Handout 6 — Information Theory and Coding
Solution 3 — October 11, 2010, SG1 – 15:15-17:00

Problem 1.

a) For n coins, there are $2n+1$ possible situations or "states".

- One of the n coins is heavier.
- One of the n coins is lighter.
- They are all of equal weight.

Each weighing has three possible outcomes - equal, left pan heavier or right pan heavier. Hence with k weightings, there are 3^k possible outcomes and hence we can distinguish between at most 3^k different "states". Hence $2n+1 \leq 3^k$ or $n \leq (3^k-1)/2$. Looking at it from an information theoretic viewpoint, each weighing gives at most $\log_2(3)$ bits of information. There are $2n+1$ possible "states", with a maximum entropy of $\log_2(2n+1)$ bits. Hence in this situation, one would require at least $\log_2(2n+1)/\log_2(3)$ weightings to extract enough information for determination of the odd coin, which gives the same result as above.

b) Split the coins into three groups of equal size. First we compare two groups. There are two possibilities.

- If the two have the same weight it means that the possible counterfeit coin is in the third group and all the other eight coins are of the same weight. Then we compare three coins from the third group with three coins of the first group. If they both have the same weight, the only possibility for the counterfeit coin is the remaining coin of the fourth group and we can decide about it by the third comparison. If they are of different weight, then we know that there exists a counterfeit coin among those three coins from the fourth group. Moreover, now we know that it is lighter or heavier than the other. By the last comparison we can find it among the three.
- If the two groups are not of the same weight, without lose of generality we can assume that the weight of the first pile is heavier than the second pile. suppose that the coins in the first pile are called A_1, A_2, A_3, A_4, in the second pile are called A_5, A_6, A_7, A_8 and in the thirst pile are called $A_9, A_{10}, A_{11}, A_{12}$. Notice that if the counterfeit coin is in the first group then it is heavier and if it is in the second group, it is lighter. Let a_i be the weight of the coin A_i. In the second comparison we make the following piles. A_1, A_2, A_3, A_5 are in the first group and A_4, A_9, A_{10}, A_{11} are in the second group. Now there are three possibilities.
  - If These two groups are of the same weight then the counterfeit coin is lighter and it is among A_6, A_7 or A_8 and we can find it using the thirds comparison.
  - If $a_1 + a_2 + a_3 + a_5 > a_4 + a_9 + a_{10} + a_{11}$ then the counterfeit coin is heavier and it is among A_1, A_2 or A_3 and we can find it using the thirds comparison.

* If $a_1 + a_2 + a_3 + a_5 < a_4 + a_9 + a_{10} + a_{11}$ then the counterfeit coin is either A_5 and is lighter or it is A_4 and it is heavier. So, for the third comparison we can compare A_5 with A_1.

Problem 2. Suppose that in the n-th coin flip, for the first time both head and tail show up. This means that in the first $n-1$ coin flip head comes and in the n-th coin flip tail comes, or vice-versa. The probability of the first event is $p^{n-1}q$ and the probability of the second event is pq^{n-1} where $q = 1-p$. So, $P(X = n) = p^{n-1}q + pq^{n-1}$ for $n \geq 2$. Therefore

$$H(X) = -\sum_{n=2}^{\infty} (p^{n-1}q + q^{n-1}p) \log (p^{n-1}q + q^{n-1}p)$$

Problem 3.

a) For every y in the range of the function f define $\mathcal{A}_y = \{x \in \mathcal{A} : f(x) = y\}$. In fact $\mathcal{A}_y$ is nothing but the inverse image of the point y under the function f. So, we have that

$$P(Y = y) = \sum_{x \in \mathcal{A}_y} P(X = x) \tag{1}$$

Therefore $P(Y = y) \geq P(X = x)$ for every $x \in \mathcal{A}_y$. Since logarithm is an increasing function we have:

$$\log(P(Y = y)) \geq \log P(X = x) \text{ for every } x \in \mathcal{A}_y. \tag{2}$$

From equations 1 and 2 we have that

$$P(Y = y)\log(P(Y = y)) = \log(P(Y = y)) \sum_{x \in \mathcal{A}_y} P(X = x) \geq \sum_{x \in \mathcal{A}_y} P(X = x) \log P(X = x) \tag{3}$$

Notice that inequality 3 holds for every y. So if we add up all of these inequalities we will get:

$$-H(Y) = \sum_{y \in \text{ range of } f} P(Y = y)\log(P(Y = y)) \geq \sum_{x \in \mathcal{A}} P(X = x) \log P(X = x) = -H(X)$$

Therefrom we have $H(Y) \leq H(X)$.

b) From the previous part it is clear that $H(X) = H(Y)$ if and only if for every y, the corresponding inequality 2 is equality. This only happens if $|\mathcal{A}_y| = 1$ for every y. That is to say, equality holds if and only if the pre-image of every element is of size 1. In the other words, $H(X) = H(Y)$ if and only if f is a one-to-one function.

Problem 4. We first observe that the length of the codeword assigned to the i has length between $\log(\frac{1}{p_i})$ and $\log(\frac{1}{p_i}) + 1$. Therefore the average length of the code satisfies the following inequality:

$$\begin{aligned} \sum_{i=1}^{m} p_i \log(\frac{1}{p_i}) \leq L &< \sum_{i=1}^{m} p_i(1 + \log(\frac{1}{p_i})) \\ &= \sum_{i=1}^{m} p_i \log(\frac{1}{p_i}) + \sum_{i=1}^{m} p_i \\ &= \sum_{i=1}^{m} p_i \log(\frac{1}{p_i}) + 1 \end{aligned}$$

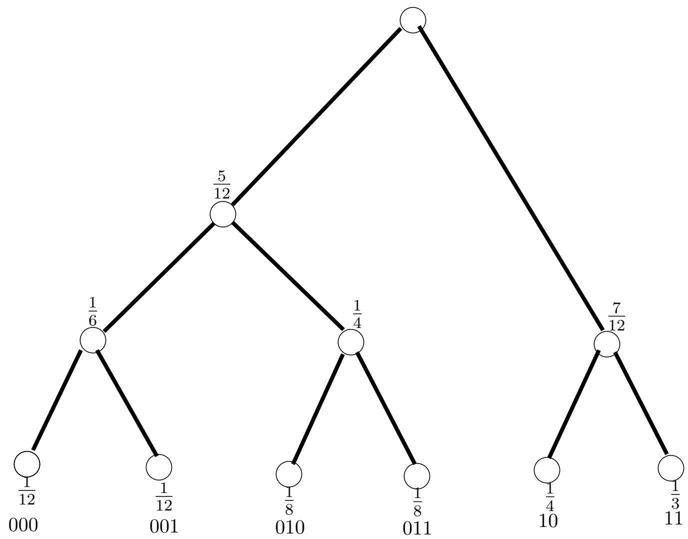

Figure 1: Binary Huffman code for the random variable X

This means that $H(X) \leq L < H(X) + 1$. To complete the solution of the problem we only need to prove the first part. Namely, we have to show that the code we constructed is a prefix-free code. For a contradiction, suppose that the code is not prefix-free. Therefore there exists two distinct indices $i \neq j$ so that the codeword assigned to i is a prefix of the codeword assigned to j. This means that the length of the length of the codeword assigned to j is at least as large as the length of the codeword assigned to i. This implies that $i < j$. (Notice that p_i's are ordered in a decreasing order and based on the code construction, the length of the codewords are non-decreasing. So i must be less than j). Based on the code construction, the code assigned to i is the binary representation of S_i rounded off to $\log\lceil \frac{1}{p_i} \rceil$ bits. Similarly, the code assigned to j is the binary representation of S_j rounded off to $\log\lceil \frac{1}{p_j} \rceil$. Since the i-th codeword is a prefix of the j-th codeword, we conclude that the S_i and S_j agree in the first k digits in their binary representation where k is $\lceil \log\lceil \frac{1}{p_i} \rceil \rceil$. This means that $S_j - S_i$ starts with $\lceil \log\lceil \frac{1}{p_i} \rceil \rceil$ zeros in its binary representation. Equivalently $S_j - S_i < 2^{-\lceil \log\lceil \frac{1}{p_i} \rceil \rceil} < 2^{-\log\lceil \frac{1}{p_i} \rceil} < p_i$. On the other hand, from the definition of S_j's it is clear that $S_j - S_i = p_i + p_{i+1} + \ldots + p_{j-1} \geq p_i$. This contradiction shows that the described code is prefix-free.

Problem 5.

a) In figure 1 we can see the Huffman code and the corresponding binary tree.

b) We first need to include a dummy symbol with probability 0. Then the solution is depicted in figure 2

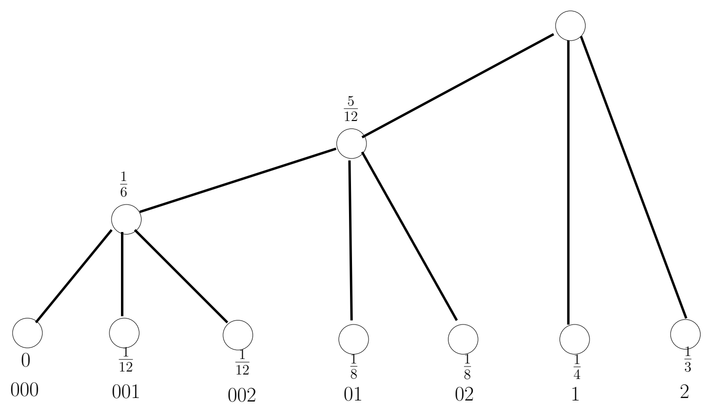

Figure 2: Ternary Huffman code for the random variable X