Existence of Codes With Given Lengths of Codewords

Since we hope to compress data, we would like codes that are uniquely decodable and whose codewords are short.

If we could make all the codewords really short, life would be really easy. Too easy.

Instead, making some codewords short will require that other codewords be long, if the code is to be uniquely decodable.

Questions: What sets of codeword lengths are possible? Is the answer to this question different for instantaneous codes than for uniquely decodable codes?

McMillan's Inequality

There is a uniquely decodable binary code with codewords having lengths $l_1, \ldots, l_I$ if and only if

$$\sum_{i=1}^{I} \frac{1}{2^{l_i}} \ \leq\ 1$$

Examples:

There is a uniquely decodable binary code with lengths 1, 2, 3, 3, since

$$1/2 + 1/4 + 1/8 + 1/8 \ =\ 1$$

An example of such a code is $\{0, 01, 011,\ 111\}$.

There is *no* uniquely decodable binary code with lengths 2, 2, 2, 2, 2, since

$$1/4 + 1/4 + 1/4 + 1/4 + 1/4 \ >\ 1$$

Kraft's Inequality

There is an instantaneous binary code with codewords having lengths $l_1, \ldots, l_I$ if and only if

$$\sum_{i=1}^{I} \frac{1}{2^{l_i}} \ \leq\ 1$$

Examples:

There is an instantaneous binary code with lengths 1, 2, 3, 3, since

$$1/2 + 1/4 + 1/8 + 1/8 \ =\ 1$$

An example of such a code is $\{0, 10, 110,\ 111\}$.

There is an instantaneous binary code with lengths 2, 2, 2, since

$$1/4 + 1/4 + 1/4 \ <\ 1$$

An example of such a code is $\{00, 10, 01\}$.

Implications for Instantaneous and Uniquely Decodable Codes

Combining Kraft's and McMillan's inequalities, we conclude that there is an instantaneous binary code with lengths $l_1, \ldots, l_I$ if and only if there is a uniquely decodable code with these lengths.

Implication: There is probably no practical benefit to using uniquely decodable codes that aren't instantaneous.

Proving the Two Inequalities

We can prove both Kraft's and McMillan's inequality by proving that for any set of lengths, $l_1, \ldots, l_I$, for binary codewords:

A) If $\sum_{i=1}^{I} 1/2^{l_i} \le 1$, we can construct an instantaneous code with codewords having these lengths.

B) If $\sum_{i=1}^{I} 1/2^{l_i} > 1$, there is no uniquely decodable code with codewords having these lengths.

(A) is half of Kraft's inequality.

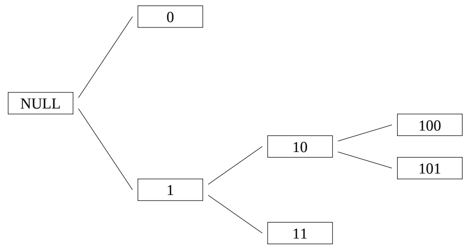
(B) is half of McMillan's inequality.

Using the fact that instantaneous codes are uniquely decodable, (A) gives the other half of McMillan's inequality, and (B) gives the other half of Kraft's inequality.

Visualizing Prefix Codes as Trees

We can view codewords of an instantaneous (prefix) code as leaves of a tree. The root represents the null string; each branch corresponds to adding another code symbol.

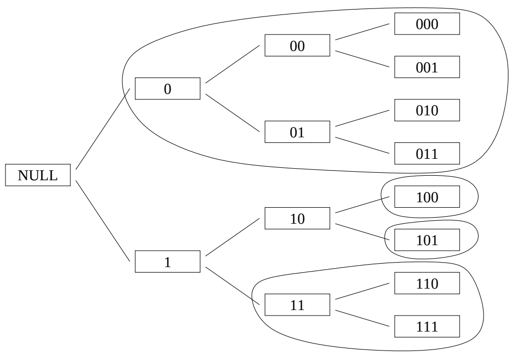
Here is the tree for a code with codewords 0, 11, 100, 101:



Extending the Tree to Maximum Depth

We can extend the tree to the depth of the longest codeword. Each codeword then corresponds to a subtree.

Here's the extension of the previous tree, with each codeword's subtree circled:



Short codewords occupy more of the tree. For a binary code, the fraction of leaves taken by a codeword of length l is $1/2^l$.

Constructing Instantaneous Codes When the Inequality Holds

Suppose that Kraft's Inequality holds:

$$\sum_{i=1}^{I} \frac{1}{2^{l_i}} \le 1$$

Order the lengths so $l_1 \le \cdots \le l_I$. In the binary tree with depth l_I, how can we allocate subtrees to codewords with these lengths?

We go from shortest to longest, $i = 1, \ldots, I$:

1) Pick a node at depth l_i that isn't in a subtree previously used, and let the code for codeword i be the one at that node.

2) Mark all nodes in the subtree headed by the node just picked as being used, and not available to be picked later.

Will there always be a node available in step (1) above?

Why the Construction Will be Possible

If Kraft's inequality holds, we will always be able to do this.

To begin, there are 2^{l_b} nodes at depth l_b.

When we pick a node at depth l_a, the number of nodes that become unavailable at depth l_b (assumed not less than l_a) is $2^{l_b-l_a}$.

When we need to pick a node at depth l_j, after having picked earlier nodes at depths l_i (with $i < j$ and $l_i \le l_j$), the number of nodes left to pick from will be

$$2^{l_j} - \sum_{i=1}^{j-1} 2^{l_j-l_i} \;=\; 2^{l_j}\left[1 - \sum_{i=1}^{j-1}\frac{1}{2^{l_i}}\right] \;>\; 0$$

Since $\sum\limits_{i=1}^{j-1} 1/2^{l_i} \;<\; \sum\limits_{i=1}^{I} 1/2^{l_i} \;\le\; 1$, by assumption.

Why Uniquely Decodable Codes Must Obey the Inequality

Let $l_1 \le \cdots \le l_I$ be the codeword lengths.

Define $K \;=\; \sum\limits_{i=1}^{I} \dfrac{1}{2^{l_i}}$.

For any positive integer n,

$$K^n \;=\; \sum_{i_1,\ldots,i_n} \frac{1}{2^{l_{i_1}}} \times \cdots \times \frac{1}{2^{l_{i_n}}}$$

The sum is over all possible combinations of values for $i_1,\ldots,i_n$ in $\{1,\ldots,I\}$. We can rewrite this in terms of possible values for $j = l_{i_1} + \cdots + l_{i_n}$:

$$K^n \;=\; \sum_{j=1}^{nl_I} \frac{N_{j,n}}{2^j}$$

$N_{j,n}$ is the number of sequences of n codewords that have total length j. If the code is uniquely decodable, $N_{j,n} \le 2^j$, so $K^n \le nl_I$, which for big enough n is possible only if $K \le 1$.