RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY

100001/EC600C
INFORMATION THEORY & CODING

MODULE 1- PART II
SOURCES & SOURCE CODING

Anila Kuriakose,
Assistant Professor, Department of ECE, RSET

Discrete Memoryless Sources

Encoding

Contents

- Basic Properties of Codes
- Construction of Instantaneous codes
- Kraft Inequality
- Code efficiency and Redundancy

Coding - Objectives

- To increase the efficiency
  - More information in shorter duration
  - Less redundancy

  → **SOURCE ENCODING**

- To reduce the transmission errors

  → **CHANNEL ENCODING**

Encoding

- Let a source be characterised by the symbols

$$\mathbf{S = \{s_1, s_2, s_3,\ldots,s_q\}} \quad \rightarrow \quad \textbf{Source Alphabet}$$

Consider another set X comprising of r symbols

$$\mathbf{X=\{x_1, x_2, x_3,\ldots,x_r\}} \quad \rightarrow \quad \textbf{Code alphabet}$$

- **Coding** →Representing every symbol of S by a sequence of symbols of X with a one to one relationship.

The sequence of symbols of X to specify a source alphabet → **Codeword**

The number of symbols in codeword → **Word length**

Basic Properties of Codes

1. Block Codes

A particular source symbol is encoded into the fixed codeword. The code can be of fixed length or variable length.

Code words are always Fixed Sequence Codes.

eg.

$S=\{s_1, s_2, s_3, s_4\}$, $X=\{0,1\}$, Code word set $C=\{01, 00, 10, 11\}$

2. Non Singular codes

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

S = {s1, s2, s3, s4}, X = {0, 1}; Codes, **C = {0, 11, 10, 01}**

-" ***non- singular in the small*** " but "***Singular in the large***". ***???***

Clue: Consider the sequence 00110.

Basic Properties of Codes

3. Uniquely Decodable Codes

A non singular code is uniquely decodable if every word in a sequence of words can be uniquely identified.

S = {s1, s2, s3, s4}, C = {0, 11, 10, 01}

$$S^2=\{s_1s_1,s_1s_2,s_1s_3,s_1s_4;\ s_2s_1,s_2s_2,s_2s_3,s_2s_4,s_3s_1,s_3s_2,s_3s_3,s_3s_4,s_4s_1,s_4s_2,s_4s_3,s_4s_4\}$$

Source Symbols	Codes	Source Symbols	Codes	Source Symbols	Codes	Source Symbols	Codes
s_1s_1	0 0	s_2s_1	1 1 0	s_3s_1	1 0 0	s_4s_1	0 1 0
s_1s_2	0 1 1	s_2s_2	1 1 1 1	s_3s_2	1 0 1 1	s_4s_2	0 1 1 1
s_1s_3	0 1 0	s_2s_3	1 1 1 0	s_3s_3	1 0 1 0	s_4s_3	0 1 1 0
s_1s_4	0 0 1	s_2s_4	1 1 0 1	s_3s_4	1 0 0 1	s_4s_4	0 1 0 1

Uniquely decodable codes → ***" The nth extension of the code be non-singular for every finite n."***

Basic Properties of Codes

4. Instantaneous Codes

- A uniquely decodable code is said to be "***instantaneous***" if the end of any code word is recognizable with out the need of inspection of succeeding code symbols.
- That is ***there is no time lag in the process of decoding***.

Source symbols	Code A	Code B	Code C
s_1	0 0	0	0
s_2	0 1	1 0	0 1
s_3	1 0	1 1 0	0 1 1
s_4	1 1	1 1 1 0	0 1 1 1

Prefix property*: *"A necessary and sufficient condition for a code to be 'instantaneous' is that no complete code word be a prefix of some other code word".

Basic Properties of Codes

Classes of Codes

X	Singular	Nonsingular, But Not Uniquely Decodable	Uniquely Decodable, But Not Instantaneous	Instantaneous
1	0	0	10	0
2	0	010	00	10
3	0	01	11	110
4	0	10	110	111

- **Prefix Property:**

A necessary and sufficient condition for a code to be instantaneous is that no complete codeword be a prefix of some other codeword.

The four symbols A, B, C, D are encoded using the following sets of codewords. In each case state whether the code is (i) non-singular, (ii) uniquely decodable and (iii) instantaneous code.

(a) {1, 01, 000, 001}

(b) {0, 10, 000, 100}

(c) {01, 01, 110, 100}

(d) {0, 01, 011, 0111}

(e) {10, 10, 0010, 0111}

Construction of Instantaneous Codes

Encode a 5 symbol source into binary instantaneous Codes.

$$S=\{s_1, s_2, s_3, s_{4,} s_5\}, X=\{0,1\}$$

1. Assign 0 to s_1

$s_1 \rightarrow 0$

2. S_2 cannot be set to 1 to satisfy the prefix property.

$s_2 \rightarrow 10$

3. Remaining codeword should start with 11

$s_3 \rightarrow 110$

4. 111 is a 3 bit prefix unused.

$s_4 \rightarrow 1110$

$s_5 \rightarrow 1111$

Construction of Instantaneous Codes

We can have more freedom if we select a 2 bit codeword for s_1.

4 prefixes are possible 00, 01, 10 and 11.

$s_1 \rightarrow$ 00

$s_2 \rightarrow$ 01

$s_3 \rightarrow$ 10

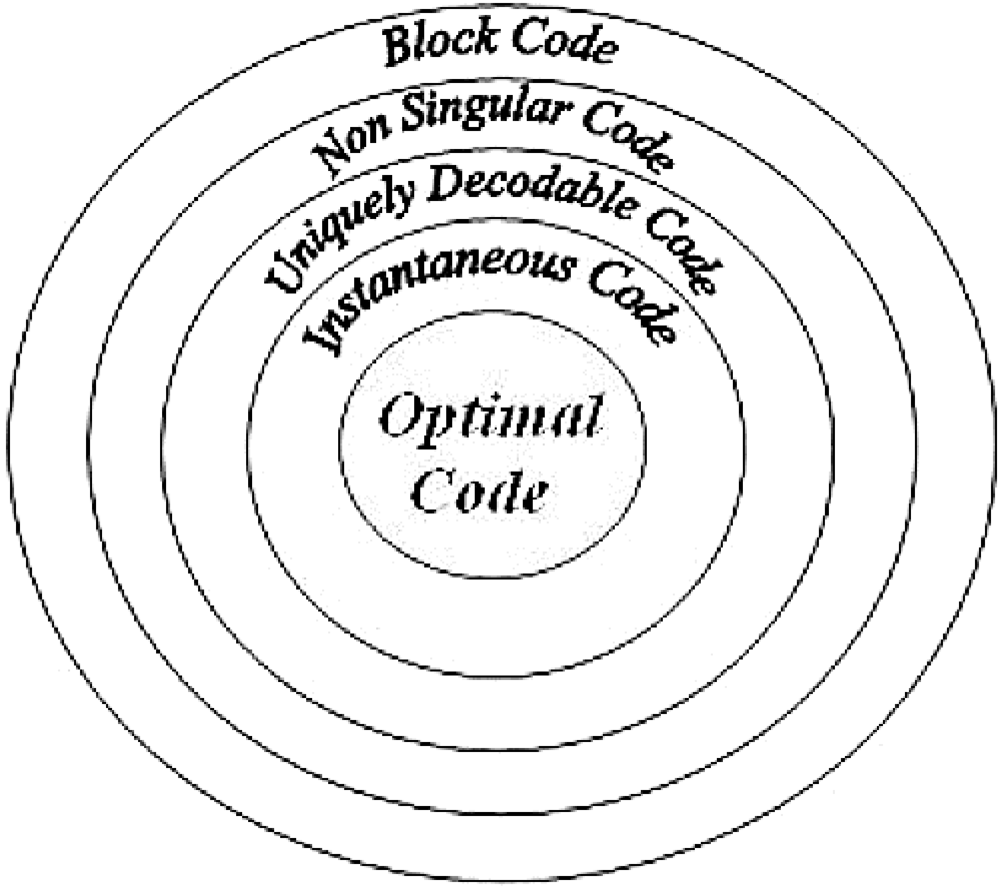
$s_4 \rightarrow$ 110 11 is used to construct codewords of length 3.

$s_5 \rightarrow$ 111

Observation: Shorter we make the first few code words, the longer we will have to make the later code words.

5. Optimal Codes:

- An instantaneous code is said to be optimal code if it has the minimum average length ‘L’



- One may wish to construct an instantaneous code by pre-specifying the word lengths. The necessary and sufficient conditions for the existence of such a code are provided by the ***'Kraft's Inequality'***.

Kraft Inequality

- Given a source $S=\{s_1,s_2,\ldots,s_q\}$.

- Let the word lengths of the codes of these symbols be $l_1, l_2,\ldots,l_q$ and the code alphabet be $X=\{x_1,x_2,\ldots,x_r\}$.

Then an instantaneous code for source exists iff

$$\boxed{\sum_{k=1}^{q} r^{-lk} \leq 1}$$

This equation is called ***Kraft inequality.***

- A six symbol source is encoded into Binary codes shown below. Which of these codes are instantaneous? Test it using Krafts inequality and prefix property.

Source symbol	Code A	Code B	Code C	Code D	Code E
s_1	0 0	0	0	0	0
s_2	0 1	1 0 0 0	1 0	1 0 0 0	1 0
s_3	1 0	1 1 0 0	1 1 0	1 1 1 0	1 1 0
s_4	1 1 0	1 1 1 0	1 1 1 0	1 1 1	1 1 1 0
s_5	1 1 1 0	1 1 0 1	1 1 1 1 0	1 0 1 1	1 1 1 1 0
s_6	1 1 1 1	1 1 1 1	1 1 1 1 1	1 1 0 0	1 1 1 1
$\sum_{k=1}^{6} 2^{-l_k}$	1	$\frac{13}{16} < 1$	1	$\frac{7}{8} < 1$	$1\frac{1}{32} > 1$

- Given $S = \{s1, s2, s3, s4, s5, s6, s7, s8, s9\}$ and $X=\{0,1\}$. Further if $l1=l2=2$ and $l3 = l4 = l5 = l6 = l7 = l8 = l9 =k$, find the minimum value of k for the code to be instantaneous and write the codes.

Code Efficiency and Redundancy

- The average length L of the code

$$L = \sum_{i=1}^{q} p_i l_i$$

p_1, p_2,…,p_q are the probabilities of the source symbols s_1,s_2,…,s_q and l_1, l_2,…,l_q are the respective codeword lengths.

The entropy $$\boldsymbol{H(S) = \sum_{k=1}^{q} P_k log(\frac{1}{P_k})}$$ **bits/symbol**

- $L \geq H(S)$ for binary codes

- $L \geq H_r(S)$ for r-ary codes (r → number of symbols in code alphabet)

$$\mathbf{H_r(S)} = \frac{\boldsymbol{H(S)}}{\boldsymbol{log_2 r}}$$

- The coding efficiency is:

$$\eta = \frac{H(S)}{L} \quad \text{or} \quad \eta = \frac{H_r(S)}{L}$$

- The coding redundancy is $R = 1 - \eta$

$$P=\left\{\frac{1}{3},\frac{1}{3},\frac{1}{6},\frac{1}{6}\right\}$$

C= { 0 , 10 , 110 , 111 }

$$H(S)=2\times\frac{1}{3}\log 3+2\times\frac{1}{6}\log 6$$

$$\log 3+\frac{1}{3}=1.918 \text{ bits/sym}$$

$$L=1.\frac{1}{3}+2.\frac{1}{3}+3.\frac{1}{6}+3.\frac{1}{6}=2 \text{binits / symbol,}$$

$$\eta_c=\frac{H(S)}{L\log r}=\frac{1.918}{2\log_2 2}=0.959 \text{ or } 95.9\%$$

$$E_c=1-\eta_c=0.041 \text{ or } 4.1\%$$

- Let the source have four messages $S = \{s1, s2, s3, s4\}$ with $P = \left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right\}$

$$H(S) = \frac{1}{2}\log 2 + \log 4 + 2\,x\,\frac{1}{8}\log 8 \;= 1.75 \text{ bits/sym.}$$

	p_k	Code	l_k
s_1	1/2	0	$l_1=1$
s_2	1/4	1 0	$l_2=2$
s_3	1/8	1 1 0	$l_3=3$
s_4	1/8	1 1 1	$l_4=3$

$$r=2,\ \eta_c = \frac{H(S)}{L\log r} = \frac{1.75}{1.75\log_2 2} = 1$$

$$\eta_c = 100\%, \qquad Ec = 1 - \eta_c = 0\%$$

$$L = \sum_{k=1}^{4} l_k p_k = 1.\frac{1}{2} + 2.\frac{1}{4} + 3.\frac{1}{8} + 3.\frac{1}{8} = 1.75 \text{binits/symbol}$$

Proof of Kraft Inequality

- Consider a zero memory source, ***S*** with ***q***-symbols {***s1***, ***s2…sq***} and symbol probabilities {***p1***, ***p2*** , … ***pq***}
- Let us encode these symbols into ***r***- ary codes (Using a code alphabet of ***r***- symbols) with word lengths ***l1***, ***l2…lq*** .
- Assume $l_1 \leq l_2 \leq \ldots \leq l_q$
- Since code alphabet has only r symbols, there can be at the most r instantaneously decodable sequences of length 1 satisfying the prefix property.
- Let n_k denote the number of messages encoded into codewords of length 'k'.
- $n_1 \leq r$

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

$$n_2 \le (r - n_1)r$$

$$n_2 \le r^2 - n_1 r$$

- The first symbol can be from only r-n_1 remaining symbols not used in forming code words of length 1 and the second symbol can be any of the r symbols.
- Similarly $\quad n_3 \le [r^2 - n_1 r - n_2]r = r^3 - n_1 r^2 - n_2 r$

- In general $\quad n_k \le r^k - n_1 r^{k-1} - n_2 r^{k-2} \ldots\ldots - n_{k-1} r$

- Multiplying throughout by r^{-k} we get

$$n_k r^{-k} + n_{k-1} r^{-(k-1)} + n_{k-2} r^{-(k-2)} + \ldots\ldots + n_1 r^{-1} \le 1$$

$$\sum_{j=1}^{k} n_j r^{-j} \le 1$$

Kraft Inequality - Proof

$\sum_{j=1}^{k} n_j \, r^{-j} \leq 1$

$\sum_{j=1}^{k} n_j \, r^{-j} = \sum n_1 \, r^{-1} + \sum n_2 \, r^{-2} + \ldots\ldots + \sum n_k \, r^{-k}$

$n_1 + n_2 + \ldots + n_k = q$

Codeword lengths are $l_1, l_2, \ldots, l_q$

Hence

$$\sum_{k=1}^{q} r^{-lk} \leq 1$$

Huffman Coding

Procedure:

1. The source symbols are arranged in the order of decreasing probabilities.

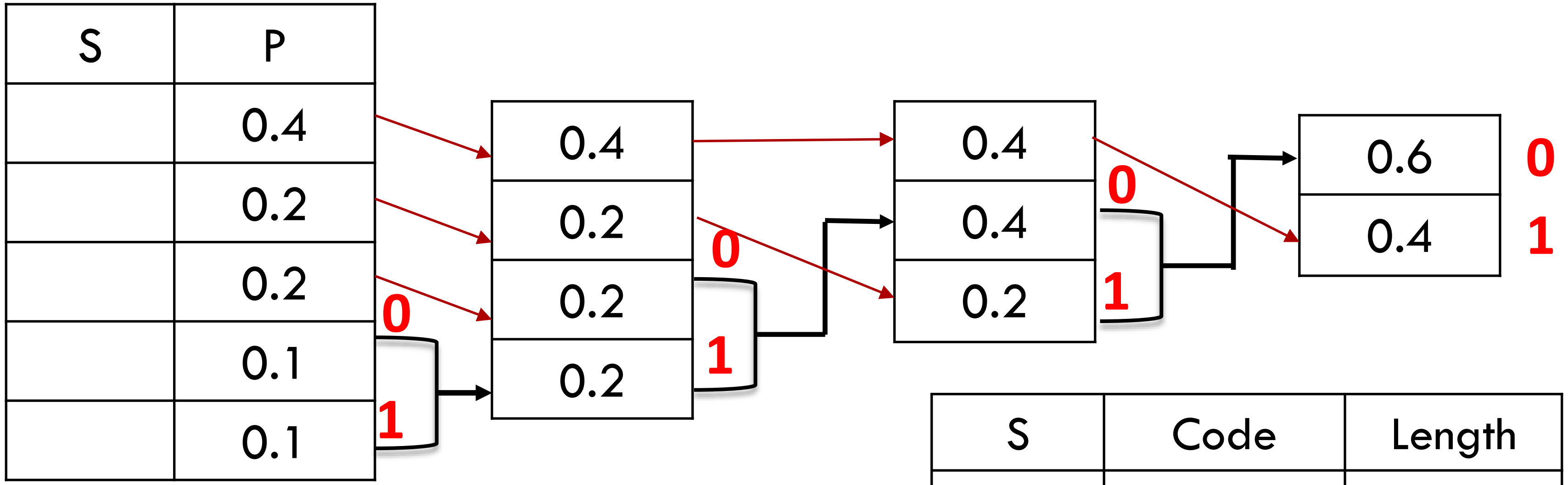
2. The two symbols of lowest probability are assigned 0 and 1.

3. These two symbols are combined into a new symbol with probability equal to the sum of the two original probabilities. The probability of the new symbol is placed in the list in the order of decreasing probabilities.

4. The procedure is repeated until we are left with a final list of symbols of only two for which a 0 and 1 are assigned.

5. The code for each source symbol is found by working backward and tracing the sequence of 0s and 1s assigned to that symbol .

Q) Given a source with symbols s_1, s_2, s_3, s_4, s_5 with probabilities **0.4, 0.2, 0.2, 0.1 and 0.1**.Construct a binary code by applying **Huffman encoding** procedure. Find **H(S),average code length, code efficiency** and **variance of the code.**



S	Code	Length
	1	1
	01	2
	000	3
	0010	4
	0011	4

Type 1 : Composite Symbol as low as possible

$$H(S) = \sum_{i=1}^{q} p(s_i)\, log_2 \frac{1}{p(s_i)}$$

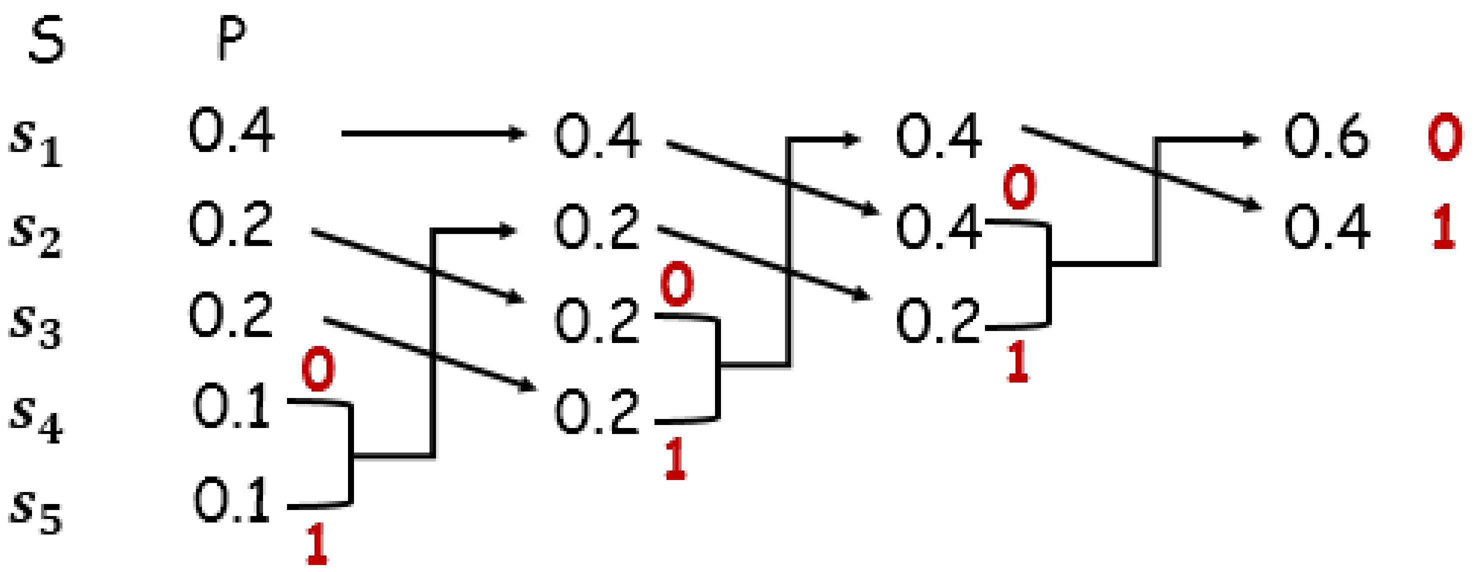
$$L = \sum_{i=1}^{q} p_i l_i$$

$$\eta_c = \frac{H(S)}{L}$$

$$R_{\eta_c} = 1 - \eta_c$$

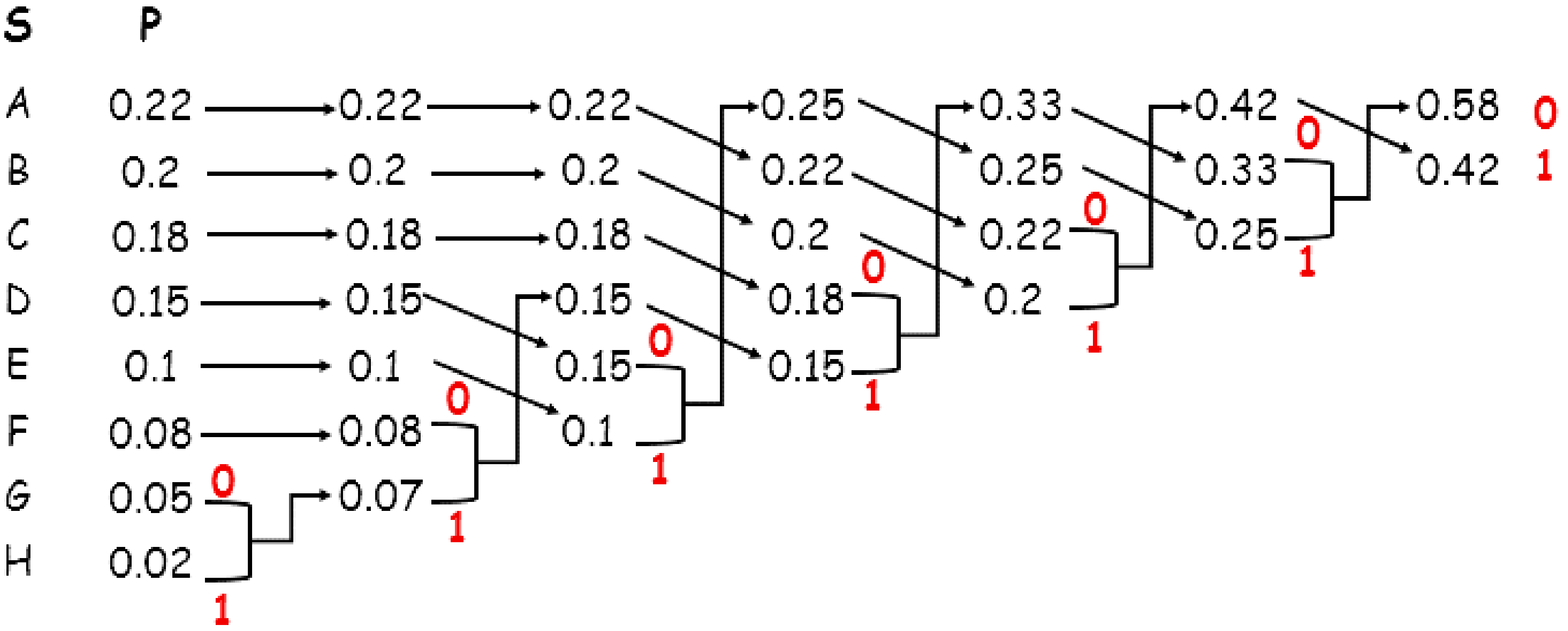
Variance, $\sigma = \sum_{i=1}^{q} p_i (l_i - L)^2$

Type 2 : Composite Symbol as high as possible



S	Code	Length
	00	2
	10	2
	11	2
	010	3
	011	3

Q. Given a source with 8 alphabets A to H with probabilities 0.22, 0.2, 0.18, 0.15, 0.1, 0.08, 0.05 and 0.02. Construct a compact binary & ternary code. Also find code efficiency and draw code tree for the ternary code.



S	Code	Length
A	10	2
B	11	2
C	000	3
D	010	3
E	011	3
F	0010	4
G	00110	5
H	00111	5

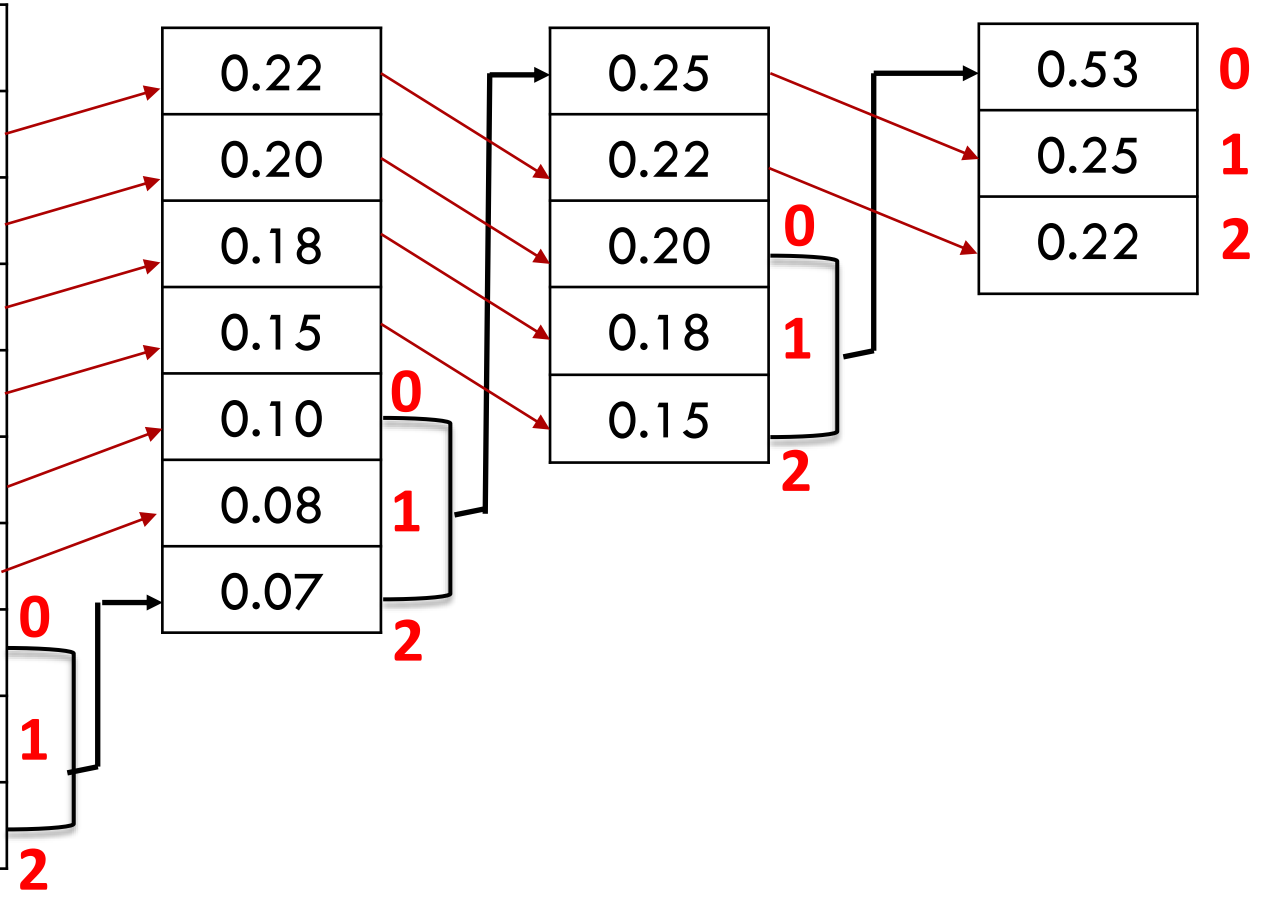
IMPORTANT

$q = r + \alpha\,(r-1)$ where **α** must always be an integer

$8 = 3 + 2\alpha$ $\alpha = 2.5$ which is not an integer

If $q=9$ $\alpha=3$. So add one dummy message symbol so that α becomes an integer.

S	P
A	0.22
B	0.20
C	0.18
D	0.15
E	0.10
F	0.08
G	0.05
H	0.02
Dummy	0



S	Code	Length
A	2	1
B	00	2
C	01	2
D	02	2
E	10	2
F	11	2
G	120	3
H	121	3

Advantage of Huffman Coding Scheme: It is an optimal source coding method.

Drawbacks

1. Impractical for real time applications – as the source symbol probabilities are not always known aprior.
2. Not the best choice for a source with memory.

Problem 2.7

- Illustrating Shannon's noiseless coding theorem.
- Consider a source $S = \{s_1, s_2, s_3\}$ with $P = \{1/2, 1/3, 1/6\}$

Solution

oding

S	P
s_1	1/2
s_2	1/3
s_3	1/6

0
1

S1	
1/2	0
1/2	1

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

Problem 2.7

- $H(S) = \sum_{i=1}^{q} p(s_i)\, log_2 \frac{1}{p(s_i)}$

  $= \frac{1}{2} log_2(2) + \frac{1}{3} log_2(3) + \frac{1}{6} log_2(6)$

  $= 1.459147917\ bits/symbol$

- $L = \sum_{i=1}^{q} p_i l_i$

  $= \frac{1}{2} + \frac{2}{3} + \frac{2}{6}$

  $= 1.5$ binits/symbol

- $\eta_c = \frac{\text{H(S)}}{\text{L}}$

  $= 97.28\%$

S	P	Length
s_1	1/2	1
s_2	1/3	2
s_3	1/6	2

Problem 2.7

s_1	s_2	s_3
1/2	1/3	1/6

- Second extension $[H(S^2)]$ of this source will have $3^2 = 9$ symbols and the corresponding probabilities are:

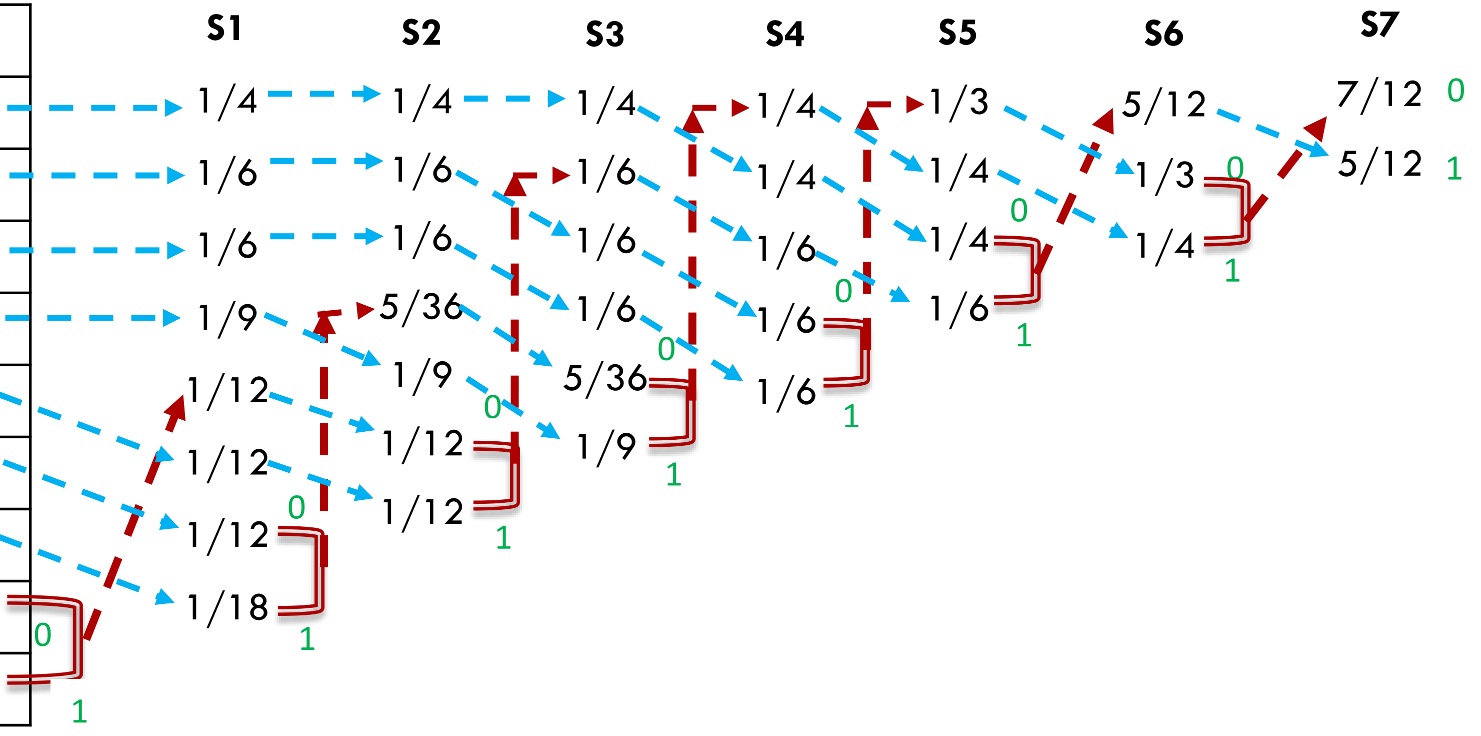
s_1s_1	1/4	s_2s_1	1/6	s_3s_1	1/12
s_1s_2	1/6	s_2s_2	1/9	s_3s_2	1/18
s_1s_3	1/12	s_2s_3	1/18	s_3s_3	1/36

- Messages are now labeled '$\mathbf{m_k}$' and are arranged in the decreasing order of probability.
- $M = \{m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9\}$
- $P = \left\{\frac{1}{4}, \frac{1}{6}, \frac{1}{6}, \frac{1}{9}, \frac{1}{12}, \frac{1}{12}, \frac{1}{18}, \frac{1}{18}, \frac{1}{36}\right\}$

Problem 2.7

s_1	s_2	s_3
1/2	1/3	1/6

S^2	P
m_1	1/4
m_2	1/6
m_3	1/6
m_4	1/9
m_5	1/12
m_6	1/12
m_7	1/18
m_8	1/18
m_9	1/36



Problem 2.7

- For the codes of second extension, we have the following:

  H $(S^2) = 2$ H(S)

- $\mathrm{H(S)} = \sum_{i=1}^{q} p(s_i)\, log_2 \frac{1}{p(s_i)}$

  $= 1.459147917\ bits/symbol$

- $L = \sum_{i=1}^{q} p_i l_i$

  =2.9722 binits/symbol

S^2	P	Code	Length
m_1	1/4	10	2
m_2	1/6	000	3
m_3	1/6	001	3
m_4	1/9	011	3
m_5	1/12	111	3
m_6	1/12	0100	4
m_7	1/18	0101	4
m_8	1/18	1100	4
m_9	1/36	1101	4

Problem 2.7

H (S^2) = 2 H(S)

- $\text{H(S)} = 1.459147917 \; bits/symbol$
- $L =$ 2.9722 binits/symbol
- $\eta_c = \dfrac{H(S^2)}{L} = \dfrac{2 \times H(S)}{L}$

$$= \frac{2 \times 1.459147917}{2.9722}$$

$$= 98.186\%$$

S^2	P	Code	Length
m_1	1/4	10	2
m_2	1/6	000	3
m_3	1/6	001	3
m_4	1/9	011	3
m_5	1/12	111	3
m_6	1/12	0100	4
m_7	1/18	0101	4
m_8	1/18	1100	4
m_9	1/36	1101	4

Inference

- An increase in efficiency of <u>0.909 %</u> (absolute) is achieved.
- This problem illustrates how encoding of extensions increase the efficiency of coding in accordance with Shannon’s noiseless coding theorem.

Shannon's First theorem (Noiseless Coding theorem)

"Given a **code with an alphabet of r-symbols** and a **source with an alphabet of q-symbols,** the **average length of the code words per source symbol** may be made as **arbitrarily close to the lower bound $H_r(S)$ i.e, H(S)/log r** as desired by encoding extensions of the source rather than encoding each source symbol individually".

- We know that each individual code word will have an integer number of code symbols.
- ***L*** be the average word length of the code
- We know,

$$l_k = \log_r \left(\frac{1}{p_k} \right)$$

- Suppose we choose ***lk*** to be the next integer value greater than logr(1/pk)

$$\log_r \frac{1}{p_k} \le l_k \le \log r \frac{1}{p_k} + 1 \qquad \text{----------------- (1)}$$

Furthermore, equation (1) can be rewritten as (change of basis from ‘r’ to ‘2’) :

$$\frac{log(1/p_k)}{logr} \le l_k \le \frac{log(1/p_k)}{logr} + 1$$

Multiplying throughout by p_k and summing for all values of k,

$$\sum_{k=1}^{q} \frac{p_k log\frac{1}{p_k}}{logr} \le \sum_{k=1}^{q} p_k l_k \le \sum_{k=1}^{q} \frac{p_k log\frac{1}{p_k}}{logr} + \sum_{k=1}^{q} p_k \quad \text{, or}$$

$$\boxed{\frac{H(S)}{logr} \le L \le \frac{H(S)}{logr} + 1} \qquad \text{-------------- (2)}$$

- To obtain better efficiency, one will use the ***n***th extension of ***S***, giving ***Ln*** as the new average word length.

- Since Eq. (2) is valid for any zero- memory source, it is also valid for S***n***, and hence, we have

$$\frac{H(S^n)}{logr} \leq L_n \leq \frac{H(S^n)}{logr} + 1 \quad \text{----------- (3)}$$

Since,

$$H(S^n) = n\,H(S) \quad \text{----------- (4)}$$

Substituting (4) in (3)

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

With n →∞

$$\lim_{n \to \infty} \frac{L_n}{n} = \frac{H(S)}{logr}$$

Noiseless Coding Theorem/ Shannon's First Fundamental Theorem