UNIT 1- Information Entropy Fundamentals

J.Premalatha
Professor/IT
Kongu Engineering College
Perundurai

Unit 1: Uncertainty, Information and Entropy – Source coding Theorem – Data Compaction – Discrete Memory less channels – Mutual Information - Channel Capacity – Channel Coding Theorem.

Uncertainty, Information and Entropy

Basic Terms:

Task: Activity or piece of work that have to do. [Ex]: Doing assignments, toss the coin , play the dice

Event: a planned or unplanned occurrence, situation, or incident . It can range from small personal events such as a birthday party to large-scale global events such as the Olympic Games or a natural disaster.

Information: An **event depends only on its probability of occurrence** and is **not dependent on its content**. The randomness of happening of an event and the probability of its prediction as a news is known as information. **The message associated with the least likelihood event contains the maximum information.**

[Examples]

Event : Planning for a trip to new Delhi on next month

A. The sun will rise [Probability =1]

B. The rain may come [Probability = ½].

C. There will be tornados [Probability= 1/100] .

A- which talks about an event **which has a probability of occurrence very close to 1, carries almost no information**.

B- carries more information than A. This has a finite probability that the rainfall may or may not occur. **This means 50% of occurrence or not**.

C- which has a **very low probability of occurrence, appears to carry a lot of information.**

Uncertainty : Consider the task[Play the dice] which have the discrete random variable X with the possible outcomes ,

i =1,2,3,…..n[For dice the outcomes are 1,2,3,4,5,6] . Suppose the dice is played for 100 times, the different probabilities for getting the outcomes.

The outcomes in ‘X’ have probabilities of $P(X = x_k) = P_k$

for k = 0,1,2,…n-1.

[Ex]: The outcome for getting 5 in dice for 100 times is 12/100. This can be written as P(X=5) = 12/100 =0.12

This set of probabilities satisfy the following condition

$$\sum_{i=0}^{n-1} P_k = 1$$

Why it is called as uncertainty?

➢Lack of certainty

➢A state of having limited knowledge **which is impossible to exactly describe the existing state, a future outcome or more than one possible outcome**

Relationship among Uncertainty, surprise , Information and Probability:

No Uncertainty –No information- No surprise – Probability is ‘1’

Uncertainty – Information – Surprise - Probability is ‘0.5’

More Uncertainty – More information – More surprise – Least Probability

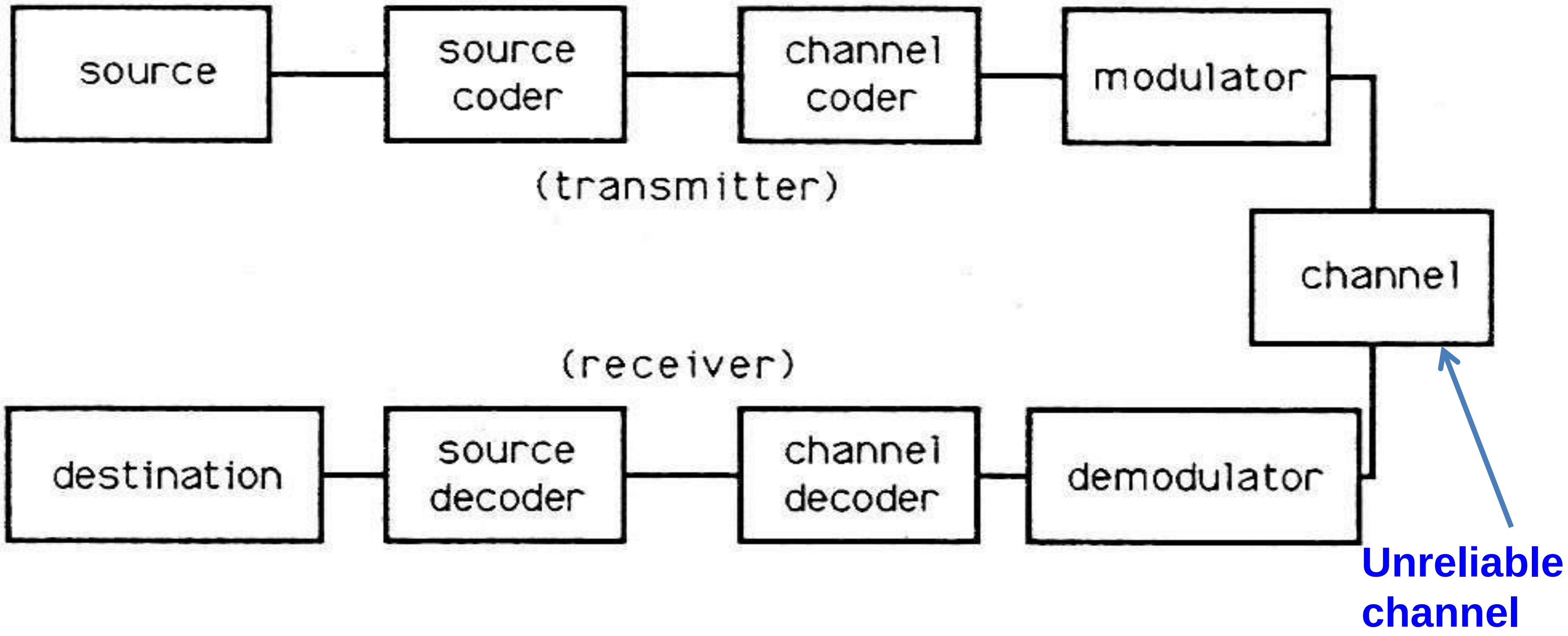
Information Theory

- It is a **branch of probability theory which may be applied to the study of the communication systems that deals with the mathematical modelling and analysis of a communication system** rather than with the physical sources and physical channels.
- Two important elements presented in this theory **are Binary Source (BS) and the Binary Symmetric Channel (BSC).**
- A **binary source is a device that generates one of the two possible symbols ‘0’ and ‘1’ at a given rate ‘r’,** measured in symbols per second
- These **symbols are called bits** (binary digits).

- The **BSC is a medium through which it is possible to transmit one symbol per time unit**. However this **channel is not reliable and is characterized by error probability ‘p’** that an output bit can be different from the corresponding input.

- **Information theory tries to analyze communication between a transmitter and a receiver through an unreliable channel** and in this approach, perform an analysis of information sources, especially **the amount of information produced by a given source, states the conditions for performing reliable transmission through an unreliable channel.**

Digital Communication System



Source coding : winzip, rar
Channel coding : CRC
Channel decoding : CRC
Source decoding : Unzip

Information (or) Self Information [I(X)]

Consider a discrete random variable X with the possible outcomes , i =1,2,3,…..n. The self Information of the event **$I(X_i)$**, is defined as

$I(X_i) \propto 1/P(X_i) \rightarrow I(X_i) = F(1/P(X_i))$ where F is the function.

$$I(x_i) = \log\left[\frac{1}{P(x_i)}\right] = -log\, P(x_i)$$

Log function is used

Properties of Information:

1. If an event has probability 1, then no information is get from the event: **$I(X) = 0$ if $P(X)=1$**
2. If there is more uncertainty in message, then more information: **$I(X_1) > I(X_2)$ for $P(X_1) < P(X_2)$**

3. Two independent events occur , then the information get from observing the events is the sum of the two information:
$I(X_1 * X_2) = I(X_1) + I(X_2)$.

4. The information is non – negative. Since **$I(X) \geq 0$ if $P(X)=0 < P(X) \leq 1$**

Unit of $I(X_i)$: The unit of $I(X_i)$ is the bit (binary unit) if b = 2, Hartley or decit if b = 10 and nat (natural unit) if b = e. It is standard to use b =2. So, the formula is

$$\boxed{I(X_i) = -\log_b P(X_i) = -\log_2 P(X_i)}$$

How to find $\log_2$ a ?

$$\log_2(a) = \frac{\ln(a)}{\ln(2)} = \frac{\log(a)}{\log(2)}$$

Problem 1: Calculate the amount of information, that a binary source which tosses a fair coin and outputs a 1 if a head appears and a 0 if a tail appears.

Solution : For this source P(1)=P(0)=0.5. The information content of each output from the source is

$$I(x_i) = -\log_2 P(x_i) = -\log_2(0.5) = 1\ bit$$

Use a ‘1’ represent H and a ‘0’ represent T.

Problem 2: Calculate amount of information for the binary source, that ‘0’ occur with probability 3/4 and ‘1’ occur with probability 1/4 .

Solution:

$$P_1 = \frac{3}{4} \qquad P_2 = \frac{1}{4}$$

$$I_1 = -\log_2 P(0.75) = 0.415\ bit$$

$$I_2 = -\log_2 P(1) = -log_2(0.25) = 2\ bits$$

Problem 3: Calculate the amount of information if $P_k = 1/8$

Solution :

$$I(x) = -log_2(0.125) = 3\ bits$$

Information Source

- An **information source is an object which produces an event**, the outcome of which is selected at random according to a probability distribution.
- The **set of source symbols is called** the **source alphabet** and the elements of the set are called **symbols** or **letters**
- **Information source can be classified** as **having memory** or being **memory-less.**
- A **source with memory** is one for which a **current symbol depends on the previous symbols.[Memory Channel**]
- A **source without memory** is one for which a **current symbol independent on the previous symbols.[Memory less Channel]**

Discrete memory-less source(DMS)

It can be characterized by the list of the symbol, the probability assignment of these symbols and the specification of the rate of generating these symbols by the source.

Information Content of a DMS

a)Information should be proportional to the uncertainty of an outcome

b)Information contained in independent outcomes should add up

Let us consider a DMS denoted by ‘x’and having alphabet $\{x_1, x_2,=, x_m\}$.

- The **information content of the symbol xi, denoted by I (*xi*)** is defined by

$$\boxed{I(x_i) = \log\left[\frac{1}{P(x_i)}\right] = -log\, P(x_i)}$$

where **p(*xi*) is the probability of occurrence of symbol *xi*.**

•For any two independent source messages x_i and x_j with probabilities P_i and P_j respectively and with joint probability $P(x_i, x_j) = P_i P_j$, the information of the messages is the addition of the information in each message. $I_{ij} = I_i + I_j$.

I(x_i) satisfies the following properties:

1. $I(x_i) = 0$ for $P(x_i) = 1$
2. $I(x_i) \geq 0$
3. $I(x_i) > I(x_j)$ if $P(x_i) < P(x_j)$
4. $I(x_i, x_j) = I(x_i) + I(x_j)$ if x_i and x_j are independent

Entropy (AverageInformation)

- The **quantity H(X) is called the entropy of source X**. It is a **measure of the average information content per source symbol**.
- The source entropy H(X) can be considered as the average amount of uncertainty within the source X that is resolved by the use of the alphabet.

$$\mathbf{H(X) = E\,[I(x_i)] = -\,\Sigma P(x_i)\, I(x_i) = -\,\Sigma P(x_i)\log_2 P(x_i)\ bits/symbol.}$$

The set of source symbols is called the **source alphabet** and the elements of the set are called **symbols** or **letters**

Source alphabet : { a, b, c,d,e,f -- x,y,z}

Let the symbols be {**abcba, bd, ghy, ghghgh, ytreytre ---}**

H(X) is a measure of the average information content per source symbol

[Ex]: Let the symbol is **abcba**

Average information = Total information / No.of. Symbols

No.of. Symbols (n) = 5 (abcba)

Total Information (I) = $n_aI_a + n_bI_b + n_cI_c$ where n_a is the number of times 'a' occur in the symbol and I_a is the self information. Similarly for n_b I_b and n_c I_c

Average information = Total information / No.of. Symbols

H(X) = $n_aI_a + n_bI_b + n_cI_c / n$ → This can be written as

$= (n_a/n)\, I_a + (n_b/n)\, I_b + (n_c/n)\, I_c$ where (n_a/n) is the probability of 'a' occur in the symbol (ie) P_a

$H(X) = P_a\, I_a + P_bI_b + P_c\, I_c$

So, the above formula can be written as,

$$H(x) = \sum_{i=1}^{n} P(x_i) I(x_i)$$

$$= -\sum_{i=1}^{n} P(x_i) \log_2 P(x_i) \text{ bits/symbol}$$

Note: i. H(x) is non negative

ii. H(x) depends only on the probabilities of the symbols of the source.

iii. H(x) represents the average information per source symbol.

Properties of Entropy:

1. Entropy H(x)=0, if and only if the probability P_k =0 or 1.
2. When P_k = 1/M for all the 'M' symbols, then the symbols are equally likely. For such source entropy is given as $H = \log_2 M$
3. Upper bound on entropy is given as $H_{max} = \log_2 M$ and Lower bound on H(x) will be zero.

Proof (or) Problem 4: Prove that entropy H(x)=0 when P_k = 0 and when P_k = 1.

Since $P_k = 1$

$$H = \sum_{k=1}^{M} P_k \log_2\left(\frac{1}{P_k}\right)$$

$$H = -\sum_{k=1}^{M} P_k \log_2 P_k$$

$$= -\sum_{k=1}^{M} 1 \times \log_2 1$$

$$= 0$$

Since $P_k = 0$

$$H = -\sum_{k=1}^{M} \log_2 P_k * P_k$$

$$= -\sum_{k=1}^{M} 0 \times \log_2 1$$

$$= 0$$

Hence entropy is zero.

Proof (or) Problem 5: find the entropy of a source having 8 equiprobable symbols.

Given: No. of symbols $= 8$ (M)

$$\therefore P_K = \frac{1}{8}$$

$$H(S) = \sum_{K=1}^{8} P_K \log_2 1/P_K = \sum_{K=1}^{8} \frac{1}{8} \log_2 \left(\frac{1}{\frac{1}{8}}\right)$$

$$= \sum_{K=1}^{8} \frac{1}{8} \log_2 8 = \sum_{k=1}^{8} \frac{1}{8} \log_2 2^3$$

$$= \sum_{K=1}^{8} \frac{3}{8} = \frac{3}{8} + \frac{3}{8} + \frac{3}{8} + \frac{3}{8} + \frac{3}{8} + \frac{3}{8} + \frac{3}{8} + \frac{3}{8}$$

$$= 8 \times \frac{3}{8} = 3 \text{ bits / symbols}$$

If the symbols are of equal probability, the entropy H(S) will be maximum and equal to

$$H(S) = \log_2 (m) = \log_2 (8) = 3 \text{ bits/symbols}$$

Problem 6 : A source transmit two independent message with probabilities of P , (1-P) respectively. Prove that the entropy is maximum when both the messages are equally likely. Plot the variation of entropy (H) as a function of probability 'P' of the message for 0, 0.2, 0.4, 0.5, 0.6 ,0.8 and 1.

Entropy $H(S) = \sum_{K=1}^{2} P_K \log_2 \frac{1}{P_K}$

Probability $P = \{P, P-1\}$

It can be viewed as, probability to occur $P_0 = P$

and probability to occur $P_1 = (1-P)$.

Now, $H(S) = \sum_{K=1}^{2} P_K \log_2 \frac{1}{P_K}$

$= P_0 \log_2 \frac{1}{P_0} + P_1 \log_2 \frac{1}{P_1}$

$$\boxed{H(S) = P \log_2 \frac{1}{P} + (1-P) \log_2 \frac{1}{(1-P)}}$$

To plot H(S): case 1: If $P=0$, Substitute in H(S)

$$H(S) = \underbrace{0\log_2\frac{1}{0}} + (1-0)\log_2\frac{1}{(1-0)}$$

$$= 0 + 1\underbrace{\log_2 1} \rightarrow \text{equal to } 0$$

$\therefore H(S) = 0$

Case 2: If $P=1$, Substitute in H(S)

$$H(S) = 1\underbrace{\log_2\frac{1}{1}}_{\text{equal to } 0} + \underbrace{0\log_2\frac{1}{0}}_{0} = 0$$

Case 3: If $P=5$, Substitute in H(S)

$$H(S) = 0.5\log_2\frac{1}{0.5} + 0.5\log_2\frac{1}{0.5}$$

$$= 0.5\log_2\frac{1}{\frac{1}{2}} + 0.5\log_2\frac{1}{\frac{1}{2}} = 0.5+0.5$$

$$= 1 \text{ bit/sym}$$

Case 4: If $p = 0.2$, then $(1-p) = 0.8$

$$H(S) = 0.2 \log_2 \frac{1}{0.2} + 0.8 \log_2 \frac{1}{0.8} = 0.72 \text{ bit/sym}$$

Case 5: If $p = 0.4$, then $(1-p) = 0.6$

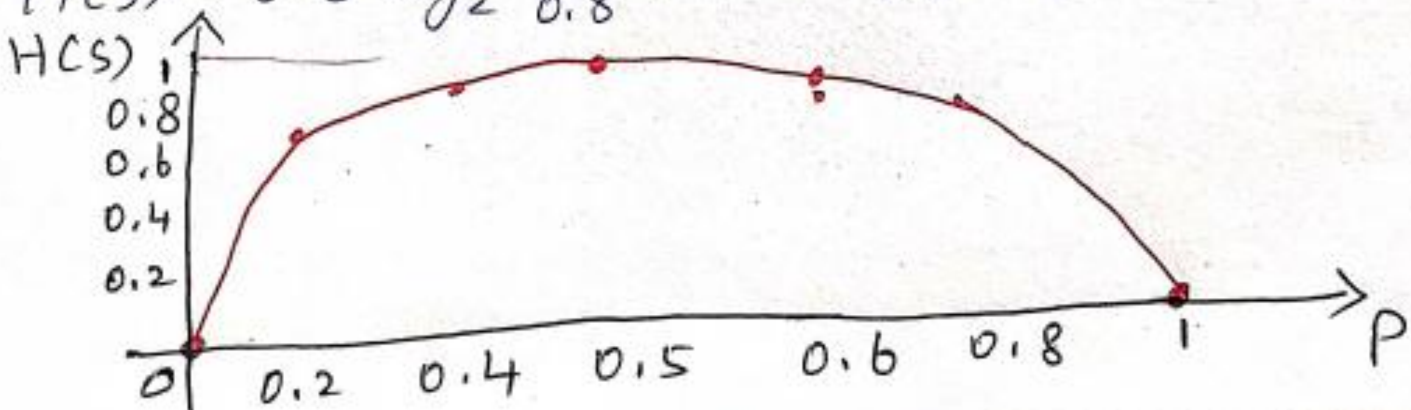
$$H(S) = 0.4 \log_2 \frac{1}{0.4} + 0.6 \log_2 \frac{1}{0.6} = 0.97 \text{ bit/sym}$$

Case 6: If $p = 0.6$, then $(1-p) = 0.4$

$$H(S) = 0.6 \log_2 \frac{1}{0.6} + 0.4 \log_2 \frac{1}{0.4} = 0.97 \text{ bit/sym}$$

Case 7: If $p = 0.8$, then $(1-p) = 0.2$

$$H(S) = 0.8 \log_2 \frac{1}{0.8} + 0.2 \log_2 \frac{1}{0.2} = 0.72 \text{ bit/sym}$$



Problem 7: calculate $H(x)$ for a discrete memoryless source with source alphabet $S = \{S_0, S_1, S_2\}$ with respective probabilities $P_0 = \frac{1}{4}$, $P_1 = \frac{1}{4}$, $P_2 = \frac{1}{2}$

$$H(x) = P_0 \log_2\left(\frac{1}{P_0}\right) + P_1 \log_2\left(\frac{1}{P_1}\right) + P_2 \log_2\left(\frac{1}{P_2}\right)$$

$$= \frac{1}{4}\log_2\left(\frac{1}{\frac{1}{4}}\right) + \frac{1}{4}\log_2\left(\frac{1}{\frac{1}{4}}\right) + \frac{1}{2}\log_2\left(\frac{1}{\frac{1}{2}}\right)$$

$$= \frac{1}{4}\log_2 4 + \frac{1}{4}\log_2 4 + \frac{1}{2}\log_2 2$$

$$= \frac{1}{4}\log_2 2^2 + \frac{1}{4}\log_2 2^2 + \frac{1}{2}\log_2 2$$

$$= \frac{2}{4} + \frac{2}{4} + \frac{1}{2} = \frac{1}{2} + \frac{1}{2} + \frac{1}{2} = \frac{3}{2} \text{ bits/symbol}$$

Extension of Discrete Memoryless Source:

Consider the discrete memoryless source having alphabet $X=\{x_1,x_2--x_m\}$. The source emits **"M" number of symbols** which occur in **groups of two or more**. These groups of symbols are produced by an extended source with a **source alphabet X^n alphabet with M^n distinct symbols.** Here "n" is the number of symbols in one group or block. Hence the probabilities of the group of symbol are obtained by product of individual symbol probabilities.

The entropy of source and H(x) are related as,

$$\mathbf{H(x^n) = n\,H(x)}$$

Problem: A source has 3 symbols with probabilities $\left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right\}$, find the second extension of the source. Also, prove that $H(X^n) = n\,H(X)$

<u>Given</u>: $S = \{x_1, x_2, x_3\}$ $P = \left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right\}$

S.no	Symbols	Probability of Symbols	Entropy ($P_i \log_2 \frac{1}{P_i}$)
1	$x_1\ x_1$	$\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$	$\frac{1}{4} \log_2 \frac{1}{\frac{1}{4}} = \frac{2}{4} = \frac{1}{2}$
2	$x_1\ x_2$	$\frac{1}{2} \times \frac{1}{4} = \frac{1}{8}$	$\frac{1}{8} \log_2 \frac{1}{\frac{1}{8}} = \frac{3}{8}$
3	$x_1\ x_3$	$\frac{1}{2} \times \frac{1}{4} = \frac{1}{8}$	"
4	$x_2\ x_1$	$\frac{1}{4} \times \frac{1}{2} = \frac{1}{8}$	"
5	$x_2\ x_2$	$\frac{1}{4} \times \frac{1}{4} = \frac{1}{16}$	$\frac{1}{16} \log_2 \frac{1}{\frac{1}{16}} = \frac{1}{4}$
6	$x_2\ x_3$	$\frac{1}{4} \times \frac{1}{4} = \frac{1}{16}$	"
7	$x_3\ x_1$	$\frac{1}{4} \times \frac{1}{2} = \frac{1}{8}$	$\frac{3}{8}$
8	$x_3\ x_2$	$\frac{1}{4} \times \frac{1}{4} = \frac{1}{16}$	$\frac{1}{4}$
9	$x_3\ x_3$	$\frac{1}{4} \times \frac{1}{4} = \frac{1}{16}$	$\frac{1}{4}$

$$H(X) = \sum_K P_K \log \frac{1}{P_K}$$

$$= \frac{1}{2}\log_2 \frac{1}{\frac{1}{2}} + \frac{1}{4}\log_2 \frac{1}{\frac{1}{4}} + \frac{1}{4}\log_2 \frac{1}{\frac{1}{4}}$$

$$= \frac{1}{2} + \frac{1}{2} + \frac{1}{2} = 3/2 = 1.5 \text{ bits/sym}$$

$$H(X^2) = 2\,H(X) = 2 \times 1.5 = 3 \text{ bits/sym}$$

<u>Proof</u>:

$$H(X^2) = \sum_K P_K \log_2 \frac{1}{P_K}$$

$$= \frac{1}{4}\log_2 \frac{1}{\frac{1}{4}} + \frac{1}{8}\log_2 \frac{1}{1/8} + \frac{1}{8}\log \frac{1}{1/8} + \frac{1}{8}\log \frac{1}{1/8} + \frac{1}{16}\log \frac{1}{\frac{1}{16}} +$$

$$\frac{1}{16}\log \frac{1}{16} + \frac{1}{8}\log \frac{1}{1/8} + \frac{1}{16}\log \frac{1}{1/16} + \frac{1}{16}\log \frac{1}{1/16}$$

$$= \frac{1}{2} + \frac{3}{8} + \frac{3}{8} + \frac{3}{8} + \frac{1}{4} + \frac{1}{4} + \frac{3}{8} + \frac{1}{4} + \frac{1}{4}$$

$$= \frac{1}{2} + \left(\cancel{4} * \frac{3}{\cancel{8}_2}\right) + \left(\cancel{4} \cancel{*} \frac{1}{\cancel{4}}\right) = \frac{1}{2} + \frac{3}{2} + 1 = \frac{1+3+2}{2} = \frac{6}{2}$$

$$= 3 \text{ bits/sym}$$

Source coding Theorem

An **important problem in communication** is the **efficient representation of data** generated by a discrete source.

Source coding Definition: The output of a discrete memory less source (DMS) is converted into a sequence of binary symbols i.e. binary code word, is called **Source Coding.**

The device that performs this conversion is called the **Source Encoder.**

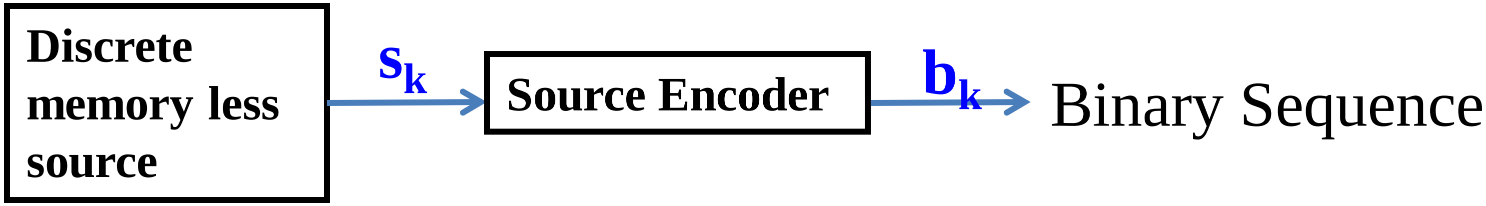
For the **source encoder to be efficient**, it **require knowledge of the statistics of the source**. If some source symbols are known to be more probable than others, assigning short code to frequent source symbols and long code to rare symbols.

[Ex]: **Morse code – Alphabets are encoded into dots and dashes.**

In English words, the letter **'E' occurs more than the letter 'Q'.** So, in Morse code the letter 'E' is encoded as single dot "." , the shortest code word and Q into "--.--" , the longest codeword.

Two functional requirements of efficient source encoder :

1. The code words produced by the encoder are in binary form.
2. The source code is uniquely decodable, so that the original sequence can be reconstructed perfectly from the encoded binary sequence



The figure depicts a discrete memory less source whose output s_k is converted by the source encoder into block of 0s and 1s, denoted by b_k .

Source Coding Theorem: Assume that the source has an alphabet with K different symbols and kth symbol s_k occurs with probability p_k , k=0,1,--K-1. Let the binary codeword assigned to symbol s_k by the encoder have length l_k, measured in bits. So, the average code-word length $\bar{L}$ of the source encoder as

$$\bar{L} = \sum_{k=0}^{K-1} p_k l_k$$

In physical terms, the parameter $\bar{L}$ represents the average number of bits per source symbol used in the source encoding process.

Let L_{min} denote the minimum possible value of $\bar{L}$.

Then the coding **efficiency** η of the source encoder is $\mathbf{L_{min} / \bar{L}}$. This can be written as $\eta = H(s)/\bar{L}$

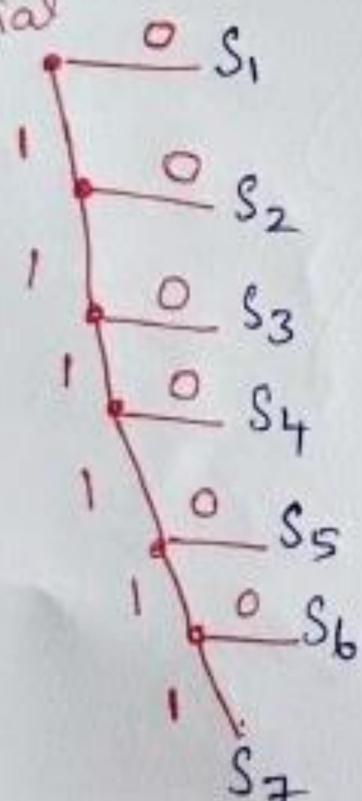
Problem: A discrete memoryless source emits symbols with probability {0.3, 0.25, 0.15, 0.1, 0.1, 0.05, 0.05}. Generate prefix codes for the symbols and evaluate coding efficiency and redundancy.

Solution! Arrange the symbols in ↓ing probability order.

$S_1 = 0.3$
$S_2 = 0.25$
$S_3 = 0.15$
$S_4 = 0.1$
$S_5 = 0.1$
$S_6 = 0.05$
$S_7 = 0.05$

Draw the binary code tree

Initial



Codewords

$S_1 = 0$
$S_2 = 10$
$S_3 = 110$
$S_4 = 1110$
$S_5 = 11110$
$S_6 = 111110$
$S_7 - 111111$

S_K	P_K	l_K length of the code
S_1	0.3	1
S_2	0.25	2
S_3	0.15	3
S_4	0.1	4
S_5	0.1	5
S_6	0.05	6
S_7	0.05	6

Average code word length

$$\bar{L} = \sum_K P_K l_K$$

$$\bar{L} = (0.3 * 1) + (0.25 * 2) + (0.15 * 3) + (0.1 * 4) + (0.1 * 5) + (0.05 * 6) + (0.05 * 6)$$

$$= 2.75 \text{ bits/symbol}$$

Efficiency $\eta = \dfrac{H(S)}{\bar{L}}$

$$H(S) = \sum_K P_K \log 1/P_K = 0.3 \log \frac{1}{0.3} + 0.25 \log \frac{1}{0.25} + 0.15 \log \frac{1}{0.15} + 0.1 \log \frac{1}{0.1} + 0.1 \log \frac{1}{0.1} + 0.05 \log \frac{1}{0.05} + 0.05 \log \frac{1}{0.05}$$

$$= 2.528 \text{ bits/symbol}$$

$$\eta = \frac{H(S)}{\bar{L}} = \frac{2.528}{2.75} = 0.919 \qquad \eta = 91.9\,\%$$

Redundancy $\gamma = (1-\eta) = 1 - 0.919) = 0.080$

$= 8\,\%$

Problem: The source symbols have probability {0.3, 0.25, 0.15, 0.1, 0.1, 0.05, 0.05}. Generate codes for the symbols using Shannon fano coding scheme and find efficiency and redundancy.

Steps:

1. Sort the symbols in ↓ing order of probability, the most probable ones to the top and the least probable ones to the right.
2. Split the list into two parts, with the total probability of both parts being as close to each other as possible.
3. Assign the value 0 to the top part and 1 to the bottom part.
4. Repeat steps 2 and 3 until all the symbols are split into individual subgroups.

S_1	0.3	0 0
S_2	0.25	0 1
S_3	0.15	1 0 0
S_4	0.1	1 0 1
S_5	0.1	1 1 0
S_6	0.05	1 1 1 0
S_7	0.05	1 1 1 1

Symbols	P_K	Codeword	Length l_K
S_1	0.3	00	2
S_2	0.25	01	2
S_3	0.15	100	3
S_4	0.1	101	3
S_5	0.1	110	3
S_6	0.05	1110	4
S_7	0.05	1111	4

Average codeword Length $\bar{L} = \sum_K l_K P_K$

$$\bar{L} = (0.3*2)+(0.25*2)+(0.15*3)+(2*0.1*3)+(2*0.05*4)$$

$=$

$$H(S) = \sum_K p_K \log P_K = 0.3 \log \frac{1}{0.3} + 0.25 \log \frac{1}{0.25} + 0.15 \log \frac{1}{0.15} + 0.1 \log \frac{1}{0.1} + 0.1 \log \frac{1}{0.1} + 0.05 \log \frac{1}{0.05} + 0.05 \log \frac{1}{0.05}$$

$H(S) =$

Efficiency $\eta = \frac{H(S)}{\bar{L}} =$

Redundancy $r = 1 - \eta =$

Problem: Encode the source symbols with Probability of $\{0.25, 0.15, 0.15, 0.1, 0.1, 0.1, 0.05, 0.05, 0.05\}$

Problem: Construct Shannon codes for the given set of symbols

Symbol	A	B	C	D	E
Probability	0.22	0.28	0.15	0.30	0.05

Answer: A = 10 ; B = 01 ; C = ~~1~~10 ; D = 00 ; E = 111

Data compaction

Any type of source contains redundant information. The transmission of redundant information consumes more resources such as power, bandwidth, storage capacity, time. In order to compensate this wastage, unnecessary information must be eliminated.

For efficient transmission, **the redundant bits should be removed through lossless data compression (or) data compaction**.

Original data can be reconstructed with no loss of information.

Data compaction is achieved by assigning short descriptions to the most frequent outcomes, of the source output and longer descriptions to the less frequent ones.

Types of Source coding schemes for Data Compaction:

1. Prefix Coding
2. Huffman Coding
3. Lempel-Ziv Coding

Prefix Coding

Classification of Code

1. Fixed – Length Codes
2. Variable – Length Codes
3. Distinct Codes
4. Prefix – Free Codes
5. Uniquely Decodable Codes
6. Instantaneous Codes
7. Optimal Codes

x_i	Code 1	Code 2	Code 3	Code 4	Code 5	Code 6
x_1	00	00	0	0	0	1
x_2	01	01	1	10	01	01
x_3	00	10	00	110	011	001
x_4	11	11	11	111	0111	0001

1. **Fixed – Length Codes:** A fixed – length code is one whose code word length is fixed. Code 1 and Code 2 of above table are fixed – length code words with length 2.
2. **Variable – Length Codes:** A variable – length code is one whose code word length is not fixed. All codes of above table except Code 1 and Code 2 are variable – length codes.
3. **Distinct Codes:** A code is distinct if each code word is distinguishable from each other. All codes of above table except Code 1 are distinct codes.
4. **Prefix Codes[Prefix Free Codes]:** A code in which no code word can be formed by adding code symbols to another code word is called a prefix- free code. In a prefix – free code, no code word is prefix of another. Codes 2, 4 and 6 of above table are prefix – free codes.

5. **Uniquely Decodable Codes:** A distinct code is uniquely decodable if the original source sequence can be reconstructed perfectly from the encoded binary sequence. A sufficient condition to ensure that a code is uniquely decodable is that no code word is a prefix of another. Thus the prefix –free codes 2, 4 and 6 are uniquely decodable codes. Prefix –free condition is not a necessary condition for uniquely decidability. Code 5 does not satisfy the prefix –free condition and yet it is a uniquely decodable code since the bit 0 indicates the beginning of each code word of the code.

6. **Instantaneous Codes:** A uniquely decodable code is called an instantaneous code if the end of any code word is recognizable without examining subsequent code symbols. The instantaneous codes have the property that no code word is a prefix of another code word. Prefix –free codes are sometimes known as instantaneous codes.

7. **Optimal Codes:** A code is said to be optimal if it is instantaneous and has the minimum average L for a given source with a given probability assignment for the source symbols

To decode a sequence of code words generated from a prefix source code, the source decoder simply starts at the beginning of the sequence and decodes one code word at a time. Basically, it sets up is equivalent to a decision tree which is graphical portrayal of the code words in the particular source code.

The tree has initial state and terminal states corresponding to source symbols. The **decoder always starts at the initial state**. The first received bit moves the decoder to the terminal state if it is ‘0’ , or else to a second decision point if it is ‘1’. Again, the second bit moves the decoder one step further down the tree, either to terminal state or further down and so on. Once each terminal state emits its symbol , the decoder is reset to its initial state. Each bit in the received encoded sequence is examined only once.

Problem: Identify the Prefix code and Non Prefix code . Also draw the decision tree decoder and find the symbols for the encoded sequence 1011111000 for the Prefix code.

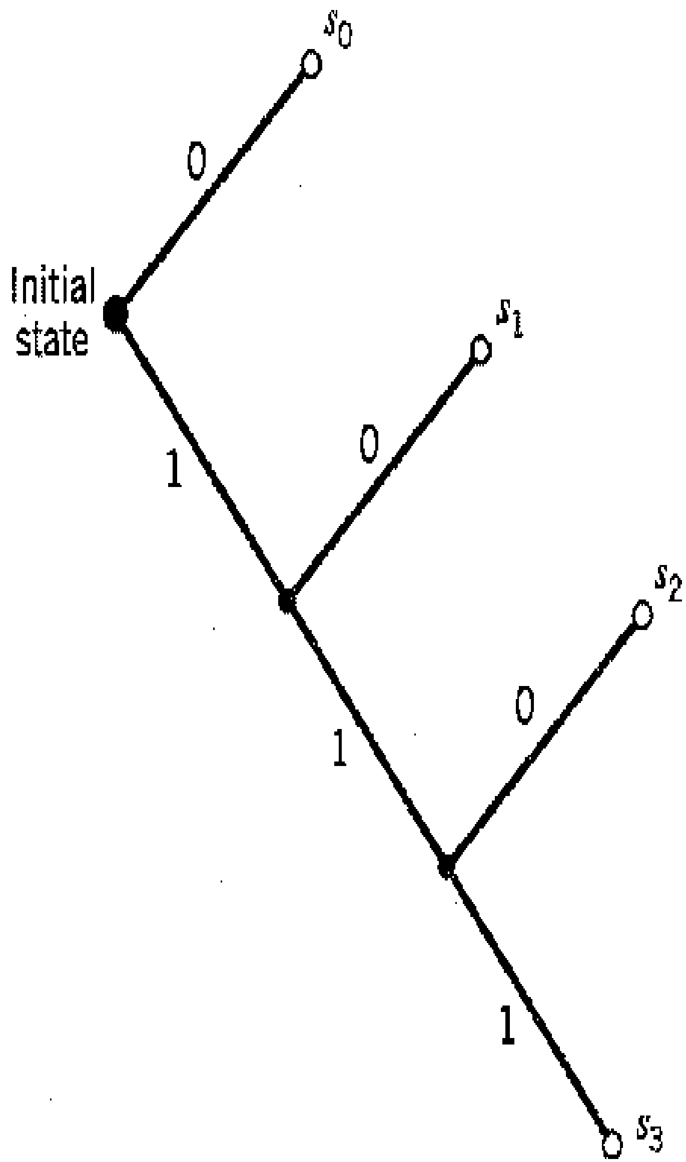
Source Symbol	*Probability of Occurrence*	*Code I*	*Code II*	*Code III*
s_0	0.5	0	0	0
s_1	0.25	1	10	01
s_2	0.125	00	110	011
s_3	0.125	11	111	0111

Code I : Non Prefix Code [‘0’ and ‘1’ are prefixed in symbols S_2 and S_3]

Code II : Prefix Code

Code III : Non Prefix Code [‘0’ is prefixed in symbols S_2 , S_3 and S_4]

Decision Tree decoder:



Decode the Sequence 1011111000

S_1 , S_3 , S_2 , S_0 , S_0

Kraft-Mcmillan Inequality

- If a prefix code for a discrete memory less source with source alphabet $\{s_0,s_1,\ldots.s_{k-1}\}$ and source statistics $\{p_0,p_1,\ldots.p_{k-1}\}$ and the code word for the symbol s_k has length l_k , $k= 0,1,2,\ldots,K-1$, then the code word lengths of the code always satisfy a certain inequality known as Kraft-McMillan Inequality , as shown as

$$\sum_{k=0}^{K-1} 2^{-l_k} \leq 1$$

- Kraft inequality assures us of the existence of an instantaneously decodable code with code word lengths that satisfy the inequality as **H(s)< $\bar{L}$<H(s)+1 [When Probability is given]**

- But it does not show us how to obtain those code words, nor does it say any code satisfies the inequality is automatically uniquely decodable.

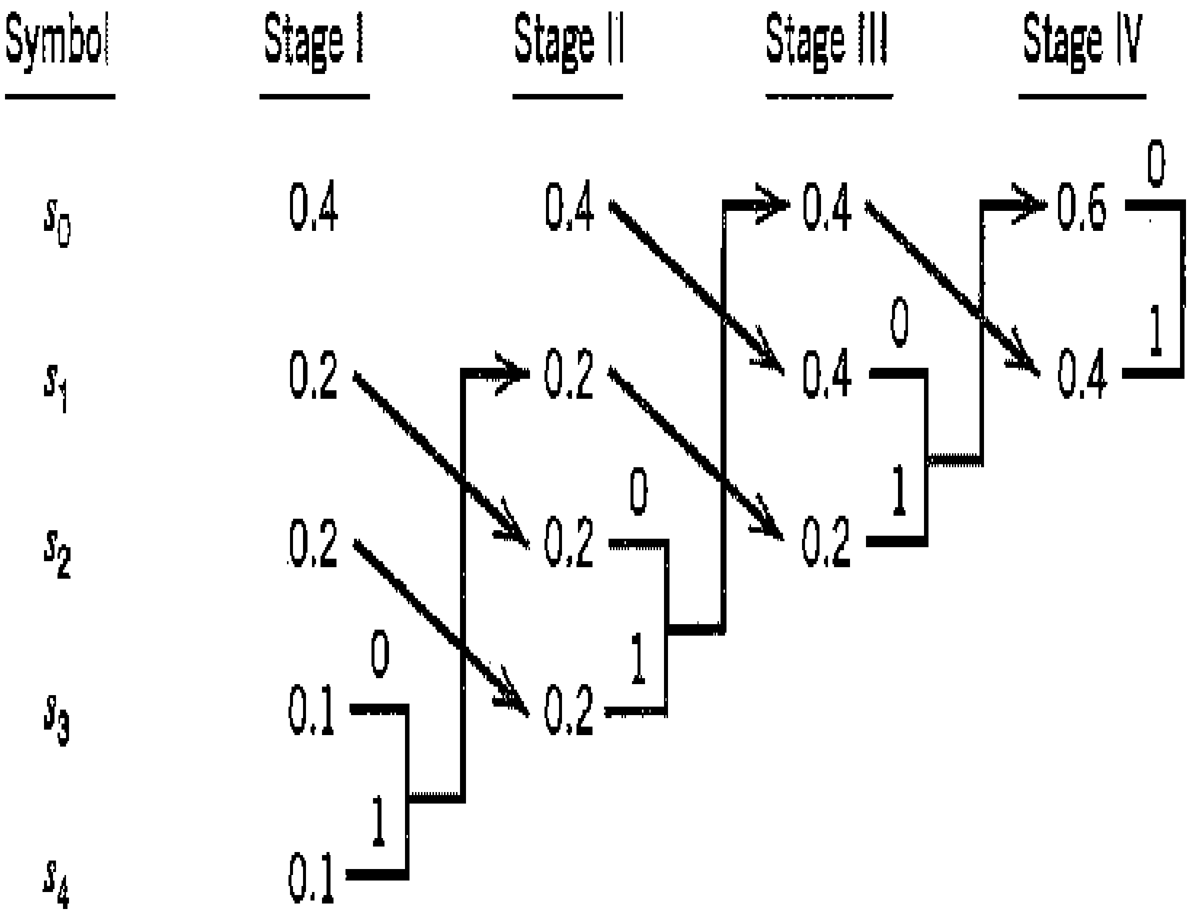
Huffman Coding : It results an optimal code. It is the code that has the highest efficiency.

The Huffman coding procedure is as follows:

1. List the source symbols in order of decreasing probability.
2. Combine the probabilities of the two symbols having the lowest probabilities and reorder the resultant probabilities, this step is called reduction 1. The same procedure is repeated until there are two ordered probabilities remaining.
3. Start encoding with the last reduction, which consists of exactly two ordered probabilities. Assign 0 as the first digit in the code word for all the source symbols associated with the first probability; assign 1 to the second probability.
4. Now go back and assign 0 and 1 to the second digit for the two probabilities that were combined in the previous reduction step, retaining all the source symbols associated with the first probability; assign 1 to the second probability.

5. Keep regressing this way until the first column is reached.

6. The code word is obtained tracing back from right to left.



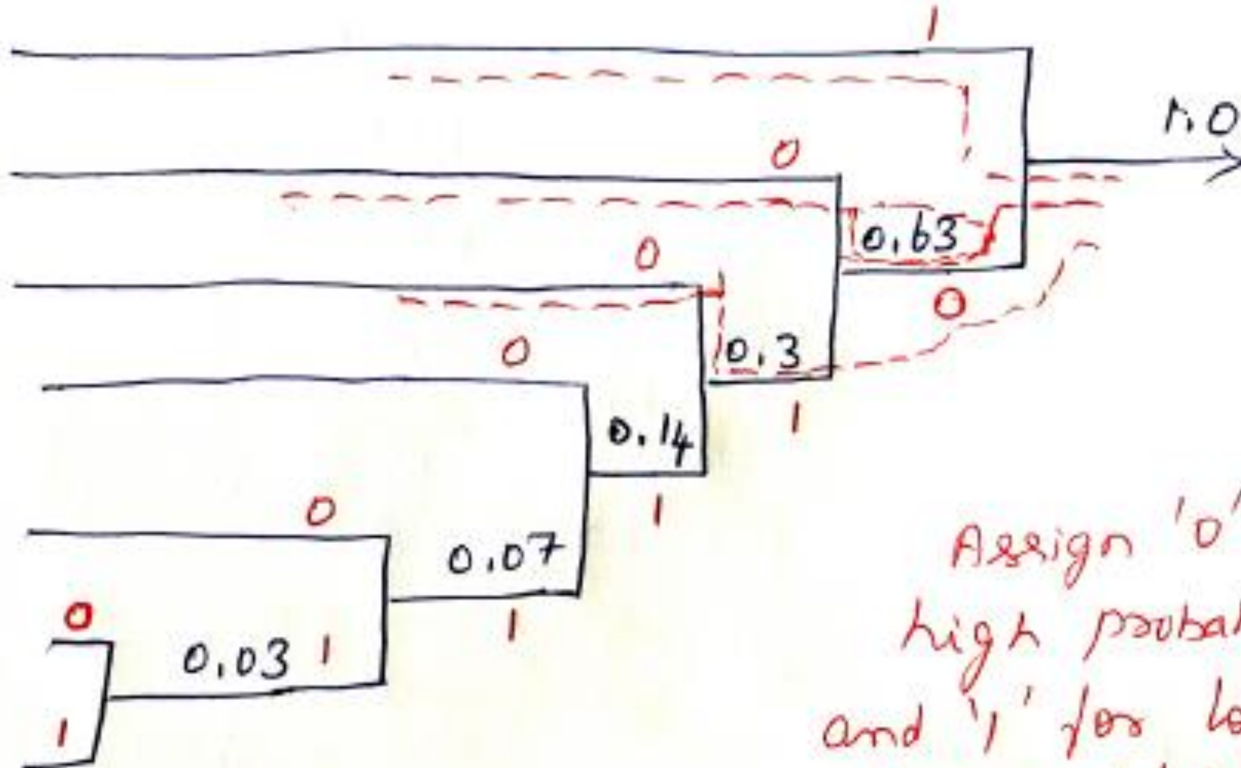
(a)

Symbol	Probability	Code word
s_0	0.4	00
s_1	0.2	10
s_2	0.2	11
s_3	0.1	010
s_4	0.1	011

(b)

Problem: consider a DMS with seven possible symbols x_i where $i = 1, 2, 3 \cdots 7$. The corresponding probabilities are $p(x_1) = 0.37$, $p(x_2) = 0.33$, $p(x_3) = 0.16$, $p(x_4) = 0.07$, $p(x_5) = 0.02$, $p(x_6) = 0.04$ and $p(x_7) = 0.01$. Find huffman code for each symbol.

Symbol	probability
x_1	0.37
x_2	0.33
x_3	0.16
x_4	0.07
x_5	0.04
x_6	0.02
x_7	0.01



Assign '0' for high probability and '1' for low probability

Assign codes from right to left,

Symbol	Codewords
x_1	1
x_2	00
x_3	010
x_4	0110
x_5	01110
x_6	011110
x_7	011111

<u>Problem</u>: How many bits required for encoding the message 'mississippi'?

<u>Problem</u>: A networking company uses a huffman code for compression before transmitting over the network. The message contains characters with their frequency are $a=5$, $b=9$, $c=12$, $d=13$, $e=16$, $f=45$. How many bits will be saved from 8bit ASCII code is used?

Problem: calculate the code efficiency using Huffman coding. Also check Kraft Inequality is satisfied or not.

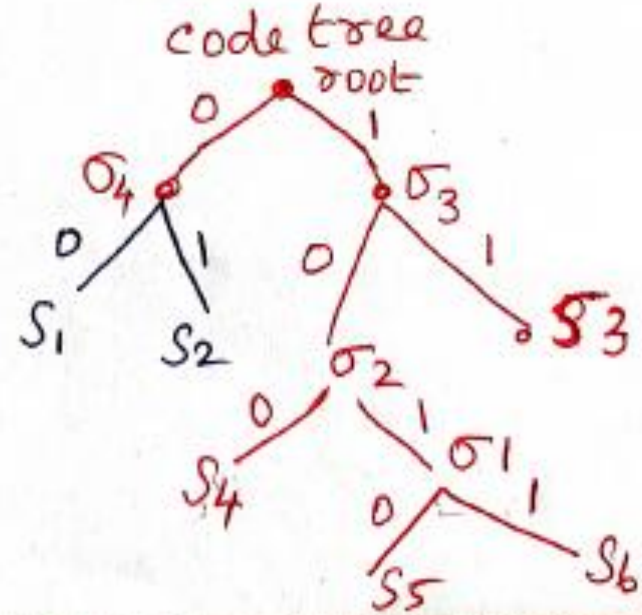
Symbols S_K	Probability P_K (C − I)	C − II	C − III	C − IV	C − V
S_1	0.3	S_1 0.3	S_1 0.3	σ_3 0.45	σ_4 0.55
S_2	0.25	S_2 0.25	S_2 0.25	S_1 0.3	σ_3 0.45
S_3	0.2	S_3 0.2	σ_2 0.25	S_2 0.25	
S_4	0.15	S_4 0.15	S_3 0.2		
S_5	0.05	σ_1 0.1			
S_6	0.05				

$\sigma_1 = 0.1$

$\sigma_2 = 0.25$

$\sigma_3 = 0.45$

$\sigma_4 = 0.55$



$S_1 = 00$

$S_2 = 01$

$S_3 = 11$

$S_4 = 100$

$S_5 = 1010$

$S_6 = 1011$

Efficiency $\eta = \frac{H(S)}{\bar{L}}$ where

$H(S) = \sum_K P_K \log \frac{1}{P_K}$ and $\bar{L} = \sum_K P_K L_K$

S_K	P_K	Codeword	length l_K	$P_K l_K$
S_1	0.3	00	2	0.6
S_2	0.25	01	2	0.5
S_3	0.2	11	2	0.4
S_4	0.15	100	3	0.45
S_5	0.05	1010	4	0.2
S_6	0.05	1011	4	0.2

$\sum P_K L_K = \bar{L}$
2.35 bits/sym

$$H(S) = \sum_{k} P_k \log \frac{1}{P_k}$$

$$= 0.3 \log \frac{1}{0.3} + 0.25 \log \frac{1}{0.25} + 0.2 \log \frac{1}{0.2} + 0.15 \log \frac{1}{0.15}$$

$$+ 0.05 \log \frac{1}{0.05} + 0.05 \log \frac{1}{0.05}$$

$$= 2.328 \text{ bits/symbol}$$

$$\eta = \frac{H(S)}{\bar{L}} = \frac{2.328}{2.35} = 0.9906$$

$$\eta = 99.06\%$$

Kraft Inequality, $H(S) < \bar{L} < H(S)+1$

$2.328 < 2.35 < 3.35$

so it satisfies the Kraft inequality.

<u>Problem</u>: Generate codewords and find code efficiency for the given symbols using Huffmann coding. Also, check for Kraft Inequality.

Symbols	x_1	x_2	x_3	x_4	x_5	x_6
Probability	0.30	0.25	0.20	0.12	0.08	0.05

<u>Answer</u>: $H(S) = 2.36$ bits/symbol

$\bar{L} = 2.38$ bits/symbol

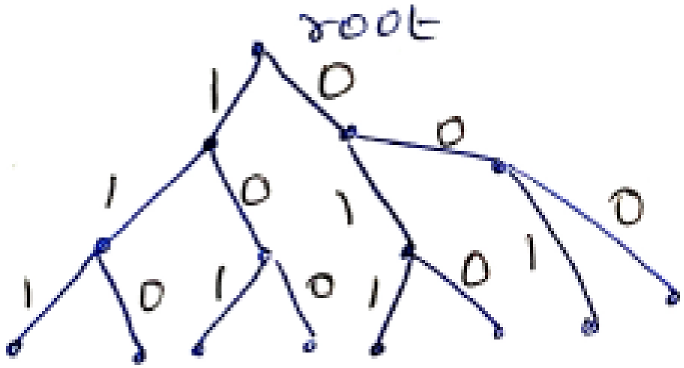
$$\eta = \frac{H(S)}{\bar{L}} = 0.99 \text{ (or) } 99\%$$

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

$$2.36 < 2.38 < 3.38$$

So, it satisfy kraft inequality

Problem: Generate your own prefix codes using binary tree for the symbols A, B, C, D. Assume that the right side tree is '0' and left side tree is '1'. Also, check your codes satisfy Kraft's inequality.



Assign codewords for the symbols from the tree,

A = 0

so, no other codewords for the symbols start with '0'.

∴ B = 10, C = 110, D = 111

Kraft's inequality: $\sum_{k=0}^{K-1} 2^{-l_k} \leq 1$ where l_k is the length & k are the no. of symbols.

i.e. the length of 'A' symbol + length of 'B' symbol + length of 'C' symbol + length of 'D' symbol.

$2^{-1} + 2^{-2} + 2^{-3} + 2^{-3} = 0.5 + 0.25 + 0.125 + 0.125 = 1$ which is ≤ 1

so, It satisfy Kraft's inequality.

Problem! Compare the no. of. bits require for encoding the message 'A BAD CAB', using the given codes. Identify the codes that produce lossy compression. Also, suggest which code is better for lossless compression.

a. Fixed length code

letter	codeword
A	000
B	001
C	010
D	011
E	100
F	101
G	110
H	111

b. variable length code 1

letter	codeword
A	0
B	1
C	00
D	01
E	10
F	11
G	000
H	001

c. variable length code 2

letter	codeword
A	00
B	010
C	011
D	100
E	101
F	110
G	1110
H	1111

Message for encoding 'A BAD CAB'.

a. fixed length code – 000 001 000 011 010 000 001 — 21 bits.

b. variable length code1 – 0 1001 0001 – 9 bits

c. variable length code2 – 00 01000100 01100010 – 18 bits

Decoding the message

a. fixed length code – $\underbrace{000}_{A}\ \underbrace{001}_{B}\ \underbrace{000}_{A}\ \underbrace{011}_{D}\ \underbrace{010}_{C}\ \underbrace{000}_{A}\ \underbrace{001}_{B}$

∴ It produce lossless compression.

b. variable length code1 – $\underbrace{0}_{A}\ \underbrace{1}_{B}\ \underbrace{00}_{C}\ \underbrace{1}_{B}\ \underbrace{0}_{A}\ \underbrace{0}_{A}\ \underbrace{01}_{D}$ (or)

$\underbrace{01}_{D}\ \underbrace{00}_{C}\ \underbrace{1}_{E}\ \underbrace{00}_{C}\ \underbrace{01}_{B}$ (or) $\underbrace{01}_{D}\ \underbrace{00}_{C}\ \underbrace{10}_{E}\ \underbrace{001}_{H}$ ∴ It produce lossy compression

c. variable length code 2 – $\underbrace{00}_{A}\ \underbrace{010}_{B}\ \underbrace{00}_{A}\ \underbrace{100}_{D}\ \underbrace{011}_{C}\ \underbrace{00}_{A}\ \underbrace{010}_{B}$

∴ It produce lossless compression.

The code better for lossless compression is

Variable length code 2. Because it produces lossless compression and take only 18 bits compare with fixed length code of 21 bits.

Problem: check the following codes satisfy Kraft's inequality. Also, identify prefix codes.

I.

symbols	probability	codewords
a	$\frac{1}{2}$	0
b	$\frac{1}{4}$	10
c	$1/8$	110
d	$1/8$	111

II

symbols	probability	code words
A	$1/2$	0
B	$1/4$	1
C	$1/4$	01
D	$1/8$	10

III

Symbols	probability	codewords
M	$1/2$	101
N	$1/4$	00
O	$1/2$	0001
P	$1/8$	1

Lempel Ziv Encoding: It is dictionary based encoding

Basic Idea: create a dictionary (a table) of strings used during communication.

* If both sender and receiver have a copy of the dictionary, then previously-encountered strings can be substituted by their index in the dictionary.

* This method have two phases.

  i) Building an index dictionary

  ii) Compressing a string of symbols

Algorithm: (1) Extract the smallest substring that cannot be found in the remaining uncompressed string.

(2) Store that substring in the dictionary as a new entry and assign it an index value.

③ Substring is replaced with the index found in the dictionary.

④ Insert the index and the last character of the substring in the compressed string.

<u>Example</u>: The binary sequence is 000101110010 100101.
It is assumed that the binary symbols 0 and 1 are already stored.

Subsequenced stored : 0, 1

Data to be parsed : 0 0 | 0 1 | 0 1 1 | 1 0 | 0 1 0 | 1 0 0 | 1 0 1

Numerical positions	1	2	3	4	5	6	7	8	9
Subsequences	0	1	00	01	011	10	010	100	101
Numerical Representations			11	12	42	21	41	61	62
Binary encoded blocks			0010	0011	1001	0100	1000	1100	1101

Problem: Apply Lempel ziv encoding for the sequence

BAABA BBB AABBBBAA

↓ String

1	
B	

← $\boxed{B}$ ← BAABA BBB AABBBB AA

1	2	
B	A	

← $\boxed{A}$ ← A ABABBBAABBBBAA

1	2	3
B	A	AB

← $\boxed{AB}$ ← AB ABBBAABBBBAA

1	2	3	4
B	A	AB	ABB

← $\boxed{ABB}$ ← ABBBAABBBBAA

1	2	3	4	5
B	A	AB	ABB	BA

← $\boxed{BA}$ ← BAABBBBAA

1	2	3	4	5	6
B	A	AB	ABB	BA	ABBB

← ABBB ← ABBBBAA

1	2	3	4	5	6	7
B	A	AB	ABB	BA	ABBB	BAA

← BAA ← BAA

Now the compressed value is,

B, A, 2<u>B</u>, 3<u>B</u>, 1<u>A</u>, 4<u>B</u>, 5<u>A</u>

The last symbol of each subsequence in the code book is an innovation symbol.

Comparison b/w Huffmann code & Lempel ziv code

Huffmann code	Lempel Ziv code
1. variable length code	fixed length code
2. 43% of compaction	55% of compaction
3. Not suitable for synchronous transmission	suitable for synchronous transmission

Lempel-Ziv Decompression

1
B

← $\boxed{B}$ ← B, A, 2B, 3B, 1A, 4B, 5A

1	2
B	A

← $\boxed{A}$ ← A, 2B, 3B, 1A, 4B, 5A

1	2	3
B	A	AB

← $\boxed{2B}$ ← 2B, 3B, 1A, 4B, 5A

1	2	3	4
B	A	AB	ABB

← $\boxed{3B}$ ← 3B, 1A, 4B, 5A

1	2	3	4	5
B	A	AB	ABB	BA

← $\boxed{1A}$ ← 1A, 4B, 5A

1	2	3	4	5	6
B	A	AB	ABB	BA	ABBB

← $\boxed{4B}$ ← 4B, 5A

1	2	3	4	5	6	7
B	A	AB	ABB	BA	ABBB	BAA

← $\boxed{5A}$ ← 5A

Lempel ziv compaction : There are 16 characters in the sequence. If ASCII encoding (8 bit) is used it takes $16 \times 8 = 128$ bits.

Encoded sequence	B	A	2B	3B	1A	4B	5A
Binary encoded blocks	0	1	0100	0110	0011	1000	1011

Lempel ziv reduce into 22 bits.

$$\text{compaction efficiency} = \frac{\text{After compression}}{\text{Before compression}}$$

$$= \frac{22}{128} = 0.1718$$

$$= 17.2\%$$

Marginal, Conditional and Joint Entropies

Using the input probabilities P (x_j), output probabilities P (y_k), transition probabilities P (y_k/x_j) and joint probabilities P (x_j, y_k), various entropy functions for a channel with j inputs and k outputs are defined

Marginal Entropy

$$H(x) = \sum_{j=0}^{J-1} p(x_j) \log_2 \left[\frac{1}{p(x_j)} \right]$$

$$H(y) = \sum_{k=0}^{K-1} p(y_k) \log_2 \left[\frac{1}{p(y_k)} \right]$$

Conditional Entropy

$$H(y/x) = \sum_{k=0}^{k-1} \sum_{j=0}^{J-1} P(x_j, y_k) \log_2 \left[\frac{1}{P(y_k / x_j)} \right]$$

$$H(x/y) = \sum_{k=0}^{k-1} \sum_{j=0}^{J-1} P(x_j, y_k) \log_2 \left[\frac{1}{P(x_j / y_k)} \right]$$

Joint Entropy

$$H(x, y) = \sum_{j=0}^{J-1} \sum_{k=0}^{k-1} P(x_j, y_k) \log_2 \left(\frac{1}{P(x_j, y_k)} \right)$$

- H (X) is the average entropy(uncertainty) of the channel input and H (Y) is the average entropy(uncertainty) of the channel output.
- The conditional entropy H (Y/X) is the average entropy(uncertainty) of the channel output given that X was transmitted.
- The conditional entropy H (X/Y) is a measure of the average uncertainty remaining about the channel input after the channel output has been observed.
- The joint entropy of two discrete random variables X and Y is merely the entropy of their pairing: (X, Y), this implies that if X and Y are independent, then their joint entropy is the sum of their individual entropies.

Properties of Entropy:

a. $H(X, Y) = H(X/Y) + H(Y)$ b. $H(X, Y) = H(Y/X) + H(X)$

c. $H(X/Y) = H(X, Y) - H(Y)$ d. $H(Y/X) = H(X, Y) - H(X)$

e. $H(Y) \geq H(Y/X)$ f. $H(X) \geq H(X/Y)$

To find joint probability P(x,y) is , P(x,y) = Matrix of x contain only diagonal elements, multiply with P(y/x) . This is equal to multiply P(x1),P(x2) to P(y/x) on respective rows. That is P(x1) multiply with first row of P(y/x) and P(x2) multiply with second row of P(y/x) and so on

Problem: The joint probability matrix for a channel is given by,

$$p(X,Y) = \begin{pmatrix} 0.08 & 0.05 & 0.02 & 0.05 \\ 0.15 & 0.07 & 0.01 & 0.12 \\ 0.10 & 0.06 & 0.05 & 0.04 \\ 0.01 & 0.12 & 0.01 & 0.06 \end{pmatrix}$$

find marginal entropy, conditional entropy & Joint entropy. Also verify the properties of entropy.

Solution: Marginal entropy $H(x), H(y)$

conditional entropy $H(X/y), H(y/x)$

Joint entropy $H(x,y)$

$$P(x,y) = \begin{array}{c|cccc|l} & y_1 & y_2 & y_3 & y_4 & \\ \hline x_1 & 0.08 & 0.05 & 0.02 & 0.05 & \rightarrow P(x_1) \\ x_2 & 0.15 & 0.07 & 0.01 & 0.12 & \rightarrow P(x_2) \\ x_3 & 0.10 & 0.06 & 0.05 & 0.04 & \rightarrow P(x_3) \\ x_4 & 0.01 & 0.12 & 0.01 & 0.06 & \rightarrow P(x_4) \end{array}$$

Row - sum of $P(x, y)$ – Input probabilities $p(x)$

$$P(x_1) = 0.08 + 0.05 + 0.02 + 0.05 = 0.2$$

$$P(x_2) = 0.15 + 0.07 + 0.01 + 0.12 = 0.35$$

$$P(x_3) = 0.10 + 0.06 + 0.05 + 0.04 = 0.25$$

$$P(x_4) = 0.01 + 0.12 + 0.01 + 0.06 = 0.2$$

Marginal Entropy $H(x) = \sum_{j=1}^{J} P(x_j) \log_2 \frac{1}{P(x_j)}$

$$= 0.2 \log_2 \left(\frac{1}{0.2}\right) + 0.35 \log \left(\frac{1}{0.35}\right) + 0.25 \log \left(\frac{1}{0.25}\right) + 0.2 \log \left(\frac{1}{0.2}\right)$$

$H(x) = 1.958$ bits/symbol

Column - sum of $P(x, y)$ – Output probabilities $p(y)$

$$P(y_1) = 0.08 + 0.15 + 0.1 + 0.01 = 0.34$$

$$P(y_2) = 0.05 + 0.07 + 0.06 + 0.12 = 0.3$$

$$P(y_3) = 0.02 + 0.01 + 0.05 + 0.01 = 0.09$$

$$P(y_4) = 0.05 + 0.12 + 0.04 + 0.06 = 0.27$$

Marginal Entropy $H(y) = \sum_{k=1}^{K} P(y_k) \log_2 \frac{1}{P(y_k)}$

$= 0.34 \log_2\left(\frac{1}{0.34}\right) + 0.3 \log_2\left(\frac{1}{0.3}\right) + 0.09 \log_2\left(\frac{1}{0.09}\right) + 0.27 \log_2\left(\frac{1}{0.27}\right)$

$H(y) = 1.872$ bits/symbol

To find conditional probability $P(y/x)$ & $P(x/y)$

We know that,

$$P(x, y) = P(y/x)\, P(x)$$

$$P(y/x) = \frac{P(x, y)}{P(x)}$$

III^{ly}

$$P(x/y) = \frac{P(x, y)}{P(y)}$$

To get $P(y/x)$ matrix, divide each row of $P(x,y)$ by $P(x)$.

$$P(y/x) = \begin{bmatrix} \frac{0.08}{0.2} & \frac{0.05}{0.2} & \frac{0.02}{0.2} & \frac{0.05}{0.2} \\ \frac{0.15}{0.35} & \frac{0.07}{0.35} & \frac{0.01}{0.35} & \frac{0.12}{0.35} \\ \frac{0.10}{0.25} & \frac{0.06}{0.25} & \frac{0.05}{0.25} & \frac{0.04}{0.25} \\ \frac{0.01}{0.2} & \frac{0.12}{0.2} & \frac{0.01}{0.2} & \frac{0.06}{0.2} \end{bmatrix} = \begin{bmatrix} 0.4 & 0.25 & 0.1 & 0.25 \\ 0.428 & 0.2 & 0.028 & 0.3 \\ 0.4 & 0.24 & 0.2 & 0.16 \\ 0.05 & 0.6 & 0.05 & 0.3 \end{bmatrix}$$

Conditional Entropy $H(y/x)$

$$H(y/x) = \sum_j \sum_k P(x_j y_k) \log \frac{1}{P(y_k/x_j)}$$

$$= \left(0.08 \log \frac{1}{0.4}\right) + \left(0.05 \log \frac{1}{0.25}\right) + \left(0.02 \log \frac{1}{0.1}\right) + \left(0.05 \log \frac{1}{0.25}\right)$$
$$+ \left(0.15 \log \frac{1}{0.428}\right) + \left(0.07 \log \frac{1}{0.2}\right) + \left(0.01 \log \frac{1}{0.028}\right) + \left(0.12 \log \frac{1}{0.3}\right)$$
$$+ \left(0.10 \log \frac{1}{0.4}\right) + \left(0.06 \log \frac{1}{0.24}\right) + \left(0.05 \log \frac{1}{0.2}\right) + \left(0.04 \log \frac{1}{0.16}\right)$$

$$+\left(0.01\log\frac{1}{0.05}\right)+\left(0.12\log\frac{1}{0.6}\right)+\left(0.01\log\frac{1}{0.05}\right)+\left(0.06\log\frac{1}{0.3}\right)$$

$= 1.712$ bits/symbol

To get $P(x/y)$ matrix, divide each column of $P(x,y)$ by $P(y)$.

$$P(x/y) = \begin{bmatrix} \frac{0.08}{0.34} & \frac{0.05}{0.3} & \frac{0.02}{0.09} & \frac{0.05}{0.27} \\ \frac{0.15}{0.34} & \frac{0.07}{0.3} & \frac{0.01}{0.09} & \frac{0.12}{0.27} \\ \frac{0.10}{0.34} & \frac{0.06}{0.3} & \frac{0.05}{0.09} & \frac{0.04}{0.27} \\ \frac{0.01}{0.34} & \frac{0.12}{0.3} & \frac{0.01}{0.09} & \frac{0.06}{0.27} \end{bmatrix} = \begin{bmatrix} 0.235 & 0.166 & 0.222 & \cancel{0.185} \; 0.185 \\ 0.441 & 0.233 & 0.111 & 0.444 \\ 0.294 & 0.2 & 0.555 & 0.148 \\ 0.029 & 0.4 & 0.11 & 0.22 \end{bmatrix}$$

Conditional Entropy $H(x/y)$

$$H(x/y) = \sum_j \sum_k P(x_j, y_k)\log\frac{1}{P(x_j/y_k)}$$

$$= \left(0.08 \log \frac{1}{0.235}\right) + \left(0.05 \log \frac{1}{0.166}\right) + \left(0.02 \log \frac{1}{0.222}\right) + \left(0.05 \log \frac{1}{0.185}\right)$$

$$+ \left(0.15 \log \frac{1}{0.441}\right) + \left(0.07 \log \frac{1}{0.233}\right) + \left(0.01 \log \frac{1}{0.111}\right) + \left(0.12 \log \frac{1}{0.444}\right)$$

$$+ \left(0.10 \log \frac{1}{0.294}\right) + \left(0.06 \log \frac{1}{0.2}\right) + \left(0.05 \log \frac{1}{0.555}\right) + \left(0.04 \log \frac{1}{0.148}\right)$$

$$+ \left(0.01 \log \frac{1}{0.029}\right) + \left(0.12 \log \frac{1}{0.4}\right) + \left(0.01 \log \frac{1}{0.11}\right) + \left(0.06 \log \frac{1}{0.22}\right)$$

$$= 1.798 \text{ bits/symbol}$$

Joint Entropy $H(X, Y)$

$$H(X, Y) = \sum_j \sum_k P(x_j, y_k) \log \frac{1}{P(x_j, y_k)}$$

$$= \left(0.08 \log \frac{1}{0.08}\right) + \left(0.05 \log \frac{1}{0.05}\right) + \left(0.02 \log \frac{1}{0.02}\right) + \left(0.05 \log \frac{1}{0.05}\right)$$

$$+ \left(0.15 \log \frac{1}{0.15}\right) + \left(0.07 \log \frac{1}{0.07}\right) + \left(0.01 \log \frac{1}{0.01}\right) + \left(0.12 \log \frac{1}{0.12}\right)$$

$$+ \left(0.10 \log \frac{1}{0.10}\right) + \left(0.06 \log \frac{1}{0.06}\right) + \left(0.05 \log \frac{1}{0.05}\right) + \left(0.04 \log \frac{1}{0.04}\right)$$

$+\left(0.01 \log \frac{1}{0.01}\right)+\left(0.12 \log \frac{1}{0.12}\right)+\left(0.01 \log \frac{1}{0.01}\right)+\left(0.06 \log \frac{1}{0.06}\right)$

$= 3.67$ bits / symbol

Marginal Entropy $H(x) = 1.958$ b/s $\quad H(y) = 1.872$ b/s

Conditional Entropy $H(x/y) = 1.798$ b/s $\quad H(y/x) = 1.712$ b/s

Joint Entropy $H(x,y) = 3.67$ b/s

Properties of Entropy

I $\quad H(X,Y) = H(x) + H(y/x)$

$3.67 = 1.958 + 1.712$ — verified

II $\quad H(x,y) = H(y) + H(x/y)$

$3.67 = 1.872 + 1.798$ — verified

III $\quad H(y) \geq H(y/x)$

$1.87 \geq 1.712$ — verified

IV $\quad H(X) \geq H(x/y)$

$1.958 \geq 1.798$ — verified

V Mutual Information

$$I(x,y) = H(x) - H(x/y)$$

(or) $I(x,y) = H(y) - H(y/x)$

$$I(x,y) = \overset{H(x)}{1.958} - \overset{H(x/y)}{1.798} = 0.16 \text{ bits/symbol}$$

(or) $I(x,y) = H(y) - H(y/x) = 1.87 - 1.712 = 0.16$ bits/symbol

Problem: The channel matrix is given as,

$$P(y/x) = \begin{pmatrix} 0.2 & 0.3 & 0.5 \\ 0.3 & 0.4 & 0.3 \\ 0.4 & 0.2 & 0.4 \end{pmatrix}$$ and the probabilities

of $P(x_1) = 0.25$, $P(x_2) = 0.35$ and $P(x_3) = 0.4$

Find $H(x)$, $H(y)$, $H(x/y)$, $H(y/x)$, $I(x,y)$

Soln: To find $P(x,y)$

$$P(x,y) = P(y/x) \cdot P(x)$$

To get P(x,y), multiply I^{st} row with $P(x_1)$, II^{nd} row with $P(x_2)$ and III^{rd} row with $P(x_3)$,

Mutual Information

- Mutual information measures the amount of information that can be obtained about one random variable by observing another.
- It is important in communication where it can be used to maximize the amount of information shared between sent and received signals.
- The mutual information denoted by I (X, Y) of a channel is defined by:

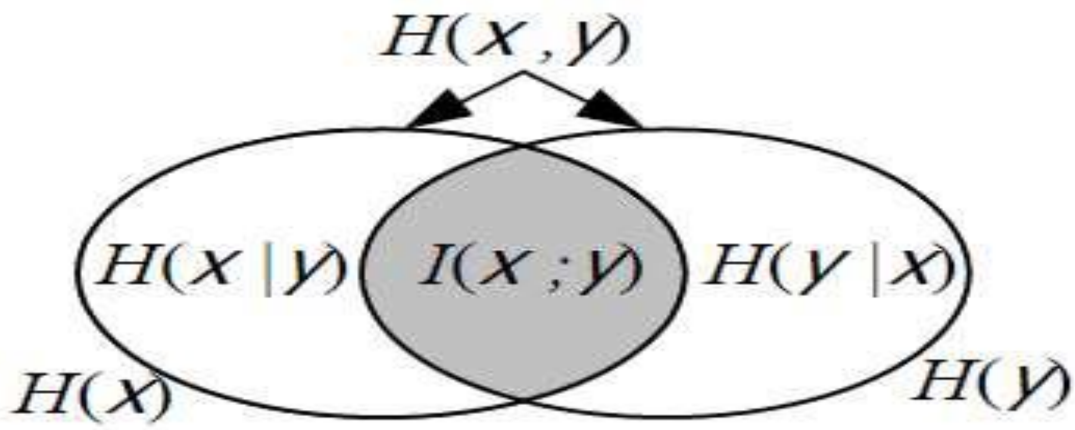
$$I(X;Y) = H(X) - H(X/Y) \text{ bits/symbol}$$

- Where H (X) represents the uncertainty about the channel input before the channel output is observed and H (X/Y) represents the uncertainty about the channel input after the channel output is observed, the mutual information I (X; Y) represents the uncertainty about the channel input that is resolved by observing the channel output.

Properties of Mutual Information I (X;Y)

1. **$I(X;Y) = I(Y;X)$**
2. **$I(X;Y) \geq 0$**
3. **$I(X;Y) = H(Y) - H(Y/X)$**
4. **$I(X;Y) = H(X) - H(X/Y)$**
5. **$I(X;Y) = H(X) + H(Y) - H(X,Y)$**

- The Entropy corresponding to mutual information [i.e. I (X, Y)] indicates a measure of the information transmitted through a channel. Hence, it is called **'Transferred information'.**



Discrete Memoryless channels

➢It is a statistical model with input(X) and output(Y) that is a noisy version of X. Both X and Y is random variables.

➢Every unit of time the channel accepts an input symbol X and in response, it emits an output symbol Y. The channel is discrete because both have finite sizes.

➢It is said to be "memory less" when the current output symbol depends only on the current input symbol and not any of the previous ones.

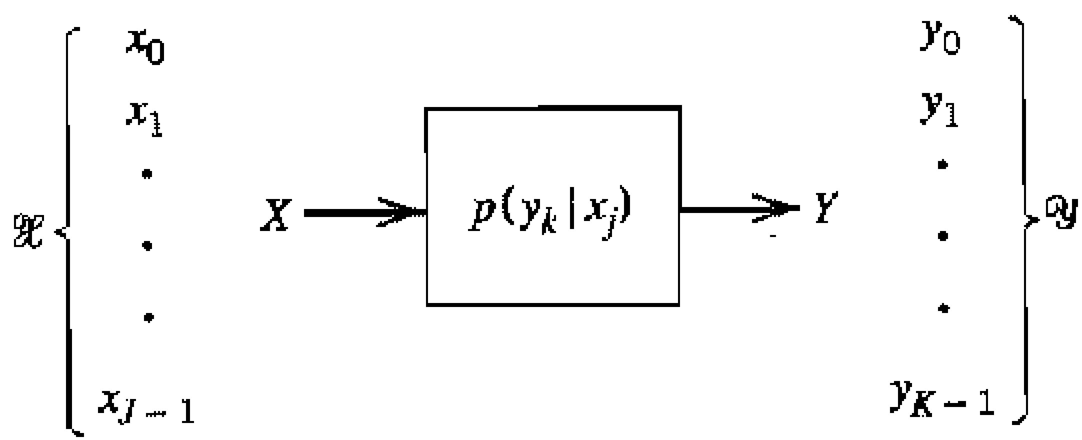


FIGURE 9.7 Discrete memoryless channel.

The channel is described in terms of an input alphabet $X = \{x_0, x_1, x_2, \dots, x_{j-1}\}$,

Output alphabet $Y = \{y_0, y_1, y_2, \dots, y_{k-1}\}$, and

a set of transition probabilities $P(y_k / x_j) = P(Y = y_k / X = x_j)$ for all j and k. This is known as Conditional probability of y_k is received, given that x_j was transmitted

The transition probabilities of the channel can be represented by a matrix

$$\mathbf{P} = \begin{bmatrix} p(y_0|x_0) & p(y_1|x_0) & \cdots & p(y_{K-1}|x_0) \\ p(y_0|x_1) & p(y_1|x_1) & \cdots & p(y_{K-1}|x_1) \\ \vdots & \vdots & & \vdots \\ p(y_0|x_{J-1}) & p(y_1|x_{J-1}) & \cdots & p(y_{K-1}|x_{J-1}) \end{bmatrix}$$

The **J by K matrix P is called the transition matrix or channel matrix. Each row of the channel matrix corresponds to a fixed channel input. Each column of the channel matrix corresponds to fixed channel output.** The **sum of the elements along any row of the matrix is always equal to one**. This means that

$$\sum_{k=0}^{K-1} p(y_k|x_j) = 1 \qquad \text{for all } j$$

The event that the channel input $X=x_j$ occurs with probability

$P(x_j) = P(X=x_j)$ for $j=0,1,\ldots,j-1$

The **joint probability distribution** of the random variables X and Y is given by **$P(x_j, y_k)$** $= P(X=x_j, Y=y_k)$

$$= P(Y=y_k|X=x_j)P(X=x_j)$$

$$= p(y_k|x_j)p(x_j)$$

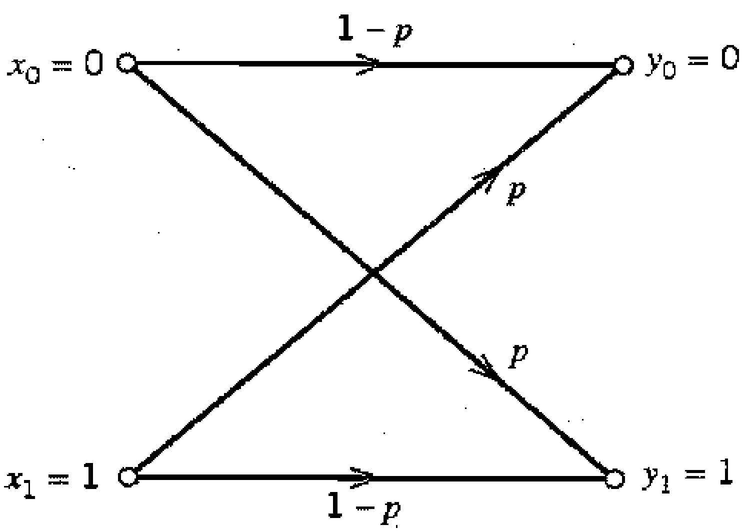
The **marginal Probability distribution** of the random variable Y is obtained by averaging of $p(x_j,y_k)$ on x_j, as shown by

$$\begin{aligned} p(y_k) &= P(Y=y_k) \\ &= \sum_{j=0}^{J-1} P(Y=y_k|X=x_j)P(X=x_j) \\ &= \sum_{j=0}^{J-1} p(y_k|x_j)p(x_j) \qquad \text{for } k=0,1,\ldots,K-1 \end{aligned}$$

Binary Symmetric Channel(BSC)

It is a special case of the discrete memoryless channel with J=K=2. This channel has two inputs(x_0=0 and x_1=1) and two outputs (y_0 = 0 and y_1 =1). This channel is symmetric because the probability of receiving a 1 if a 0 is sent is the same as the probability of receiving a 0 if a 1 is sent. The probability of error is denoted by p. The transition probability diagram of a binary symmetric channel is

Channel Diagram



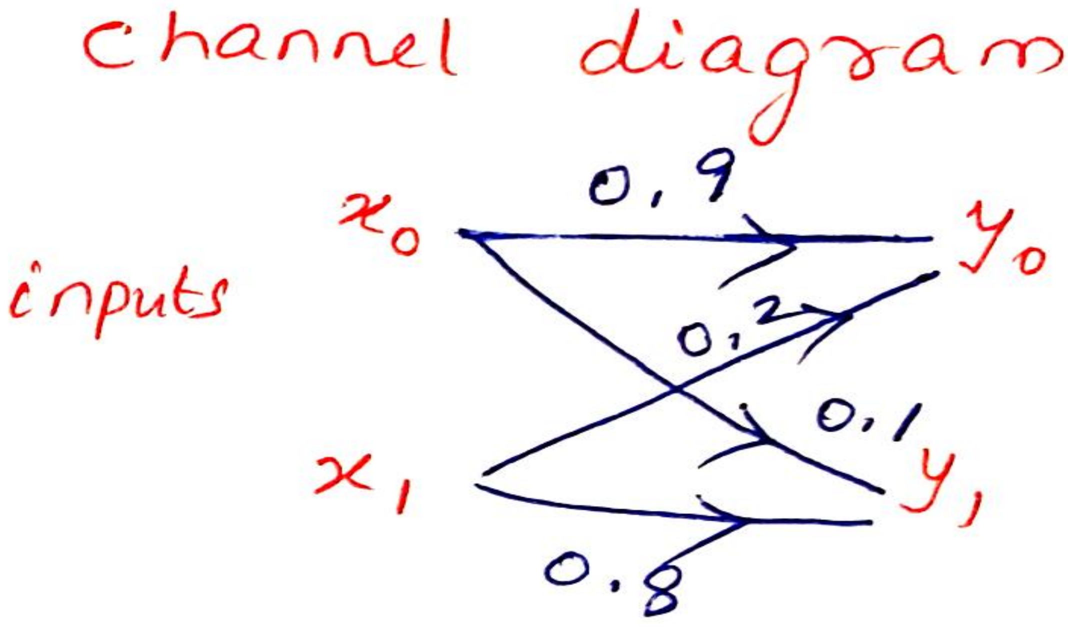
The transition matrix P(y/x) is

$$P(y/x) = \text{inputs} \begin{array}{c c} & \begin{array}{cc} \multicolumn{2}{c}{\text{Outputs}} \\ y_0 & y_1 \end{array} \\ \begin{array}{c} x_0 \\ x_1 \end{array} & \begin{pmatrix} 1-P & P \\ P & 1-P \end{pmatrix} \end{array}$$

Problem 1 : The channel transition matrix is given by $\begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$

i) Draw the channel diagram.
ii) Determine the probabilities associated with output assuming equi probable input.

Solution:
Channel Diagram:



ii) The given data is

$$P(y/x)=\begin{bmatrix} P(y_0/x_0) & P(y_1/x_0) \\ P(y_0/x_1) & P(y_1/x_1) \end{bmatrix}=\begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$$

Inputs are equi probable $P(x_0)=0.5$ and $P(x_1)=0.5$

To get P(x,y), multiply I[st] row with $P(x_1)$ and II[nd] row with $P(x_2)$

$$P(x,y)=\begin{bmatrix} P(y_0/x_0)P(x_0) & P(y_1/x_0)P(x_0) \\ P(y_0/x_1)P(x_1) & P(y_1/x_1)P(x_1) \end{bmatrix}=\begin{bmatrix} 0.9*0.5 & 0.1*0.5 \\ 0.2*0.5 & 0.8*0.5 \end{bmatrix}$$

$$P(x,y)=\begin{bmatrix} P(y_0/x_0)P(x_0) & P(y_1/x_0)P(x_0) \\ P(y_0/x_1)P(x_1) & P(y_1/x_1)P(x_1) \end{bmatrix}=\begin{bmatrix} 0.45 & 0.05 \\ 0.1 & 0.4 \end{bmatrix}$$

Outputs are the addition of each columns $P(y_0)=0.55$ and $P(y_1)=0.45$

Problem 2: A channel has the full channel matrix

$$P[Y/X] = \begin{bmatrix} 1-P & P & 0 \\ 0 & P & 1-P \end{bmatrix}$$

i) Draw the channel diagram

ii) If the source has equally likely input, compute the probabilities associated with the channel outputs for P = 0.2. [Hint: Substitute P=0.2 in Transition matrix

Problem 3: A discrete source transmits message X_1, X_2, X_3 with the probabilities 0.3,0.4 and 0.3 . The channel matrix is,

$$P[Y/X] = \begin{bmatrix} 0.8 & 0.2 & 0 \\ 0 & 1 & 0 \\ 0 & 0.3 & 0.7 \end{bmatrix}$$

i) Draw the channel diagram

ii) Calculate all entropies

Channel capacity

Consider a discrete memoryless channel with input symbols x, output symbols y and the transition probabilities $P(y/x)$, then the mutual information is,

$$\boxed{I(x,y) = H(y) - H(y/x)}$$

. The mutual information depends on the type of channel is used. But, the inputs $p(x)$ is independent of the channel. So, the channel capacity is defined as the maximum mutual information $I(x,y)$ on any channel. The channel capacity is commonly denoted as C. So, the capacity $C = \max[I(x,y)]$ (or) $I(x,y)_{max}$

Types of Channels:

1. Symmetric channel [In this Binary symmetric Channel is very popular)
2. Lossless channel
3. Deterministic channel
4. Noiseless channel

Capacity of Symmetric channel

This channel has same number of input & output symbols. Suppose if there are '3' input & output symbols and error probabilities are P_1, P_2, P_3. The transition matrix is permutation of I^{st} row elements. (ie) The transition matrix of symmetric channel is,

$$P(Y/X) = \begin{pmatrix} P_1 & P_2 & P_3 \\ P_2 & P_3 & P_1 \\ P_3 & P_1 & P_2 \end{pmatrix}$$

The input probabilities are $P(x_1), P(x_2), P(x_3)$ then the channel capacity $C = I(x,y)_{max}$

where $I(x,y)_{max} = \left[H(Y) - H(Y/X)\right]_{max}$

$$P(x,y) = P(y/x) * P(x)$$

$$= \begin{pmatrix} P_1 & P_2 & P_3 \\ P_2 & P_3 & P_1 \\ P_3 & P_1 & P_2 \end{pmatrix} \begin{matrix} \xleftarrow{*} P(x_1) \\ \xleftarrow{*} P(x_2) \\ \xleftarrow{*} P(x_3) \end{matrix} = \begin{pmatrix} P_1 P(x_1) & P_2 P(x_1) & P_3 P(x_1) \\ P_2 P(x_2) & P_3 P(x_2) & P_1 P(x_2) \\ P_3 P(x_3) & P_1 P(x_3) & P_2 P(x_3) \end{pmatrix}$$

Then the entropy $H(y/x) = \sum_j \sum_k P(x_j, y_k) \log \frac{1}{P(y_k/x_j)}$

$$= P_1 P(x_1) \log \frac{1}{P_1} + P_2 P(x_1) \log \frac{1}{P_2} + P_3 P(x_1) \log \frac{1}{P_3} +$$

$$P_2 P(x_2) \log \frac{1}{P_2} + P_3 P(x_2) \log \frac{1}{P_3} + P_1 P(x_2) \log \frac{1}{P_1} +$$

$$P_3 P(x_3) \log \frac{1}{P_3} + P_1 P(x_3) \log \frac{1}{P_1} + P_2 P(x_3) \log \frac{1}{P_2}$$

$$= P_1 \log \frac{1}{P_1} \big(\underbrace{P(x_1) + P(x_2) + P(x_3)}_{1}\big) + P_2 \log \frac{1}{P_2} \big(\underbrace{P(x_1) + P(x_2) + P(x_3)}_{1}\big)$$

$$+ P_3 \log \frac{1}{P_3} \big(\underbrace{P(x_1) + P(x_2) + P(x_3)}_{}\big)$$

The addition of inputs are equal to 1.

So, $\boxed{H(y/x) = P_1 \log \frac{1}{P_1} + P_2 \log \frac{1}{P_2} + P_3 \log \frac{1}{P_3}}$

$$C = I(x, y)_{max}$$

$$C = \left[H(y) - H(y/x) \right]_{max}$$

$$= H(y)_{max} - H(y/x)_{max}$$

If there are 'M' symbols, then maximum entropy $H(x)$ (or) $H(y)$ is equal to $\log_2 M$

$$\therefore C = \log_2 M - \left[P_1 \log \frac{1}{P_1} + P_2 \log \frac{1}{P_2} + P_3 \log \frac{1}{P_3} \right]$$

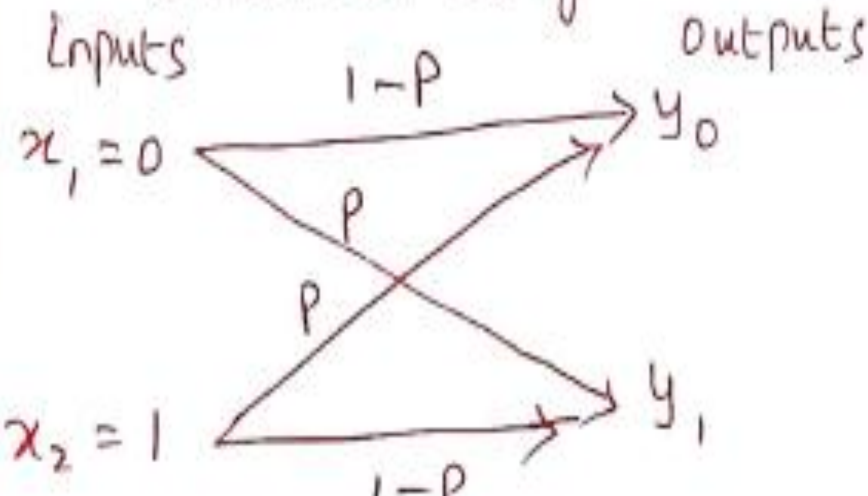
$\therefore$ The capacity of a symmetric channel is,

$$\boxed{C = \log_2 M - \left[\text{Entropy of first row elements of } P(y/x) \text{ matrix} \right]}$$

Binary Symmetric Channel [BSC]

This channel has two inputs ($x_1 = 0$, $x_2 = 1$) and two outputs ($y_1 = 0$, $y_2 = 1$). This channel is symmetric because the probability of receiving a '1' if a '0' is sent is the same as the probability of receiving a '0' if a '1' is sent.

Channel diagram



Channel matrix

$$\begin{array}{c c} & \begin{array}{cc} y_1 = 0 & y_2 = 1 \end{array} \\ \begin{array}{c} x_1 = 0 \\ x_2 = 1 \end{array} & \begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix} \end{array}$$

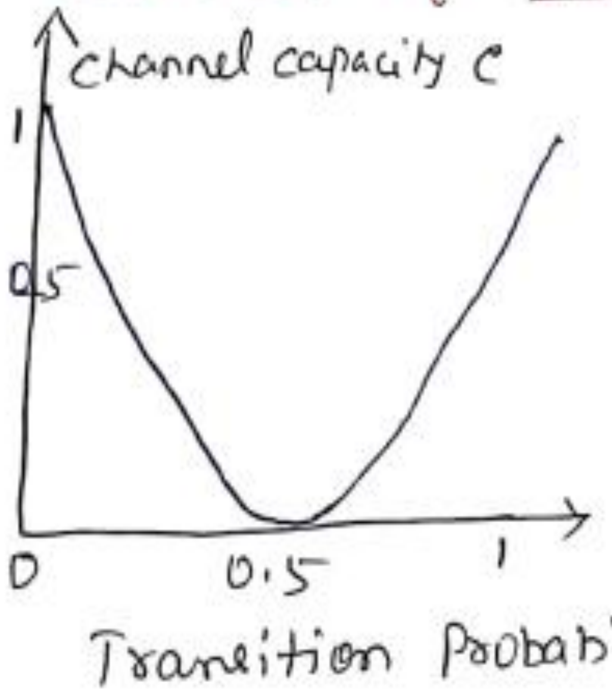
Channel capacity $C = \log_2 M - [\text{Entropy of first row elements}]$

$$C = \log_2 M - \left[(1-P)\log\frac{1}{(1-P)} + P\log\frac{1}{P}\right]$$

Here $M = 2$ because two inputs

$$\therefore C = \log_2 2 - H(P) = 1 - H(P) \Rightarrow \boxed{C = 1 - H(P)}$$

Variation of channel capacity with transition probability



At $P=0$, $C = 1 - \left[(1-P)\log\frac{1}{(1-P)} + P\log\frac{1}{P}\right]$

$$= 1 - \left[(1-0)\log\frac{1}{1} + 0\log\frac{1}{0}\right]$$

$$\boxed{C = 1}$$

At $P=1$, $C = 1 - \left[0\log\frac{1}{0} + 1\log 1\right]$

$$\boxed{C = 1}$$

At $P=0.5$, $C = 1 - \left[0.5\log\frac{1}{0.5} + 0.5\log\frac{1}{0.5}\right]$

$$\boxed{C = 1 - 1 = 0}$$

From the curve it is observed that,

1. When the channel is noise free [(ie) $p = 0$], the channel capacity 'C' attains its maximum value of 1 bit per channel, which is exactly the information in each channel input. At this value of p, the entropy $H(p)$ attains its minimum value of zero.

2. When the channel is noisy [(ie) $p = 1/2$], the channel capacity 'C' attains its minimum value of zero, whereas the entropy $H(p)$ attains its maximum value of one. In that case, the channel is said to be useless.

Channel coding Theorem

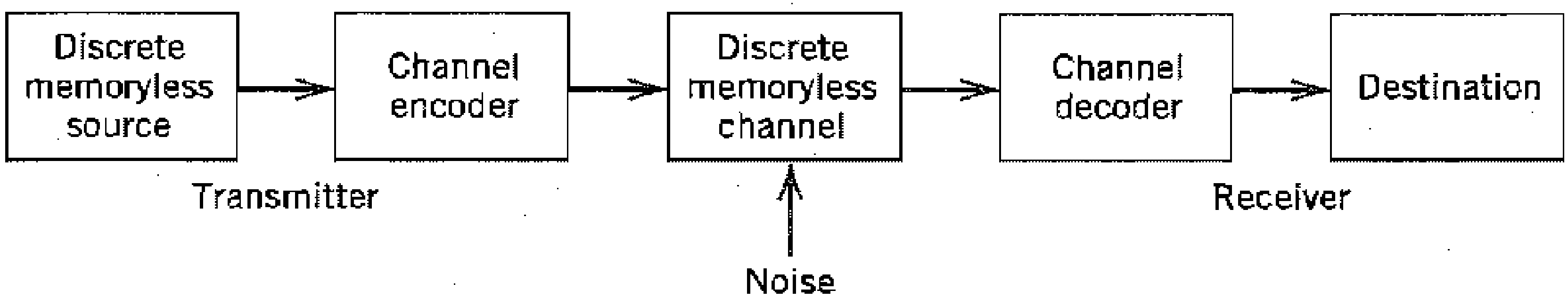


FIGURE 9.11 Block diagram of digital communication system.

Suppose the discrete memoryless source has the source alphabet S , entropy H(S) and it emits symbols once every T_s seconds. So, the average information rate is H(S)/ T_s bits per second. The decoder delivers decoded symbols to the destination at the same source rate of one symbol every T_s seconds. The discrete memoryless channel has a channel capacity equal to C bits per channel. Assume that the channel is being capable of being used once every T_c seconds. Hence , the channel capacity per unit time is C/ T_c bits per second which represents the maximum rate of information transfer over the channel.

Channel Coding Theorem Definition : It is stated as: ‘If the **information rate of a given source does not exceed the capacity of a given channel then there exists a coding technique that makes possible transmission through the unreliable channel with a low error rate.**

$$\boxed{H(S)/T_s \leq C/T_c}$$

The parameter C/T_c is called critical rate.

Suppose $H(S)/T_s > C/T_c$ **,** then it is not possible to transmit over the channel and reconstruct with an arbitrarily small probability or error.

Book Exercise Problems:

Problem 1: Consider a discrete memoryless source alphabet $A=\{S_0,S_1,S_2\}$ and source statistics {0.7,0.15,0.15}

a) Calculate the entropy of the source.

b) Calculate the entropy of the second-order extension of the source.

(a) The entropy of the source is

$$H(S) = 0.7 \log_2 \frac{1}{0.7} + 0.15 \log_2 \frac{1}{0.15} + 0.15 \log_2 \frac{1}{0.15}$$

$$= 0.258 + 0.4105 + 0.4105$$

$$= 1.079 \text{ bits}$$

(b) The entropy of second-order extension of the source is

$$H(S^2) = 2 \times 1.079 = 2.158 \text{ bits}$$

Problem 2: It may come as a surprise, but the number of bits needed to store text is much less than that required to store its spoken equivalent. Can you explain the reason for it?

The entropy of text is defined by the smallest number of bits . On average English text may be represented by 3bits per character, because of redundancy built into the English language. But, the spoken equivalent of English text has less redundancy; So, its entropy is higher than 3 bits. Therefore as per source coding theorem that the number of bits required to store text is smaller than the number of bits required to store its spoken equivalent.

Problem 3: Consider the four codes listed below:

Symbol	Code I	Code II	Code III	Code IV
S_0	0	0	0	00
S_1	10	01	01	01
S_2	110	001	011	10
S_3	1110	0010	110	110
S_4	1111	0011	111	111

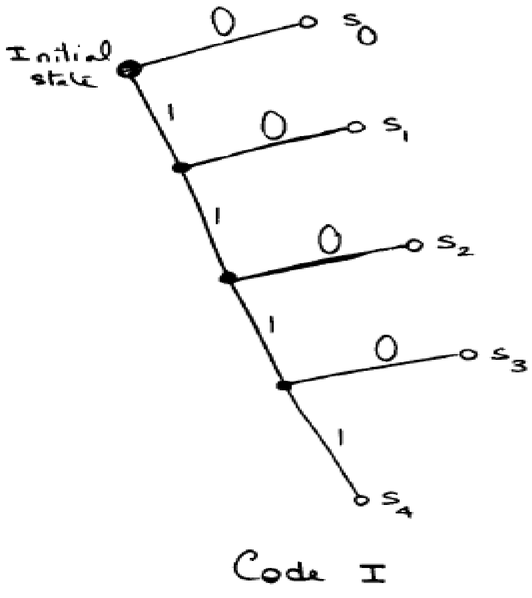
a. Two of these four codes are prefix codes. Identify them, and construct their individual decision trees.

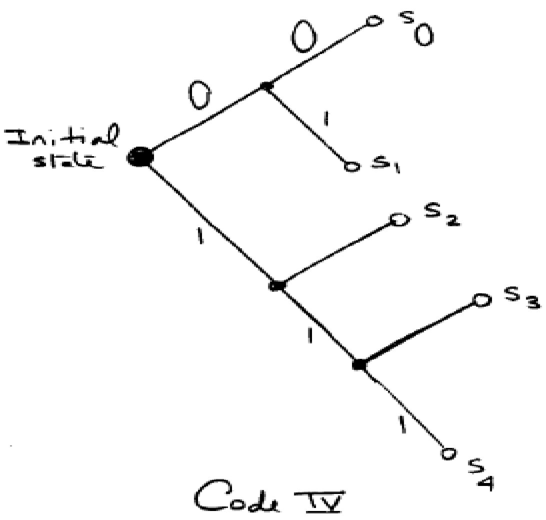
b. Apply the Kraft-McMillian inequality to codes I,II,III,and IV. Discuss your results in the light of those obtained in part(a).

Answer :

a.

A prefix code is defined as a code in which no code work is the prefix of any other code word. By inspection, we see therefore that codes I and IV are prefix codes, whereas codes II and III are not.





Symbol	Code I	Code II
S_0	0	00
S_1	10	01
S_2	110	10
S_3	1110	110
S_4	1111	111

(b) : Kraft-McMillian inequality to codes I,II,III,and IV

b. Kraft - McMillian inequality to codes I, II, III, IV

[$l_k \rightarrow$ length of code]

Code I :- $\boxed{\sum_{k=0}^{k-1} 2^{-l_k} \leq 1}$

$2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-4} \leq 1$

$0.5 + 0.25 + 0.125 + 0.0625 + 0.0625 \leq 1$

$\boxed{1 \leq 1.}$

Code I satisfies Kraft - McMillian inequality.

Code II : $\sum_{k=0}^{k-1} 2^{-l_k} \leq 1$

$2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-4} \leq 1$

$0.5 + 0.25 + 0.125 + 0.0625 + 0.0625 \leq 1$

$\boxed{1 \leq 1}$ Satisfies.

Code III: $\sum_{K=0}^{K-1} 2^{-l_K} \leq 1$

$2^{-1} + 2^{-2} + 2^{-3} + 2^{-3} + 2^{-3} \leq 1$

$0.5 + 0.25 + 0.125 + 0.125 + 0.125 \leq 1$

$\boxed{1.125 \nleq 1}$ Not satisfies.

Code IV: $\sum_{K=0}^{K-1} 2^{-l_K} \leq 1$

$2^{-2} + 2^{-2} + 2^{-2} + 2^{-3} + 2^{-3} \leq 1$

$0.25 + 0.25 + 0.25 + 0.125 + 0.125 \leq 1$

$\boxed{1 \leq 1}$ satisfies

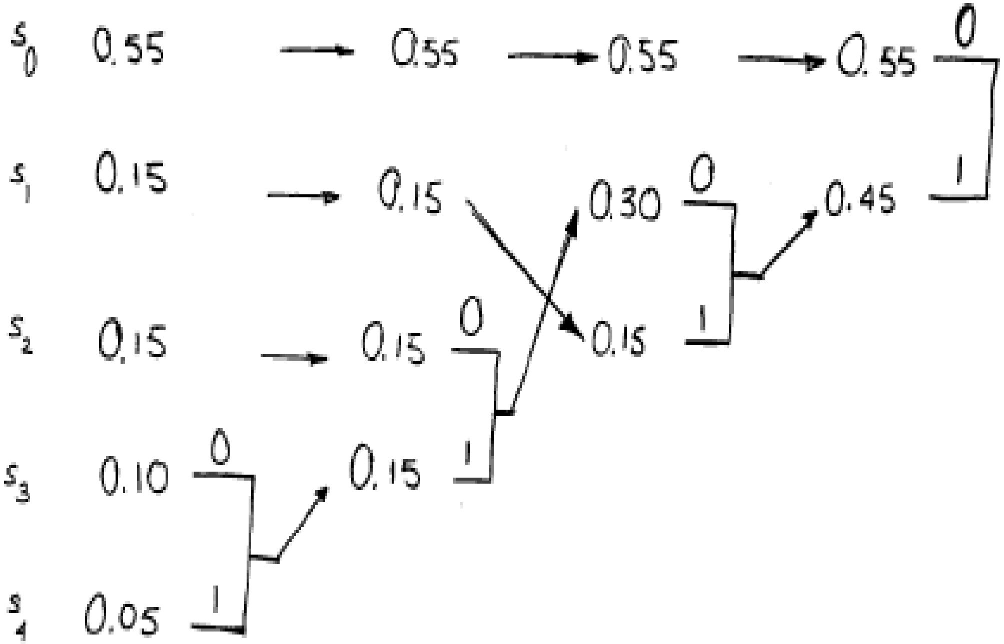
Problem 4: Consider a sequence of letters of the English alphabet with their probabilities of occurrence as given here:

Letter	**a**	**i**	**l**	**m**	**n**	**o**	**p**	**y**
Probability	**0.1**	**0.1**	**0.2**	**0.1**	**0.1**	**0.2**	**0.1**	**0.1**

Compute two difference Huffaman codes for this alphabet. In one case, move a combined symbol in the coding procedure as high as possible, and in the second case, move it as low as possible. Hence, for each of the two codes, find the average code-word length and the variance of the average code-word length over the ensemble of letters.

We may construct two different Huffman codes by choosing to place a combined symbol as low or as high as possible when its probability is equal to that of another symbol.

We begin with the Huffman code generated by placing a combined symbol as low as possible:



The source code is therefore

s_0	0
s_1	1 1
s_2	1 0 0
s_3	1 0 1 0
s_4	1 0 1 1

The average code-word length is therefore

$$\bar{L} = \sum_{k=0}^{4} p_k l_k$$

$$= 0.55(1) + 0.15(2) + 0.15(3) + 0.1(4) + 0.05(4)$$

$$= 1.9$$

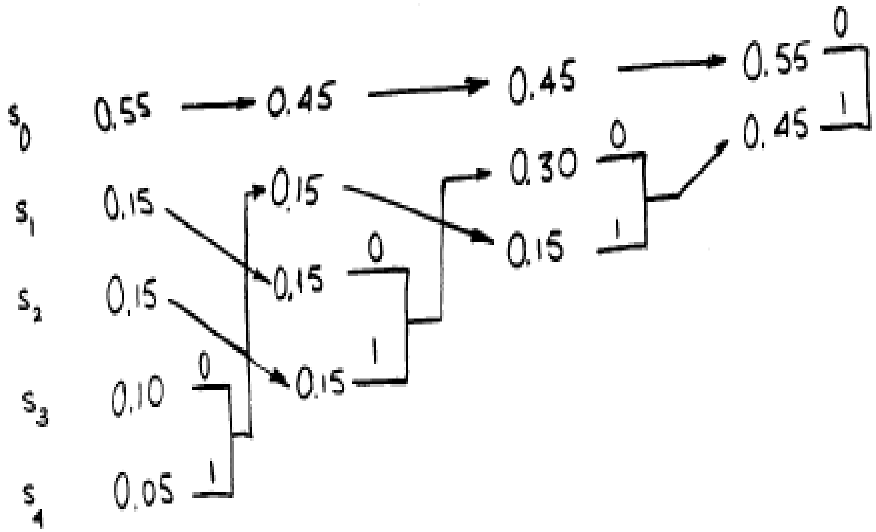
The variance of $\bar{L}$ is

$$\sigma^2 = \sum_{k=0}^{4} p_k (l_k - \bar{L})^2$$

$$= 0.55(-0.9)^2 + 0.15(0.1)^2 + 0.15(1.1)^2 + 0.1(2.1)^2 + 0.05(2.1)^2$$

$$= 1.29$$

Next placing a combined symbol as high as possible, we obtain the second Huffman code:



Correspondingly, the Huffman code is

s_0	0
s_1	1 0 0
s_2	1 0 1
s_3	1 1 0
s_4	1 1 1

The average code-word length is

$$\bar{L} = 0.55(1) + (0.15 + 0.15 + 0.1 + 0.05)\,(3)$$
$$= 1.9$$

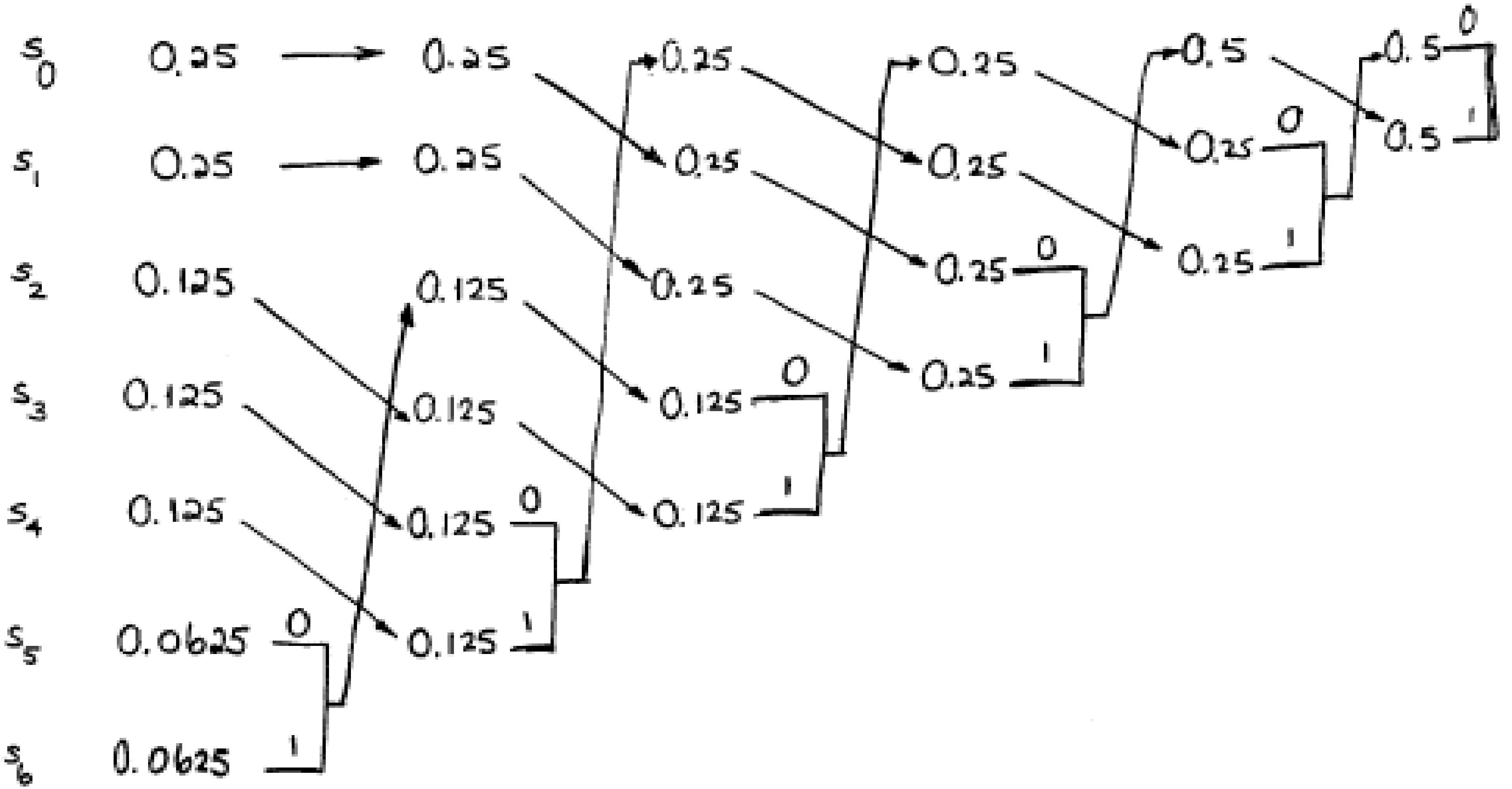
The variance of $\bar{L}$ is

$$\sigma^2 = 0.55(-0.9)^2 + (0.15 + 0.15 + 0.1 + 0.05)\,(1.1)^2$$
$$= 0.99$$

The two Huffman codes described herein have the same average code-word length but different variances.

Problem 5: A discrete memoryless source has an alphabet of seven symbols whose probabilities of occurrence are as described here: Generate Huffman codes and find efficiency.

Symbol	S_0	S_1	S_2	S_3	S_4	S_5	S_6
Probability	0.25	0.25	0.125	0.125	0.125	0.0625	0.0625



The Huffman code is therefore

s_0	1 0
s_1	1 1
s_2	0 0 1
s_3	0 1 0
s_4	0 1 1
s_5	0 0 0 0
s_6	0 0 0 1

The average code-word length is

$$\bar{L} = \sum_{k=0}^{6} p_k l_k$$

$$= 0.25(2)(2) + 0.125(3)(3) + 0.0625(4)(2)$$

$$= 2.625$$

The entropy of the source is

$$H(S) = \sum_{k=0}^{6} p_k \log_2\left(\frac{1}{p_k}\right)$$

$$= 0.25(2) \log_2\left(\frac{1}{0.25}\right) + 0.125(3) \log_2\left(\frac{1}{0.125}\right)$$

$$+ 0.0625(2) \log_2\left(\frac{1}{0.0625}\right)$$

$$= 2.625$$

The efficiency of the code is therefore

$$\eta = \frac{H(S)}{\bar{L}} = \frac{2.625}{2.625} = 1$$

Problem 6: Consider a discrete memory less source with alphabet $\{S_0, S_1, S_2\}$ and statistics $\{0.7, 0.15, 0.15\}$ for its output.

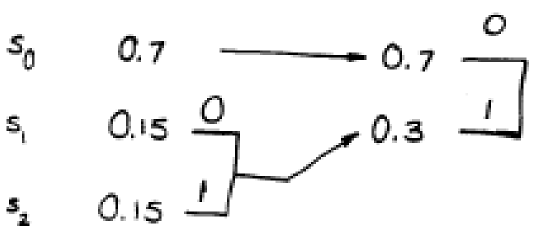
a) Apply the Huffman algorithm to this source. Hence, show that the average code-word length of the Huffman code equals 1.3 bits/symbol.

b)Let the source be extended to order two. Apply the Huffman algorithm to the resulting extended source, and show that the average code-word length of the new code equals 1.1975 bits/symbol.

c)Compare the average code-word length calculated in part (b) with the entropy of the original source.

Answer:

(a)



The Huffman code is therefore

s_0	0
s_1	1 0
s_2	1 1

The average code-word length is

$$\bar{L} = 0.7(1) + 0.15(2) + 0.12(2)$$

$$= 1.3$$

(b) For the extended source we have

Symbol	s_0s_0	s_0s_1	s_0s_2	s_1s_0	s_2s_0	s_1s_1	s_1s_2	s_2s_1	s_2s_2
Probability	0.49	0.105	0.105	0.105	0.105	0.0225	0.0225	0.0225	0.0225

Applying the Huffman algorithm to the extended source, we obtain the following source code:

s_0s_0	1
s_0s_1	0 0 1
s_0s_2	0 1 0
s_1s_0	0 1 1
s_2s_0	0 0 0 0
s_1s_1	0 0 0 1 0 0
s_1s_2	0 0 0 1 0 1
s_2s_1	0 0 0 1 1 0
s_2s_2	0 0 0 1 1 1

The corresponding value of the average code-word length is

$$\overline{L_2} = 0.49(1) + 0.105(3)(3) + 0.105(4) + 0.0225(4)(4)$$

$$= 2.395 \text{ bits/extended symbol}$$

$$\frac{\overline{L_2}}{2} = 1.1975 \text{ bits/symbol}$$

(c) The original source has entropy

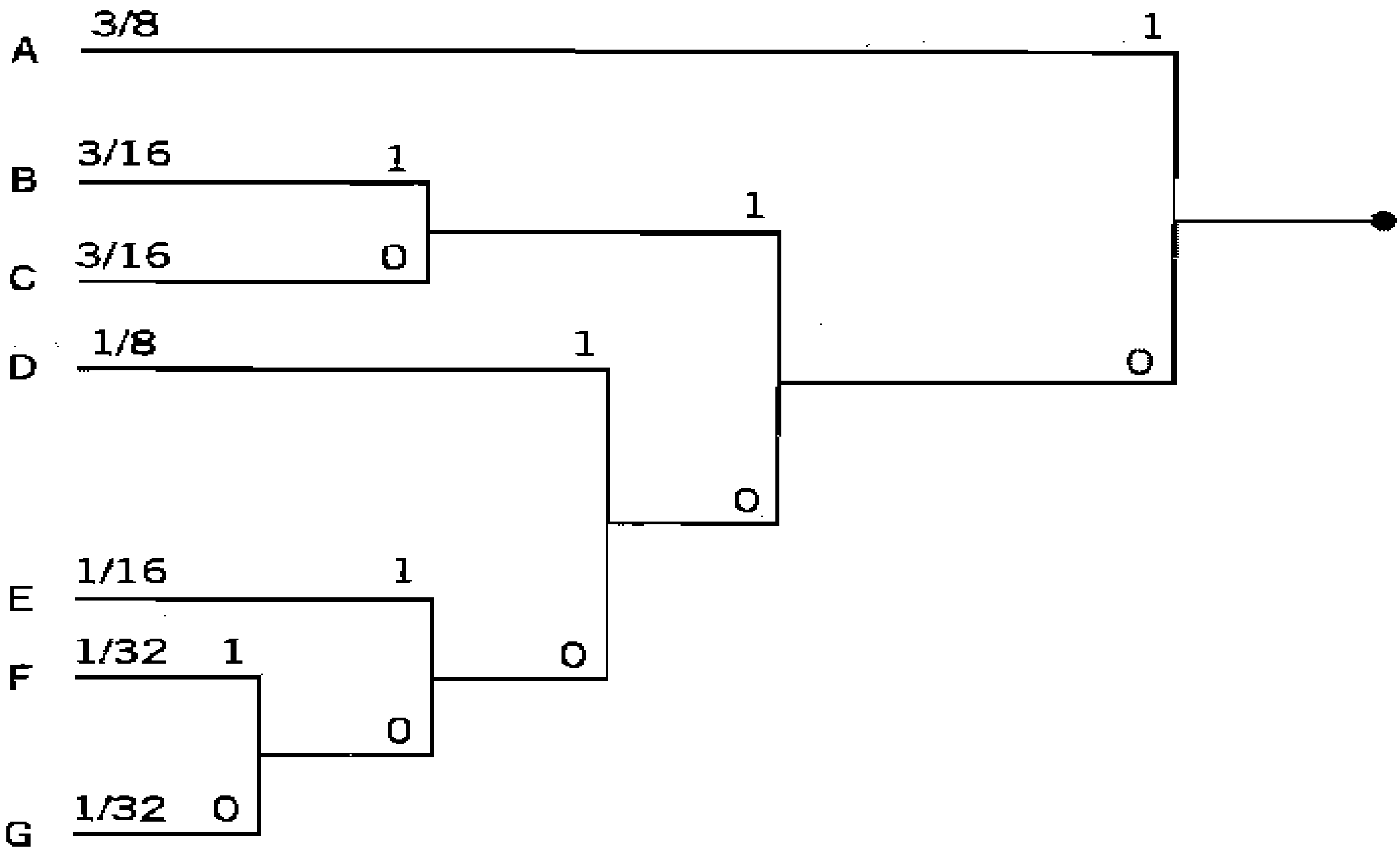
$$H(S) = 0.7 \log_2\left(\frac{1}{0.7}\right) + 0.15(2) \log_2\left(\frac{1}{0.15}\right)$$

$$= 1.18$$

$$H(S) \leq \frac{L_n}{n} \leq H(S) + \frac{1}{n}$$

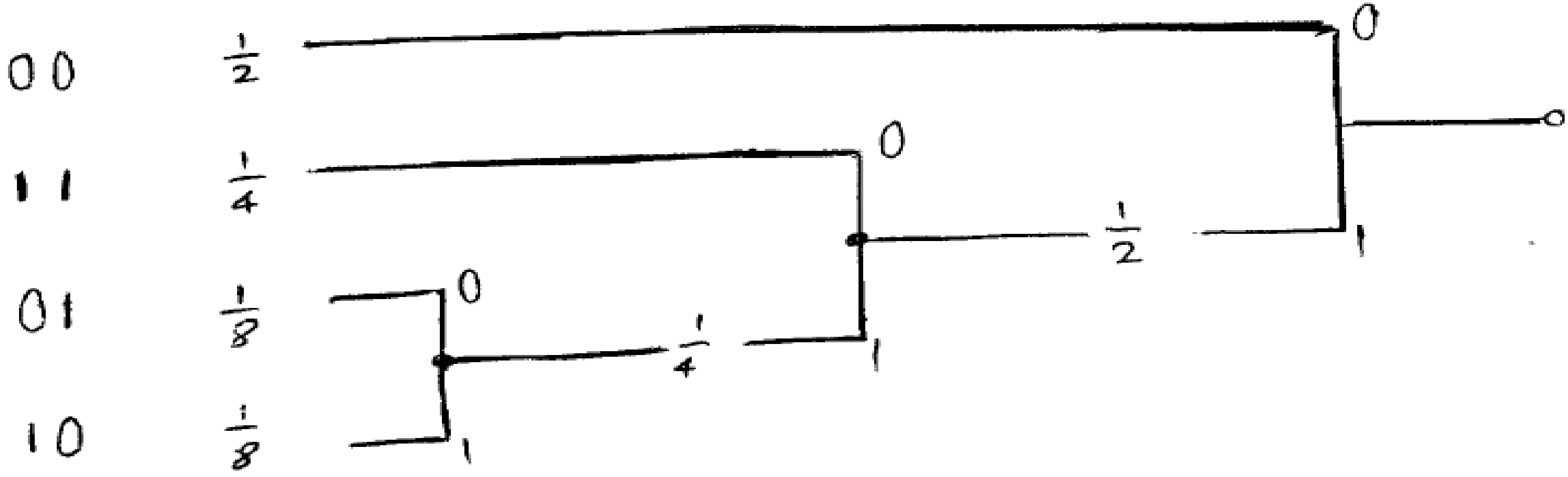
This is a condition which the extended code satisfies.

Problem 7: Figure shows a Huffman tree, What is the code word for each of the symbols A,B,C,D,E,F and G represented by this Huffman tree? What are their individual code-word lengths?



Symbol	Huffman Code	Code-word length
A	1	1
B	0 1 1	3
C	0 1 0	3
D	0 0 1	3
E	0 0 1 1	4
F	0 0 0 0 1	5
G	0 0 0 0 0	5

Problem 8: A computer executes four instructions that are designated by the code words (00,01,10,11). Assuming that the instructions are used independent with probabilities (1/2,1/8,1/8,14), calculate the percentage by which the number of bits used for the instructions may be reduced by the use of an optimum source code. Construct a Huffman code to realize the reduction.



Computer code	Probability	Huffman Code
0 0	$\frac{1}{2}$	0
1 1	$\frac{1}{4}$	1 0
0 1	$\frac{1}{8}$	1 1 0
1 0	$\frac{1}{8}$	1 1 1

The number of bits used for the instructions based on the computer code is equal to

$$2\left(\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\frac{1}{8}\right)=2 \text{ bits}$$

On the other hand, the number of bits used for instructions based on the Huffman code, is equal to

$$1 \times \frac{1}{2}+2 \times \frac{1}{4}+3 \times \frac{1}{8}+3 \times \frac{1}{8}=\frac{7}{4} \quad = 1.75$$

The percentage reduction in the number of bits used for instruction, realized by adopting the Huffman code, is therefore

Number of bits used by Huffman code / Number of bits used by computer code = (1.75/2) × 100 = 87.5 ; Therefore the percentage reduction is 12.5

Problem 9: Consider the following binary sequence

111010011000101101 00..

Use the Lempel-Ziv algorithm to encode this sequence. Assume that the binary symbols 0 and 1 are already in the codebook. Also, decode to get the original sequence.

<u>Initial step</u>

Subsequences stored: 0

Data to be parsed: 1 1 1 0 1 0 0 1 1 0 0 0 1 0 1 1 0 1 0 0 ...

<u>Step 1</u>

Subsequences stored: 0, 1, 11

Data to be parsed: 1 0 1 0 0 1 1 0 0 0 1 0 1 1 0 1 0 0 ..

<u>Step 2</u>

Subsequences stored: 0, 1, 11, 10

Data to be parsed: 1 0 0 1 1 0 0 0 1 0 1 1 0 1 0 0 ....

<u>Step 3</u>

Subsequences stored: 0, 1, 11, 10, 100

Data to be parsed: 1 1 0 0 0 1 0 1 1 0 1 0 0 ...

<u>Step 4</u>

Subsequences stored: 0, 1, 11, 10, 100, 110

Data to be parsed: 0 0 1 0 1 1 0 1 0 0 ...

Step 5

Subhsequences stored: 0, 1, 11, 10, 100, 110, 00

Data to be parsed: 1 0 1 1 0 1 0 0 ....

Step 6

Subsequences stored: 0, 1, 11, 10, 100, 110, 00, 101

Data to be parsed: 1 0 1 0 0 ...

Step 7

Subsequences stored: 0, 1, 11, 10, 100, 110, 00, 101, 1010

Data to be parsed: 0 ....

Numerical positions	1	2	3	4	5	6	7	8	9
Subsequences	0,	1,	11,	10,	100,	110,	00,	101,	1010
Numerical representations			22,	21,	41,	31,	11,	42,	81
Binary encoded blocks			0101,	0100,	**1000**	0110,	0010,	1001,	10000
Actual Transmission	**00000**	**00001**	**00101**	**00100**	**01000**	**00110**	**00010**	**01001**	**10000**

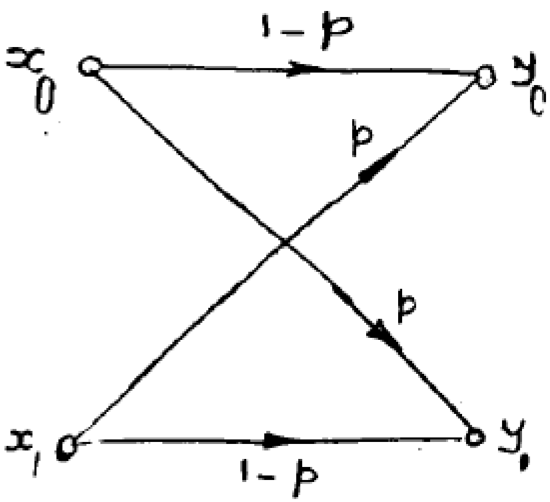
Received bits	**00000**	**00001**	**00101**	**00100**	**01000**	**00110**	**00010**	**01001**	**10000**
Numerical positions	1	2	3	4	5	6	7	8	9
Numerical Rep	0	1	22	21	41	31	11	42	81
Sub sequence	0	1	11	10	100	110	00	101	1010

Original Data : 1110100110001011010**0**

Decoded Data : 111010011000101101 0

Compression efficiency = After LempvelZiv/ before Lempvelziv = 45/(20*8) = 28.5%

Problem 10: Consider the transition probability diagram of a binary symmetric channel. The input binary symbols 0 and 1 occur with equal probability. Find the probabilities of the binary symbols 0 and 1 appearing at the channel output.



$$p(x_0) = p(x_1) = \frac{1}{2}$$

$$p(y_0) = (1 - p)p(x_0) + p\, p(x_1)$$

$$= (1 - p)\left(\frac{1}{2}\right) + p\left(\frac{1}{2}\right)$$

$$= \frac{1}{2}$$

$$p(y_1) = p\, p(x_0) + (1 - p)\, p(x_1)$$

$$= p\left(\frac{1}{2}\right) + (1 - p)\left(\frac{1}{2}\right)$$

$$= \frac{1}{2}$$

Problem 11 : Consider the transition probability diagram of a binary symmetric channel. Assuming that the input binary symbols 0 and 1 occur with probabilities ¼and ¾ respectively. Find the probabilities of the binary symbols 0 and 1 appearing at the channel output.

Input symbol '0' $p(x_0) = \frac{1}{4}$

Input symbol '1' $p(x_1) = \frac{3}{4}$

Output symbol '0' $p(y_0) = (1 - p)\left(\frac{1}{4}\right) + p\left(\frac{3}{4}\right)$

$$= \frac{1}{4} + \frac{p}{2}$$

Output symbol '1' $p(y_i) = p\left(\frac{1}{4}\right) + (1 - p)\left(\frac{3}{4}\right)$

$$= \frac{3}{4} - \frac{p}{2}$$