ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
School of Computer and Communication Sciences

Handout 5 — Information Theory and Coding
Solutions to homework 2 — Sep. 29, 2015

PROBLEM 1.

(a) Code I is prefix-free, Code II is not.

(b) Both codes are uniquely decodable: Code I because it is instantaneous, Code II because the 1's at the beginning of each code word act as markers that separates the codewords and the decoding can be performed by counting the 0's between the 1's.

(c) Since each codeword of code II begins with the letter 1 and since the letter 1 only appears at the beginning of codewords, this letter acts as an indicator of start of a codeword.

PROBLEM 2. Since the class of instantaneous codewords is a subset of the class of uniquely decodable codewords, it follows that $\bar{M}_2 \leq \bar{M}_1$. On the other hand, let $\{l_i\}$ be the codeword lengths of the uniquely decodable code for which $\bar{M} = \bar{M}_2$. Since $\{l_i\}$ satisfies the Kraft's inequality, there exists an instantaneous code with these codeword lengths. For this instantaneous code $\bar{M} = M_2$ and we see that $\bar{M}_1 \leq \bar{M} = M_2$, and we conclude that $\bar{M}_1 = \bar{M}_2$.

PROBLEM 3.

(a) $\{00, 01, 100, 101, 1100, 1101, 1110, 1111\}$.

(b) First note that if any two number differ by 2^{-k}, their binary expansion will differ somewhere in the first k bits after the 'point'. (Think of the decimal case: if $a = 0.375\ldots$ and b differs by more than 10^{-3} by it, then b's expansion cannot start with 0.375.)

Next observe that that for $i > j$

$$Q_i - Q_j = \sum_{k=j}^{i-1} P(a_k) \geq P(a_j) \geq 2^{-l_j}.$$

So, the binary expansion of Q_i and Q_j must differ somewhere in the first l_j bits. Since codewords for i and j are at least l_j bits long, neither codeword can be a prefix of the other.

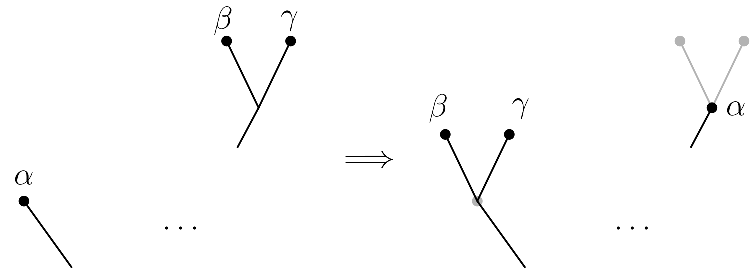
The bound on the average codeword length follows from

$$-\log_2 P(a_i) \leq l_i < -\log_2 P(a_i) + 1.$$

This method of coding is also known as Shannon coding and predates Huffman coding.

PROBLEM 4.

(a) Consider the longest and the shortest codewords. We know that there are at least two longest codewords, suppose their length is l. Suppose the shortest codewords has length s. Suppose that s and l differ by 2 or more. To show that this cannot be the case for an optimal code, consider the transformation shown below:



We see that the transformation decreases the length of two codewords (for letters β and γ) by $l-(s+1)=l-s-1$, whereas it increases the length of one codeword (for the letter α) by $(l-1)-s=l-s-1$. But since $l-s-1>0$, and since all the codewords are equally likely, this would have decreased the average codeword length, contradicting the optimality of the Huffman code. Thus, the longest and shortest codeword lengths can differ by at most 1, and these lengths must be j and $j+1$. (If some other two consecutive depths were used we would either not have enough leaves, or have too many leaves).

(b) Let the number of codewords of length k be m_k, $k=j,j+1$. Since the Huffman procedure yields a complete tree (no leaf is unoccupied) all intermediate nodes have two children. Thus, the 2^j nodes at level j of the tree are either codewords (m_j of them) or each of their two children are codewords ($m_{j+1}/2$ of them). Thus

$$m_j + m_{j+1}/2 = 2^j,$$

and also $m_j + m_{j+1} = x2^j$. From these two equations we find

$$m_j = (2-x)2^j \quad \text{and} \quad m_{j+1} = (x-1)2^{j+1}.$$

(c) By the result of (b) the average codeword length is

$$[jm_j + (j+1)m_{j+1}]/(x2^j) = j + 2(x-1)/x.$$

PROBLEM 5. *An* optimal set of codewords for the two sources are as follows:

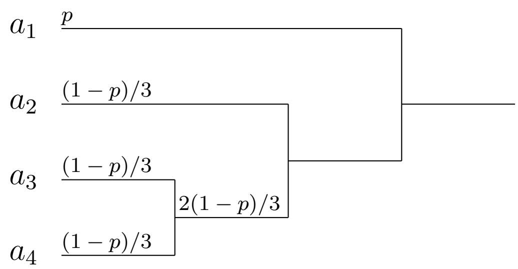
Source I		Source II	
Binary	Ternary	Binary	Ternary
00	0	00	0
01	10	01	1
100	11	100	21
101	12	101	20
110	20	110	220
111	21	1110	221
		1111	222

with average codeword lengths 2.5, 1.7, 2.55, 1.65 digits/symbol, in the order the codes appear in the table.

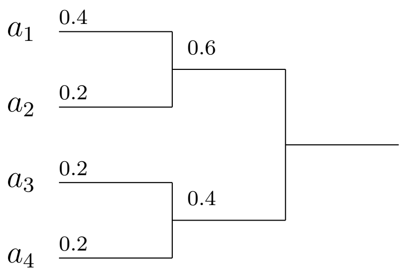
Note that for the ternary code for Source I, we need to add to the symbols of the source an extra symbol of probability zero so that the number of symbols equal 1 modulo $D - 1$.

PROBLEM 6.

(a) Let $p = P(a_1)$, thus $P(a_2) = P(a_3) = P(a_4) = (1 - p)/3$. By the Huffman construction (see figure below) we must have $p > 2(1 - p)/3$, i.e., $q = 2/5$ in order to have $n_1 = 1$.



(b) With $P(a_1) = q$, the figure below illustrates that a Huffman code exists with $n_1 > 1$.



(c) & (d) For $K = 2$, n_1 is always 1. For $K = 3$, $n_1 = 1$ is guaranteed by $P(a_1) > P(a_2) \geq P(a_3)$. Now take $K \geq 4$ and assume $P(a_1) > 2/5$ and $P(a_1) > P(a_2) \geq \cdots \geq P(a_K)$. The Huffman procedure will combine a_{K-1} and a_K to obtain a super-symbol with probability

$$P(a_{K-1}) + P(a_K) < 2\frac{3/5}{K-1} \leq 2/5.$$

Thus, in the reduced ensemble a_1 is still the most likely element. Repeating the argument until $K = 3$, we see that $P(a_1) > q$ guarantees $n_1 = 1$ in all cases.

(e) For $K < 3$ no such q' exists. For $K \geq 3$, we claim $q' = 1/3$. Assume a_1 remains unpaired until the 2nd to last stage (otherwise there is nothing to prove). At this stage we have three nodes, and $P(a_1) < q'$ must be strictly less than one of the other two (otherwise all three would have been less than 1/3). Thus a_1 will be combined with one of them, leading to $n_1 > 1$.