EXERCISE FOR CSE202 – WEEK 12

This exercise studies the optimality of Huffman coding among different families of codes. First, a *prefix-free* code is a set S of finite binary words none of which is a prefix of another one.

Question 1. *Show that for such a set, the following inequality holds,*

$$\sum_{s\in S} 2^{-|s|} \leq 1,$$

where $|s|$ denotes the length of the string s.

Solution. Since the code is prefix-free, storing its strings in a trie results in a binary trie where each of the strings corresponds to a leaf. It is thus sufficient to prove that in any binary tree B,

$$\sum_{\ell \text{ leaf of } B} 2^{-\operatorname{depth}(\ell)} \leq 1.$$

The proof is by induction on the number of leaves of the binary tree. The inequality clearly holds for a tree reduced to one leaf (which has to be at depth 0). Assume it holds for all binary trees with n leaves. Take a binary tree with $n+1$ leaves. One of its internal nodes has only leaves for children (whose number is at most 2). The binary tree $B^\star$ obtained by replacing this internal node with a leaf satisfies the inequality. Replacing that leaf by the original internal node increases the sum by at most 0: 2^{-d} becomes either 2^{-d-1} or $2\times 2^{-d-1}$. $\square$

Question 2. *Conversely, given positive integers $(\ell_1,\dots,\ell_n)$ such that*

$$\sum_{k=1}^{n} 2^{-\ell_k} \leq 1, \tag{1}$$

show that there exists a prefix-free code with n words of lengths $\ell_1,\dots,\ell_n$. [Indication: proceed by induction.]

Solution. It is sufficient to show how to build a binary tree with leaves at depths $\ell_1,\dots,\ell_n$. Without loss of generality, assume $\ell_1 \geq \cdots \geq \ell_n$. If $\ell_1 = 0$, then $n=1$ and the tree is reduced to a leaf. Otherwise, there exists $m \geq 1$ such that $2^{-\ell_1}+\cdots+2^{-\ell_m} \leq 1/2$ and $2^{-\ell_{m+1}}+\cdots+2^{-\ell_n} \leq 1/2$. By induction on n, there exist two binary trees one with leaves at depths $(\ell_1-1,\dots,\ell_m-1)$ and the other one with leaves at depths $(\ell_{m+1}-1,\dots,\ell_n-1)$. The binary tree obtained with those two trees as children of the root answers the question. $\square$

A (not necessarily prefix-free) code is called *uniquely decodable* when its words $w_1,\dots,w_n$ have the property that any equality between concatenations of the form $w_{i_1}w_{i_2}\cdots w_{i_k} = w_{j_1}w_{j_2}\cdots w_{j_m}$ implies $(i_1,\dots,i_k)=(j_1,\dots,j_m)$.

Question 3. *Show that prefix-free codes are uniquely decodable.*

Solution. Since no word is a prefix of another one, the equality implies $w_{i_1} = w_{j_1}$ and the result follows by induction on m. $\square$

Let $\ell_1, \dots, \ell_n$ be the lengths of the words of a uniquely decodable code and consider the polynomial $P = x^{\ell_1} + \cdots + x^{\ell_n}$, whose coefficient of x^j is the number of code words of length j.

Question 4. *Show that the coefficient of x^j in P^m is the number of distinct strings of length j obtained by concatenation of m of the words of the code.*

Solution. The concatenations $w_{i_1} \cdots w_{i_m}$ are all distinct since the code is uniquely decodable. Thus the sum of $x^{\ell_{i_1}+\cdots+\ell_{i_m}}$ over all such strings has for coefficient of x^j the number of distinct strings of length j of that type. This sum can be rewritten

$$\sum_{(i_1,\dots,i_m)} x^{\ell_{i_1}+\cdots+\ell_{i_m}} = (x^{\ell_1} + \cdots + x^{\ell_n})^m. \qquad \square$$

Question 5. *If the code is binary (the alphabet has size 2), show that $P(1/2)^m \leq m \max(\ell_i)$. [Indication: bound each of the coefficients.]*

Solution. The number of distinct words of length j over a binary alphabet is bounded by 2^j, thus $P(1/2)^m$ is bounded by its degree, which is bounded by m times the length of the longest word in the code. $\square$

Question 6. *Deduce that the lengths of the words in a decodable binary code satisfy the inequality* (1)*.*

Solution. As m tends to infinity, the right-hand side of the inequality in the previous question grows only linearly, which implies $P(1/2) \leq 1$, and that is exactly inequality (1). $\square$

Question 7. *Conclude that the codes constructed by Huffman's algorithm are optimal not only among prefix-free codes but more generally among all uniquely decodable binary codes.*

Solution. An optimal uniquely decodable binary code satisfies the inequality. By question 2, there exists a prefix-free code with those same word lengths. Thus one can replace the words of the original code by words of identical lengths from a prefix-free one, giving a (necessarily optimal) prefix-free code, which has therefore the same weight as the one found by Huffman's algorithm. $\square$