Solutions: Homework Set # 1

Problem 1

When a prefix code satisfies the Kraft's inequality with equality, every sequence of code alphabet symbols corresponds to a sequence of codewords, since the probability that a random generated sequence begins with a codeword is at most

$$\sum_{i=1}^{m} D^{-l_i} = 1$$

If the code does not satisfy the prefix condition, then at least one codeword, say $C(m_1)$ is a prefix of another. Then the probability that a random generated sequence begins with a codeword is at most

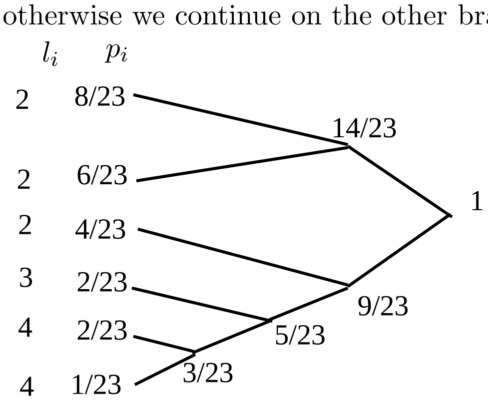
$$\sum_{i=1}^{m-1} D^{-l_i} \leq 1 - D^{-l_m} \leq 1$$

which shows that not every sequence of code alphabet symbols is the beginning of a sequence of codewords.

Problem 2

(a) We will simply start with the most probable until we find the bad one. (But don't taste the last one, it is useless!) I will taste bottle 1 first ($prob = \frac{8}{23}$).

(b) In that case, we can use Huffman coding. So the strategy would be to mix wines of the first and the second bottles and taste the mixture. If it was bad, we taste one of them, otherwise we continue on the other branch of the Huffman tree.



$$
\begin{aligned}
L &= 2.\frac{8}{23} + 2.\frac{6}{23} + 2.\frac{4}{23} + 3.\frac{2}{23} + 4.\frac{2}{23} + 4.\frac{1}{23} \\
&= \frac{16+12+8+6+8+4}{23} \\
&= \frac{54}{23}
\end{aligned}
$$

(c) No, it is optimal as we saw in part (c) that it is possible to find the bad wine with less average number of tastings.

Problem 3

(a) We want to minimize $C = \sum p_i c_i l_i$ subject to $\sum 2^{-l_i} \le 1$. We will assume equality in the constraint and define $r_i = 2^{-l_i}$ and $Q = \sum p_i c_i$. Also define $q_i = (p_i c_i)/Q$. Then $\boldsymbol{q}$ forms a probability distribution and we can write C as

$$
\begin{aligned}
C &= \sum p_i c_i n_i \\
&= Q \sum q_i \log \frac{1}{r_i} \\
&= Q \left(\sum q_i \log \frac{q_i}{r_i} - \sum q_i \log q_i \right) \\
&= Q \left(D(\boldsymbol{q}||\boldsymbol{r}) + H(\boldsymbol{q}) \right).
\end{aligned}
$$

Since the only freedom is in the choice of r_i, we can minimize C by choosing $\boldsymbol{r} = \boldsymbol{q}$ or

$$
l_i^* = -\log \frac{p_i c_i}{\sum p_j c_j},
$$

where we have ignored any integer constraints on l_i. The minimum cost C^* for this assignment of codewords is

$$
C^* = QH(\boldsymbol{q}).
$$

(b) If we use $\boldsymbol{q}$ instead of $\boldsymbol{p}$ for the Huffman procedure, we obtain a code minimizing expected cost.

(c) Now we can account for the integer constraints. Let

$$
l_i = \lceil -\log q_i \rceil.
$$

Then

$$
-\log q_i \le l_i < -\log q_i + 1.
$$

Multiplying both side by $p_i c_i$ and summing over i, we get the relationship

$$
C^* \le C_{\text{Huffman}} < C^* + Q.
$$

Problem 4

(a) Assume each of the codewords have a length multiple of m. We can associate to each m bits a number from 0 to $2^m - 1$. Thus such a code can be converted to a 2^m-arry code in a natural way. Thus the procedure would be to design a 2^m-ary Huffman code and convert it to a binary code. Sine the Huffman codes are optimal, the reader can easily verify the optimality in this case.

(b) We have

$$H_{2^m}(X) = -\sum p_i \log_{2^m}(p_i)$$

Thus

$$H_m(X) = \frac{H_2(X)}{m}$$

By the above procedure we have

$$H_2(X) \leq \mathbb{E}\{l_m\} < H_2(X) + m$$

(c) Let the source have 2^m alphabets with uniform distribution and the rest is clear.

(d) If we have a code in a way that each codeword length is a multiple of m, then

$$l_i \geq m \Rightarrow \mathbb{E}\{l\} \geq m,$$

since

$$H_2(X) = \mathbb{E}\{l\},$$

we must have

$$H_2(X) \geq m.$$

This means that X must have at least 2^m alphabets.

(e) By the following procedure, the hypothesis is clear: To each codeword derived by huffman procedure add redundant bits such that the length of the codeword is a multiple of m (at most $m-1$ redundant bits are enough). This would result in a uniquely decodable code which its codeword lengths are multiples of m.

(f) Let $m = 2$ and take a source producing 7 symbols with probabilities$p_1 = \frac{\epsilon}{6}, p_2 = \frac{\epsilon}{6}, p_3 = \frac{\epsilon}{6}, p_4 = \frac{\epsilon}{6}, p_5 = \frac{\epsilon}{6}, p_6 = \frac{\epsilon}{6}, p_7 = 1-\epsilon$ where ϵ is very small. The code of (a) will be designed as in figure 1:

$$\begin{aligned} 00 &\leftrightarrow 0000 \\ 01 &\leftrightarrow 0001 \\ 02 &\leftrightarrow 0010 \\ 03 &\leftrightarrow 0011 \\ 1 &\leftrightarrow 01 \\ 2 &\leftrightarrow 10 \\ 3 &\leftrightarrow 11 \end{aligned}$$

and thus

$$l_m = 2 + \frac{4\epsilon}{3}$$

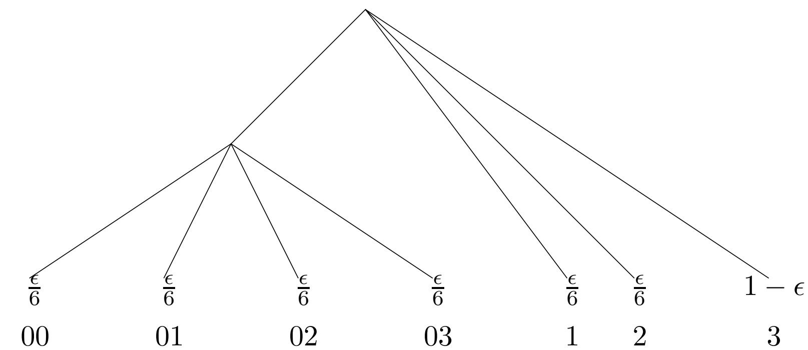


Figure 1: 4-ary Huffman of problem 4 part (f)

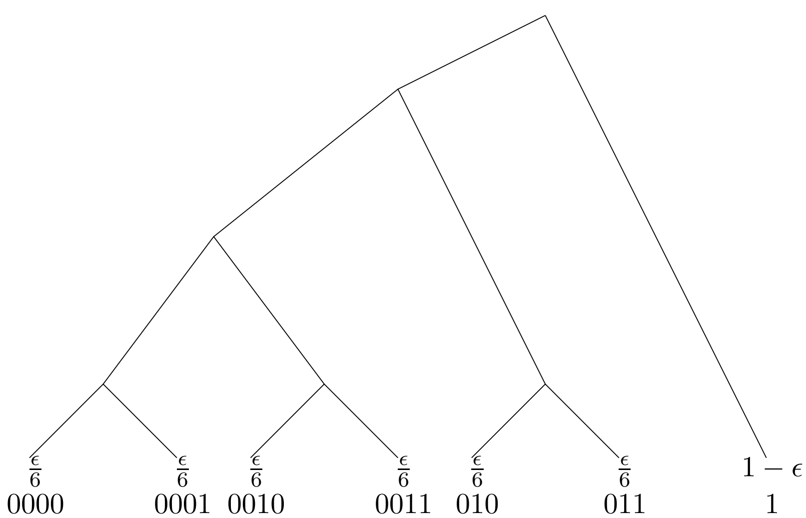


Figure 2: binary Huffman of problem 4 part (f)

If no codeword length constraints were to be satisfied, the Huffman code would give the average codeword length equal to

$$l_H = 1 + \frac{8\epsilon}{3}$$

Thus

$$l_m - l_H = 1 - \frac{4\epsilon}{3} = m - 1 - \frac{4\epsilon}{3}$$

$\frac{4\epsilon}{3}$ could be as small as possible but not equal to zero ofcourse, as it would change the Huffman tree structure.

Problem 5

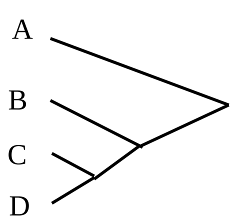
(a) Regardless of what we have as the probability distribution, we have $Pr[A] = \frac{1}{2}$ and $Pr[B] = \frac{1}{4}$. Specifically,

$$\begin{aligned}
p(A) &= \lambda \frac{1}{2} + (1-\lambda)\frac{1}{2} = \frac{1}{2} \\
p(B) &= \lambda \frac{1}{4} + (1-\lambda)\frac{1}{4} = \frac{1}{4} \\
p(C) &= \lambda \frac{1}{16} + (1-\lambda)\,0 = \frac{\lambda}{16} \\
p(D) &= \lambda \frac{1}{16} + (1-\lambda)\,0 = \frac{\lambda}{16} \\
p(E) &= \lambda \frac{1}{16} + (1-\lambda)\frac{2}{16} = \frac{1}{8} - \frac{\lambda}{16} \\
p(F) &= p(E) = \frac{1}{8} - \frac{\lambda}{16}
\end{aligned}$$

For $0 < \lambda < 1$, $p(E) = p(F) > p(C) = p(D)$.
If $\lambda = 1, p(E) = p(F) = p(C) = p(D)$ (model 1),
If $\lambda = 0$ model 2, obviously.

So for $0 < \lambda < 1$, we add $p(C) + p(D) = \frac{\lambda}{8}$. Is this smaller than $\frac{1}{8} - \frac{\lambda}{16}$?
$\frac{\lambda}{8} < \frac{1}{8} - \frac{\lambda}{16} \Rightarrow \frac{3}{16}\lambda < \frac{1}{8}$, $\lambda < \frac{2}{3}$. So for $0 < \lambda < \frac{2}{3}$, Huffman procedure goes on by adding;$\frac{\lambda}{8} + \frac{1}{8} - \frac{\lambda}{16} = \frac{\lambda}{16} + \frac{1}{8} > \frac{1}{8} - \frac{\lambda}{16}$, but smaller than $\frac{1}{4}$.

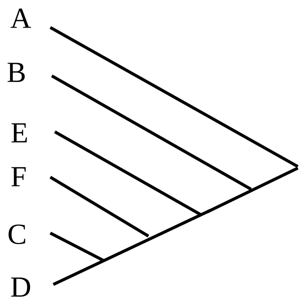
To sum up: For $\lambda = 0$,



which means $l(A) = 1, l(B) = 2, l(C) = l(D) = 3, l(E) = l(F) = 0$

$$\Rightarrow L = \frac{1}{2}1 + \frac{1}{4}2 + \frac{1}{8}3 + \frac{1}{8}3 = 1.75$$

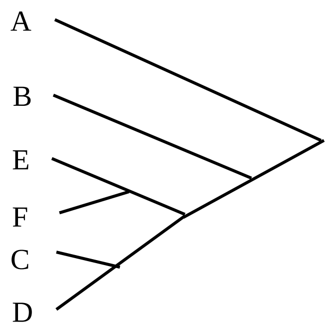
For $0 < \lambda < \frac{2}{3}$,



which means $l(A)=1, l(B)=2, l(C)=l(D)=5, l(E)=3, l(F)=4$

$$\Rightarrow L=\frac{1}{2}+\frac{1}{4}2+\left(\frac{1}{8}-\frac{\lambda}{16}\right)3+\left(\frac{1}{8}-\frac{\lambda}{16}\right)4+5\times 2\frac{\lambda}{16}=\frac{15}{8}+\frac{3}{16}\lambda$$

For $\frac{2}{3}<\lambda\leq 1$,



which means $l(A)=1, l(B)=2, l(C)=l(D)=l(E)=l(F)=4$

$$\Rightarrow L=\frac{1}{2}+\frac{1}{4}2+\left(\frac{1}{8}-\frac{\lambda}{16}\right)4+\left(\frac{1}{8}-\frac{\lambda}{16}\right)4+4\times 2\frac{\lambda}{16}=2$$

(b) If the model is known, then the optimal strategies are the ones we found for $\lambda=0$ or $\lambda=1$ in part (a). Average length $L=2(\lambda)+1.75(1-\lambda)$.

(c) They think the model 1 is valid, so according to this they construct their codes like we have shown in part (a). Then

$$L=\frac{1}{2}1+\frac{1}{4}.2+0.4+0.4+\frac{1}{8}.4+\frac{1}{8}.4=2$$

The average length for the true model is 1.75 as found. So $L_{false}-L_{true}=2-1.75=0.25bits$.

$$\begin{aligned} D\left(p\left(x\right)\|q\left(x\right)\right) &= \sum_x p\left(x\right)\log\frac{p\left(x\right)}{q\left(x\right)} \\ &= \sum_x p\left(x\right)\log\frac{1}{q\left(x\right)}-\sum_x p\left(x\right)\log\frac{1}{p\left(x\right)} \end{aligned}$$

Let's find $D\left(P_2\|P_1\right)$ for this question (since the real model is model 2).

$$D\left(P_2\|P_1\right)=\frac{1}{2}\log 1+\frac{1}{4}\log 1+\frac{1}{8}\log\frac{1/8}{1/16}+\frac{1}{8}\log\frac{1/8}{1/16}=\frac{1}{4}$$

We see that $D\left(P_2\|P_1\right)=\frac{1}{4}=L_{false}-L_{true}$, which is expected. Apart from any rounding effects due to the log function, D distance is the difference between the average false code and the average true code.

Problem 6

(a) Note that the process is a (first-order) Markov chain since the the probability of being in each state (building) for the next time only depends on the current state (building).

(b) The transition matrix for this process would be

$$P = \begin{array}{c} \\ \text{IN} \\ \text{CO} \\ \text{SG} \end{array} \begin{array}{c} \begin{array}{ccc} \text{IN} & \text{CO} & \text{SG} \end{array} \\ \begin{pmatrix} 0 & 2/3 & 1/3 \\ 2/6 & 2/6 & 2/6 \\ 1/3 & 2/3 & 0 \end{pmatrix} \end{array},$$

where P_{ij} is the probability of going to state j given that we are in state i.

(c) The stationary distribution is a vector $\Pi = (\Pi_{\text{IN}} \quad \Pi_{\text{CO}} \quad \Pi_{\text{SG}}) = (p_1, p_2, p_3)$, where $\Pi P = \Pi$.

$$\begin{aligned} &\frac{1}{3}p_2 + \frac{1}{3}p_3 = p_1 \\ &\frac{2}{3}p_1 + \frac{1}{3}p_2 + \frac{2}{3}p_3 = p_2 \\ &\frac{1}{3}p_1 + \frac{1}{3}p_2 = p_3 \\ &p_1 + p_2 + p_3 = 1 \end{aligned}$$

$$\begin{aligned} &p_2 + p_3 = 3p_1 \\ &2p_1 + p_2 + 2p_3 = 3p_2 \\ &p_1 + p_2 = 3p_3 \\ &p_1 + p_2 + p_3 = 1 \end{aligned}$$

$\Rightarrow \Pi = \left(\frac{1}{4} \quad \frac{1}{2} \quad \frac{1}{4}\right)$.