EE326 Introduction to Information Theory and Coding

- TA in charge of Homework #2: Minyeol Bae (willy3690@kaist.ac.kr)

5.7

(a) Let Q be the minimum number of questions required to determine the set of all defective objects. Then, as we learned in class, we have

$$\begin{aligned} E[Q] &\geq H(X_1, ..., X_n) \\ &\overset{(a)}{=} \sum_{i=1}^{n} H(X_i) \\ &= \sum_{i=1}^{n} H(p_i) \end{aligned}$$

where (a) is because $X_1, ..., X_n$ are independent.

(b) As we learned in class, the Huffman code determines the compact sequence of questions. Then, the last question corresponds to the two least probable sequences.

Because $p_1 > p_2 > \cdots > p_n$, one can easily find the two least probable sequences that are $00\ldots00$ and $00\ldots01$ where

$$\begin{aligned} P(00\ldots00) &= (1-p_1)(1-p_2)\cdots(1-p_{n-1})(1-p_n) \\ P(00\ldots01) &= (1-p_1)(1-p_2)\cdots(1-p_{n-1})p_n \end{aligned}$$

Since these two sequences, $00\ldots00$ and $00\ldots01$, differ only in the value of the last symbol, X_n, the question to distinguish them is simply: "Is $X_n = 1$?".

(c) As we learned in class, we have

$$\begin{aligned} E[Q] &\leq H(X_1, ..., X_n) \\ &= \sum_{i=1}^{n} H(p_i) + 1 \end{aligned}$$

5.11

(i) A suffix condition code is uniquely decodable.

pf) Let C be an arbitrary suffix condition code.

Suppose that C is not uniquely decodable. Then, there exists a concatenation of the codeword W that can be decomposed in two different ways:

$$W = C(x_1)C(x_2)\cdots C(x_n) = C(y_1)C(y_2)\cdots C(y_m)$$

where $x_1x_2\cdots x_n$ and $y_1y_2\cdots y_m$ are two different source strings of symbols.
Without loss of generality, $C(x_n) \neq C(y_m)$.
(If not, simply remove $C(x_n)$ and $C(y_m)$ until $C(x_n) \neq C(y_m)$.)

Now, consider the string W, which is terminated by both $C(x_n)$ and $C(y_m)$. Since $C(x_n) \neq C(y_m)$, their lengths must be different. Assume, without loss of generality, that $l(y_m) > l(x_n)$.

Since $C(x_n)$ and $C(y_m)$ form the last $l(x_n)$ and $l(y_m)$ symbols of W, respectively, $C(x_n)$ must be a suffix of $C(y_m)$. This directly contradicts to the suffix condition.

Therefore, C is uniquely decodable.

(ii) The minimum average length over all codes satisfying the suffix condition is the same as the average length of the Huffman code.

pf) Let C_s be an arbitrary suffix condition code and C_H is a Huffman code. Because the Huffman code is optimal,

$$L(C_s) \overset{(a)}{\geq} L(C_H)$$

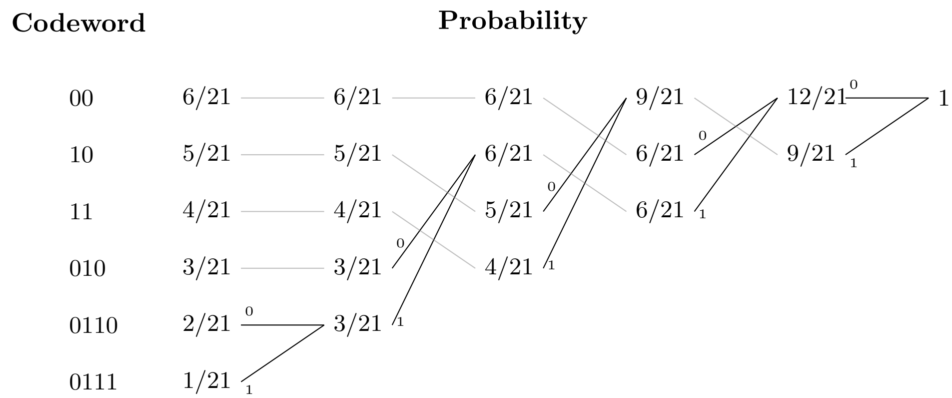
Now, check the equality condition of (a) can be achievable.
Let f be a function that reversing the order of the input codeword (e.g. $f(01) = 10$). Assume that $f(C) = \{f(c) : c \in C\}$ for any code C.

Since Huffman code is a prefix free code, $f(C_H)$ is a suffix condition code, further, $L(f(C_H)) = L(C_H)$. Thus, the equality condition of (a) is achievable.

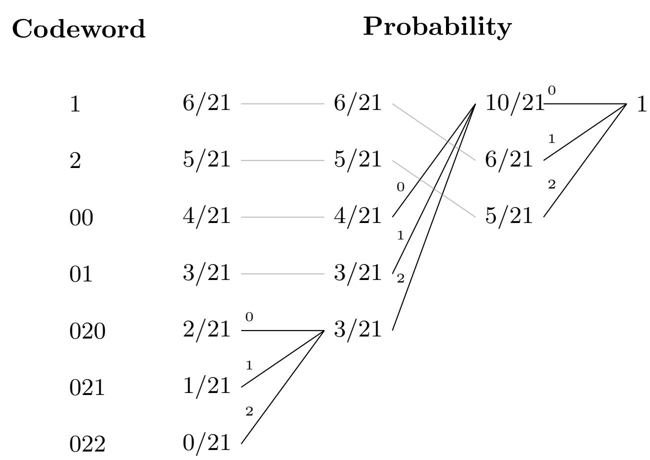
Because there exists a suffix condition code with minimum average length of $L(C_H)$, the minimum average length over all codes satisfying the suffix condition is exactly equal to the average length of the Huffman code.

5.14

(a)



(b)



(c)

$$L_{binary} = \frac{6}{21} \times 2 + \frac{5}{21} \times 2 + \frac{4}{21} \times 2 + \frac{3}{21} \times 3 + \frac{2}{21} \times 4 + \frac{1}{21} \times 4 = \frac{51}{21}$$

$$L_{ternary} = \frac{6}{21} \times 1 + \frac{5}{21} \times 1 + \frac{4}{21} \times 2 + \frac{3}{21} \times 2 + \frac{2}{21} \times 3 + \frac{1}{21} \times 3 + \frac{0}{21} \times 3 = \frac{34}{21}$$

5.25

Assume that x_i is the message symbol of probability p_i. In this context, we will refer to both the message symbols (x_is) and the combined (merged) symbols in the Huffman tree construction as nodes.

(a)

(i) Base Case: $m = 2$

If $m = 2$, the two symbols x_1 and x_2 are assigned the codeword 0 and 1. Thus, the codeword for x_1 trivially has a length of 1.

(ii) Base Case: $m = 3$

If $m = 3$, since $p_1 > p_2 \geq p_3$, the symbols x_2 and x_3 are merged in the first stage of Huffman code construction. Then, since there are only 3 message symbols, merging x_1 with the combined node is the final step. As a result, the codeword of x_1 has length 1.

(iii) Induction Step

Hypothesis: If $m = k \geq 3$, $l(x_1) = 1$.

Induction step: Consider $m = k + 1$. In the first stage of Huffman code construction, the two lowest probability symbols, x_k and x_{k+1}, will be merged. The resulting combined node has a probability of $p_k + p_{k+1}$.

Claim: $p_k + p_{k+1} < p_1$

pf of the claim) Since $p_2 \geq \cdots \geq p_k \geq p_{k+1}$,

$$1 = p_1 + \sum_{i=2}^{k} p_i + p_{k+1} > \frac{2}{5} + (k-1)p_k + p_{k+1}$$
$$\therefore p_k < \frac{1}{k-1}\left(\frac{3}{5} - p_{k+1}\right)$$

Similarly,

$$1 = p_1 + \sum_{i=2}^{k+1} > \frac{2}{5} + kp_{k+1}$$
$$\therefore p_{k+1} < \frac{3}{5k}$$

Thus,

$$\begin{aligned} p_k + p_{k+1} &< \frac{1}{k-1}\left(\frac{3}{5} - p_{k+1}\right) + p_{k+1} \\ &< \frac{3}{5(k-1)} + \frac{k-2}{k-1} p_{k+1} \\ &< \frac{3}{5(k-1)} + \frac{k-2}{k-1} \times \frac{3}{5k} \\ &= \frac{6}{5k} \overset{(a)}{\leq} \frac{2}{5} < p_1 \end{aligned}$$

where (a) is because $k \geq 3$.

In the later stages, the code construction proceeds with the set of k probabilities $(p_1, p_2, ..., p_{k-1}, p_k + p_{k+1})$ where p_1 remains the largest probability. Thus, by the hypothesis, the codeword for x_1 constructed in the later stages has length 1. Since x_1 was not involved in the first merge, the final codeword assigned to x_1 also retains a length of 1.

Therefore, by mathematical induction, the codeword for x_1 has a length 1.

(b) Since $p_1 < \frac{1}{3}$, $m \geq 4$.

Suppose that $l(x_1) = 1$. The codeword length of a symbol increases by one for each merging step in which the symbol is involved. Since all x_i must be merged in the final stage, to achieve $l(x_1) = 1$, x_1 must only be merged in the final stage.

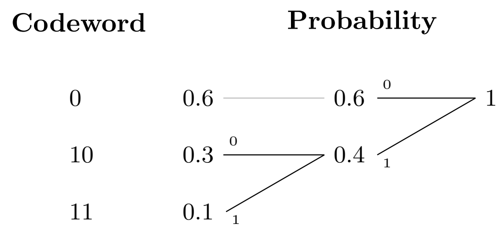
Now, consider the stage with 3 remaining nodes. Because this stage is not a final stage, x_1 remains without merging. Thus, the three remaining probabilities must be p_1, p_2', p_3' where p_2' and p_3' are the combined probabilities of the other symbols. Let S_2' and S_3' be the nodes corresponding to p_2' and p_3', respectively.

Since $p_1 < \frac{1}{3}$, $p_2' + p_3' > \frac{2}{3} > 2p_1$. Consequently, at least one of p_2' or p_3' must be larger than p_1. Without loss of generality, $p_2' > p_1$ and $p_2' > p_3'$.

Then, because p_2' is the largest probability, the two least probability symbols, x_1 and S_3' will be merged in the next stage. This merge involves x_1 before the final stage, which contradicts the requirement for x_1 to have length-1 codeword. Therefore, $l(x_1)$ must be greater than or equal to 2.

5.33

(a)



Thus, the lengths of the binary Huffman codewords are (1, 2, 2).

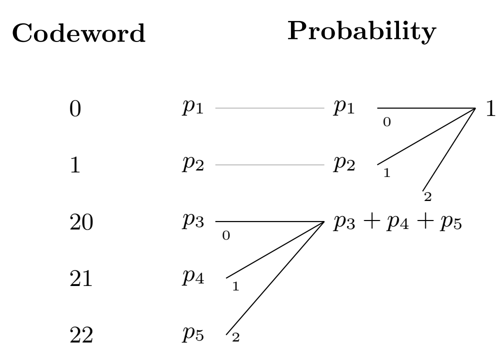
$$\left\lceil \log\left(\frac{1}{0.6}\right)\right\rceil = 1, \quad \left\lceil \log\left(\frac{1}{0.3}\right)\right\rceil = 2, \quad \left\lceil \log\left(\frac{1}{0.1}\right)\right\rceil = 4$$

Thus, the lengths of binary Shannon codewords are (1, 2, 4).

(b) Since X takes only three values, for all $D \geq 3$, the lengths of the D-ary Huffman code are (1, 1, 1). Then, if the longest length codeword of the D-ary Shannon code, $\log_D(\frac{1}{0.1}) = \log_D 10$ is 1, the expected Huffman codeword length is the same as the expected Shannon codeword length. Therefore, the smallest D is 10.

5.42

(a) The codeword lengths (1, 2, 2, 2, 2) are not possible for a 3-ary Huffman code with five symbols.
Construction of a 3-ary Huffman code with five symbols requires two merging stages: first, combining the three least probable symbols, and second, merging the remaining three nodes. This process always results in word lengths of (1, 1, 2, 2, 2) (see the following figure). Because (1, 2, 2, 2, 2) is not equal to (1, 1, 2, 2, 2), it cannot be the set of word length of a 3-ary Huffman code.



(b) The codeword lengths of (2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3) can be the word lengths of a 3-ary Huffman code. See the following example.