{i_2}, S_{i_3}, \ldots S_{i_n})$

$$H(S^n) = \sum_{S^n} P(\sigma_i) \log_2 \frac{1}{P(\sigma_i)}$$

where, $P(\sigma_i) = P(S_{i_1})\, P(S_{i_2}) \cdots P(S_{i_n})$

$$\therefore H(S^n) = \sum_{S^n} P(S_{i_1})P(S_{i_2})\cdots P(S_{i_n}) \log_2 \frac{1}{P(S_{i_1})P(S_{i_2})\cdots P(S_{i_n})}$$

$$= \sum_{S^n} P(S_{i_1})P(S_{i_2})\cdots P(S_{i_n}) \log_2 \frac{1}{P(S_{i_1})} + \sum_{S^n} P(S_{i_1})P(S_{i_2})\cdots P(S_{i_n}) \log_2 \frac{1}{P(S_{i_2})} + \cdots\cdots$$

$$= \sum_{i_1=1}^{q} P(S_{i_1}) \log_2 \frac{1}{P(S_{i_1})} \sum_{i_2=1}^{q} P(S_{i_2}) \sum_{i_3=1}^{q} P(S_{i_3}) \cdots \sum_{i_n=1}^{q} P(S_{i_n})$$

$$= H(S)$$

$\therefore$ $H(S^n) = n\,H(S)$

All. to Sourle coding theorem, $H(S^n) \le \bar{R}_n \le H(S^n) + 1$

$\rightarrow$ $n\,H(S) \le \bar{R}_n < n\,H(S) + 1$

$\rightarrow$ $$H(S) \le \frac{\bar{R}_n}{n} < H(S) + \frac{1}{n}$$

$$\lim_{n\to\infty} \frac{\bar{R_n}}{n} = H(S)$$ This particular source is k/a DMS.

(*) Hence higher order extension is more efficient than lower order extension.

For large value of n, Huffman code is not easy to handle & hence not that much effective

So theoritically higher order extension is much more efficient but practically it is not the same case.

May 2019

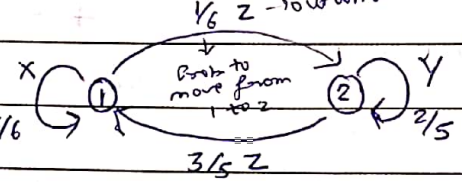
DMS (Discrete M Source)

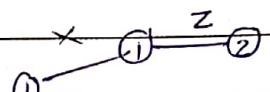
Markov model → we have Prev. info, on the basis of this info we predict next information or outcome.

eg) $A \to B/C$

Markov model particularly represented in terms of state diagram.

eg)





Prob occurence of XZ ~~P(x1 state)~~

$P(XZ) = P(1/1)\ P(2/1)\ P(1)$

State Prob. of 1, $P(1) = \frac{5}{6}P(1) + \frac{3}{5}P(2)$ — (a) { Summation of all intensity source on 1

$P(2) = \frac{1}{6}P(1) + \frac{2}{5}P(2)$ — (b)

Also we know that, $P(1) + P(2) = 1$ — (c)

Solving (a) & (c) $P(1) = \frac{5}{6}P(1) + \frac{3}{5}$ $\quad \because P(2) = P(1) \times \frac{3}{5}$

$P(1) - \frac{5}{6}P(1) + \frac{3}{6}P(1) = \frac{3}{5}$

$P(1)\left(\frac{30 - 25 + 18}{30}\right) = \frac{3}{5}$ $\quad \therefore P(1) = \frac{18}{23}$

$P(1) = 18/23$

$P(2) = 5/23$

State Entropy $H(S_i) = \sum_{j=1}^{s} P(j/i) \log_2 \frac{1}{P(j/i)}$, s is the total no. of states

$\Rightarrow H(S_1) = P(2/1) \log_2 \frac{1}{P(2/1)} + P(1/1) \log_2 \frac{1}{P(1/1)}$ { considering all edges originating from S_i

State Entropy of (1) $= \frac{1}{6} \log_2 \frac{1}{1/6} + \frac{5}{6} \log_2 \frac{1}{5/6}$

$= 0.65$ bits / Symbol

Similarly, $H(S_2) = \frac{2}{5} \log_2 \frac{1}{2/5} + \frac{3}{5} \log_2 \frac{1}{3/5}$

$= 0.92$ bits / Symbol

Source Entropy of this Markov's model, $H(S) = \sum_{i=1}^{s} P(S_i) H(S_i)$

$= P(S_1) H(S_1) + P(S_2) H(S_2) = 0.7192$ bits / Symbol

1) To Get messages of length 1, Possible ways are following :

Entropy of L length message $G_L = \frac{1}{L} \sum P(msg_L) \log \frac{1}{P(msg_L)}$

now, $P(X) = P(1) \times P(1/1) = \frac{18}{23} \times \frac{5}{6} = \frac{15}{23}$

$P(z) = P(1) P(2/1) + P(2) \cdot P(1/2)$

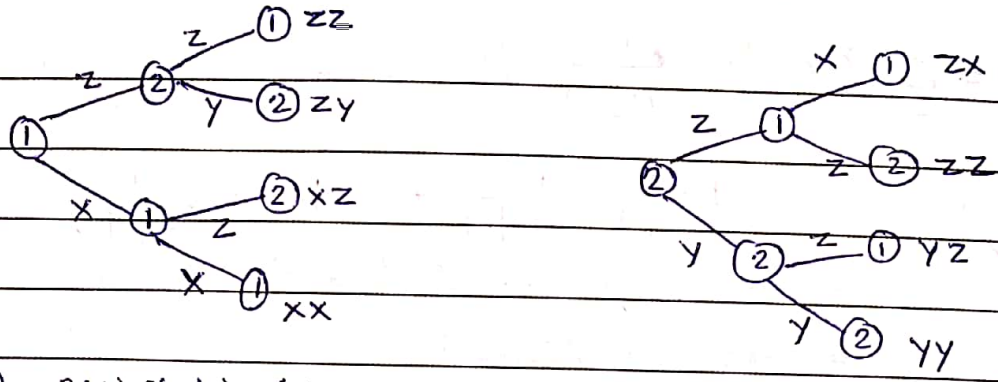
$= 3\cancel{18}/23 \times 1/\cancel{6} + \cancel{5}/23 \times \cancel{3}/5 = 6/23$

Similarly, $P(Y) =$

$\therefore G_1 = \frac{1}{1} P(X) \log_2 \frac{1}{P(X)} + P(Y) \log_2 \frac{1}{P(Y)} + P(z) \log_2 \frac{1}{P(z)}$

$= 1.2142$ bits / Symbol

2) To get message of length 2,



now $P(zz) = P(1)\,P(2|1)\,P(1|2)$ Similarly find all $P(-)$

$$G_2 = \frac{1}{2} \sum P(msg_2) \log \frac{1}{P(msg_2)} = 1.0597 \text{ bits/symbol}$$

Conclusion: $G_1 > G_2 > G_3 > \cdots\cdots > H$

i.e, $\lim_{L \to \infty} G_L = H(S)$

Shanon Coding: If an event occurs with prob. P_i, how many bits are required to represent it?

$$L_i = \left\lceil \log_2 \frac{1}{P_i} \right\rceil \text{ bits.} \quad \text{—(1)}$$

Algorithm:- 1) Arrange Prob. in non rising order.

2) Compute the length L_i for the codeword corresponding to each symbol S_i given by (1)

3) Define the following parameters

$$Q_1 = 0$$

$$Q_2 = Q_1 + P_1 = P_1$$

$$Q_3 = Q_2 + P_2 = P_1 + P_2$$

$$Q_4 = Q_3 + P_3 = P_1 + P_2 + P_3$$

$$\vdots$$

$$Q_{n+1} = Q_n + P_n = 1$$

4) Expand ξ_i in binary till l_i no. of places after decimal point.

5) The numbers after decimal places in the binary representation of ξ_i are the codewords ~~from~~ for the corresponding symbol s_i.

Suppose we have source $(A, B, C, D) = (0.1, 0.2, 0.3, 0.4)$

$\Rightarrow (D, C, B, A) = (0.4, 0.3, 0.2, 0.1)$

$\Rightarrow \because l_D = \lceil \log_2 \frac{1}{0.4} \rceil \quad l_C = \lceil \log_2 \frac{1}{0.3} \rceil \quad l_B = \lceil \log_2 \frac{1}{0.2} \rceil \quad l_A = \lceil \log_2 \frac{1}{0.1} \rceil$

$= 2 \qquad = 2 \qquad = 3 \qquad = 4$

now,

$\xi_1 = 0 \; ; \; \xi_2 = P_1 = 0.4 \; ; \; \xi_3 = 0.7 \; ; \; \xi_4 = 0.9$

In binary representation, $\xi_1 = 0 = (0.000\ldots)_2 \qquad \xi_2 = 0.4 = (0.01\ldots)_2$

$\xi_3 = 0.7 = (0.101\ldots)_2 \qquad \xi_4 = 0.9 = (0.1110\ldots)_2$

$\because l_D = 2$ so we'll select 2 digits after decimal point in ξ_1.

So, $D = 00 \qquad B = 101$

$C = 01 \qquad A = 1110$

now, $\eta = \frac{H(S)}{\bar{L}} \times 100\ \%$ (1.84)

$= \dfrac{0.4 \log_2 \frac{1}{0.4} + 0.3 \log_2 \frac{1}{0.3} + 0.2 \log_2 \frac{1}{0.2} + 0.1 \log_2 \frac{1}{0.1}}{0.4 \times 2 + 0.3 \times 2 + 0.2 \times 3 + 0.1 \times 4}$

$= \dfrac{0.159 + 0.157 + 0.139 + 0.1}{0.8 + 0.6 + 0.6 + 0.4} \times 100$

$\simeq 76.95\ \%$

0.529
0.521
0.464
0.332

$= \dfrac{0.529 + 0.521 + 0.464 + 0.332}{0.8 + 0.6 + 0.6 + 0.4} \times 100$

$= 76.6 + \%$

Au. to Huffman code, $D = 1$, $C = 00$, $B = 010$, $A = 011$

$$\bar{R} = 1 \times 0.4 + 2 \times 0.3 + 3 \times 0.2 + 3 \times 0.1$$

$$= 1.9$$

$$\eta = \frac{1.84}{1.9} \times 100$$

$$= 96\%.$$

(*)

So, Huffman code is more compact than Shanon's code.

r-ary Huffman code:

Compute a parameter $\alpha = \frac{q - r}{(r-1)}$

we may need to add dummy nodes to create a r-ary tree.

$$q' = r + \lceil \alpha \rceil (r-1)$$

total dummy nodes = $q' - q$

eg)

S_1	0.16
S_2	0.14
S_3	0.13
S_4	0.12
S_5	0.10
S_6	0.10
S_7	0.06
S_8	0.06
S_9	0.05
S_{10}	0.04
S_{11}	0.04

Here, we need to design a 2-ternary tree.

i.e, $q = 11$ (here 11 nodes are there)

$r = 4$

$$\alpha = \frac{11-4}{4-1} = \frac{7}{3} = 2.33$$

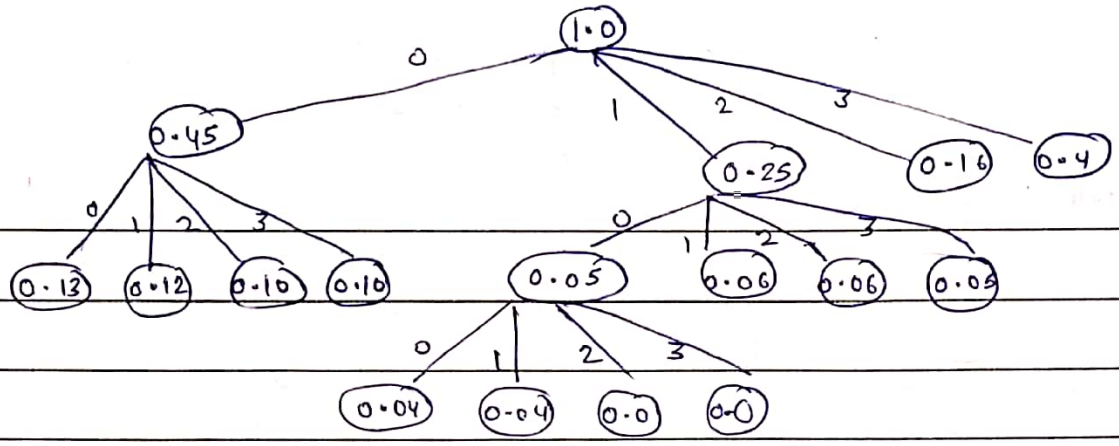
$$q' = 4 + 3(4-1) = 4 + 9 = 13$$

$\therefore$ # dummy nodes = $13 - 11$

$= 2$

Add 2 dummy nodes

$S_{12} = 0.00$ & $S_{13} = 0.00$



~~Q) A 10 feet = x~~

16 Aug, 2019

$$H(S) \leq \bar{R} < H(S) + 1$$

$$H(S^n) \leq \bar{R}_n < H(S^n) + 1$$

$$\Rightarrow \quad H(S) \leq \bar{R} < H(S) + \frac{1}{n}$$

$\therefore$ If $n \to \infty$ then $\bar{R} \simeq H(S)$

eg) $P(a) = 0.8 \qquad P(b) = 0.2$

$\therefore H(S) = 0.8 \log \frac{1}{0.8} + 0.2 \log \frac{1}{0.2} = 0.72$ bits/symbol

Theoritically Extended source coding (Huffman code) is good but practically implementing the Huffman code is not a good approach bcoz there will be huge no. of code symbols.

So, we go for another coding approach i.e Arithmetic coding -

Arithmetic Coding (AC) :- encode → aacb

(a, b, c, d) = (0.4, 0.2, 0.3, 0.1)

In Huffman coding approach, # symbols possible = 4^5 (for msg sym = 5)

now a ∈ range of 0.6 to 1.0

a	1.0	
	0.6	cumulative prob.
b		
c	0.4	
	0.1	
d	0.0	

find d = diff. = UB − LB (same for each interval stage in a first step)

Range 'a' = LB + d × P(a)

0.64 1) $0.6 + 0.1\,(UB - LB)$ | 2) $0.64 + 0.1 \times 0.36$

$= 0.6 + 0.04 \times 0.1$ $= 0.64$

	1.00	1.00	1.00	0.90
a	0.4 a			
b	0.6 0.2	0.84	0.936	0.8824
c	0.4 0.3	0.76	0.90 c	0.8736 ⋆
d	0.1 0.1	0.64	0.856 ~~0.8464~~	0.8604
	0.0	0.6	0.84	0.856

$tag = \frac{UB + LB}{2}$

$0.84 + 0.016 (100 - 0.84)$ | $0.84 + 0.1536$ (~~0.16~~)

$= 0.856$

3rd iteration:

1) $0.856 + (0.9 - 0.856) \times 0.1$ $= 0.8604$

2) $0.8604 + 0.4^{3} \times (0.0396)$ $= 0.8736$

3) $0.8736 + 0.2\,(0.044)$ $= 0.8824$

4) $0.8824 + 0.4\,(0.0176)$ $= 0.9$

for aacb, send any value b/w 0.8736 & 0.8824

say (0.879)

⋆ If we have given 0.879 with source prob. then we can decode 0.879 as following :-

we start scaling we find that $0.879 \in [0.6, 1.0]$ then 1st symbol is a then iterate b/w 0.6 to 1.0 then we find $0.879 \in [0.84, 1.0]$

i.e, 2nd symbol is a & so on.

drawback : Floating point operations for scaling are computation intensive.

eg) $0.7 = 0.(1011001)001\ldots_2$ $\frac{1}{2}$

$\downarrow$

$(0.6992)_{10}$

Instead of randomly selecting the tag value, we can choose specific values say for 0.6992 only 8 bits are sufficient to represent but for 0.7 large no. of bits are required.

$0.8 = 0.11001100110$

$0.6 = 0.10011001100$

From 2nd position onwards we are getting diff. bit positions.

So we can take (0.11) as suitable tag value.

eg) UB 0.(1011)10110000

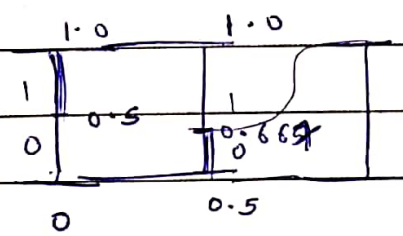
LB 0.(1011)00110111 0

From 5th position onwards, we are moving more closer to UB.

So Possible tag value will be (0.10111).

Adaptive AC: let us say we want to encode 1011011 0

symbols (0,1) = 2 So. initial Prob.(0) = Prob.(1) = 1/2



C = counter variable Initially, C(0) = C(1) = 1	In 2nd phase, C(1) = 2 ∴ P(1) = 2/3 C(0) = 1 P(0) = 1/3

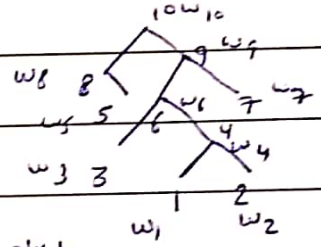
Adaptive Huffman Coding :

Sibling Property

node numbers are assigned as

1) A node with higher ~~number~~ weight will have a higher node number.

2) A Parent node will always have a higher node number than its children.



(*) Bottom to Top, left to right

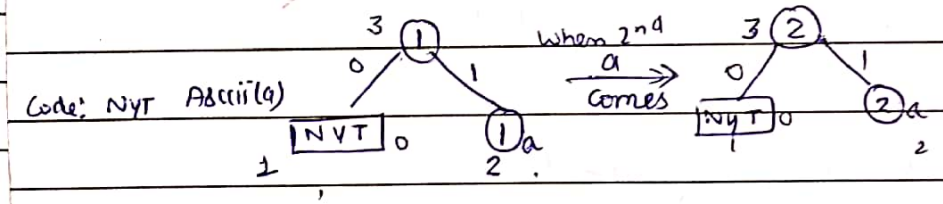
(*) Sibling property says that nodes are arranged in order of rising weights.

i.e, $w_1 \leq w_2 \leq \cdots \leq w_{10}$

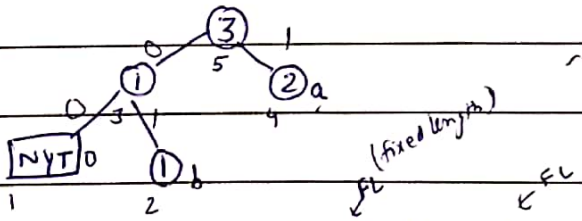
Not yet Transmitted

Initial node is NYT with weight = 0

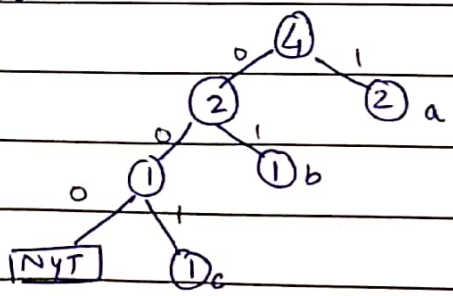
Suppose we want to transmit aabcdad.



→ now b comes 1st time, make b as right sibling of NYT.

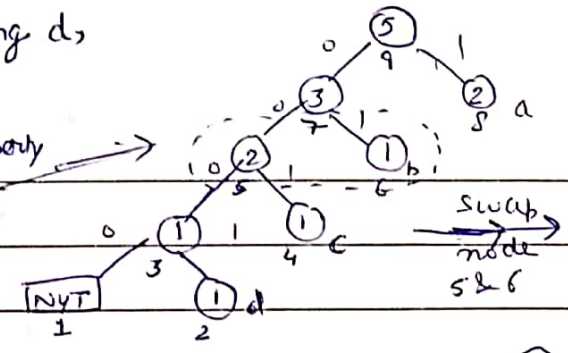


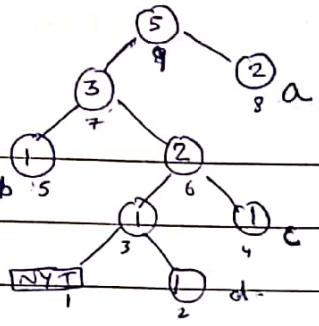
→ After inserting c,



Code: 000

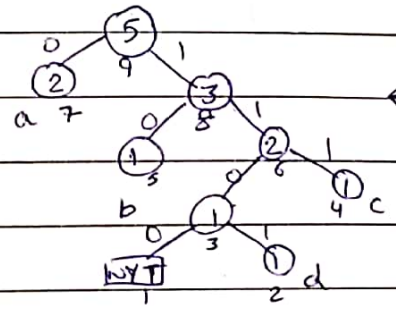
→ After inserting d,



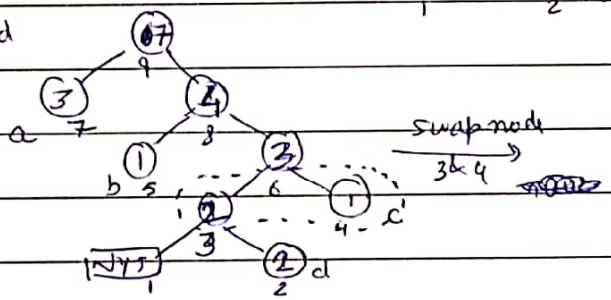


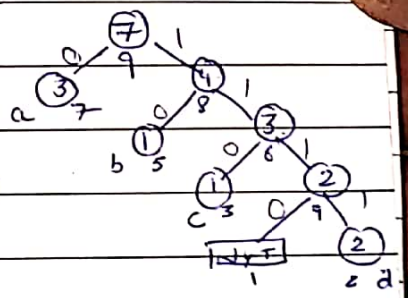
Swap node 5 & 6

code(d): 1100 fixed(d)
NYT



→ After inserting a & d





⊛ Performance of extended huffman codes : $H(S) \leq \bar{R} < H(S) + \frac{1}{n}$

LZW decoding Algo

Initialize dictionary

Decode first a index to w

Repeat not end of indices

{ Step1 : Decode next index to S

Step 2 : Add w with first character of S to the dictionary.

Step3 : Update w by S

}

dictionary based coding < static / Adaptive

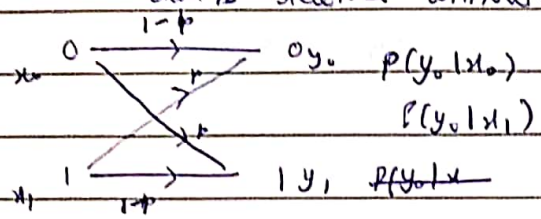
LZ77 LZ78 LZW

Information channel :=

BSC (Binary symmetric channel)

0 ——— 0
1 ——— 1 (same data at sender & receiver end)

→ data is received without errors.



→ crossover Prob. : p

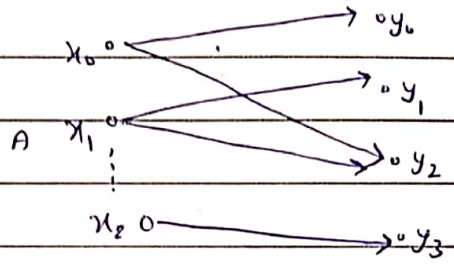
A → Set of source symbol

B → " " outcome "

P → Transition probability

(A, B, P)

$P(y_j|x_i)$



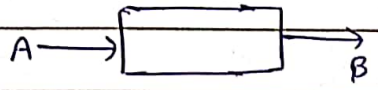
Channel Matrix :

	y_0	y_1	y_2	y_3
x_0	$P(y_0\|x_0)$	$P(y_1\|x_0)$	$P(y_2\|x_0)$	$P(y_3\|x_0)$
x_1	$P(y_0\|x_1)$	- - -	.	
x_2	$P(y_0\|x_2)$	- - -	.	

Assumptions

→ A particular channel should be stationary.

→ Channel should be memory less.



H(A) H(B)

Conclusions on channel Matrix

1) $\sum_{j=1}^{K} P(y_j/x_i) = 1$ i.e, Summation of each row = 1 in channel matrix.

2) $P(y_j) = \sum_{i=1}^{L} P(x_i) \cdot P(y_j/x_i)$

$= \sum_{i=1}^{L} P(x_i, y_j)$ (All. to Bay's theorem)

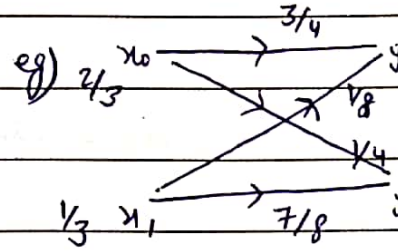
Joint Probability

$$H(A) = \sum_i P(x_i) \log \frac{1}{P(x_i)}$$

$\log_b a$

$$H(B) = \sum_j P(y_j) \log \frac{1}{P(y_j)}$$

eg)



$P(x_0) = 2/3$ $\quad P(x_1)$ $\quad P(y_0) = 1/3$

$$P(y_0) = P(x_0) P(y_0|x_0) + P(x_1) P(y_0|x_1)$$
$$= \frac{2}{3} \times \frac{3}{4} + \frac{1}{3} \times \frac{1}{8} = \frac{13}{24}$$

$$P(y_1) = P(x_0) P(y_1|x_0) + P(x_1) P(y_1|x_1) \quad \text{or} \quad 1 - \frac{13}{24}$$
$$= \frac{11}{24}$$

now, $H(X) = \frac{2}{3} \log \frac{1}{2/3} + \frac{1}{3} \log \frac{1}{1/3} = 0.918$

$$H(Y) = \frac{13}{24} \log \frac{1}{13/24} + \frac{11}{24} \log \frac{1}{11/24} = 0$$

(*) Avg. Entropy of x with observation of y $= H(X|Y)$

$$H(X|Y_0) = P(x_0|y_0) \log \frac{1}{P(x_0|y_0)} + P(x_1|y_0) \log \frac{1}{P(x_1|y_0)}$$

$$H(x|y_1) = P(x_0|y_1) \log \frac{1}{P(x_0|y_1)} + P(x_1|y_1) \log \frac{1}{P(x_1|y_1)}$$

we know $P(x_0|y_0) = \frac{P(x_0 y_0)}{P(y_0)} = \frac{P(y_0|x_0)\, P(x_0)}{P(y_0)}$

$H(X|Y_0) = 0.391$ $\quad H(X|Y_1) = 0.946$

$\therefore H(X|Y) = 0.645 \Rightarrow H(X|Y) < H(X)$ $\Rightarrow$ we loss some info

$\therefore$ Conditional Entropy $H(X|Y) = P(Y_1) H(X|Y_1) + P(Y_0) H(X|Y_0)$

$$\Rightarrow H(X|Y) = P(Y_1) \sum_i P(x_i|y_1) \log \frac{1}{P(x_i|y_1)} + P(y_0) \sum_i P(x_i|y_0) \log \frac{1}{P(x_i|y_0)}$$

$$\boxed{H(X|Y) = \sum_i \sum_j P(y_j)\, P(x_i|y_j) \log \frac{1}{P(x_i|y_j)}}$$

$$\Rightarrow \boxed{H(X|Y) = \sum_i \sum_j P(x_i, y_j) \log \frac{1}{P(x_i|y_j)}}$$

$$\boxed{\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \end{aligned}} \leftarrow \text{mutual Information}$$

mutual information represent the loss that happens during channel transmission.

Channels are categorized as

1) Noise free $\Rightarrow H(X|Y) = 0$
2) Noisy channel $\Rightarrow I(X;Y) > 0$ i.e $H(X) > H(X|Y)$
3) Ambiguous channel $\Rightarrow I(X;Y) = 0$ i.e $H(X) = H(X|Y)$

$$H(X,Y) = \sum_i \sum_j P(x_i, y_j) \log_2 \frac{1}{P(x_i, y_j)}$$

$$= \sum_i \sum_j P(x_i, y_j) \log_2 \frac{1}{P(x_i|y_j) P(y_j)}$$

$$= \sum_i \sum_j P(x_i, y_j) \log \frac{1}{P(y_j)} + \sum_i \sum_j P(x_i, y_j) \log \frac{1}{P(x_i|y_j)}$$

$$= \sum_i \left(P(x_i, y_j) \log \frac{1}{\sum_i P(x_i, y_j)} \right) + H(x|y)$$

$$H(X,Y) = H(Y) + H(X|Y)$$

Similarly, $H(X,Y) = H(X) + H(Y|X)$

$$I(X;Y) = H(X) - H(X|Y)$$

$$= \sum_i P(x_i) \log \frac{1}{P(x_i)} - \sum_i \sum_j P(x_i, y_j) \log \frac{1}{P(x_i|y_j)}$$

$$= \sum_i P(x_i) \log \frac{1}{P(x_i)} \sum_j P(y_j|x_i) - \sum_i \sum_j P(x_i, y_j) \log \frac{1}{P(x_i|y_j)}$$

$$= \sum_i \sum_j P(x_i, y_j) \log \frac{1}{P(x_i)} - \sum \sum P(x_i, y_j) \log \frac{1}{P(x_i|y_j)}$$

$$= \sum_i \sum_j P(x_i, y_j) \log \frac{P(x_i, y_j)}{P(x_i) P(y_j)}$$

Properties :-

1) The mutual Information of a channel is symmetric.

$$I(X;Y) = I(Y;X)$$

$$I(X;Y) = H(X) - H(X|Y)$$

$$= \sum_i P(x_i) \log \frac{1}{P(x_i)} - \sum_i \sum_j P(x_i, y_j) \log \frac{1}{P(x_i|y_j)}$$

$\because \sum_j P(y_j|x_i) = 1 \Rightarrow$

$$= \sum_i \sum_j P(x_i)\, P(y_j|x_i) \log \frac{1}{P(x_i)} - \sum_i \sum_j P(x_i, y_j) \log \frac{1}{P(x_i|y_j)}$$

$$= \sum_i \sum_j P(x_i, y_j) \log \frac{1}{P(x_i)} - \text{"}$$

$$\Rightarrow I(X;Y) = \sum_i \sum_j P(x_i, y_j) \log \frac{P(x_i|y_j)}{P(x_i)}$$

Similarly, $$I(Y;X) = \sum_i \sum_j P(x_i, y_j) \log \frac{P(y_j|x_i)}{P(y_j)}$$

2) Mutual Information is always nonnegative.

$$I(X;Y) \geq 0$$

Proof :- we know, $$I(X;Y) = \sum_i \sum_j P(x_i, y_j) \log_2 \frac{P(x_i|y_j)}{P(x_i)}$$

$$= \sum_i \sum_j P(x_i, y_j) \log_2 \frac{P(x_i, y_j)}{P(x_i)\, P(y_j)}$$

$$= \log_2 e \sum_i \sum_j P(x_i, y_j) \ln \frac{P(x_i, y_j)}{P(x_i) P(y_j)}$$

$\because \ln x \leq x-1$

$\Rightarrow -\ln x \geq 1-x$

$\therefore I(X;Y) \geq \log_2 e \left\{ \sum_i \sum_j P(x_i, y_j) \left(1 - \frac{P(x_i)P(y_j)}{P(x_i, y_j)}\right)\right\}$

$\Rightarrow I(X;Y) \geq \log_2 e \left\{ \underbrace{\sum_i \sum_j P(x_i, y_j)}_{I} - \underbrace{\sum_i \sum_j P(x_i)P(y_j)}_{I} \right\}$

$\boxed{I(X;Y) \geq 0}$

3) $I(X;Y) = H(X) + H(Y) - H(X;Y)$

Proof → we know, $I(X;Y) = H(X) - H(X|Y)$ — ①

$H(X,Y) = H(Y) + H(X|Y)$ — ②

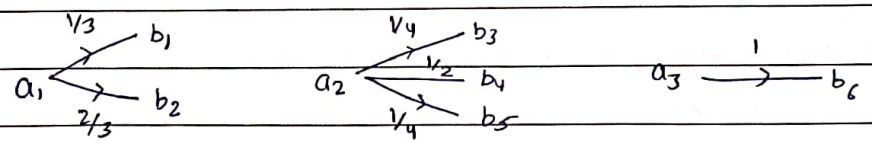
from ① & ②, $\boxed{I(X;Y) = H(X) + H(Y) - H(X;Y)}$

→ channel capacity : $C = \text{Max}\ (I(X;Y))$ w.r.t. $P(X)$

Theorem: $\frac{H(X)}{T_s} < \left(\frac{C}{T_c}\right)$ → critical rate — Shanon's Second theorem

(*) Reliable communication is possible if above condⁿ is satisfied

Q7



Find out whether channel is noisy, noise free or Ambiguous.

$H(A/B) = \sum P(b) \sum P(a|b) \log \frac{1}{P(a|b)}$

$P(a_1|b_1) = \frac{P(a_1, b_1)}{P(b_1)} = \frac{\sum P(b_1|a_1) P(a_1)}{\sum_j P(a_j) P(b_j|a_j)}$

$= \frac{\frac{1}{3} P(a_1)}{P(a_1) \times \frac{1}{3} + P(a_2) \times 0} = 1$

$\therefore P(a_2|b_1) = P(a_3|b_1) = 0$

$\therefore H(A|B) = \sum P(b) \sum P(a|b) \log \frac{1}{P(a|b)} = \sum P(b) \sum 1 \log \frac{1}{1}$

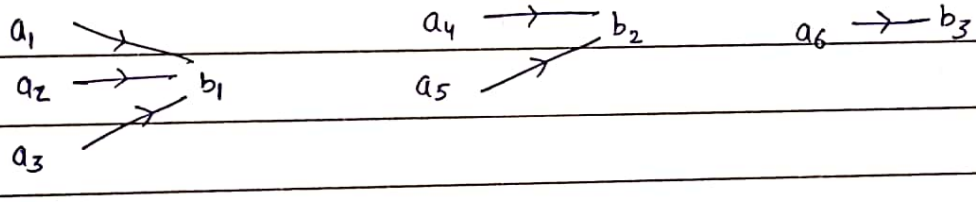
$H(A|B) = 0$

Channel matrix of this particular channel will be

	b_1	b_2	b_3	b_4	b_5	b_6
a_1	1/3	2/3	0	0	0	0
a_2	0	0	1/4	1/2	1/4	0
a_3	0	0	0	0	0	1

(*) Each column has only one nonzero entry. Such a channel is noise-free.

Deterministic channel



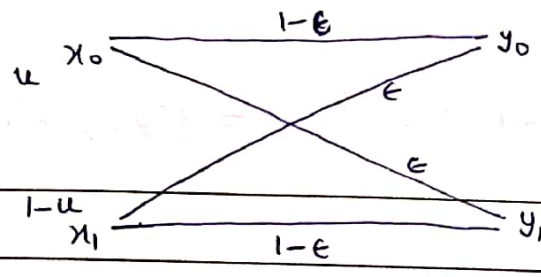
$H(A|B) = 0$ noisefree channel

Channel matrix:

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

Every row has only one nonzero entry.

⇒ channel is noise free & deterministic channel.



we have to find the channel capacity of this system.

We know, $I(X;Y) = H(Y) - H(Y|X)$

$P(y_0) = P(x_0)P(y_0|x_0) + P(x_1)P(y_0|x_1)$

$\Rightarrow P(y_0) = u(1-\epsilon) + (1-u)\epsilon = A$ —①

$P(y_1) = P(x_0)P(y_1|x_0) + P(x_1)P(y_1|x_1)$

$\Rightarrow P(y_1) = u\epsilon + (1-u)(1-\epsilon) = B$ —②

$\therefore H(Y) = A\log\frac{1}{A} + B\log\frac{1}{B}$

$H(Y|X) = \sum_i \sum_j P(x_i)\, P(y_j|x_i) \log\frac{1}{P(y_j|x_i)}$

$y=0$: $H(Y|X) = P(x_0)P(y_0|x_0)\log\frac{1}{P(y_0|x_0)} + P(x_1)P(y_0|x_1)\log\frac{1}{P(y_0|x_1)}$

$= u(1-\epsilon)\log\frac{1}{1-\epsilon} + (1-u)\epsilon\log\frac{1}{\epsilon}$

$y=1$: $H(Y|X) = P(x_0)P(y_1|x_0)\log\frac{1}{P(y_1|x_0)} + P(x_1)P(y_1|x_1)\log\frac{1}{P(y_1|x_1)}$

$= u\epsilon\log\frac{1}{\epsilon} + (1-u)(1-\epsilon)\log\frac{1}{1-\epsilon}$

$\therefore H(Y|X) = \epsilon(u+1-u)\log\frac{1}{\epsilon} + (1-u+u)(1-\epsilon)\log\frac{1}{1-\epsilon}$

$H(Y|X) = \epsilon\log\frac{1}{\epsilon} + (1-\epsilon)\log\frac{1}{1-\epsilon}$ —③

Now $I(X;Y) = H(Y) - H(Y|X)$

$\Rightarrow I(X;Y) = -A\log A + -B\log B - H_\epsilon$ —④

$$C = \underset{\text{w.r.t.} u}{\text{Max}}\ I(y;x)$$

$$I' = -\frac{A}{A}A' - A'\log A - \frac{B}{B}B' - B'\log B$$

now, $A' = 1-\epsilon-\epsilon = 1-2\epsilon$

$B' = \epsilon - (1-\epsilon) = 2\epsilon - 1$

$\therefore I' = -(1-2\epsilon) - (1-2\epsilon)\log A + (1-2\epsilon) + (1-2\epsilon)\log B$

$= (1-2\epsilon)\log\frac{B}{A}$

Put $I' = 0$ $\Rightarrow$ $\boxed{\epsilon = \frac{1}{2}}$ $\log B = \log A$ $\Rightarrow B = A$

$\Rightarrow B = A$

so, $u(1-\epsilon) + \epsilon(1-u) = u\epsilon + (1-u)(1-\epsilon)$

After Solving this equation we get, $\boxed{u = \frac{1}{2}}$

$C = H(Y) - H(Y|X)$ at $u = \frac{1}{2}$

$\because H(Y) = A\log\frac{1}{A} - B\log\frac{1}{B}$

$A = \frac{1}{2}(1-\epsilon) + \epsilon\times\frac{1}{2} = \frac{1}{2}(1-\epsilon+\epsilon) = \frac{1}{2}$

$B = \frac{1}{2}\epsilon + \frac{1}{2}(1-\epsilon) = \frac{1}{2}$

$\therefore C = \frac{1}{2}\log 2 + \frac{1}{2}\log 2 - H\epsilon = 1 - H_\epsilon$

$\Rightarrow$ $\boxed{C = 1 - \left(\epsilon\log\frac{1}{\epsilon} + (1-\epsilon)\log\frac{1}{1-\epsilon}\right)}$

Channel Coding

→ Error detection
→ " correction

Parity checking — Even Parity
Odd Parity

We generally Conside Even Parity Code.

Channel Coding — Block Code
Convolution

Block Codes :- a type of error correcting code.

Length of Codeword ← (n, k) → no. of ~~bt~~ information bits $n > k$

⇒ $n-k$ are no. of redundant bits

(n, 1) Repetition code : n bits repeated 1 times.

eg) (3, 1) 0 — 000
1 — 111

(5, 1) 0 → 00000
1 — 11111

Tx Rx
0 ⇒ 000 —— 000 decoding based on majority bits or majority of voting
either 0 or 1 will be major

⇒0

000 —— 010 ⇒0

000 —— 011

So upto 1 bit of error can be handled for 3 bits.

$P = 10^{-6}$ very less value of Prob.

eg) (3, 2) even parity codes

00	If we want to construct even parity codes
01	
10	
11	

k	n
00	000
01	011
10	101
11	110

Tx 01 → 011 ; 00 → 000 ; 10 → 101

Rx 010 → 00 ; 000 → 00 ; 101 → 10

(*) out of 3 bit codewords i.e, from 8 possible codewords 4 are valid codewords (VC) & 4 are invalid.

000	VC
001	X
010	X
011	VC
100	X
101	VC
110	VC
111	X

decoding ~~failure~~ erroneous (De): decoding is possible but result may is not correct.

- when ~~more than~~ error is fnt at 2 bits say LSB & MSB.

110

~~deco~~ decoding failure (Df): Invalid codeword is received & hence unable to decode.

decoding ~~failure~~: whenever a code crosses the limit of error that can be corrected.

Q) For a given (5,4) even parity code. find the Prob. of correct decoding, Prob. of De, Prob. of Df. Prob. of error is 0.1.

P_c ↓ P_e ↓ P_f

Soln →

$$P_c = {}^5C_0\, p^0 (1-p)^5$$

$$P_e = {}^5C_2\, p^2 (1-p)^3 + {}^5C_4\, p^4 (1-p)^1$$

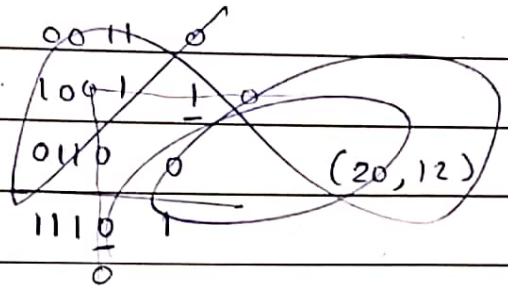
$$P_f = {}^5C_1\, p^1 (1-p)^4 + {}^5C_3\, p^3 (1-p)^2 + {}^5C_5\, p^5 (1-p)^0$$

or we can use $P_c + P_e + P_f = 1$

Product Code: Basically represent in 2-D form.

0011 1011 0110 →

0	0 1 1	0	
1	0 0 1	1	0
0	1 1 0	0	(20, 12)
1	1 1 0	1	
	0		



Hamming Codes:

Parity bits added locate the error position.

For index 0 to 10 we use 4 bits along with codeword.

Condition: $m \le 2^{n-K}$

1011 011
K = 7

let $n-K = r$

$m \le 2^r$

$\therefore K + r \le 2^r \Rightarrow 7 + r \le 2^r$

Putting $r = 4$, above condition is satisfied.

Hence 4 bit parity is needed.

D7	D6	D5	P_8	D4	D3	D2	P_4	D1	P_2	P_1
1	0	1	0	1	0	1	0	1	1	1
11	10	9	8	7	6	5	4	3	2	1

2^i → Position of parity

$i = 0, 1, 2, 3$

For rest of the position we can put the data

now we compute the parity,

P_1 : Read one bit skip 1 bit
P_2 : " 2 " " 2 "
P_4 : " 4 " " 4 "
P_8 : " 8 " " 8 "

$P_1 = 1+1+1+1+1+0 = 1$ $\quad P_2 = 1+1+1 = 1$

$P_4 = 1+1 = 0$ $\quad P_8 = 1+1 = 0$

Suppose error is present at 5th position

1	0	1	0	1	0	0	0	1	1	1

$P_8\ P_4\ P_2\ P_1$

$(0\ 1\ 0\ 1)_2 = (5)_{10}$ $\Rightarrow$ error is present at 5th location.

So flip the bit at 5th location.

(*) In (5,2) error correction limit is 2 bits.

(3,2) even parity

00	000
01	011
10	101
11	110

$a = a_1 a_2 a_3 \cdots a_n$

$b = b_1 b_2 b_3 \cdots b_n$

Receiver predict based on the received product what actual code was sent

$a = D(b)$

↳ decoding function

Prob of a when b is received at receiver end. $= P(a|b)$

↓
backward Prob

$$\therefore \quad P(a|b) = \frac{P(b|a) \cdot P(a)}{P(b)}$$

where, $P(b|a) = \sum_i P(b|a_i)$

$$P(b) = \sum_i P(a_i)\, P(b_j|a_i)$$

we can correctly predict the value of a with prob. $P(a/b)$

$\therefore$ Prob. of unsuccessful prediction of $a = 1 - P(a|b)$

Minimum error decoding Rule :=

$$D_{ME}(b) = a^*$$

$$P(a^*|b) \geq P(a_i|b) \quad \forall i$$

$$\Rightarrow \frac{P(b|a^*)\,P(a^*)}{P(b)} \geq \frac{P(b|a_i)\,P(a_i)}{P(b)}$$

$$\Rightarrow P(b|a^*)\,P(a^*) \geq P(b|a_i)\,P(a_i)$$

Maximum likelihood decoding rule :=

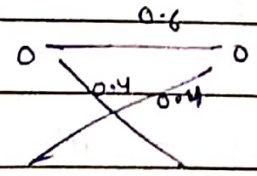
Sometime source prob. is unknown then we assume that a^* & a_i are equiprobable.

$$P(b|a^*) \geq P(b|a_i) \quad \forall i$$

Q) $P = \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}$

		$P(a_i)$
a_1	000	0.4
a_2	011	0.2
a_3	101	0.1
a_4	110	0.3

$b = 111$



what was the actual message when $b = 111$ is received i.e $D(111)$?

$= a$

we know, ~~P(a*)~~ $P(b|a^*) \geq P(b|a_i) \quad \forall i$

$P(b|a_1) = P(1|0)\,P(1|0)\,P(1|0) = 0.4 \times 0.4 \times 0.4 = 0.064$

$P(b|a_2) = P(1|0)\,P(1|1)\,P(1|1) = 0.144$

$P(b|a_3) = 0.144$

$P(b|a_4) = 0.144$

max. Prob is 0.144 with 3 diff. choices on 111 mapped to 3 diff. positions hence we are unable to correct the error with maximum likelihood decoding rule.

Now, use ~~min~~ min. error decoding rule.

$P(a^*)\,P(b|a^*) \geq P(a_i)\,P(b|a_i) \quad \forall i$

$P(a_1)P(B|a_1) = P(a_1)\,P(1|0)\,P(1|0)\,P(1|0) = 0.4 \times 0.064 = 0.0256$

$P(a_2)\,P(B|a_2) = 0.2 \times 0.144 = 0.0288$

$P(a_3)\,P(B|a_3) = 0.1 \times 0.144 = 0.0144$

$P(a_4)\,P(B|a_4) = 0.3 \times 0.144 = 0.0432$ → max. Prob.

hence $b = 111$ can be mapped to $a_4 = 110$ with min. error decoding rule.

Hamming distance decoding rule :=

$a = 110110$

$b = 101111$

$d(a,b) = 3$ $\qquad$ $d(a,b) = HW(a \oplus b)$ → Hamming weight

Hamming weight = no. of non zero elements.

$$P(b|a) = \varepsilon^{D}(1-\varepsilon)^{N-D}$$

$\varepsilon \rightarrow$ error Probability or crossover Probability

(*) If ~~$\varepsilon > 0.5$~~ $\varepsilon < 0.5$ then reliable communication is possible.

our objective is to maximize $P(b|a)$

$\therefore \quad d(a^*, b) \leq d(a_i, b) \quad \forall i$

Q) dist with b_1 & b_2

000	1*	3	$b_1 = 010$ $b_2 = 111$
001	2	2	In 000 & 011 hamming dist. is min for b_1
011	1*	1	So error detection is possible
111	2	0*	

If $b = 110$ then

000	2
001	3
011	2
111	1*

error correction is only possible when we have single choice for min. hamming distance.

Linear Codes : codeword is represented in this particular way.

~~$C = I \cdot G$~~ $\boxed{C = i \cdot G}$

C → Codeword, i → information vector, G → Generation matrix

Consider the generator matrix for (5,3) linear code $G = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$

000	→	00000
001	→	01110
111	→	10011

dey^z

1) In linear code, one codeword will consist of 0 vector.

2) ~~One~~ If there exist 2 codewords c_i & c_j then $c_k = c_i + c_j$ i.e, Summation of any 2 valid codeword produce a new valid codeword.

3) Min. Hamming weight $d(Y_n)$ = min. Hamming weight of non zero codewords.

C_0	0 0 0
C_1	0 1 1
C_2	1 0 1
C_3	1 1 0

	C_0	C_1	C_2	C_3
C_0	C_0	C_1	C_2	C_3
C_1	C_1	C_0	C_3	C_2
C_2	C_2	C_3	C_0	C_1
C_3	C_3	C_2	C_1	C_0

→ Every parity code is linear code

Eg) For (3,2) odd parity code,

C_0	0 0	→ 0 0 1
C_1	0 1	→ 0 1 0
C_2	1 0	→ 1 0 0
C_3	1 1	→ 1 1 1

C_0	0 0 1
C_1	0 1 0
	0 1 1 → Invalid

Hence, odd parity code is ~~invalid~~ not linear code.

⊛ Even parity code is linear code.

⊛ ∵ It is always prefer to consider even parity in Hamming code.

linear code words → Zero codeword : for even parity → C_0

→ Nonzero code word " → C_1, C_2, C_3

⊛ $\boxed{d(Y_n) = \min_{HW} d(C_i, C_j)} \quad i \neq j$

Eg)

	C_0	C_1	C_2	C_3
C_0	0	2	2	2
C_1	2	0	2	2
C_2	2	2	0	2
C_3	2	2	2	0

$i = j$

For $i \neq j$, $d(Y_n)$ = min Ham. dis. = 2

∴ In (3,2) even parity code , $d(\chi_n) = 2$

Q) 1) construct (7,4) Hamming code with even parity.

2) Prove that (7,4) Hamming code is a linear code.

3) ~~Find~~ write the Algorithmic steps to compute the Basis vector of any given set.

4) Find the basis vector of (7,4) Hamming code.

Basis vector :- consider n vectors : $v_1, v_2, \ldots, v_n$

then basis vectors of given vectors are: $\bar{v_1}, \bar{v_2}, \ldots, \bar{v_m}$ where $m < n$

$$v_1 = \omega_1 \bar{v_1} + \omega_2 \bar{v_2} + \ldots + \omega_m \bar{v_m}$$

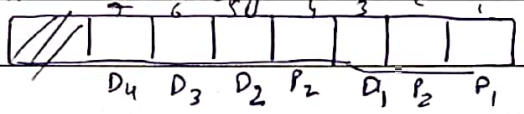
(*) ~~Gener~~ Generator matrix is a collection of basis vector.

Systematic Code word : Identification of info. bits & parity bits is easy.

eg)

00	000
01	011
10	101
11	110

eg) In Hamming code



Q) (5,3) $G = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$

110 → 11101 (not a systematic codeword)

∴ Adjust G as $G = [I \mid P]$

then G is able to construct all systematic codewords

Apply elementary row operation $R_3 \to R_3 + R_2$ on G_1

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

$R_1 \to R_1 + R_3$ $\quad G'' = \left(\begin{array}{ccc|cc} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{array}\right) \Leftarrow$ Keep on apply elementary row operation until I matrix is not received.

$\qquad\qquad\qquad\qquad\qquad\qquad\quad I \qquad P$

(*) Canonical representation of G

now, $110 \longrightarrow 11010$ (→ now it is a systematic codeword)

Syndrome value: (S) $\quad e \longrightarrow v$

$$S = v \cdot H^T$$

H → Parity check matrix

if $S = 0$, then v is a valid codeword

if $S \neq 0$, then error is present in particular codeword.

Relation b/w G & H: $\quad G \cdot H^T = 0$

~~$G = [I|P] \cdot H^T = 0$~~ $\quad \because G = [I|P]$

$$\Rightarrow \quad H = [P^T | I]$$

$$G_{3\times 5} = [I_3 | P]$$

for $G = \left[\begin{array}{ccc|cc} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{array}\right]$

$$\Rightarrow \quad H = \left[\begin{array}{ccc|cc} 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{array}\right]_{2\times 5}$$

now $G H^T$ will be equal to 0.

(*) For $G_{K\times n}$ dimension of H are $(n-K) \times n$.

$\therefore$ ~~$G_{K\times n} \cdot H_{K-n\times n}$~~ $G_{K\times n} H^{T}_{n\times n-K}$ is a zero matrix.

$\therefore$ For valid codeword, there is a zero matrix or $s=0$.

Theorem: Error detection property

A Block code K_n can detect upto t error if & only if its $d(K_n) > t$

Proof: 'b' is the received codeword & a^* is the valid codeword of 'b' and rest of the codewords are represented by a_i.

$d(b, a_i) > 0$ —(i) bcoz if $d(b, a_i) = 0$ then there would be no error.

Using triangle inequality,

$$d(a^*, b) + d(b, a_i) \geq d(a^*, a_i)$$

$$d(b, a_i) \geq d(a^*, a_i) - d(a^*, b) \quad \text{—(ii)}$$

Combine (i) & (ii): $d(a^*, a_i) - d(a^*, b) > 0$

let $d(a^*, b) = t$ and we know ~~$d(b$~~ $d(K_n) = d(a^*, a_i)$

$\therefore$ $d(K_n) - t > 0$

$d(K_n) > t$, Hence proved.

Theorem: A Block code K_n corrects upto t errors iff $d(K_n) > 2t$.

Proof: 'b' = received codeword

a^* = valid codeword corresponding to b

a_i = rest of all codewords

a^* (Tx) —— b (Rx)

let $d(b, a^*) = t$

i) $d(b, a^*) < d(b, a_i) \quad \forall i$

then only we can map b to a^*

Using triangle inequality, $d(a^*, b) + d(b, a_i) \geq d(a^*, a_i)$

$\Rightarrow$ $d(a^*, b) \geq d(a^*, a_i) - d(b, a_i)$

ii) $\Rightarrow \quad d(a^*, a_i) - d(b, a_i) \leq d(a^*, b)$ Just reverse of prev. stmt.

For eq. (i) to hold given ii)

$$d(a^*, a_i) \geq 2d(a^*, b)$$

$\Rightarrow \quad d(K_n) > 2t$, Hence Proved.

—x—

$$d(a^*, b) = t \quad \text{—(i)}$$

$$d(a^*, b) < d(a_i, b) \quad \forall i \quad \text{—(ii)}$$

using Δ inequality, $d(a^*, b) + d(b, a_i) \geq d(a^*, a_i)$

$\Rightarrow \quad d(b, a_i) \geq d(a^*, a_i) - d(a^*, b)$ —(iii)

To satisfy the inequality (ii), $d(a^*, a_i) - d(a^*, b) > d(a^*, b)$

$\Rightarrow \quad d(a^*, a_i) > 2d(a^*, b)$

$\Rightarrow \quad \boxed{d(K_n) > 2t}$

$\Rightarrow$ For (3, 2) codewords,

		Hamming dist. d	$d(K_n)$
0 0	0 0 0	— 0	min. = 2
0 1	0 1 1	— 2	$d(K_n) > 2t$
1 0	1 0 1	— 2	$2 > 2t$
1 1	1 1 0	— 2	Hence, 1 bit of error can be detected.

The Standard array :

w_0, w_1, w_2 --- are all possible combinations of error vectors

c_0, c_1, c_2, -- are all codewords.

	c_0	c_1	c_2	..... c_{n-1}
w_0	$c_0 + w_0$	$c_1 + w_0$	$c_2 + w_0$	- - - - - - .
w_1				
⋮				

(*) V is a valid codeword if $V \cdot H^T = 0$

because $V = i \cdot G$

$\Rightarrow i \cdot G H^T = 0$

and $GH^T = 0$ if V is a valid codeword.

whenever a error is present, syndrome solely depends on it.

$VH^T = (C+e) \cdot H^T$

$= cH^T + eH^T \quad (cH^T = 0)$

$= eH^T$

If size of matrix is large then searching & other mechanism will not be feasible.

Syndrome Error Table := $S = V \cdot H^T = e \cdot H^T$

For (7,4) Hamming code

All possible error patterns in (7,4) Hamming code

0000001	error is that at 1st LSB	
0000010	" " " " 2nd LSB	
0000100		
0001000	$+e_3$	$\cdot H^T$
0010000		$\cdot H^T$
0100000		$\cdot H^T$
1000000		$\cdot H^T$

multiply these error patterns with H^T we receive Syndrome vectors.

$\Rightarrow$ Syndrome ~~is~~ solely depend on error pattern.

we Know, $V = C + e$

$\Rightarrow C = V + e$ (in modulo 2 addition)

say $C = V + e_3$

Q) For (6,3) linear code $G = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{array}\right]$

	Codewords	
0 0 0	→ 000000	
0 0 1	→ 001110	3
0 1 0	→ 010011	3
0 1 1	→ 011101	4
1 0 0	→ 100101	3
1 0 1	→ 101011	4
1 1 0	→ 110110	4
1 1 1	→ 111000	3

$\Rightarrow d(K_n) = 3$

$\Rightarrow$ Upto 1 bit of error can be detected.

$$H = P^T I = \left[\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$$

$$\Rightarrow H^T = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

So syndrome table will be calculated as.

000001	001
000010	010
000100	100
001000	110
010000	011
100000	101

error syndromes by $S_n = V_1 H^T$

now for $\underline{v} = 100110$

$$c = \begin{array}{r} 100110 \; v \\ 0100000 \; e \\ \hline 110110 \end{array}$$

So if 1 bit of error is there, it can be corrected.

$\therefore V \cdot H^T \neq 0$ gives 011

Cyclic Codes :=

101110 represented as Polynomial $x^5+x^3+x^2+x$

5 4 3 2 1 0

or $1\cdot x^5+0\cdot x^4+1\cdot x^3+1\cdot x^2+1\cdot x+0\cdot x^0$

Here, $C(x) = i(x)\cdot g(x)$

eg) In (7,4) cyclic code $g(x) = x^3+x+1$

Information bits	Polynomial
0001	1
0010	x
0011	$x+1$
0100	x^2
⋮	⋮

x ⟶ $C(x) = i(x)\cdot g(x) = x(x^3+x+1) = x^4+x^2+x$

$= 0\,0\,0\,1\,0\,1\,0$

(*) In linear code if $vH^T = 0$ ⇒ v is a valid codeword

but In ~~Circular~~ Cyclic code, $R_{g(x)}\cdot C(x) = 0$ then $C(x)$ is valid

Now, $R_{g(x)}\cdot V(x) = R_{g(x)}(C(x)+e(x))$

∴ Syndrome $S(x) = R_{g(x)}\cdot e(x)$

$g(x)$: For an (n,k) binary cyclic code, the generator polynomial has the form $g(x) = g_{n-k}x^{n-k} + g_{n-k-1}x^{n-k-1} + \cdots + g_2x^2 + g_1x + g_0$

where the coefficient $g_{n-k} = g_0 = 1$ and rest of the $g_i = 0/1$ (either 0 or 1)

The generator polynomial is a unique polynomial from which all the ~~codewords~~ codepolynomials can be generated.

Theorem 1: The nonzero code polynomial of min. degree in a cyclic code C is unique.

Proof: $C(x) = C_{n-1}x^{n-1} + C_{n-2}x^{n-2} + \cdots + C_2x^2 + C_1x + C_0$

$\Rightarrow$ max degree of Polynomial is $n-1$ for (n,k) cyclic code

$$i(x) = i_{k-1}x^{k-1} + \cdots + i_1x + i_0$$

now suppose we consider $g(x) = g_{n-k}x^{n-k} + \cdots + g_1x + g_0$

By contradiction, we assume there exist multiple polynomials with min. degree $n-k$

$\therefore g'(x) = g'_{n-k}x^{n-k} + \cdots + g'_1x + g'_0$

if we add $g(x)$ & $g'(x)$ i.e, $g''(x) = g(x) + g'(x)$

$\Rightarrow g''(x) =$

↓

valid Poly^n have degree less than $n-k$ → contradiction

Hence, $g(x)$ is always unique with min. degree $n-k$.

—x—

~~Theorem 2:~~ ⊛ If we perform a cyclic operation on a cyclic codeword, we get another valid cyclic codeword.

$C_{n-1}\ C_{n-2}\ \cdots\ C_2\ C_1\ C_0$ (valid cyclic) $\xrightarrow{\text{left rotate}}$ $C_{n-2}\ C_{n-3}\ \cdots\ C_1\ C_0\ C_{n-1}$ (valid cyclic)

eg) $C(x) = x^5 + x^3 + x^2 + 1$ $(6,k)$.

1 0 1 1 0 1

↳ 0 1 1 0 1 1

$R_{x^6+1}(x \cdot C(x))$

$x(C(x)) = x^6 + x^4 + x^3 + x$

$= 1\ 0\ 1\ 1\ 0\ 1\ 0$ (length becomes 7)

So take Residual R_{x^6+1}

i.e, $x^6+1 \overline{)\ x^6 + x^4 + x^3 + x}$

$x^6 + 1$

$x^4 + x^3 + x + 1$

↓

0 1 1 0 1 1

(*) For i bit circular shift $R_{x^n+1}(x^i \cdot c(x))$

eg) for 1 bit cyclic shift in (6,K) $R_{x^6+1}(x \cdot c(x))$

" 2 " " " " " $R_{x^6+1}(x^2 \cdot c(x))$

Theorem 2: Let $g(x) = x^r + g_{r-1}x^{r-1} + \cdots + g_2x^2 + g_1x + g_0$ be the nonzero codepolynomial of min degree r in (n, K) cyclic Polynomial. The const. term g_0 must be equal to 1.

Proof: $\therefore g(x) = x^r + g_{r-1}x^{r-1} + \cdots + g_2x^2 + g_1x$ (By having $g_0 = 0$)

$= x(x^{r-1} + g_{r-1}x^{r-2} + \cdots + g_2x + g_1)$

$= x \cdot g'(x)$

So $g'(x)$ is also a valid codeword (Polynomial) with degree $r-1$ which is less than r

$\Rightarrow$ contradiction

Hence g_0 must be equal to 1.

Theorem 3: Let $g(x) = x^r + g_{r-1}x^{r-1} + \cdots + g_2x^2 + g_1x + g_0$ be the nonzero codepolynomial of min-degree in a (n, K) cyclic code C. A binary Polynomial of degree $n-1$ or less is a codepolynomial iff it is a multiple of $g(x)$.

Proof: Initially we assume $v(x)$ is a valid codeword which is not perfectly divisible by $g(x)$

$\therefore v(x) = \underbrace{a(x)g(x)}_{c_1} + r(x)$ where $r(x) \neq 0$

(valid codeword) (valid codeword)

$\Rightarrow r(x) = c_1(x) + v(x)$ (Adding 2 valid codeword (VC) gives a VC)

$= c_2(x)$

degree $(v(x))$ < degree $(g(x))$ $\rightarrow$ contradiction.

Hence, valid codewords must be perfectly divisible by $g(x)$.

$x^n g(x)$ → Cyclic shifting of a code n times.

$x^n g(x) = (x^n + 1) + g'_n(x)$

$\Rightarrow x^n + 1 = x^n g(x) + g'_n(x)$

$\Rightarrow x^n + 1 = x^n g(x) + a(x) \cdot g(x)$ {if $g'_n(x)$ is valid then $g'_n(x) = a(x) \cdot g(x)$}

$x^n + 1 = g(x)\,[x^n + a(x)]$

eg) (7, 4) Cyclic codes

$n = 7$ ∴ $g(x)$ must be a factor of $x^7 + 1$

$\Rightarrow x^7 + 1 = (x+1)(x^6 + x^5 + x^4 + x^3 + x^2 + x + 1)$

In Galois field both + and − are same.

Put $x = 1$ we get 7 so $(x+1)$ is not a factor.

Above Polyn can be written as deg(2) × deg(4)

i.e, $x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 = (a_1x^2 + a_2x + a_3)(b_1x^4 + b_2x^3 + b_3x^2 + b_4x + b_5)$

deg(2) & deg(4) combination is not possible to solve.

So take deg(3) × deg(3)

$\Rightarrow x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 = (x^3 + x + 1)(x^3 + x^2 + 1)$

∴ $x^7 + 1 = (x+1)(x^3 + x + 1)(x^3 + x^2 + 1)$

$g(x) = x^{n-k} + g_{n-k-1}x^{n-k-1} + \cdots + g_2x^2 + g_1x + 1$

$n = 7$, $k = 4$ ∴ $g(x)$ will be a polyn of deg. 3.

∴ $g(x) = x^3 + x + 1$ or $x^3 + x^2 + 1$

∴ So generator polynomial of (7, 4) cyclic code are, $x^3 + x + 1$ and $x^3 + x^2 + 1$.

eg) (7, 3) cyclic code

$n = 7$, $K = 3$, $n - K = 4$

$\therefore g(x) = (x+1)(x^3+x+1)$

or $(x+1)(x^3+x^2+1)$

Q) How many cyclic codes are possible for (15, 11) ?

Ans $x^{15}+1 = (x+1)(x^2+x+1)(x^4+x+1)(x^4+x^3+1)(x^4+x^3+x^2+x+1)$

$n - K = 4$ . . . deg(4) deg(4) deg(4)

f_1 f_2 f_3

So 3 cyclic codes are possible.

Q) (15, 7) $n - K = 8$

$f_1 f_2$ $f_1 f_3$ $f_2 f_3$

Possible 8 deg. Polynomials

Q) (15, 12) $n - K = 3$

$(x+1)(x^2+x+1)$

Encoding :- $C(x) = i(x) \cdot g(x)$

(7, 4) → $g(x) = x^3 + x + 1$

1010 $i(x) \cdot g(x)$

$i(x) = x^3 + x$ $= (x^3+x)(x^3+x+1)$

$= x^6 + x^3 + x^2 + x$

1010 → 1001110 } not a systematic code word.

x^3+x $x^6 + x^3 + x^2 + x$

Construction of codeword in a systematic form :=

(7, 4)

$i(x) = 1010 = x^3 + x$

$x^{n-k} \cdot i(x) = x^3 \cdot (x^3 + x) = x^6 + x^4$

$\therefore$ 1010 $\longrightarrow$ 1010000 } not perfectly divisible by $g(x)$.

$v(x) = a(x)\, g(x) + r(x)$ — (1)

$x^3 + x + 1 \;\big|\; x^6 + x^4 \;\big|\; x^3 + 1$

$x^6 + x^4 + x^3$

$x + 1$

$\Rightarrow x^6 + x^4 = (x^3 + 1)(x^3 + x + 1) + (x + 1)$

from (1) $a(x)g(x) = v(x) + r(x)$

one of the $c(x)$ $= x^6 + x^4 + x + 1$

$\therefore$ $c(x) =$ 1010 011 This is perfectly divisible by $g(x)$ and in a systematic form.

i p

Steps for $i(x) = i \mid p$

1) $x^{n-k} \cdot i(x)$

2) $r(x) = R_{g(x)}\, x^{n-k} \cdot i(x)$

3) $c(x) = x^{n-k} \cdot i(x) + r(x)$

Decoding :=

$s(x) = R_{g(x)}(v(x))$

if $s(x) = 0$ valid code word.

otherwise $v(x) = c(x) + e(x)$

$\therefore s(x) = R_{g(x)} \cdot (v(x)) = R_{g(x)} [c(x) + e(x)]$

$= R_{g(x)} \cdot e(x)$

$\Rightarrow$ In a cyclic code syndrome solely depends on error pattern.

Comparison of linear codes & cyclic codes :=

	linear code	cyclic code
1)	$C = i \cdot G$	$C(x) = i(x) \cdot g(x)$
2)	$GH^T = 0$	$g(x)\, h(x) = x^n + 1$ (Parity check Polynomial)
3)	$v \cdot H^T = 0$ then v is valid CW	$R_{\frac{x^n+1}{g(x)}} v(x) = 0$ then $v(x)$ is valid CW. OR $R_{g(x)} v(x) = 0$

★) Cyclic code is a subset of linear code. linear code is a subset of block code.

Generator matrix :=

we know, $g(x) = g_n x^n + g_{n-1} x^{n-1} + \cdots + g_2 x^2 + g_1 x + g_0$

$$G = \begin{bmatrix} g_n & g_{n-1} & g_{n-2} & \cdots & g_2 & g_1 & g_0 & 0 & \cdots & 0 & 0 & \cdots \\ 0 & g_n & g_{n-1} & & g_3 & g_2 & g_1 & g_0 & 0 & \cdots & & \\ 0 & 0 & g_n & & g_4 & g_3 & g_2 & g_1 & g_0 & & & \\ \vdots & \vdots & \vdots & & & & & & & & & \end{bmatrix}$$

eg) (7,4) cyclic code $g(x) = x^3 + x + 1$

$= g_3 x^3 + g_2 x^2 + g_1 x + g_0$ $\quad g_3 = g_1 = g_0 = 1$, $g_2 = 0$

$$G = \begin{bmatrix} g_3 & g_2 & g_1 & g_0 & 0 & 0 & 0 \\ 0 & g_3 & g_2 & g_1 & g_0 & 0 & 0 \\ 0 & 0 & g_3 & g_2 & g_1 & g_0 & 0 \\ 0 & 0 & 0 & g_3 & g_2 & g_1 & g_0 \end{bmatrix}$$

now Parity check Polyn^n $h(x) = \frac{x^7+1}{g(x)}$

$= x^4 + x^2 + x + 1$

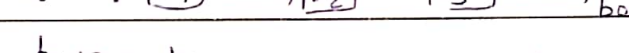
$= h_4x^4 + h_3x^3 + h_2x^2 + h_1x + h_0\ 1$

where $h_4 = 1$ $h_3 = 0$ $h_2 = 1$ $h_1 = 1$ $h_0 = 1$

order of H is $(n-k) \times n$

$$\therefore H = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & h_4 & 0 & 0 \\ 0 & h_0 & h_1 & h_2 & h_3 & h_4 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & h_4 \end{bmatrix}$$

Forward LFSR :=



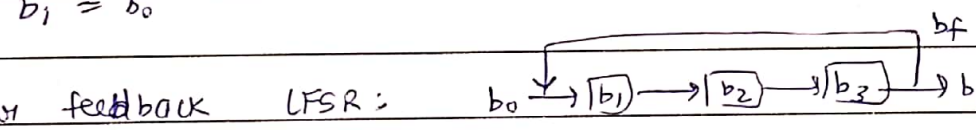
$b_{out} = b_3$ Initially $b_{out} = b_1 = b_2 = b_3 = b_0 = 0$

$b_3 = b_2$

$b_2 = b_1$

$b_1 = b_0$

linear feedback LFSR :



$b_{out} = b_3$

$b_3 = b_2$

$b_2 = b_1$

$b_1 = b_0 + b_f$

$r(x) = R_{g(x)}\, x^{n-k} i(x)$

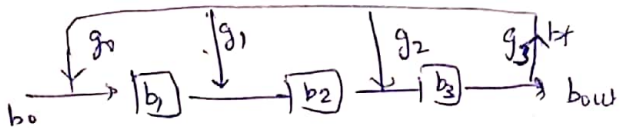
$C(x) = x^{n-k} \cdot i(x) + r(x)$

eg) $v(x) = x^5 + x^3 + x^2 + 1$ $g(x) = x^3 + x + 1$

for $\frac{v(x)}{g(x)}$, remainder will be of form $a_0 + a_1x + a_2x^2$

$\therefore g(x)$ can be written as, $g(x) = g_3x^3 + g_2x^2 + g_1x + g_0$

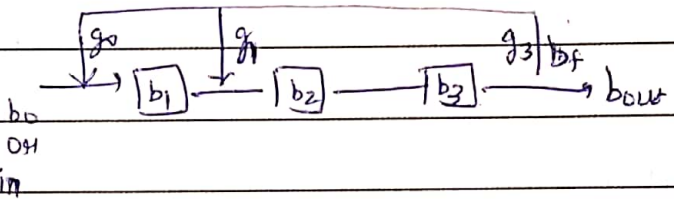
$g_3 = 1$ $g_2 = 0$ $g_1 = 1$ $g_0 = 1$



g_i are connected if $g_i \neq 0$

So when $g_3 = g_1 = g_0 = 1$, $g_2 = 0$

then ckt will be



$b_{out} = b_f = b_3$

$b_3 = b_2$

$b_2 = b_1 + b_f$

$b_1 = b_0 + b_f$

we have, $v(x) = x^5 + x^3 + x^2 + 1$

In binary, 1 0 1 1 0 1

b_{in}	b_1	b_2	b_3	b_f	b_{out}
—	0	0	0	0	
MSB 1	1	0	0	0	0
0	0	1	0	0	0
1	1	0	1	0	0
1	0	0	0	1	1
0	0	0	0	0	0 → $q(x)$
LSB 1	1	0	0	0	0

$r(x)$ = actual content in shift registers

$\therefore v(x) = q(x)\, g(x) + r(x)$

$= x^2 (x^3 + x + 1) + 1$

$= x^5 + x^3 + x^2 + 1$

Cyclic Encoding :-

$$x(x) = R_{g(x)}\, x^{n-k}\, i(x)$$

eg) (7,4) $g(x) = x^3 + x + 1$

Consider $i(x) = 1010 = x^3 + x$

$$x^3 + x + 1 \;\overline{)\; x^6 + x^4 \;(\; x^3 + 1}$$

b_{in}	b_1	b_2	b_3	b_f	b_{out}
–	0	0	0	0	0
1	1	0	0	0	0
0	0	1	0	0	0
1	1	0	1	0	0
0	1	0	0	1	1
–	0	1	0	0	0
–	0	0	1	0	0
–	1	0	1	1	1

$n-K = 7-4 = 3$

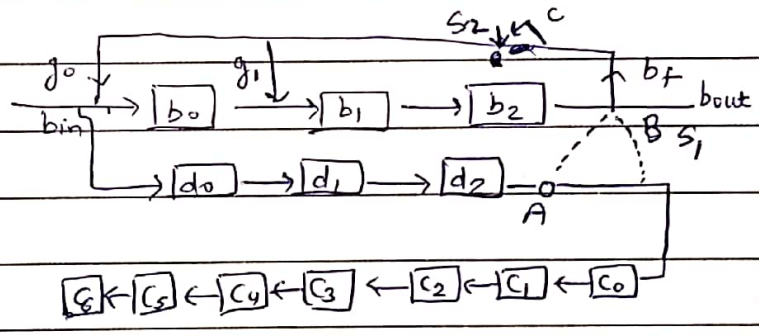
So another 3 times we continue this process

Encoding Using LFSR :=

I/P is provided at b_0 (LSB) ⇒ lower order LFSR

(7, 4) $g(x) = x^3+x+1$ $(g_3=1,\ g_1=1,\ g_0=1,\ g_2=0)$

$i(x) = x^2+1$



d_0, d_1, d_2 delay registers $= n-K$

$b_f = b_{out} = b_2$

$b_2 = b_1$

$b_1 = b_0 + b_f$

$b_0 = b_{in} + b_f$

$C_6 = C_5$

$C_5 = C_4$

$C_0 = d_2$ (Till 7 shifts (n))

$d_2 = d_1$

$d_1 = d_0$

$d_0 = b_{in}$

Initially S_2 is connected at C junction

After n shifts " " disconnected from C junction.
and S_1 also disconnected. (shift from A to B)

$x^{n-K} \cdot i(x) = x^3(x^2+1) = x^5+x^3$

$R_{g(x)}\dfrac{x^5+x^3}{x^2} = x^2$

$\therefore C(x) = x^5+x^3+x^2$

(*) Take $i(x) = x^3+x^2+1$ (1101) for table

	b_{in}	b_0	b_1	b_2	b_f (bout)	d_0	d_1	d_2	c_0	c_1	c_2	c_3	c_4	c_5	c_6
		0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0
2	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0
3	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0
4	1	0	1	1	1	1	0	1	1	0	0	0	0	0	0
5	-	1	1	1	1	0	1	0	1	1	0	0	0	0	0
6	-	1	0	1	1	0	0	1	0	1	1	0	0	0	0
7	-	1	0	0	1	0	0	0	1	0	1	1	0	0	0
									Info bits						
8		0	1	0	-	-	-	-	0	1	0	1	1	0	0
9		0	0	1	-	-	-	-	0	0	1	0	1	1	0
10		0	0	0	-	-	-	-	1	0	0	1	0	1	1

Lower order i/p based elimination encoding

$\therefore$ For $i(x) = x^3 + x^2 + 1$

$x^{n-k} \cdot i(x) = x^3(x^3 + x^2 + 1)$

$= x^6 + x^5 + x^3$

$x^3 + x + 1 \overline{)x^6 + x^5 + x^3}$ $\quad x^3 + x^2 + x + 1$

$x^6 + x^4 + x^3$

$x^5 + x^4$

$x^5 + x^3 + x^2$

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

$x^4 + x^2 + x$

$x^3 + x$

$x^3 + x + 1$

1

$C_0 = b_2$

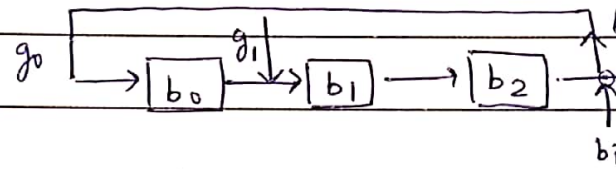
$x^6 + x^5 + x^3 + 1$

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

Higher Order LFSR encoding :=

(7,4) $g(x) = x^3 + x + 1$ $\quad i(x) = x^3 + x^2 + 1$

I/P is provided at MSB ⇒ Higher order LFSR.



→ use no. of shifts

→ doesn't require delay registers

bin	b_0	b_1	b_2	b_f	b_{out}
-	0	0	0	0	0
1	1	1	0	1	1
1	1	0	1	1	1
0	1	0	0	1	1
1	1	0	0	1	1

$R_{g(x)}\, x^{n-k} \cdot i(x)$

$b_{out} = b_f = b_2 + bin$

$b_2 = b_1$

$b_1 = b_0 + b_f$

$b_0 = b_f$

So no. of shifting operations to obtain $R_{g(x)}\, x^{n-k} \cdot i(x) = k$

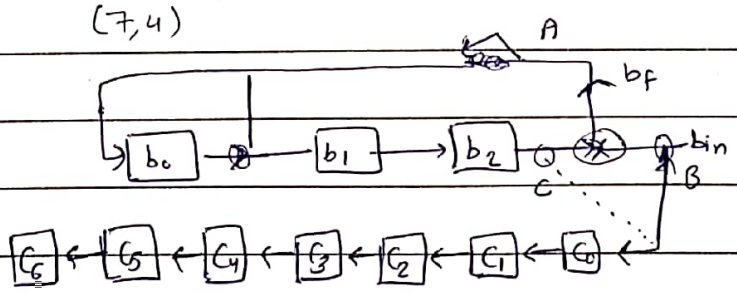
k
n-k
n-k
2n-k

(7,4) $2 \times 7 - 4 = 10$ shift operations

LFSR based Cyclic Codes :=

Consider, $g(x) = x^3 + x + 1$

(7,4)



Take $i(x) = x^3 + x^2 + 1$

$b_f = b_2 + bin$

$b_2 = b_1$

$b_1 = b_0 + b_f$

$b_0 = b_f$

$C_0 = bin$

$C_6 = C_5$

$C_5 = C_4$

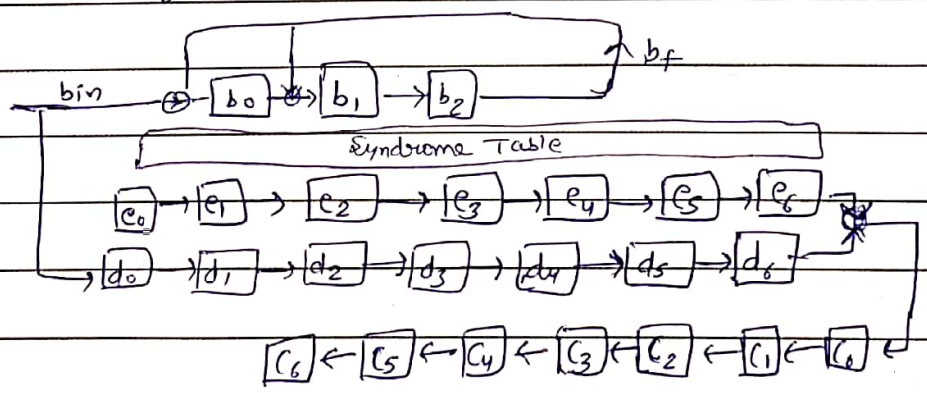
!

b_{in}	b_0	b_1	b_2	C_0	C_1	C_2	C_3	C_4	C_5	C_6	b_f	
–	0	0	0	0	0	0	0	0	0	0		
1	1	1	0	1	0	0	0	0	0	0	1	
1	1	0	1	1	1	0	0	0	0	0	1	
0	1	0	0	0	1	1	0	0	0	0	1	
1	1	0	0	1	0	1	1	0	0	0	1	
	↓ remainder bits			↓ Information bits								
–	0	1	0	0	1	0	1	1	0	0	–	A & B disconnect
–	0	0	1	0	0	1	0	1	1	0	–	Ccomect
–	0	0	0	1	0	0	1	0	1	1	–	

$\therefore$ C = 1101001

Cyclic Code decoding :=

(7,4) $g(x) = x^3 + x + 1$



It is a conventional approach where $R_{g(x)}\ v(x)$ is computed.

→ Not efficient

$s(x) = R_{g(x)}\ v(x)$

$s(x) = R_{g(x)}\ e(x)$

$s'(x) = R_{g(x)}\ e'(x)$

$= R_{g(x)}\ x\, s(x)$

$s''(x) = R_{g(x)}\ x(s'(x))$ $\qquad e''(x) = x^2 e(x)$

$= R_{g(x)}\ x^2 s(x)$

eg) $x^4 \quad g(x) = x^3 + x + 1$

$s(x) = x^2 + x \qquad s'(x) = x^2 + x + 1$