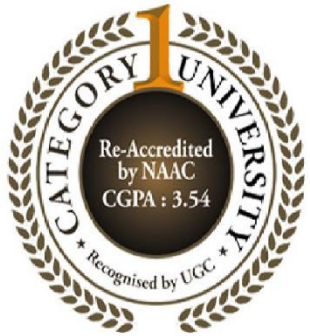
SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 OF THE UGC ACT, 1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA



Course Code: ECE403

Course Name: Information Theory & Coding

Course Objectives

- This course enables the learners to realize the fundamental concepts of information theory, various types of communication channel and its capacity for data transfer
- The course also analyzes various types of source coding and channel coding techniques and their significance for efficient and reliable communication

Syllabus

UNIT I – INFORMATION THEORY and SOURCE CODING

Block Diagram of a Communication System – Fundamental Problems of Communication – Information and Entropy – Properties of Entropy – Binary Memoryless Source – Extension to Discrete Memoryless Source - Elements of Encoding – Properties of Code – Kraft-Macmillan Inequality – Code Length – Code Efficiency – Source Coding Theorem – Source Coding Techniques – Shannon encoding - Shannon-Fano Encoding - Huffman's Encoding, Arithmetic Coding, Run-Length Encoding, Lempel-Ziv Encoding and Decoding

UNIT II – NOISY CHANNEL CODING

Measure of Information for two dimensional discrete finite probability scheme – marginal, conditional and joint entropies – Interpretation of different entropies for a two port communication system – Basic relationships among different entropies – Discrete Memoryless Channel – Mutual Information – Properties – Channel Capacity – Channel Classification – Channel Coding Theorem

Entropy in the continuous case – Definition and Properties – Capacity of a band-limited Gaussian channel – Hartley-Shannon's Law – Ideal System – Definition – Bandwidth Efficiency Diagram

Syllabus

UNIT III – BLOCK CODES, CYCLIC CODES and CONVOLUTIONAL CODES

Block Codes: Introduction – Hamming Code – Linear Block Codes – Syndrome decoding – Minimum Distance Consideration

Cyclic Codes: Generator Polynomial – Parity-Check Polynomial – Encoder for Cyclic Codes – Calculation of the Syndrome

Convolutional Codes: Convolutional Encoder Representations (State Diagram, Code trellis, Code tree) – Viterbi decoding. Trellis Coded Modulation

UNIT IV – BCH, RS, LDPC and TURBO CODES

General Principles – Definition and Construction of Binary BCH codes – Error Syndromes in finite fields – Decoding of SEC and DEC – binary BCH codes – Error Location Polynomial – Peterson-Gorenstein-Zieler Decoder – Reed-Solomon Codes – Reed Solomon Encoding and decoding – Introduction to LDPC and Turbo Codes

References

- **TEXTBOOKS:**
  - Bernard Sklar and Prabitra Kumar Ray, ***Digital Communications***, 2nd Edition, Pearson Education, 2011
  - Simon Haykin, ***Communication Systems***, 5th Edition, John Wiley and Sons, 2010
  - F.M.Reza, ***An introduction to information theory***, McGraw Hill Inc., 1994
- **REFERENCES:**
  - B.P.Lathi, ***Modern Digital and Analog Communication Systems***, 4th Edition, Oxford University Press, 2012
  - Salvatore Gravano, ***Introduction to Error Control Codes***, Oxford University Press, 2011
  - R.P.Singh and S.D.Sapre, ***Communication Systems - Analog and Digital***, 2nd Edition, Tata McGraw Hill, 2008
  - Peter Sweeney, ***Error Control Coding from Theory to Practice***, 2nd Edition, Wiley, 2002
- **ONLINE MATERIAL:**
  - NPTEL : http://www.youtube.com/watch?v=f8RvFlr5wRk

Learning Outcomes

- **UNIT I:**
  - Remember the basics notions in information theory like self-information, entropy and its types
  - Implement various types of source coding algorithms and classify them
- **UNIT II:**
  - Analyse various types of communication channels and its channel capacity
- **UNIT III:**
  - Design and interpret various types of error control codes like linear block codes, cyclic codes, convolutional codes and trellis coded modulation
- **UNIT IV:**
  - Design and interpret about BCH Code and Reed Solomon Code

Contents

- **UNIT I – Information Theory and Source Coding**
  - Block diagram of a communication system
  - Fundamental problems of communication
  - Information and entropy
    - Properties of entropy
    - Binary memoryless source
    - Extension to discrete memoryless source
  - Elements of encoding
    - Properties of code
    - Kraft-Macmillan Inequality
    - Code length
    - Code Efficiency

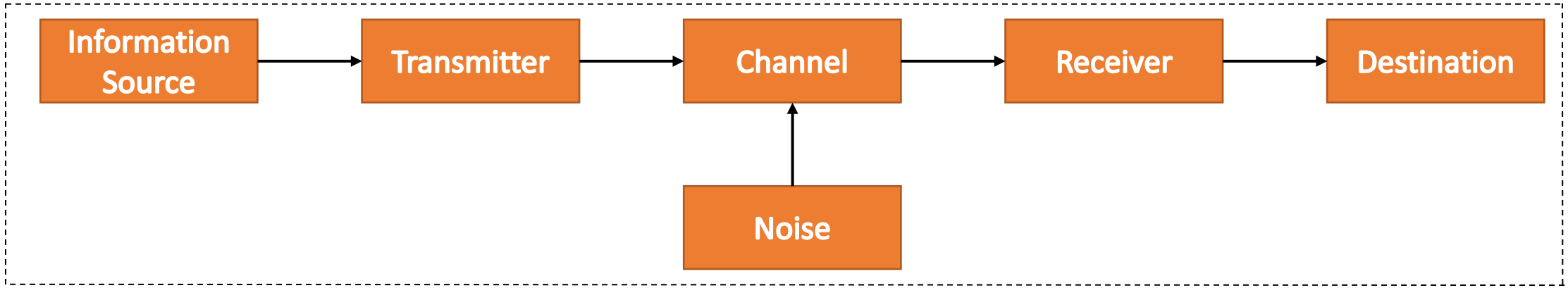
Contents

- **UNIT I – Information Theory and Source Coding**
  - Source Coding Theorem
  - Source Coding Techniques
    - Shannon encoding
    - Shannon-Fano encoding
    - Huffman's encoding
    - Arithmetic Coding
    - Run-Length Encoding
    - Lempel-Ziv encoding and decoding

Introduction

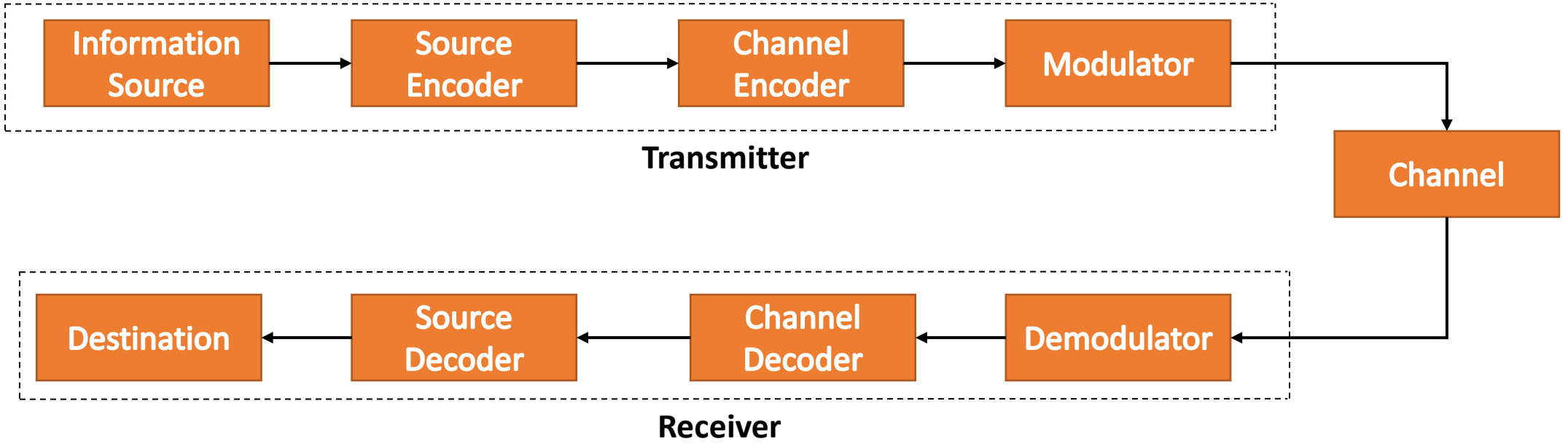
- Information Theory deals with a mathematical modeling and analysis of a Communication System
- It determines the capacity of the system to transfer essential information from the source to destination
- Information Theory provides the following limitations
    1. Minimum number of bits/symbol required to completely specify the source
    2. Maximum rate at which reliable communication can take place

Basic Block Diagram of a Communication System



- **Transmitter:** Convert the message signal produced by the source of information into a form suitable for transmission over the channel. However, as the transmitted signal propagates along the channel, it is distorted due to channel imperfections
- **Channel:** Physical medium that connects the transmitter and the receiver. noise and interfering signal (originating from other sources) are added to the channel output, with the result that the received signal is a corrupted version of the transmitted signal
- **Receiver:** Reconstructs a recognizable form of the original message signal for an end user or information sink

Block Diagram of Communication System



- The basic block representation of Communication System consists of:
  - **Transmitter**
  - **Channel**
  - **Receiver**

Block Diagram of Communication System

- **Information Source**
  - Binary stream of bits (0's and 1's)
- **Source Encoder**
  - Compresses the data into minimum number of bits in order to have effective utilization of the bandwidth
  - It is done by removing redundant information (bits)
- **Channel Encoder**
  - Performs the process of error correction as the noise in the channel might alter the information
  - It adds redundant bits to the transmitted data called the error correcting bits

Block Diagram of Communication System

- **Modulator**
  - This facilitates the transmission of signal over long distances

  (e.g.,) ASK, FSK. PSK, QAM, QPSK

- **Channel**
  - Medium of data transfer
  - Source of various types of noise

- **Demodulator**
  - The received signal is demodulated to extract the original signal from the carrier

Block Diagram of Communication System

- **Channel Decoder**
  - The distortions occurred during the transmission are corrected by the decoder
- **Source Decoder:**
  - The source decoder recreates the source output
- **Destination:**
  - Output at the receiver end of the communication system

Limitations of a Communication System

- Bandwidth
- Noise
- Equipment
- Minimum number of bits/symbol
- Channel capacity ($R \leq C$)

Information

- Information encountered in a Communication system are statistically defined
- Most significant feature is unpredictability
- An information source is an object that produces an event, the outcome of which is selected at random according to a probability distribution
- Information source can either be with memory or memory less
- Source with memory depends on previous information (symbol)
- Memoryless source produces symbols independent of previous symbols
- **Information is a non negative quantity - Important property**

Properties of Self information

- **I(x_i) satisfies the following properties:**
  - If receiver knows message being transmitted then information is zero,

    I(x_i) = 0 for P(x_i) = 1

  - If uncertainty is more information is more and probability is less

I(x_i) >I(x_j) if P(x_i) <P(x_j)

- I(x_i, x_j) = I(x_i) + I(x_j) , if x_i and x_j are independent
- If there are m = 2^N equally likely messages, then amount of information carried by each message will be “N” bits

Self Information

- Lack of Information, that gives the amount of uncertainty
- System can be binary, analog or digital
- The unit of choice of self information is based on the information
- Let us consider a DMS denoted by 'x' and having alphabet x = $\{x_1, x_2, \ldots\ldots, x_m\}$

  $I(x_i) = - log_\gamma P(x_i)$

γ	$\mathbf{I(x_i)}$	**Unit of $\mathbf{I(x_i)}$**	**Name of Sequence**
2	$I(x_i) = - log_2 P(x_i)$	Bits	Binary
3	$I(x_i) = - log_3 P(x_i)$	Triples	Ternary
4	$I(x_i) = - log_4 P(x_i)$	Quadruples	Quaternary
10	$I(x_i) = - log_{10} P(x_i)$	Decits or Hartley	Decadron
e	$I(x_i) = - log_e P(x_i)$	Nats	Natural

Properties of Self information

- **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
  - nat (natural unit) if b = e
  - It is standard to use b = 2 since efficiency is more

$log_2\ a$ = ln a / ln 2 = log a / log 2

Base Conversions

- $\log_2 X = \frac{\log_{10} X}{\log_{10} 2} = \frac{\log_{10} X}{0.3010}$
- $\log_3 X = \frac{\log_{10} X}{\log_{10} 3} = \frac{\log_{10} X}{0.4771}$
- $\log_2 3 = \frac{\log_{10} 3}{\log_{10} 2} = \frac{0.4771}{0.3010} = 1.585$
- $\log_2 2 = \frac{\log_{10} 2}{\log_{10} 2} = 1$
- $\log_2 \left(\frac{1}{P(A)}\right) = 3.322 \text{ x } \log_{10} \left(\frac{1}{P(A)}\right)$

Problems on Self Information

- **A source produces one of the four possible symbols during each interval having probabilities P(X_1)=0.5, P(X_2)=0.25, P(X_3)=P(X_4)=0.125 . Obtain the information content of each of these symbols.**

Soln.,

$$I(X) = log_2 \left(\frac{1}{P(X)} \right)$$

Therefore.,

$$I(X_1) = log_2 \left(\frac{1}{0.5} \right) = 1 \text{ bit} \; ; \; I(X_2) = log_2 \left(\frac{1}{0.25} \right) = 2 \text{ bits} \; ; \; I(X_3) = I(X_4) = log_2 \left(\frac{1}{0.125} \right) = 3 \text{ bits}$$

- **Calculate the amount of information if binary digits occur with equal likelihood in a binary PCM System.**

Soln., Number of Symbols in PCM = 2 (0 and 1)

i.e., $P(X_1) = P(X_2) = 0.5$

$$I(X_1) = I(X_2) = log_2 \left(\frac{1}{0.5} \right) = 1 \text{ bit}$$

Problems on Self Information

- **If the receiver knows the message being transmitted, the amount of information carried will be Zero. PROVE THE STATEMENT**

Soln., If the receiver is knowing the message transmitted then, P(X) = 1

$I(X) = log_2 \left(\frac{1}{P(X)} \right) = log_2 \left(\frac{1}{1} \right) = 0$

- **A card is selected at random from a deck and found that it is from red suite, how much information is received? How much more information is needed to completely specify the card?**

Soln., Information Received: $P(A_1) = \frac{26}{52} = 0.5$; $I(A_1) = log_2 \left(\frac{1}{0.5} \right) = 1$ bit

Information Received: $P(A_2) = \frac{1}{52} = 0.019$; $I(A_2) = log_2 \left(\frac{1}{0.019} \right) = 5.7$ bits

Therefore, $I(A_2) - I(A_1) = 4.7$ bits

Problems on Self Information

- **A single TV picture can be thought of an array of black, white and gray dots roughly 500 rows and 600 columns. Suppose that each of these dots may take on any one of 10 distinguishable levels, What is the amount of information provided by one picture?**

Soln., Total number of possible dots = 500 x 600 = 300000

Total number of pics possible = 10 x 10 x ……. x 10 = 10^{300000}

Probability of one pic, P(A) = $\frac{1}{10^{300000}}$

I(A) = $log_2 10^{300000}$ = 3 x $10^5 log_2$ 10

= 996578.43 = 1 Mb

= 9.965 x 10^5 bits

Discrete Memoryless Source (DMS)

- If the emitted symbols are statistically independent, i.e., any symbol being produced does not depend upon the symbols that have been produced already , we say that the source has no memory and is called as **D**iscrete **M**emoryless **S**ource

- **Information content of a DMS:**
  - The amount of information contained in a symbol (X_r) emitted by the DMS is closely related to the amount of uncertainty of that symbol
  - A mathematical measure of information should be a function of the probability of the outcome and should satisfy the following axioms

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

    (b) Information contained in independent outcomes should add up

Discrete Memoryless Source (DMS)

- A discrete source emits sequence of symbols from fixed finite source alphabet,

$$\{S\} = \{s_0, s_1, s_2, \ldots\ldots, s_q\} \qquad \textbf{[1]}$$

- Probabilities of emitted signals,

$$\{P\} = \{p_0, p_1, p_2, \ldots\ldots, p_q\} \qquad \textbf{[2]}$$

$$\sum_{k=1}^{q} P_k = 1$$

- Suppose we consider a long sequence of symbols, where n symbols is made up n_1 symbols of type s_1, n_2 symbols of type s_2 and n_q symbols of type s_q. The amount of information associated with each symbol of the source is given by,

$$I(S_k) = log_2 \left(\frac{1}{P(s_k)} \right) \qquad \textbf{[3]}$$

Discrete Memoryless Source (DMS)

- Average information is given by,

$$H(S) = \frac{1}{n}\sum_{k=1}^{q} I(s_k).\ n_k \qquad \textbf{[4]}$$

$$H(S) = \sum_{k=1}^{q} p_k\ .\ I(s_k) \qquad](P_k = \frac{n_k}{n}) \qquad \textbf{[5]}$$

$$\textbf{H(S)} = \sum_{k=1}^{q} \boldsymbol{p_k}\ \ \boldsymbol{log_2}\ (\frac{1}{\boldsymbol{p_k}})\ \textbf{bits/symbol} \qquad \textbf{[6]}$$

H(S) is called as entropy or average information or average uncertainty

Entropy

- Entropy is a measure of the uncertainty in a random variable
- The entropy H, of a discrete random variable X is a measure of the amount of uncertainty associated with the value of X
- Entropy is found maximum, when the uncertainty is maximum i.e., when all the alphabet of source X are equiprobable
- The quantity H(X) is called the entropy of source X

Entropy

- It is a measure of the average information content per source symbol denoted by,

$H(X) = E[I(x_i)] = -\Sigma P(x_i) I(x_i) = -\Sigma P(x_i) log_2 P(x_i)$

- Its unit is bits/symbol
- **Entropy for Binary Source:**

  $H(X) = -1/2\ log_2\ 1/2 - 1/2\ log_2\ 1/2 = 1$ bit/symbol

- The source entropy H(X) satisfies the relation: $\mathbf{0 \leq H(X) \leq log_2\ m}$, where m is the size of the alphabet source X

Properties of Entropy

- **Continuity Property:**
  - If the probability of occurrence of events X_k are slightly changed, the measurement of uncertainty associated with the system varies accordingly in a continuous manner

  p_k = P (X_k) ; 0 $\leqslant$ $p_k \leqslant$ 1

  H(X_k) = - $\sum_1^N p_k \ log_2 \ p_k$ bits/symbol

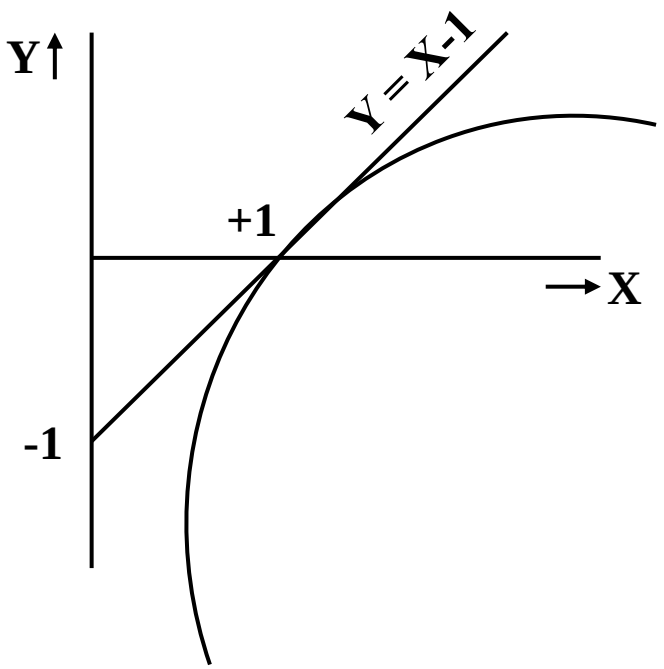
  - As, p_k is continuous between the limits 0 and 1, H(X) also varies continuously

- **Symmetry Property:**
  - H(X) is functionally symmetric in every p_k ; H(p_k , 1-p_k) = H(1-p_k, p_k) for all k = 1 to q
  - As the entropy is the sum of weighted averages its value remains the same even when the position of the values are interchanged. The value of the entropy function remains same irrespective of location of probabilities

Properties of Entropy

- **Minimum value of H(X) is Zero:** $0 \leq H(X) \leq log_2 N$; where N is total number of symbols
- **Extremal Property:**
  - Entropy has its maximum value when all the events are equally likely
- **Additive Property:**
  - $H_2(p_1,p_2,p_3,\ldots,p_{N-1}, q_1,q_2,q_3,\ldots,q_m) = H_1(p_1,p_2,p_3,\ldots,p_N) + p_N H_3(q_1/p_N, q_2/p_N,\ldots, q_N/p_N)$

Logarithmic Inequalities



- Plot of ln X, always lie below the straight line Y = X-1 and equality exists when X= +1

- Plot of straight line, Y=X-1
- Log function, Y= ln (X)
- $\ln X \leq (X-1)$; equality if X=1

Multiply by -1.,

$$\ln\left(\frac{1}{X}\right) \geq (1-X)\ ;$$ equality if X=1

- The figure shows plots of a straight line Y=X-1 and a log function Y= ln X on the same set of co=ordinate axis
- Notice that any point on the straight line will always be found above the log function for any given value of X
- The straight line is tangent to log function at X=1

Proof for Extremal Property

- **Statement:** For a zero memory information source with **'q'** symbol alphabet the entropy becomes maximum if and only if all the source symbols are equally probable

$$\mathbf{H(S)_{max} = \log q} \qquad ; \text{if } p_k = \frac{1}{q} \text{ for all k=1 to q}$$

- **Proof:** Consider a memory less source with q symbol alphabet, {S} = $\{s_1, s_2, \ldots\ldots, s_q\}$ with probabilities {P} = $\{p_1, p_2, p_3, \ldots\ldots, p_q\}$. Entropy of source is given by,

$$H(S) = \sum_{k=1}^{q} p_k \, log_2 \left(\frac{1}{p_k} \right) \qquad \textbf{[1]}$$

Consider log q – H(S)

$$= 1 \,.\, \log q - \sum_{k=1}^{q} p_k \, log_2 \left(\frac{1}{p_k} \right) \qquad \textbf{[2]}$$

Proof for Extremal Property

$(\sum_{k=1}^{q} P_k = 1)$

$$= \sum_{k=1}^{q} p_k \log q - \sum_{k=1}^{q} p_k \log \frac{1}{p_k} \quad \textbf{[3]}$$

$$=> log_2 q - H(S) = \sum_{k=1}^{q} p_k \, log_2 (q \, p_k) \quad \textbf{[4]}$$

Changing the base to e, $log_2 X = log_2 e \, . \, log_e X$

$$log_2 q - H(S) = log_2 e \, [\sum_{k=1}^{q} p_k \ln q. \, p_k] \quad \textbf{[5]}$$

- Apply logarithmic inequality $\ln (\frac{1}{X}) \geq (1\text{-}X)$ **[6]**

$$X = \frac{1}{q \, . \, p_k}$$

$$log_2 q - H(S) \geq log_2 e \sum_{k=1}^{q} p_k \, . \, (1 - \frac{1}{q \, . \, p_k}) \quad \textbf{[7]}$$

Proof for Extremal Property

Equality holds on only if X=1, $q \,.\, p_k = 1 \Rightarrow p_k = \frac{1}{q}$ **[8]**

$$log_2\, q - H(S) \leq log_2\, e \left(\sum_{k=1}^{q} p_k - \sum_{k=1}^{q} \frac{1}{q} \right) \quad \textbf{[9]}$$

$$log_2\, q - H(S) \geq 0 \quad \textbf{[10]}$$

$H(S) \leq \log q$

$H(S)_{max} = \log q$

Equality if $q \,.\, p_k = 1$ **[11]**

Proof for Additive Property

- **Statement:** Partitioning of symbols or events into sub-symbols or sub events cannot decrease the entropy
- **Proof:** Consider a memoryless information source with 'q' symbol alphabets $\{S\} = \{s_0, s_1, s_2, \ldots\ldots, s_q\}$ with associated probabilities $\{p_1, p_2, p_3, \ldots\ldots, p_q\}$. Suppose we split symbol S_q into 'm' sub-symbols such that,

$$S_q = \sum_{j=1}^{m} s_{q_j} \quad ; \quad P\{S_{q_j}\} = p_{q_j}$$

$$P_q = \sum_{j=1}^{m} p_{q_j}$$

$$H(S) = H(p_1, \ldots\ldots, p_{q-1}, p_{q1}, p_{q2}, \ldots, p_{qm}) \qquad \textbf{[1]}$$

$$H(S) = \sum_{k=1}^{q-1} p_k \log\left(\frac{1}{p_k}\right) + \sum_{j=1}^{m} p_{q_j} \log\left(\frac{1}{p_{q_j}}\right) \qquad \textbf{[2]}$$

$$= \sum_{k=1}^{q} p_k \log\left(\frac{1}{p_k}\right) - p_q \log\frac{1}{p_q} + \sum_{j=1}^{m} p_{q_j} \log\left(\frac{1}{p_{q_j}}\right) \qquad \textbf{[3]}$$

Proof for Additive Property

Since, $p_q = \sum_{j=1}^{m} p_{q_j}$

$$H(S) = \sum_{k=1}^{q} p_k \log\left(\frac{1}{p_k}\right) - \sum_{j=1}^{m} p_{q_j} \log\left(\frac{1}{p_q}\right) + \sum_{j=1}^{m} p_{q_j} \log\left(\frac{1}{p_{q_j}}\right) \quad \textbf{[4]}$$

$$H(S) = \sum_{k=1}^{q} p_k \log\left(\frac{1}{p_k}\right) + \sum_{j=1}^{m} p_{q_j}\left(-\log\left(\frac{1}{p_q}\right) + \log\left(\frac{1}{p_{q_j}}\right)\right)$$

$$H(S) = \sum_{k=1}^{q} p_k \log\left(\frac{1}{P_k}\right) + \sum_{j=1}^{m} p_{q_j} \log\frac{p_q}{p_{q_j}} \quad \textbf{[5]}$$

Proof for Additive Property

Multiply and Divide 2nd part of RHS by p_q,

$$H(S) = \sum_{k=1}^{q} p_k \log\left(\frac{1}{p_k}\right) + p_q \sum_{j=1}^{m} \frac{p_{q_j}}{p_q} \cdot \log \frac{p_q}{p_{q_j}} \qquad \textbf{[6]}$$

$$\boxed{H(S) = H(p_1, p_2, \ldots, p_q) + p_q \; H\left(\frac{p_{q_1}}{p_q}, \frac{p_{q_2}}{p_q}, \ldots, \frac{p_{qm}}{p_q}\right)} \qquad \textbf{[7]}$$

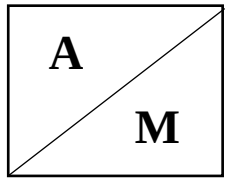
Since, the entropy functions are essentially non-negative, we have

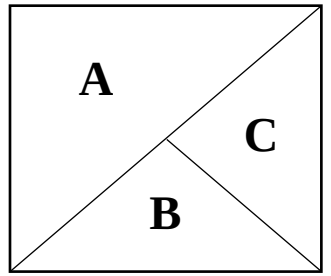
$$\boxed{H(p_1, p_2, \ldots, p_{q-1}, p_{q1}, p_{q2}, \ldots, p_{qm}) \geqslant H(p_1, p_2, \ldots, p_q)}$$

i.e., partitioning of symbols into sub-symbols cannot decrease entropy

Numerical Problem on Additive Property

- **A sample space of events is shown with {P} = $\{\frac{1}{5}, \frac{4}{15}, \frac{8}{15}\}$. Evaluate**
  - **(i) Average uncertainty associated with the scheme**
  - **(ii) Average uncertainty pertaining to the following probability scheme [A , M = B U C] , $[\frac{B}{M}, \frac{C}{M}]$**
  - **(iii) Verify the rule of additivity**





Soln., (i) $H(S) = \sum_{k=1}^{3} p_k \log\left(\frac{1}{p_k}\right) = \{ \left(\frac{1}{5}\log\left(\frac{1}{\frac{1}{5}}\right)\right) + \left(\frac{4}{15}\log\left(\frac{1}{\frac{4}{15}}\right)\right) + \left(\frac{8}{15}\log\left(\frac{1}{\frac{8}{15}}\right)\right) \}$

= 1.456 bits/symbol **[1]**

(ii) A , M = B U C ; M is divided into B U C sub-symbols

Average uncertainty, H [A , M = B U C]

$P(M) = P(B \cup C) = P(B) + P(C) = \frac{4}{15} + \frac{8}{15} = \frac{12}{15} = \frac{4}{5} = 0.8$

Numerical Problem on Additive Property

- $H(A, M) = H(\frac{1}{5}, \frac{4}{5}) = \{ (\frac{1}{5}\log(\frac{1}{\frac{1}{5}}) + (\frac{4}{5}\log(\frac{1}{\frac{4}{5}}) \}$

  $= 0.46 + 0.2575 = 0.721$

  $H[A, M = B \cup C] = 0.721$ bits/symbol

- $H(\frac{B}{M}, \frac{C}{M})$:

$$P(\frac{B}{M}) = \frac{P(B)}{P(M)} = \frac{\frac{4}{15}}{\frac{4}{5}} = \frac{1}{3} \quad ; \quad P(\frac{C}{M}) = \frac{P(C)}{P(M)} = \frac{\frac{8}{15}}{\frac{4}{5}} = \frac{2}{3}$$

- $H(\frac{B}{M}, \frac{C}{M}) = H(\frac{1}{3}, \frac{2}{3}) = \{ (\frac{1}{3}\log(\frac{1}{\frac{1}{3}}) + (\frac{2}{3}\log(\frac{1}{\frac{2}{3}}) \}$

  $= 0.528 + 0.39 = 0.918$ bits/symbol

Numerical Problem on Additive Property

(iii) Rule of Additivity:

$$H(S) = H(P_1, P_2, \ldots, P_q) + P_q\, H\left(\frac{P_{q1}}{P_q}, \frac{P_{q2}}{P_q}, \ldots, \frac{P_{qm}}{P_q}\right)$$

$$P = \{ P(A), P(M) \}$$

$$H(S) = H(P(A), P(M)) + P(M)\, H\left(\frac{P(B)}{P(M)}, \frac{P(C)}{P(M)}\right)$$

$$H(S) = 0.721 + (0.8)(0.918)$$

$$H(S) = 1.456 \text{ bits/symbol} \qquad \textbf{[2]}$$

[1] = [2]

Hence, Rule of additivity is verified

Entropy of a Zero Memory Binary Source

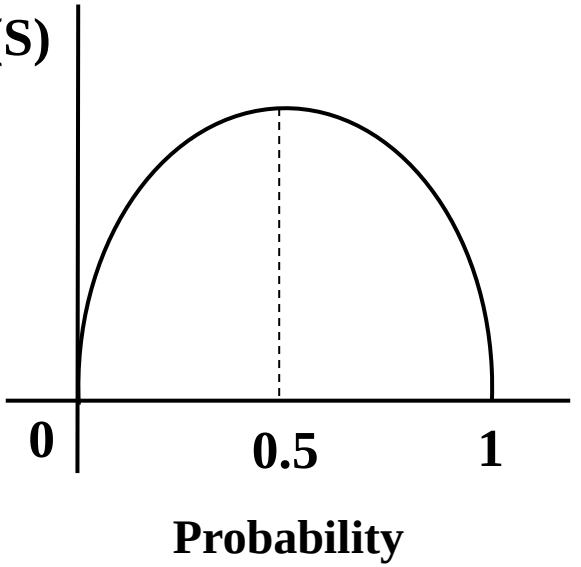
- For a zero memory binary source with source alphabet {S} = {0,1} with probabilities {P} = {p,q} ; p + q = 1

$$H(S) = \sum_{k=1}^{2} p_k \log \left(\frac{1}{p_k} \right)$$

$$= p \log \frac{1}{p} + q \log \frac{1}{q}$$

$$= - p \log p - q \log q$$

$$= - p \log p - (1-p) \log (1-p)$$



- The sketch showing variation of H(S) with probability is shown. If the output of source is certain, then the source provides no information
- The maximum entropy of the source occurs if 0 and 1 are equally likely

Information Rate

- Suppose two sources have equal entropies but one is faster than the other producing more number of symbols/unit time., In a given period, more information will be transmitted by the faster source than the other
- If the time rate at which X emits symbols is 'r_s' (symbols s), the information rate R of the source is given by,

$$\mathbf{R = r_s \ . H(X)\ b/s}$$ [(symbols / second) * (information bits/ symbol)]

where, R is the information rate

r_s is the symbol rate

H(X) is the Entropy or average information

$\boldsymbol{r_s} = \frac{1}{\tau}$ symbol/sec ; τ is the average symbol duration

$\tau = \sum_{k=1}^{q} P_k \tau_k$ seconds/symbol ; τ_k is the duration of k^{th} symbol

Numerical Problem

- **An event has 5 possible outcomes with probabilities 0.5, 0.25, 0.125, 0.0625 and 0.0625. Find the entropy of the system and also find the rate of information if there are 16 outcomes per second**

Soln., $H(S) = -\Sigma P(s_i) \log_2 P(s_i) = -\{(0.5 \log_2 0.5) + (0.25 \log_2 0.25) + (0.125 \log_2 0.125) + 2 \times (0.0625 \log_2 0.0625)\} =$ **1.875 bits/symbol**

r = 16

$R = 16 \times 1.875 \approx$ **30 bits/second**

Numerical Problem

- **A continuous signal is bandlimited to 5kHz. The signal is quantized in 8 levels of a PCM system with probabilities 0.25, 0.2, 0.2, 0.1, 0.1, 0.05, 0.05, 0.05 . Calculate the entropy and rate of information**

Soln.,

H(S) = - Σ P(s_i) log_2P(s_i) = - { (0.25 log_2 0.25) + 2 x (0.2 log_2 0.2) + 2 x (0.1 log_2 0.1) + 3 x (0.05 log_2 0.05) } = **2.7412 bits/symbol**

r = f_s = 10000 bits

R = 10000 x 2.7412 = **27412 bits/second**

Extension of Discrete Memoryless Source (or) Zero Memory Source

- It is useful to consider blocks rather than individual symbols, with each block consisting of 'n' successive source symbols
- Each such block is being produced by an extended source with a source alphabet 's^n' that has 'k^n' distinct block where, 'k' is the number of symbols in the source alphabet of the original source
- In the case of discrete memoryless source, the source symbols are statistically independent. Hence, the probability of source symbol in 's^n' in the product of the probabilities 'n' source symbols in 's' constituting the particular source symbol in 's^n'

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

Numerical Problem (Contd.,)

- **Consider a discrete memoryless source with source alphabet, $\{S_0, S_1, S_2\}$ with {P} = { 0.25, 0.25, 0.5 }**

Prove that, H(S^2) = 2 H(S)

Soln., k = 3 , n = 2

Extended source consists of 3^2 = 9 symbols

Blocks	S_0S_0	S_0S_1	S_0S_2	S_1S_1	S_1S_2	S_1S_0	S_2S_2	S_2S_1	S_2S_0
Probability	0.0625	0.0625	0.125	0.0625	0.125	0.0625	0.25	0.125	0.125

H(S) =- Σ P(s_i) log_2P(s_i) = - { (0.25 log_2 0.5) + (0.25 log_2 0.25) + (0.5 log_2 0.5) = **1.5**

2 x H(S) = **3**

H(S^2) = - Σ P(s_i) log_2P(s_i) = - { 4 x (0.0625 log_2 0.5) + (0.25 log_2 0.25) + 4 x (0.125 log_2 0.125) } = **3**

Hence Proved, **H(S^2) = 2 H(S)**

Encoding

- Encoding is the procedure for associating words constructed from a finite alphabet of a language with given words of another language in a one to one manner
- **Classification of Codes:**
- **Fixed Length Codes:**

  A fixed length code is defined as a code whose word length is fixed

  E.g.: $\{S_0, S_1, S_2, S_3\}$ – {00, 01, 10, 11}
- **Variable Length Codes:**

  A fixed length code is defined as a code whose word length is not fixed

  E.g.: $\{S_0, S_1, S_2, S_3\}$ – {0, 01, 110, 111}

Classification of Codes

- **Distinct Codes:**

  A distinct code is defined as the one in which each codeword is distinguishable from each other

  E.g.: $\{S_0, S_1, S_2, S_3\}$ – {0, 1, 00, 11}

- **Uniquely Decipherable Encoding:**

  A code is said to be uniquely decipherable if any sequence of code word can be interpreted in only one way

  E.g.: $\{S_0, S_1, S_2, S_3\}$ – {0, 01, 010, 101} – Not Uniquely Decipherable

  $\{S_0, S_1, S_2, S_3\}$ – {0, 10, 110, 111} – Uniquely Decipherable

Classification of Codes

- **Prefix Free Codes:**

  A code in which no codeword can be formed by adding code symbols to another codeword is called a prefix free code. In a Prefix-free code no codeword is a prefix of another

  E.g.: $\{S_0, S_1, S_2, S_3\}$ – {0, 1, 10, 11} – Not Prefix Code

  E.g.: $\{S_0, S_1, S_2, S_3\}$ – {0, 10, 110, 111} – Prefix Code

- **Non Singular:**

  A block code is said to be non-singular, if all the code words of the word set are distinct

  E.g.: $\{S_0, S_1, S_2, S_3\}$ – {0, 01, 10, 11}

Classification of Codes

- **Instantaneous Code:**

  A code word having a property that no codeword is a prefix of another codeword is said to be instantaneous

  E.g.: $\{S_0, S_1, S_2, S_3\}$ – {0, 01, 011, 0111} – Not Instantaneous

  $\{S_0, S_1, S_2, S_3\}$ – {0, 100, 101, 11} – Instantaneous

- **Optimal Code:**

  An instantaneous code is said to be optimal if it has minimum average length for a source with the given probability of assignment for the source symbol

Code Length

- **Codeword Length:**
  - Let X be a DMS with finite entropy H(X) and an alphabet $\{x_1, x_2, \ldots\ldots, x_m\}$ with corresponding probabilities of occurrence P(x_i) (i = 1, …. , m)
  - Let the binary code word assigned to symbol xi by the encoder have length n_i, measured in bits
  - The length of the code word is the number of binary digits in the code word

Average Codeword Length

- **Average Codeword Length:**
  - The average code word length L, per source symbol is given by,

$$L = \sum_{i=1}^{k} p_i \cdot n_i$$

  - The parameter L represents the average number of bits per source symbol used in the source coding process

Code Efficiency and Redundancy

- **Code Efficiency:**

  Efficiency is defined as the ratio of the average information per symbol of encoded language to the maximum possible average information per symbol denoted by,

$$\eta = \frac{H(S)}{L \, .log_2 \, r} ; \qquad \text{If r = 2 then., } \eta = \frac{H(S)}{L}$$

  where, H(S) - Entropy

  L - Average Length

- **Redundancy:**

  Redundancy = 1 – Efficiency

Numerical Problem

- **Let us consider a source having four messages, S = $\{s_0, s_1, s_2, s_3\}$ – {0, 10, 110, 111} with probabilities 0.5, 0.25, 0.125, 0.125 . Calculate the efficiency and redundancy of the code.**

Soln.,

Symbols	Prob.	Code
s_0	0.5	0
s_1	0.25	10
s_2	0.125	110
s_3	0.125	111

- H(S) = - Σ P(s_i) log_2P(s_i) = - { (0.5 log_2 0.5) + (0.25 log_2 0.25) + 2 x (0.125 log_2 0.125)} = 1.75 bits/symbol
- L = { (1 x 0.5) + (2 x 0.25) + 2 x (3 x 0.125) } = 1.75 bits
- Efficiency = H(S)/L = 100%
- Redundancy = 0

Numerical Problem

- **Let us consider a source having four messages, S = $\{s_0, s_1, s_2, s_3\}$ – {0, 10, 110, 111} with probabilities 1/3, 1/3, 1/6, 1/6. Calculate the efficiency and redundancy of the code.**

Soln.,

Symbols	Prob.	Code
s_0	1/3	0
s_1	1/3	10
s_2	1/6	110
s_3	1/6	111

- H(S) = - Σ P(s_i) log_2P(s_i) = - { 2 x (1/3 log_2 1/3) + 2 x (1/6 log_2 1/6)

  = 1.918 bits/symbol
- L = { (1 x 1/3) + (2 x 1/3) + 2 x (3 x 1/6) } = 2 bits
- Efficiency = H(S)/L = 95.9%
- Redundancy = 4.1%

Numerical Problem

- The output of a discrete source is given by, {x} = {$x_1, x_2, \ldots, x_6$} with probabilities {P} = {$2^{-1}, 2^{-2}, 2^{-4}, 2^{-4}, 2^{-4}, 2^{-4}$ }is encoded in the following ways:

(i) Determine which of these codes are uniquely decodable?

(ii) Determine which of these codes have prefix property?

(iii) Find average length of each uniquely decodable code?

	C_1	C_2	C_3	C_4	C_5	C_6
x_1	0 (1)	1 (1)	0 (1)	111 (3)	1 (1)	0 (1)
x_2	10 (2)	011 (3)	10 (2)	110 (3)	01 (2)	01 (2)
x_3	110 (3)	010 (3)	110 (3)	101 (3)	0011 (4)	011 (3)
x_4	1110 (4)	001 (3)	1110 (4)	100 (3)	0010 (4)	0111 (4)
x_5	1011 (4)	000 (3)	11110 (5)	011 (3)	0001 (4)	01111 (5)
x_6	1101 (4)	110 (3)	111110 (6)	010 (3)	0000 (4)	011111 (6)

Numerical Problem (Contd.,)

	C_1	C_2	C_3	C_4	C_5	C_6
x_1	0 (1)	1 (1)	0 (1)	111 (3)	1 (1)	0 (1)
x_2	10 (2)	011 (3)	10 (2)	110 (3)	01 (2)	01 (2)
x_3	110 (3)	010 (3)	110 (3)	101 (3)	0011 (4)	011 (3)
x_4	1110 (4)	001 (3)	1110 (4)	100 (3)	0010 (4)	0111 (4)
x_5	1011 (4)	000 (3)	11110 (5)	011 (3)	0001 (4)	01111 (5)
x_6	1101 (4)	110 (3)	111110 (6)	010 (3)	0000 (4)	011111 (6)
Instantaneous	No	No	Yes	Yes	Yes	No
Uniquely Decodable	No	No	Yes	Yes	Yes	Yes

Average Length., $L_3 = (1x2^{-1})+(2x2^{-2})+(3x2^{-4})+(4x2^{-4})+(5x2^{-4})+(6x2^{-4})$ = **2.2125 bits/symbol**

$L_4 = (3x2^{-1})+(3x2^{-2})+(3x2^{-4})+(3x2^{-4})+(3x2^{-4})+(3x2^{-4})$ = **3 bits/symbol**

$L_5 = (1x2^{-1})+(2x2^{-2})+(4x2^{-4})+(4x2^{-4})+(4x2^{-4})+(4x2^{-4})$ = **2 bits/symbol**

$L_6 = (1x2^{-1})+(2x2^{-2})+(3x2^{-4})+(4x2^{-4})+(5x2^{-4})+(6x2^{-4})$ = **2.2125 bits/symbol**

Kraft Inequality

- Given a source {S} = $\{S_1, S_2, \ldots\ldots, S_q\}$. Let the word length of the codes corresponding to these symbols be $\{l_1, l_2, \ldots\ldots, l_q\}$ and let the code alphabet {x} = $\{x_1, x_2, \ldots\ldots, x_q\}$ then an instantaneous code for the source exists if and only if,

$$\boxed{\sum_{k=1}^{q} r^{-l_k} \leqslant 1} \qquad \textbf{[1]}$$

Proof: Let us assume that the word length be arranged in ascending order such that,

$$l_1 \leqslant l_2 \leqslant \cdots\cdots\cdots \leqslant l_q \qquad \textbf{[2]}$$

Since, the code alphabet has only ‘r’ symbols, we can have at most ‘r’ instantaneously decodable sequence of length ‘1’ so as to satisfy the prefix property

Kraft Inequality

- Let n_k be the actual number of messages encoded into the codeword of length 'k', then

$$n_1 \leqslant \text{r} \qquad \textbf{[3]}$$

- The number of actual instantaneous codes of word length 2 must obey the rule,

$$n_2 \leq (\text{r}\text{-}n_1) \text{ . r} \qquad \text{i.e.,} \qquad n_2 \leq r^2\text{-}n_1 \qquad \textbf{[4]}$$

- As the first symbol can only be (r-n_1) symbols that are not used in forming the code words of length 1 and second symbol of sequence can be any one of the 'l' code alphabet symbols

Kraft Inequality

- Similarly, the actual number of codes of length 3 that are distinguishable from each other and from n_1 and n_2 words must obey,

$$n_3 \leqslant ((\mathrm{r}\text{-}n_1)\ \mathrm{r} - n_2)\ \mathrm{r}$$

$$n_3 \leqslant r^3 - n_2\, r^2 - n_1\ \mathrm{r} \qquad \textbf{[5]}$$

- The first two symbols may be selected in (r-n_1) r - n_2 ways and the third symbol element in ‘r’ ways, then we can write,

$$n_k \leqslant r^k - n_1 r^{k-1} - n_2 r^{k-2}\ , \ldots\ldots\ , n_{k-1}\ \mathrm{r} \qquad \textbf{[6]}$$

Kraft Inequality

Multiply [6] by r^{-k} and rewriting,

$$n_k r^{-k} + n_{k-1} r^{-(k-1)} + \ldots\ldots + n_1 r^{-1} \leqslant 1$$

$$\sum_{j=1}^{k} n_j \, r^{-j} \leqslant 1 \qquad ; \left(\sum_{j=1}^{m} W_j \, D^{-j} \leqslant 1\right) \qquad \textbf{[7]}$$

$$\sum_{j=1}^{k} n_j \, r^{-j} = r^{-1} + \underbrace{r^{-1} + \ldots + r^{-1}}_{n_1 \text{ times}} + \underbrace{r^{-2} + \ldots + r^{-2}}_{n_2 \text{ times}} + \ldots\ldots + \underbrace{r^{-k} + r^{-k} + \ldots + r^{-k}}_{n_k \text{ times}} \; n_1 \text{ times} \qquad \textbf{[8]}$$

$$= \sum_{j=1}^{n_1} r^{-1} + \sum_{j=1}^{n_2} r^{-2} + \ldots\ldots\ldots + \sum_{j=1}^{n_k} r^{-k} \qquad \textbf{[9]}$$

$n_1 + n_2 + \ldots\ldots + n_k = \mathrm{q}$;

$$\boxed{\sum_{k=1}^{q} r^{-l_k} \leqslant 1} \qquad \textbf{[10]}$$

Kraft Inequality

- This inequality just tells us whether an instantaneous code exists or not wherein it does not show how to construct the code or it does not guarantee that any code that has word lengths satisfying the inequality to be instantaneous itself
- **A symbol code is encoded into binary code shown below. Which of these are Instantaneous?**

Source Symbol	Code A	Code B	Code C	Code D	Code E
S_1	00	0	0	0	0
S_2	01	10000	10	1000	10
S_3	10	1100	110	1110	110
S_4	110	1110	1110	111	1110
S_5	1110	1101	11110	1011	11110
S_6	1111	1111	11111	1100	1111

Soln., Using Kraft Inequality, given r=2 ;

Code A: $(3 \text{ x } 2^{-2} + 2^{-3} + 2 \text{ x} 2^{-4}) =$ **1**

Code B: $(2^{-1} + 2^{-5} + 4 \text{ x } 2^{-4}) =$ **0.78125**

Code C: $(2^{-1} + 2^{-2} + 4 \text{ x } 2^{-4}) =$ **1**

Code D: $(2^{-1} + 2^{-3} + 4 \text{ x } 2^{-4}) =$ **0.8125**

Code E: $(2^{-1} + 2^{-2} + 2^{-3} + 2 \text{ x } 2^{-4} + 2^{-5}) =$ **1.031**
Code E doesn't satisfy prefix property hence not instantaneous

Kraft Macmillan Inequality

- Kraft inequality applies to prefix codes which are special cases of uniquely decodable codes. The same inequality is necessary for uniquely decodable codes and was proved by Macmillan

- **Statement:** The Kraft Macmillan inequality states that we can construct a uniquely decodable code with word length $l_1, l_2, \ldots\ldots, l_q$ iff these lengths satisfy the condition,

$$\sum_{k=1}^{q} r^{-l_k} \leq 1 \qquad \textbf{[1]}$$

where, ‘r’ is the number of symbols in the code alphabet

Kraft Macmillan Inequality

- **Proof:** Consider the quantity,

$$\left(\sum_{k=1}^{q} r^{-l_k}\right)^n = \left(r^{-l_1} + r^{-l_2} + r^{-l_3} + \ldots\ldots + r^{-l_q}\right)^n \qquad \textbf{[2]}$$

Expanding **[2]** we will have q_n terms each of the form $r^{-l_{k1}} + r^{-l_{k2}} + \ldots\ldots + r^{-l_{kn}} = r^{-j}$; l_{k1} +

$$l_{k1} + \ldots + l_{kn} = \text{j}$$

Suppose 'l' is the maximum word length of the codes, then it follows that 'j' can be assigned some set of values from n to nl

Let 'N_j' be the number of terms of the form r^{-j}, then Eqn., **[2]** can be written as,

$$\left(\sum_{k=1}^{q} r^{-l_k}\right)^n = \sum_{\text{j=n}}^{\text{nl}} N_j\, r^{-j} \textbf{[3]}$$

Kraft Macmillan Inequality

- N_j is also the number of strings of 'n' code words that can be formed so that each string has a length of exactly 'j' symbols. If a code is uniquely decodable then, $N_j \leq r^j$, the number of distinct r-ary code sequence of length 'j'

$$\left(\sum_{k=1}^{q} r^{-l_k}\right)^n \leq \sum_{j=n}^{nl} r^j \, r^{-j} \qquad \textbf{[6]}$$

$$\leq nl - n + 1 \qquad \textbf{[7]}$$

For a long sequence,

$$\left(\sum_{k=1}^{q} r^{-l_k}\right)^n \leq nl \qquad \textbf{[8]}$$

Kraft Macmillan Inequality

Taking n^{th} root on both sides of the inequality,

$$\sum_{k=1}^{q} r^{-l_k} \leq (nl)^{1/n} \text{ ; for all 'n'} \qquad \textbf{[9]}$$

Since, x > 1 , x^n> nl ; if we take 'n' large

$$\lim_{n\to\infty} (nl)^{1/n} = 1$$

Eqn., **[9]** holds for any integer

$$\boxed{\sum_{k=1}^{q} r^{-l_k} \leq 1} \qquad \textbf{[10]}$$

Numerical Problem

- Let {X} = $\{x_1 , x_2 , \dots , x_7\}$. After encoding we get a set of messages with following lengths $n_1 = 2 , n_2 = 2 , n_3 = 3 , n_4 = 3 , n_5 = 3 , n_6 = 4 , n_7 = 5$. Length of the i^{th} code $[n_i = 2 , 2 , 3 , 3 , 3 , 4 , 5]$; $[w_i = 0 , 2 , 3 , 1 , 1 , 0 , 0]$. Prove that $\sum_{i=1}^{N} D^{-n_i} = \sum_{j=1}^{m} W_j D^{-j}$

Soln.,

LHS: $\sum_{i=1}^{N} D^{-n_i} = D^{-2} + D^{-2} + D^{-3} + D^{-3} + D^{-3} + D^{-4} + D^{-5}$

$= 2 D^{-2} + 3 D^{-3} + D^{-4} + D^{-5}$ **[1]**

RHS: $\sum_{j=1}^{m} W_j D^{-j} = 2 D^{-2} + 3 D^{-3} + D^{-4} + D^{-5}$ **[2]**

LHS = RHS

Numerical Problem

- **Find the smallest number of letters in the alphabet 'D' for dividing a code with a prefix property such that [w] = [0,3,0,5] . Devise such a code**

Soln., $\sum_{j=1}^{m} W_j D^{-j} \leqslant 1$

Therefore., $0\ D^{-1} + 3\ D^{-2} + 0\ D^{-3} + 5\ D^{-4} \leqslant 1$

$$3\ D^{-2} + 5\ D^{-4} \leqslant 1$$

$$\frac{3}{D^2} + \frac{5}{D^4} \leqslant 1$$

$$3\ D^2 + 5 \leqslant D^4$$

$$D^4 - 3\ D^2 + 5 \geqslant 0$$

$$(t - 4.19)\ (t + 1.19) \geqslant 0$$

$$D^2 - 4.19 \geqslant 0$$

$$\mathbf{D \geqslant 2.04}$$

Numerical Problem

When D = 3; $\{X\} = \{0,1,2\}$

Total number of codes = 0 + 3 + 0 + 5 = 8

Number of codes with length :

1 = 0 ; 2 = 3 ; 3 = 0 ; 4 = 5

Conditions to devise the code: (i) D = 3

(ii) 3 codes with length 2 ; 5 codes with length 4

(iii) Prefix Property

One such way of constructing code:

m_1 = 00 ; m_2 = 01 ; m_3 = 10 ; m_4 = 2000

m_5 = 2010 ; m_6 = 2200 ; m_7 = 2202 ; m_8 = 2222

Numerical Problem

- **Show all possible sets of binary codes with prefix property for encoding the messages m_1, m_2, m_3 in words not more than 3 digits long.**

Soln.,

- $\sum_{j=1}^{m} W_j D^{-j} \leqslant 1$
- D = 2
- $W_1 2^{-1} + W_2 2^{-2} + W_3 2^{-3} \leq 1$
- $W_1 + W_2 + W_3 = 3$

- Possible Sets.,

W_1	W_2	W_3
1	1	1
1	2	0
1	0	2
0	3	0
0	0	3
0	2	1
0	1	2

Shannon's source coding Theorem (or) Noiseless Coding Theorem

- In Information Theory, Shannon's noiseless coding theorem places an upper and lower limit on the minimum possible expected length of the code words as a function of entropy of the input word and size of the code alphabet
- **Statement:** Let 'S' be the zero memory source with 'q' symbols, $\{S\} = \{s_1, s_2, \ldots, s_q\}$ and symbol probabilities $\{P\} = \{p_1, p_2, \ldots, p_q\}$ respectively. If 'S' ensemble is encoded in a sequence of uniquely decodable characters taken from the code alphabet of 'r' symbols, then

$$\frac{H(S)}{\log r} \leqslant L < \frac{H(S)}{\log r} + 1 \qquad [1]$$

Shannon's source coding Theorem (or) Noiseless Coding Theorem

- **Proof:** Consider a zero memory source with 'q' symbols, {S} = $\{s_1, s_2, \ldots, s_q\}$ and symbol probabilities {P} = $\{p_1, p_2, \ldots, p_q\}$ respectively. Let us encode the symbols into r-ary codes with word lengths $l_1, l_2, \ldots, l_q$, we shall find lower bound for average length of the codewords

Let $Q_1, Q_2, \ldots, Q_q$ be any set of numbers such that $Q_k \geq 0$ and $\sum_{k=1}^{q} Q_k = 1$ **[2]**

Consider the quantity, $$H(S) - \sum_{k=1}^{q} p_k \log \left(\frac{1}{Q_k} \right) \quad \textbf{[3]}$$

$$= \sum_{k=1}^{q} p_k \log \left(\frac{1}{p_k} \right) - \sum_{k=1}^{q} p_k \log \left(\frac{1}{Q_k} \right) = \boldsymbol{\sum_{k=1}^{q} p_k \log \left(\frac{Q_k}{p_k} \right)} \quad \textbf{[4]}$$

Shannon's source coding Theorem (or) Noiseless Coding Theorem

- Changing the base and applying log inequality, ($\log_b X = \log_b r \cdot \log_r X$; $\ln X \leqslant X-1$)

$$=>\log_2 e \sum_{k=1}^{q} p_k \ln\left(\frac{Q_k}{p_k}\right) \quad \textbf{[5]}$$

$$\leqslant \log_2 e \left(\sum_{k=1}^{q} p_k \ln\left(\frac{Q_k}{p_k} - 1\right)\right) \quad \textbf{[6]}$$

$$\leqslant \log_2 e \left[\underbrace{\sum_{k=1}^{q} Q_k}_{\mathbf{1}} - \underbrace{\sum_{k=1}^{q} P_k}_{\mathbf{1}}\right] \quad => \leqslant \mathbf{0} \quad \textbf{[7]}$$

$$H(S) - \sum_{k=1}^{q} P_k \log\left(\frac{1}{Q_k}\right) \leqslant \mathbf{0} \quad \textbf{[8]}$$

The equality holds iff, $Q_k = P_k$

Eqn., **[8]** is valid for any set of numbers Q_k thatare non-negative and sum to unity. We may choose,

$$Q_k = \frac{r^{-l_k}}{\sum_{k=1}^{q} r^{-l_k}} \quad \textbf{[9]}$$

Shannon's source coding Theorem (or) Noiseless Coding Theorem

Applying Eqn., **[9]** in **[8]**,

$$H(S) \leqslant \sum_{k=1}^{q} P_k \log (r^{l_k} \sum_{k=1}^{q} r^{-l_k}) \quad \textbf{[10]}$$

$$H(S) \leqslant \sum_{k=1}^{q} P_k [l_k \log r + \log \sum_{k=1}^{q} r^{-l_k}] \quad \textbf{[11]}$$

$$H(S) \leqslant \log r \sum_{k=1}^{q} P_k\, l_k + \log \sum_{k=1}^{q} r^{-l_k} \quad \textbf{[12]}$$

Consider, $L = \sum_{k=1}^{q} P_k\, l_k$ =>Average codeword length

$$H(S) \leqslant L \log r + \log \sum_{k=1}^{q} r^{-l_k} \quad \textbf{[13]}$$

Second term in Eqn., **[13]** is either negative or almost zero, from Kraft's inequality

$$H(S) \leqslant L \log r \quad \textbf{[14]}$$

$$\boxed{L \geqslant \frac{H(S)}{\log r}} \quad \textbf{[15]}$$

Shannon's source coding Theorem (or) Noiseless Coding Theorem

- **Code Efficiency Derivation:**

Lower bound is possible when, (i) $\sum_{k=1}^{q} r^{-l_k} = 1$; (ii) $P_k = r^{-l_k}$ **[16]**

For equality, we choose, $l_k = \log_r \left(\frac{1}{P_k}\right)$ i.e., $\left(\log_r \left(\frac{1}{P_k}\right) = \frac{\log_2 \left(\frac{1}{P_k}\right)}{\log_2 (r)} \right)$ **[17]**

l_k must be an integer for all k-1 to q;

Eqn., **[15]** => Lower bound on L, average word length of code expressed as a fraction of code bit/ source symbol

- We know that each codewords will have integer number of code bits. There is a problem where in we need to find what to select for the value of l_k , the number of code symbols in codeword 'k' corresponding to a source symbol 'S_k' when the quantity in Eqn., **[17]** is not an integer

Shannon's source coding Theorem (or) Noiseless Coding Theorem

- Suppose we choose l_k to be next nearest integer value to be greater than $\log_r\left(\frac{1}{P_k}\right)$,

$$\log_r\left(\frac{1}{P_k}\right) \leqslant l_k \leqslant \log_r\left(\frac{1}{P_k}\right) + 1 \qquad \textbf{[18]}$$

Eqn., **[18]** satisfies Kraft's inequality

$$\frac{1}{P_k} \leqslant r^{l_k} \Rightarrow P_k \geqslant r^{-l_k} \qquad \textbf{[19]}$$

$$\sum_{k=1}^{q} P_k = 1 \geqslant \sum_{k=1}^{q} r^{-l_k} \qquad \textbf{[20]}$$

We know that,

$$\log_r\left(\frac{1}{P_k}\right) = \frac{\log_2\left(\frac{1}{P_k}\right)}{\log_2(r)}$$

Eqn., **[18]** =>

$$\frac{\log\frac{1}{P_k}}{\log r} \leqslant l_k < \frac{\log\frac{1}{P_k}}{\log r} + 1 \qquad \textbf{[21]}$$

Shannon's source coding Theorem (or) Noiseless Coding Theorem

Multiplying Eqn., **[21]** by P_k and summing for all values 'k'

$$\frac{\sum_{k=1}^{q} P_k \log\left(\frac{1}{P_k}\right)}{\log r} \leqslant \sum_{k=1}^{q} p_k\, l_k < \frac{\sum_{k=1}^{q} P_k \log\left(\frac{1}{P_k}\right)}{\log r} + \sum_{k=1}^{q} P_k \qquad \textbf{[22]}$$

$$\boxed{\frac{H(S)}{\log r} \leqslant L < \frac{H(S)}{\log r} + 1} \qquad \textbf{[23]}$$

For binary codes, r=2,

$$\boxed{\mathbf{H(S) \leq L < H(S) + 1}} \qquad \textbf{[24]}$$

Shannon's source coding Theorem (or) Noiseless Coding Theorem

- To obtain better efficiency, we can use the n^{th} extension of source 'S'

Eqn., **[23]** is valid for any zero memory source. It is also valid for S^n ,

$$\textbf{[23]} \Rightarrow \frac{H(S^n)}{log\ r} \leqslant L < \frac{H(S^n)}{log\ r} + 1 \qquad \textbf{[25]}$$

We know that, $H(S^n) = n\ .\ H(S)$

$$\Rightarrow \frac{n\ .\ H(S^n)}{log\ r} \leqslant L < \frac{n\ .\ H(S^n)}{log\ r} + 1$$

$$\Rightarrow \quad \boxed{\frac{\boldsymbol{H(S)}}{\boldsymbol{log\ r}} \leq \frac{\boldsymbol{L}}{\mathbf{n}} < \frac{\mathbf{H(S)}}{\mathbf{log\ r}} + \frac{\mathbf{1}}{\mathbf{n}}} \qquad \textbf{[26]}$$

For binary codes, $$H(S) \leq \frac{L}{\mathrm{n}} < H(S) + \frac{1}{\mathrm{n}} \qquad \textbf{[27]}$$

$$\lim_{n\to\infty} \frac{L_n}{n} = \frac{H(S)}{\log r}$$ (Lower and Upper bounds converge)

Source Coding Techniques

- **Source Coding Techniques:**

1. Shannon Encoding
2. Shannon Fano Encoding
3. Huffman's Encoding
4. Arithmetic Coding
5. Run-Length Encoding
6. Lempel-Ziv Encoding and Decoding

Shannon Encoding Procedure

- **STEP 1:** List the source symbols $\{S\} = \{s_1, s_2, \dots, s_q\}$ in the order of decreasing probability $\{P\} = \{p_1, p_2, p_3, \dots, p_q\}$ of occurrence such that $p_1 > p_2 > \cdots > p_q$
- **STEP 2:** Compute the sequence,

$$\propto_1 = 0$$

$$\propto_2 = p_1$$

$$\propto_3 = p_1 + p_2$$

$$\propto_{q+1} = p_q + p_{q-1} + \dots\dots + p_1$$

- **STEP 3:** Determine the set of integers 'l_k' which are the smallest integer solution of the inequality

$$2^{l_k} \cdot p_k \geqslant 1 \text{ , k} = 1, 2, \dots\dots, \text{q}$$

- **STEP 4:** Expand the decimal number of $\propto$ in binary form to l_k places and neglect the expansion beyond l_k digits
- **STEP 5:** Removal of decimal points results in desired code

Numerical Problem on Shannon Encoding Procedure

- **Consider the following ensemble {S} = {$s_1, s_2, \dots, s_4$} with {P} = {0.4, 0.3, 0.2, 0.1}. Encode the symbols using Shannon's binary encoding procedure and calculate the efficiency and redundancy of the code**

Soln.,

- Arrange the probabilities, $\boldsymbol{p_1 > p_2 > p_3 > p_4}$ => **0.4 > 0.3 > 0.2 > 0.1**
- Compute the sequences $\propto$,

$$\propto_1 = 0$$
$$\propto_2 = p_1 = 0.4$$
$$\propto_3 = p_1 + P_2 = 0.7$$
$$\propto_4 = p_1 + p_2 + p_3 = 0.9$$
$$\propto_5 = p_1 + p_2 + p_3 + p_4 = 1.0$$

- Find 'l_k' => $2^{l_k} \cdot p_k \geq 1$

$$2^{l_1} \cdot p_1 \geq 1 \; ; \; l_1 \geq 1.321 \Rightarrow l_1 = 2$$
$$2^{l_2} \cdot p_2 \geq 1 \; ; \; l_2 \geq 1.730 \Rightarrow l_2 = 2$$
$$2^{l_3} \cdot p_3 \geq 1 \; ; \; l_3 \geq 2.320 \Rightarrow l_3 = 3$$
$$2^{l_1} \cdot p_4 \geq 1 \; ; \; l_4 \geq 3.320 \Rightarrow l_4 = 4$$

Numerical Problem on Shannon Encoding Procedure

- Expand '$\propto$' into binary form,

$$\propto_1 = 0 = (0.0000)_2$$
$$\propto_2 = 0.4 = (0.0110)_2$$
$$\propto_3 = 0.7 = (0.10110)_2$$
$$\propto_4 = 0.9 = (0.11100)_2$$
$$\propto_5 = 1.0 = (1.0)_2$$

- Removal of decimal point,

Symbol	Code	Length
S_1	00	2
S_2	01	2
S_3	101	3
S_4	1110	4

- H(S) = - Σ P(s_i) log_2 P(s_i)
  = - { (0.4 log_2 0.4) + (0.3 log_2 0.3) + (0.2 log_2 0.2) + (0.1 log_2 0.1) }
  = 1.845 bits/symbol
- L = Σ $\{p_k \, . \, l_k\}$ = { (2 x 0.4) + (2 x 0.3) + (3 x 0.2) + (4 x 0.1) }
  = 2.4 bits
- Efficiency = $\frac{H(S)}{L}$ = 76%
- Redundancy = 24%

Shannon Fano Encoding Procedure

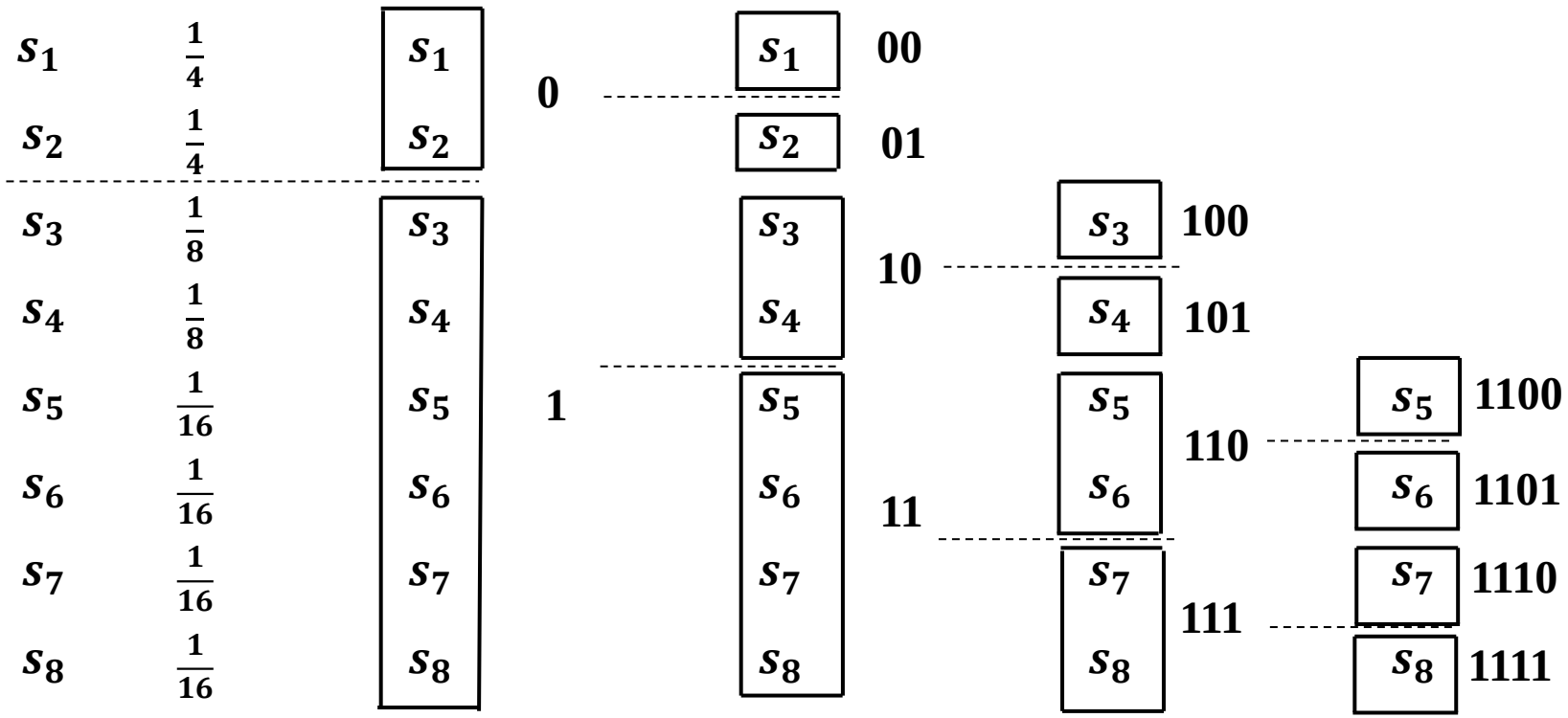
- **STEP 1:** List the source symbols in the order of decreasing probabilities
- **STEP 2:** Partition this ensemble into almost two equiprobable groups (‘r’ groups for r-ary coding) for binary coding
- **STEP 3:** Assign ‘0’ to one group and ‘1’ to the other group (assign a code symbol each to each group respectively, from code alphabet). This forms the stating code symbols of the codes
- **STEP 4:** Repeat steps 2 & 3 on each of the sub-groups until the sub-groups contain only one source symbol to determine the succeeding code symbols of the code word

Problem on Shannon Fano Encoding

- Consider the message ensemble {S} = {$s_1, s_2, \ldots\ldots, s_8$} and {P} = {$\frac{1}{4}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}$} with {X} = {0,1}

Construct a binary code using Shannon Fano encoding procedure. Calculate η and E_c .

Soln.,



Symbols	Codes (Length)
s_1	00 (2)
s_2	01 (2)
s_3	100 (3)
s_4	101 (3)
s_5	1100 (4)
s_6	1101 (4)
s_7	1110 (4)
s_8	1111 (4)

Numerical Problem on Shannon Fano Encoding

- Entropy, H(S) = - Σ P(s_i) log_2 P(s_i)

  = -{{ (2) x (0.25 log_2 (0.25)) } + { (2) x (0.125 log_2 (0.125)) } + { (4) x (0.0625 log_2 (0.0625)) }

  = 2.75 bits/symbol

- Length, L = { {2 x (2 x 0.25)} + {2 x (3 x 0.125)} + {4 x (4 x 0.0625)} }

  = 2.75 bits/symbol

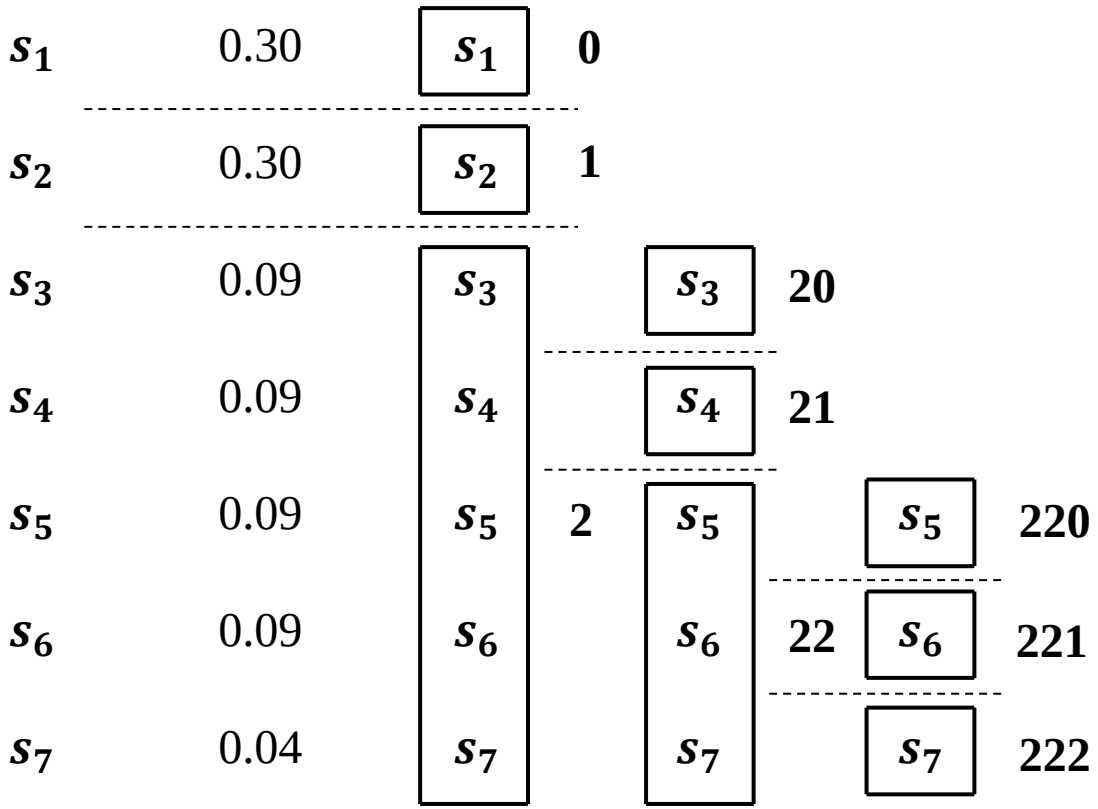
- Efficiency, η = $\frac{H(S)}{L}$ = 100 %

- Redundancy = 0

Numerical Problem

- **Construct a trinary code for symbols with {P} ={0.3,0.3,0.09,0.09,0.09,0.09,0.04} and {X} = {0,1,2} using Shannon Fano Encoding Procedure**

Soln.,



Symbol	Code	Length
s_1	0	1
s_2	1	1
s_3	20	2
s_4	21	2
s_5	220	3
s_6	221	3
s_7	222	3

- H(S) = 2.477 bits/symbol
- L = 1.62 trinits/symbol (r=3)
- Efficiency = $\frac{H(S)}{L \cdot \log r} = \frac{2.477}{1.62\ log_2\ (3)} = 96.53\%$
- Redundancy = 3.47%

Huffman Encoding Procedure

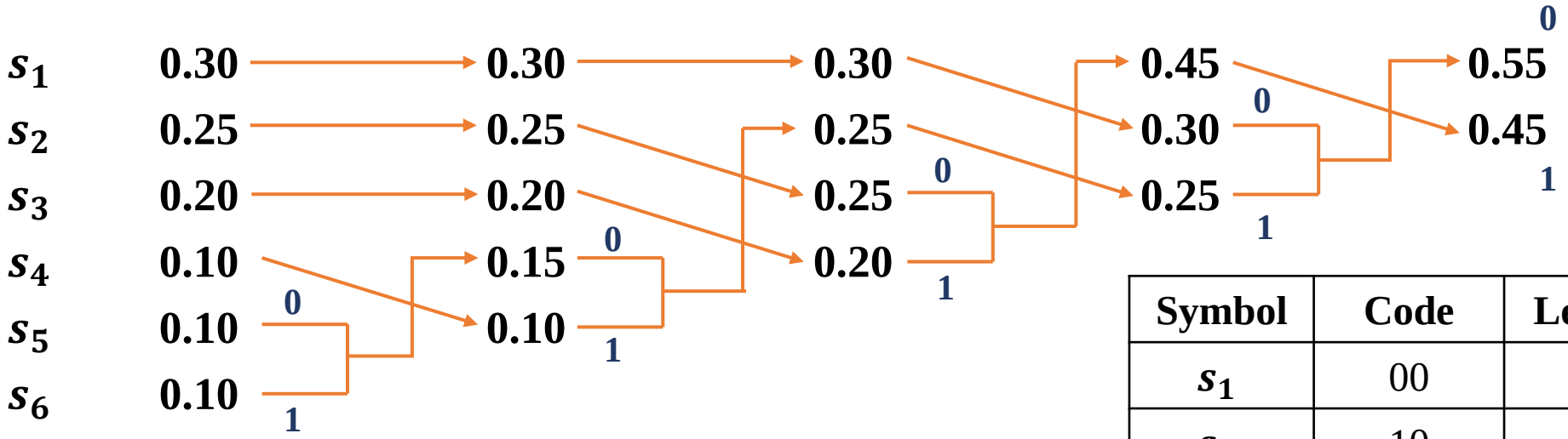
- Huffman Encoding gives the proper position for optimal code
- The code with minimum average length 'L' would be more efficient and to have minimum redundancy associated with it
- A compact code is one which achieves this objective. Huffman has suggested a simple method that guarantees an optimal code
- **Procedure:**
- **STEP 1:** List the source symbol in decreasing order of probabilities
- **STEP 2:** Check if $q=r+\propto(r-1)$ is satisfied and find the integer '$\propto$'. Otherwise add suitable number of dummy symbols of zero probability of occurrence to satisfy the equation (This step is not needed for binary codes)

Huffman Encoding Procedure

- **STEP 3:** Club the last ‘r’ symbols into a single composite symbol whose probabilities of occurrence is equal to sum of probabilities of occurrence of ‘r’ symbols involved in this step
- **STEP 4:** A new list of events is recorded again to be in the order of decreasing probability
- **STEP 5:** Repeat steps **3** and **4** on the resulting set of symbols until in the final step exactly ‘r’ symbols are left
- **STEP 6:** Assign codes to the last ‘r’ composite symbols and work backwards to the original source to arrive at the optimal code
- **STEP 7:** Discard the codes of dummy symbols (This step is not needed for binary codes)

Numerical Problem on Huffman Encoding Procedure

- Construct a Huffman code for symbols {S} = {s_1, s_2,……, s_6} and {P} = {0.3, 0.25, 0.2, 0.1, 0.1, 0.05} with {X} = {0, 1}
- **Soln.,**



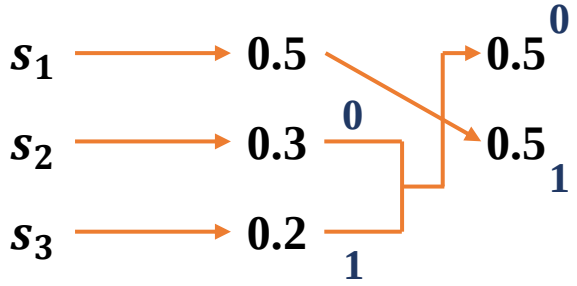
Symbol	Code	Length
s_1	00	2
s_2	10	2
s_3	11	2
s_4	011	3
s_5	0100	4
s_6	0101	4

- **H(S)** = - Σ P(s_i) log_2 P(s_i) = 2.365 bits/symbol
- **L** = 2.4 bits/symbol
- **Efficiency** = H(S)/L = 98.54 %
- **Redundancy** = 0.0146

Huffman Code – Second Order Extension

- **Apply Huffman binary encoding procedure for the given symbols{S} = {s_1, s_2, s_3} with {P} = {0.5,0.3,0.2} . Find the entropy and efficiency. If the same technique is applied for the second order extension of the source calculate its entropy and efficiency.**

Soln.,



Symbol	Code	Length
s_1	1	1
s_2	00	2
s_3	01	2

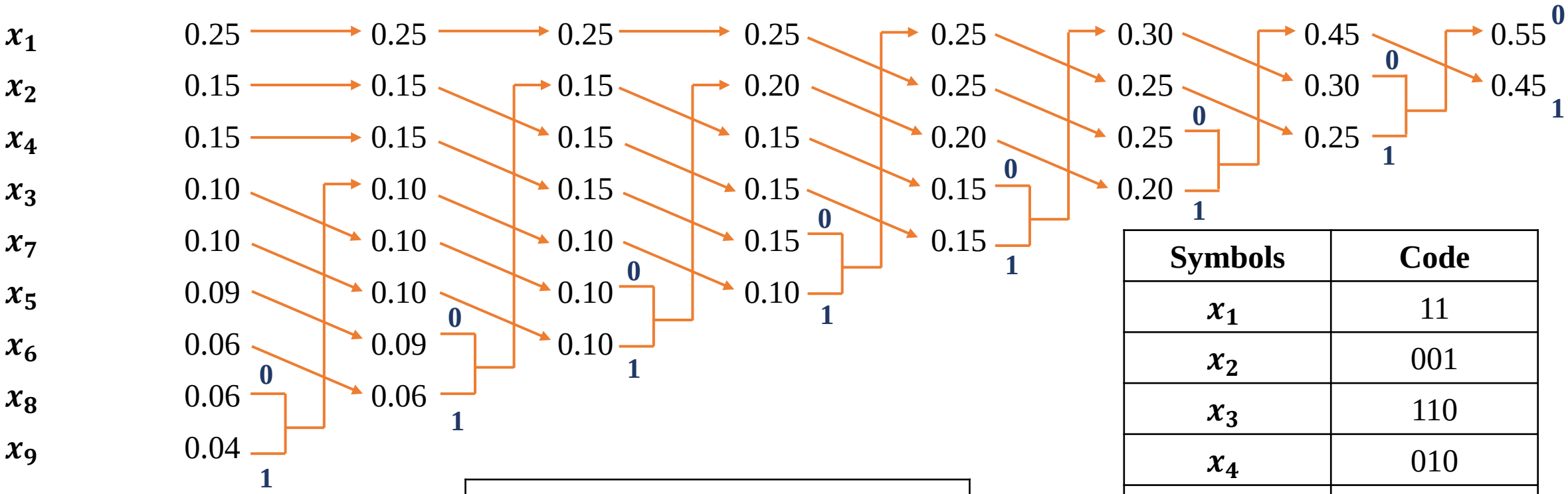
- **H(S)** = - Σ P(s_i) log_2 P(s_i) = 1.485 bits/symbol
- **L = 1.5** bits/symbol
- **Efficiency** = H(S)/L = 99 %
- **Redundancy** = 1%

Huffman Code – Second Order Extension

- Second Extension:

x_1	s_1s_1	0.25
x_2	s_1s_2	0.15
x_3	s_1s_3	0.1
x_4	s_2s_1	0.15
x_5	s_2s_2	0.09
x_6	s_2s_3	0.06
x_7	s_3s_1	0.10
x_8	s_3s_2	0.06
x_9	s_3s_3	0.04

Huffman Code – Second Order Extension



- H(S) = 2.969 bits/symbol
- L = 3 binits/symbol
- Efficiency = 98.96%
- Redundancy = 1.04%

Symbols	Code
x_1	11
x_2	001
x_3	110
x_4	010
x_5	0000
x_6	0001
x_7	111
x_8	0110
x_9	0111

Numerical Problem

- **Construct a Huffman code for the symbols {S} = $\{s_1, s_2, \ldots, s_6\}$ with {P} = $\{\frac{1}{3}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}, \frac{1}{12}, \frac{1}{12}\}$ and {X} = {0,1,2}**

Soln., Solving $q = r + \propto (r-1)$; $q = 6 ; r = 3$

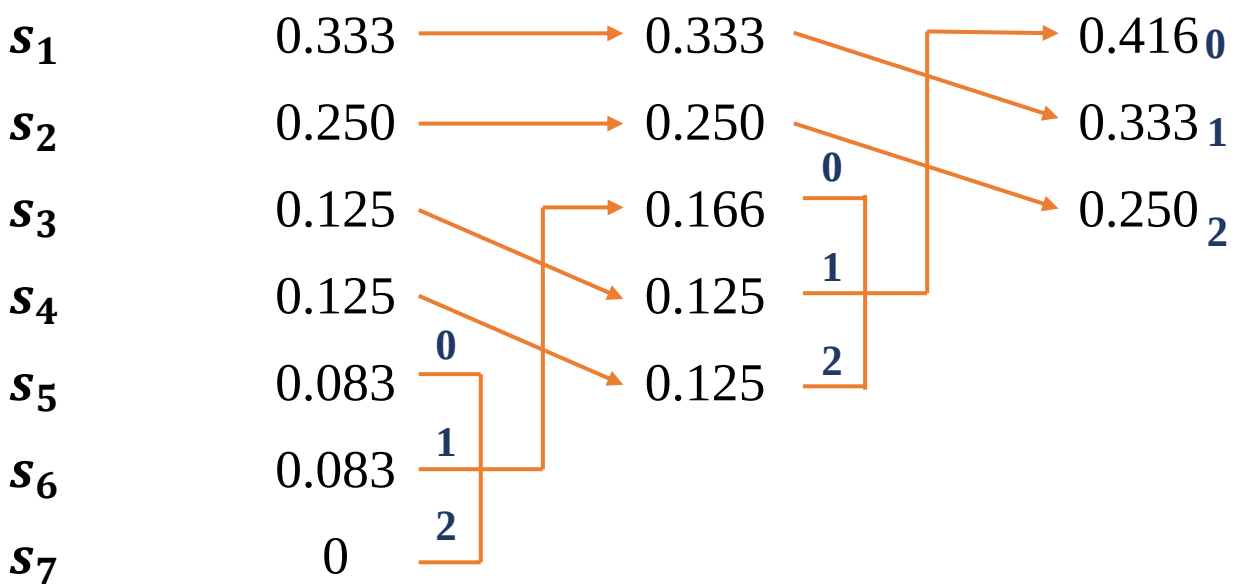
$$\propto = \frac{q-r}{r-1} = \frac{6-3}{3-1} = \frac{3}{2} \text{ (not an integer)}$$

Therefore, we add a dummy variable with probability zero

Now, $q = 7 ; r = 3$

$$\propto = \frac{q-r}{r-1} = \frac{7-3}{3-1} = \frac{4}{2} = 2 \text{ } (\propto \text{ is an integer})$$

Numerical Problem (Contd.,)



Symbol	Code	Length
s_1	1	1
s_2	2	1
s_3	01	2
s_4	02	2
s_5	000	3
s_6	001	3
s_7	002	3

- H(S) = 2.375 bits/symbol
- Length, L = 1.5833 trinits/symbol
- Efficiency = 94.54%
- Redundancy = 5.46%

Arithmetic Coding

- **Procedure:**
- **STEP 1:** Divide the given probabilities in the same order ranging from 0 to 1
- **STEP 2:** Expand the first symbol to be coded. The new range is defined by calculating its limits

$$d = \text{Upper Limit} - \text{Lower Limit of the symbol encoded}$$

$$\text{New Range (Lower Limit of each symbol)} = \text{Lower Limit} + d\ (\text{Probability of the symbol})$$

- **STEP 3:** Repeat the procedure for successive symbols until the final symbol in the given sequence is encoded
- **STEP 4:** The tag value is generated by calculating

$$\text{Tag} = (\text{Upper Limit} + \text{Lower Limit}) / 2$$

Arithmetic Coding

- **STEP 5: Decoding Process:** The tag value is used to decode the symbols assigned with its probabilities
- **STEP 6:** The probabilities of the symbols are arranged in the given format ranging from 0 to 1
- **STEP 7:** The range in which tag value is present is now formed as new range having the lower bound and upper bound. The new lower limit of each symbol within this range is calculated by

$$d = \text{Upper bound} - \text{Lower bound}$$

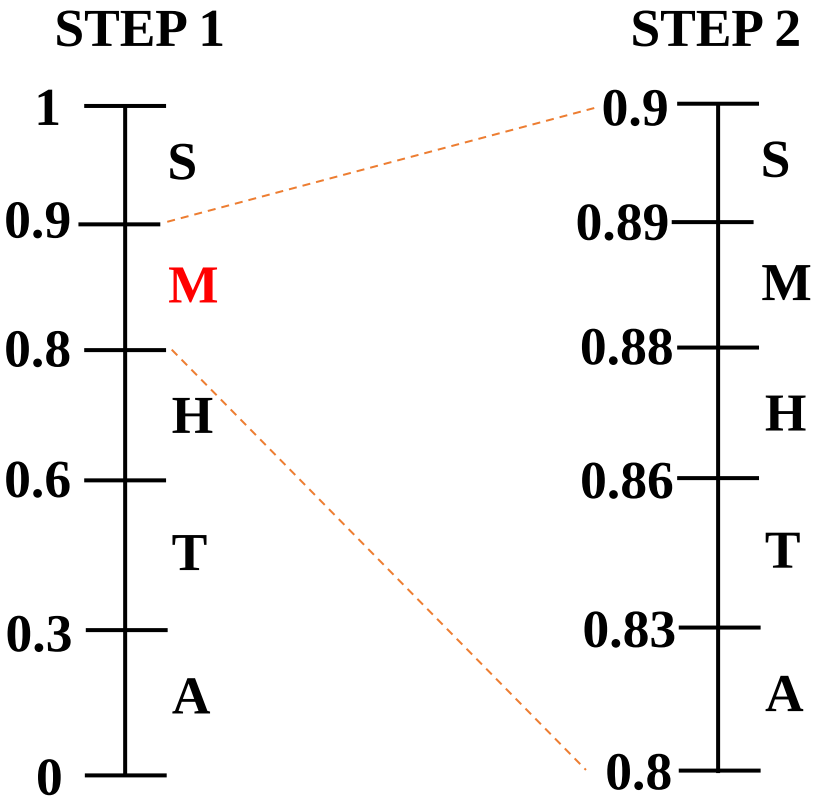
$$\text{New Range (Lower Limit of each symbol)} = \text{Lower Limit} + d\ (\text{Probability of the symbol})$$

- **STEP 8:** This procedure continues until all the symbols are decoded
- **STEP 9:** The symbol within the tag value decoded at each stage is stored as a sequence forming the final decoded data

Arithmetic Coding

- **Using Arithmetic Coding, encode the message MATHS with probabilities A = 0.3 ; T = 0.3 ; H = 0.2 ; M = 0.1 ; S = 0.1 . Generate the Tag value**

Soln.,

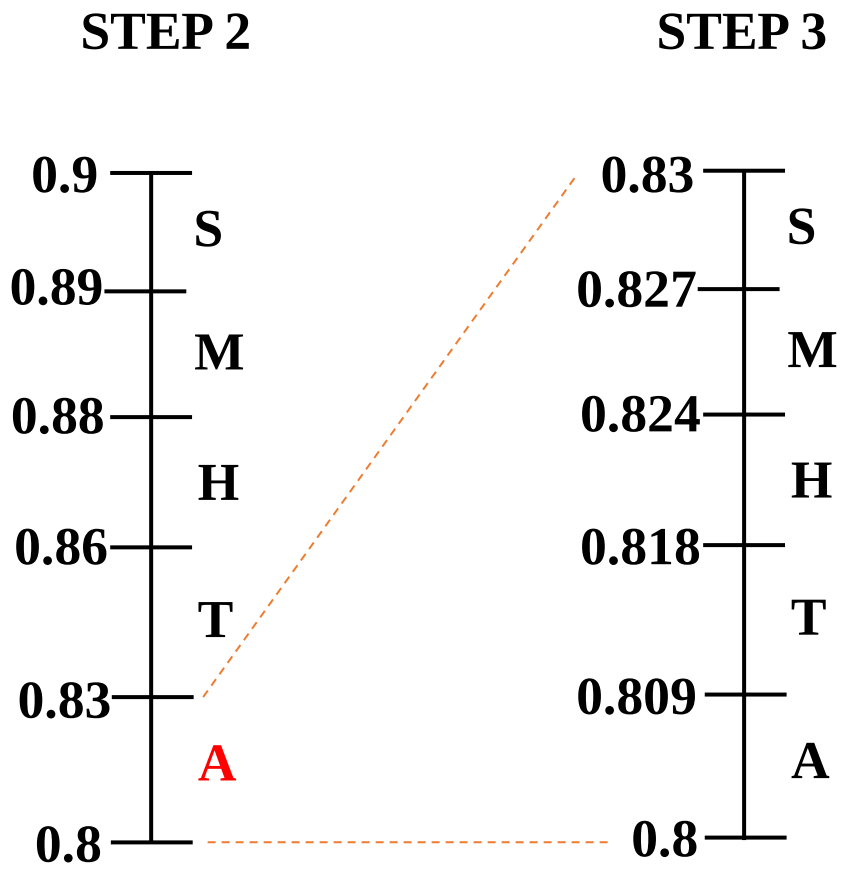


- d = Upper bound – Lower bound = 0.9 – 0.8 = 0.1
- Range of each "Symbol" =

  Lower Limit + d (Prob. Of Symbol)
- Range of "A" = 0.8 + 0.1 (0.3) = 0.83
- Range of "T" = 0.83 + 0.1 (0.3) = 0.86
- Range of "H" = 0.86 + 0.1 (0.2) = 0.88
- Range of "M" = 0.88 + 0.1 (0.1) = 0.89
- Range of "S" = 0.89 + 0.1 (0.1) = 0.9

Arithmetic Coding

Soln.,

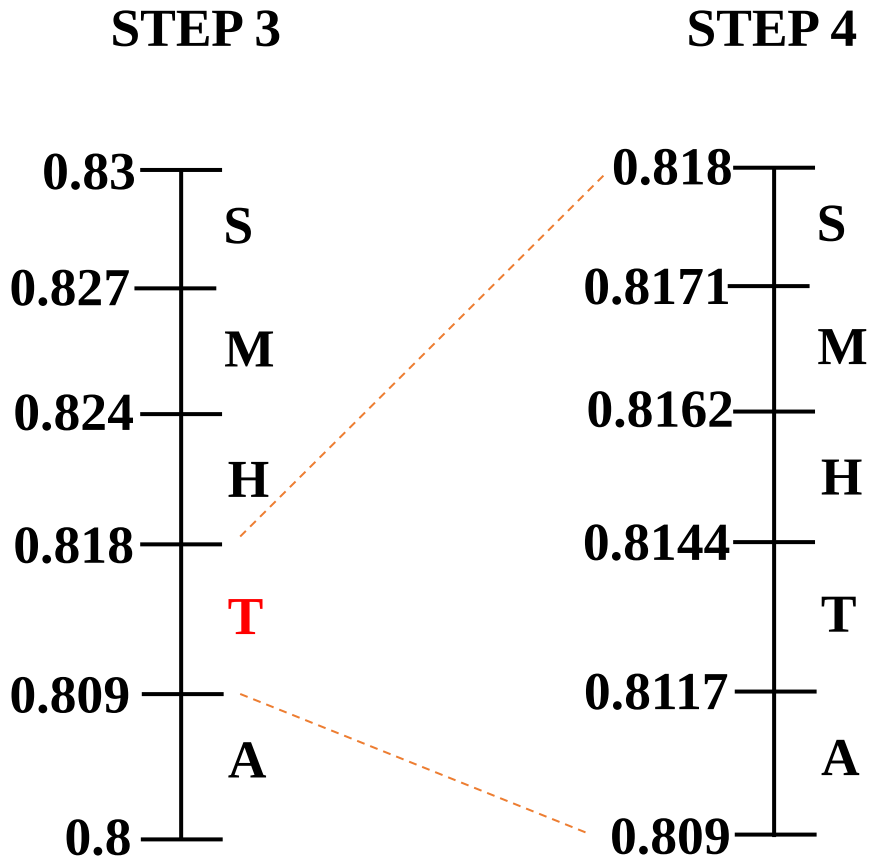


- d = Upper bound – Lower bound = 0.83 – 0.8 = 0.03
- Range of each "Symbol" =

  Lower Limit + d (Prob. Of Symbol)
- Range of "A" = 0.8 + 0.03 (0.3) = 0.809
- Range of "T" = 0.809 + 0.03 (0.3) = 0.818
- Range of "H" = 0.818 + 0.03 (0.2) = 0.824
- Range of "M" = 0.824 + 0.03 (0.1) = 0.827
- Range of "S" = 0.827 + 0.03 (0.1) = 0.83

Arithmetic Coding

Soln.,

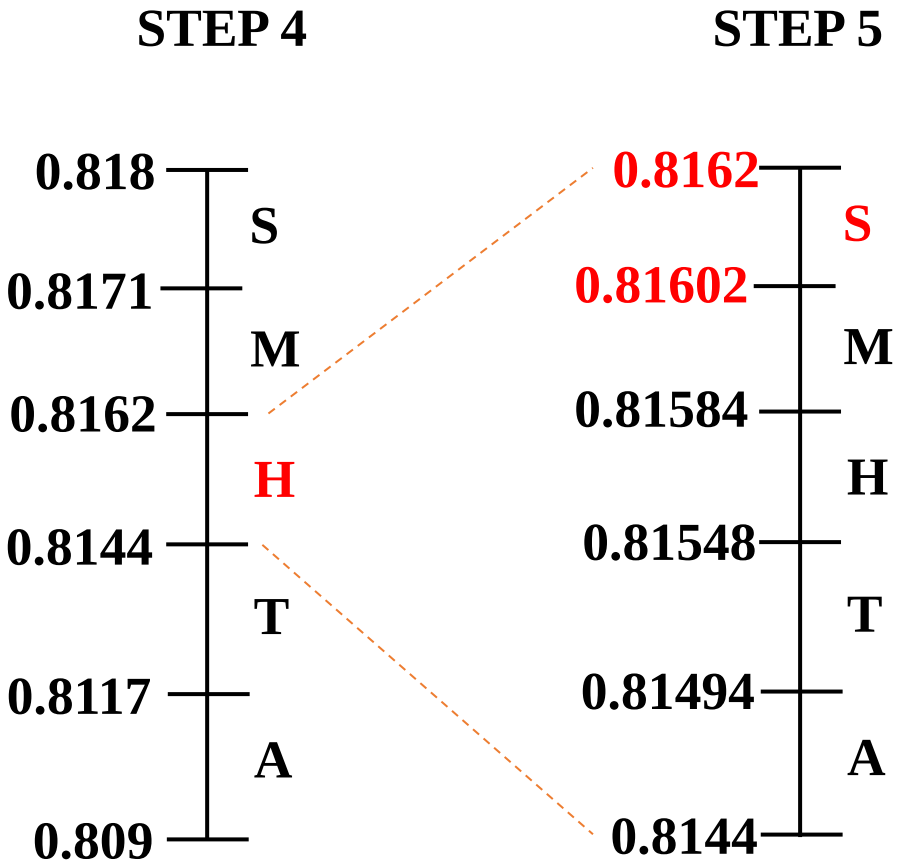


- d = Upper bound – Lower bound = 0.818 – 0.809 = 0.009
- Range of each "Symbol" =

  Lower Limit + d (Prob. Of Symbol)
- Range of "A" = 0.809 + 0.009 (0.3) = 0.8117
- Range of "T" = 0.8117 + 0.009 (0.3) = 0.8144
- Range of "H" = 0.8144 + 0.009 (0.2) = 0.8162
- Range of "M" = 0.8162 + 0.009 (0.1) = 0.8171
- Range of "S" = 0.8171 + 0.009 (0.1) = 0.818

Arithmetic Coding

Soln.,



- d = Upper bound – Lower bound = 0.8162 – 0.8144 = 0.0018
- Range of each "Symbol" =

  Lower Limit + d (Prob. Of Symbol)
- Range of "A" = 0.8144 + 0.0018 (0.3) = 0.81494
- Range of "T" = 0.81494 + 0.0018 (0.3) = 0.81548
- Range of "H" = 0.81548 + 0.0018 (0.2) = 0.81584
- Range of "M" = 0.81584 + 0.0018 (0.1) = 0.81602
- Range of "S" = 0.81602 + 0.0018 (0.1) = 0.8162

Arithmetic Coding

- The arithmetic codeword from the encoding process is obtained in the range

0.81602 < CODEWORD < 0.8162

- The TAG value is generated by

$$TAG = (\text{Upper Limit} + \text{Lower Limit}) / 2$$

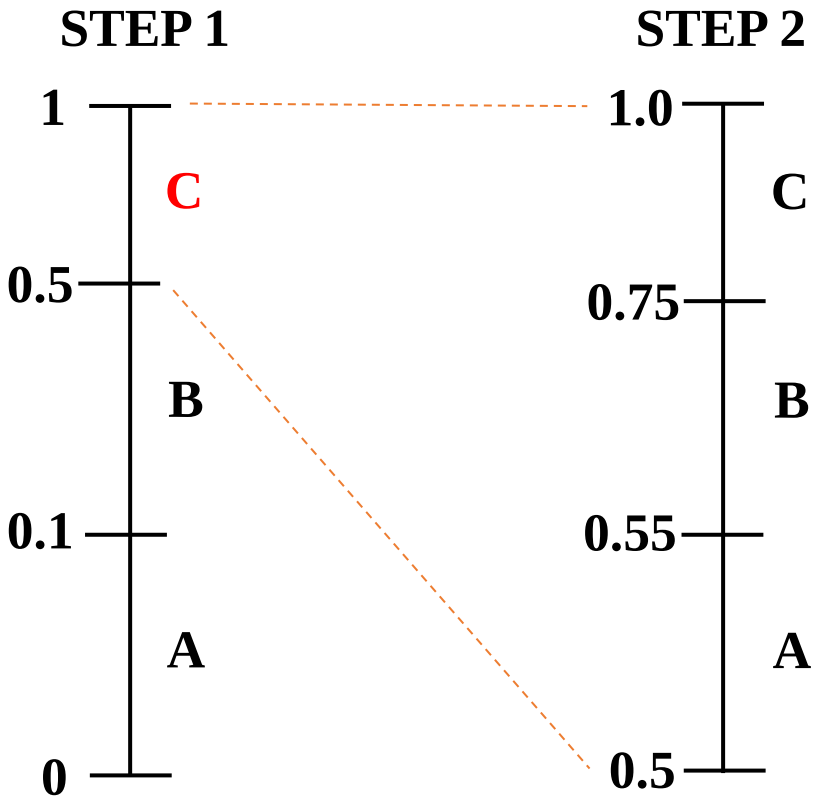
$$= (0.8162 + 0.81602) / 2$$

TAG = 0.81611

Arithmetic Coding

- **Using Arithmetic Coding, decode the message with tag value 0.572 given in the source with probabilities**
  A = 0.1 ; B = 0.4 ; C = 0.5

Soln.,



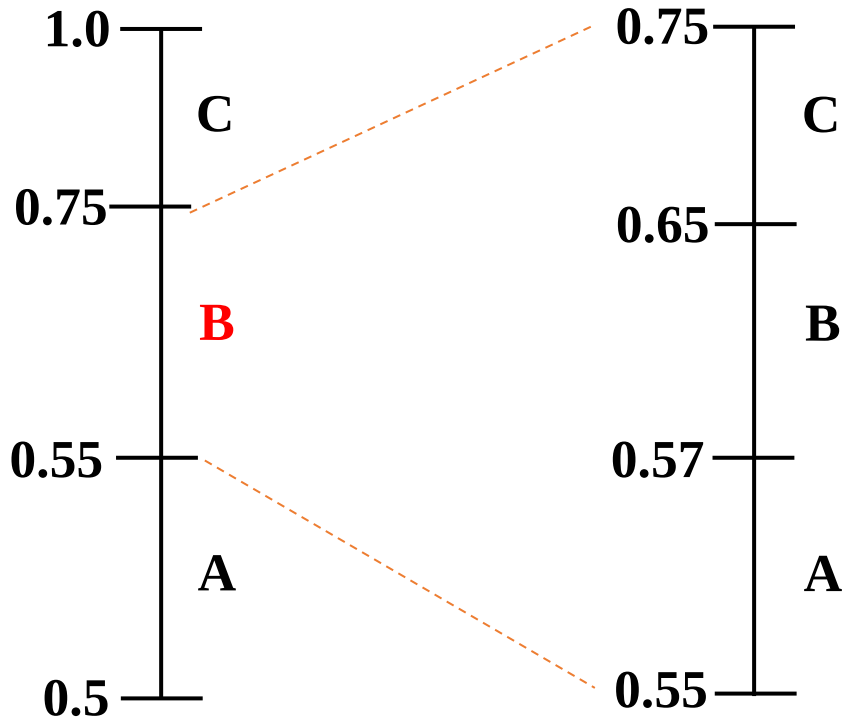
- d = Upper bound – Lower bound = 1.0 – 0.5 = 0.5
- Range of each "Symbol" =

  Lower Limit + d (Prob. Of Symbol)
- Range of "B" = 0.5 + 0.5 (0.1) = 0.55
- Range of "C" = 0.55 + 0.5 (0.4) = 0.75
- Range of "A" = 0.75 + 0.5 (0.5) = 1.0

Arithmetic Coding

- **Using Arithmetic Coding, decode the message with tag value 0.572 given in the source with probabilities**
  A = 0.1 ; B = 0.4 ; C = 0.5

Soln.,



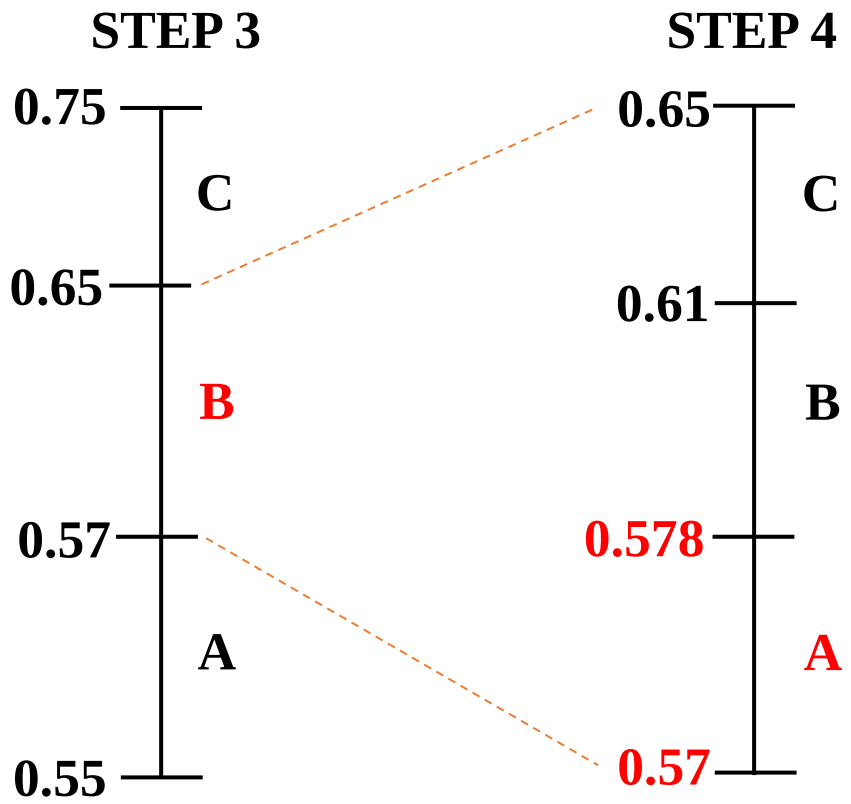
- d = Upper bound – Lower bound = 0.75 – 0.55 = 0.2
- Range of each "Symbol" =

  Lower Limit + d (Prob. Of Symbol)
- Range of "B" = 0.55 + 0.2 (0.1) = 0.57
- Range of "C" = 0.57 + 0.2 (0.4) = 0.65
- Range of "A" = 0.65 + 0.2 (0.5) = 0.75

Arithmetic Coding

- **Using Arithmetic Coding, decode the message with tag value 0.572 given in the source with probabilities**
  A = 0.1 ; B = 0.4 ; C = 0.5

Soln.,



- d = Upper bound – Lower bound = 0.65 – 0.57 = 0.08
- Range of each "Symbol" =

  Lower Limit + d (Prob. Of Symbol)
- Range of "B" = 0.57 + 0.08 (0.1) = 0.578
- Range of "C" = 0.578 + 0.08 (0.4) = 0.61
- Range of "A" = 0.61 + 0.08 (0.5) = 0.65
- The decoded word is:

C B B A

Run Length Encoding

- **Procedure:**
- **STEP 1:** The sequence at the output of a discrete source is written as number of times the bit occurs as a sequence
- **STEP 2:** Every sequence is represented as (bit value, Number of occurrence in the sequence)
- **STEP 3:** If the maximum number of occurrence for all the sequence is denoted as ‘n’ then length of occurrence in bits is $\log_2(n)$ bits (If n is decimal, take the next integer)
- **STEP 4:** The number of occurrence in each sequence is replaced by its binary form with length of $\log_2(n)$ bits
- **STEP 4:** The final sequence of the compressed form is written as the encoded output

Run Length Encoding

- **Using Run Length Encoding Technique, encode the given bit stream 00000111110010000101 .**

Soln., Original Bit Stream : 00000111110010000101

STEP 1: Grouping bits as per successive occurrence

00000 11111 00 1 0000 1 0 1

STEP 2: Arranging in the form (Bit Value, Number of Occurrence)

(0,5) (1,5) (0,2) (1,1) (0,4) (1,1) (0,1) (1,1)

STEP 3: Finding the length of occurrence in bits

Maximum number of occurrence = 5

Length of occurrence in bits = $\log_2(5) = 2.32 \approx 3$ bits

STEP 4: Representing each occurrence value in its corresponding 3 bit representation

(0,5) (1,5) (0,2) (1,1) (0,4) (1,1) (0,1) (1,1)

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

(0,101) (1,101) (0,010) (1,001) (0,100) (1,001) (0,001) (1,001)

STEP 5: Encoded bit stream

0101110100101001010010010011001

Run Length Encoding

- **Using Run Length Encoding Technique, encode the given bit stream.**
  000000111111111111110000000000000111111111

Soln., Original Bit Stream : 000000111111111111110000000000000111111111

STEP 1: Grouping bits as per successive occurrence

000000 11111111111111 0000000000000 111111111

STEP 2: Arranging in the form (Bit Value, Number of Occurrence)

(0,6) (1,14) (0,13) (1,9)

STEP 3: Finding the length of occurrence in bits

Maximum number of occurrence = 14

Length of occurrence in bits = $\log_2(14) = 3.8 \approx 4$ bits

STEP 4: Representing each occurrence value in its corresponding 3 bit representation

(0,6) (1,14) (0,13) (1,9)

↓ ↓ ↓ ↓

(0,0110) (1,1110) (0,1101) (1,1001)

STEP 5: Encoded bit stream

001101111001101111001

Run Length Encoding

- **Using Run Length Encoding Technique, encode the given bit stream AAAAABBBBCCCDEEEFFFFGG.**

Soln., Original Bit Stream : AAAAABBBBCCCDEEEFFFFGG

STEP 1: Grouping bits as per successive occurrence

AAAAA BBBB CCC D EEE FFFF GG

STEP 2: Arranging in the form (Bit Value, Number of Occurrence)

(A,5) (B,4) (C,3) (D,1) (E,3) (F,4) (G,2)

Encoded bit stream

A5B4C3D1E3F4G2

Lempel Ziv Encoding

- The major difficulty in using Huffman code is that symbol probabilities must be known or estimated and both encoder and decoder must know the coding tree
- If a tree is constructed from a unusual alphabet, a channel connecting encoder and decoder , must also deliver a coding tree as the header (for the compressed file)
- This overhead would reduce the compression efficiency
- The Lempel Ziv algorithm is designed to be independent of source probability. It is a variable length to a fixed length algorithm

Lempel Ziv Encoding

- **Procedure:**
- **STEP 1:** The sequence at the output of a discrete source is divided into variable length blocks which are called phrases
- **STEP 2:** A new phrase is introduced every time, a block of letters from the source differs from some previous phrases in last letter
- **STEP 3:** Phrases are listed in the dictionary which shows the location of the existing phrase
- **STEP 4:** When encoding a new phrase, simple specify the location of existing phrase in the dictionary and append a new letter

Numerical Problem on Lempel Ziv Encoding

- **Encode the given sequence using Lempel Ziv algorithm: 10101101001001110101000011001110101100011011**

Soln., 1 | 0 | 10 | 11 | 01 | 00 | 100 | 111 | 010 | 1000 | 011 | 001 | 110 | 101 | 10001 | 1011

Dictionary Location	Dictionary Content	Codeword
Ref: 0000		
0001	1	00001
0010	0	00000
0011	10	00010
0100	11	00011
0101	01	00101
0110	00	00100
0111	100	00110
1000	111	01001
1001	010	01010
1010	1000	01110
1011	011	01011
1100	001	01101
1101	110	01001
1110	101	00110
1111	10001	10101
	1011	11101