Huffman Coding

Also known as Minimum Redundancy code or Optimum code. This is the procedure for obtaining a compact code with least redundancy. Steps are:

Step 1: Source symbols are listed in non-increasing order of probabilities.

Step 2: Consider the equation $q = r + (r-1)\alpha$ where

q = number of source symbols and

r = number of different symbols used in the code alphabet.

from the equation calculate the quantity 'α' and it should be an integer. If it is not, then "dummy symbols" with zero probabilities are added to 'q', to make α an integer.

For binary codes, α will always be an integer. So this step is not need for Huffman binary codes.

Step 3: The last 'r' symbols of source "s" are combined into a "single composite symbol" by adding their probabilities to get a reduced source "Sa". Then the symbols of "Sa" are arranged in the non increasing order.

Step 4: The last "r" symbols of "Sa" are combined to form another composite symbol by adding their probabilities to get a reduced source "Sb". Then the symbols of "Sb" are arranged in the non increasing order.

Step 5: The process is continued till we arrive a last source having 'r' symbols (if 'α' is an integer, then this condition will automatically met).

Step 6: The last source with 'r' symbols are now encoded with 'r' different code symbols ie 0, 1, 2 .... (r−1).

Step 7: As we "comming backwards" we will recompose the codeword depending on the level which have been combined to get the last reduced source symbol.

In binary coding, the last two symbols are encoded with '0' and '1'. As comming backwards either '0' may be recomposed as '00' and '01' or '1' may be as '10' and '11' depending on the level we have combined.

Step 8 : As we coming backward, the recomposition of one code word is done in order to form new codewords.

Step 9 : This procedure continues till we get codewords corresponding to each of the source symbols. If any dummy symbols are used, they are discarded.

Q8. Given the msgs s_1, s_2, s_3 and s_4 with respective probabilities 0.4, 0.3, 0.2 and 0.1. Construct a binary code by applying Huffman encoding procedure. Determine the efficiency and Redundancy of the code?

Sol: Step 1 : Arrange the symbols in non-increasing order of probabilities.

s_1	s_2	s_3	s_4
0.4	0.3	0.2	0.1
$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$
P_1	P_2	P_3	P_4

Step 2 : Calculate 'α' using $q = r + (r-1)\alpha$

Given $q = 4$ and $r = 2$ (binary code).

$$\alpha = ?$$

$$4 = 2 + (2-1)\alpha$$

$$\alpha = 2 \text{ ; An integer.}$$

Here α shows the number of reduced sources. or the number of stages we have to proceed to reduce the given sources to 'r' symbols.

Step 3 : Form reduced source "s_a" by combining last "$r = 2$" symbols.