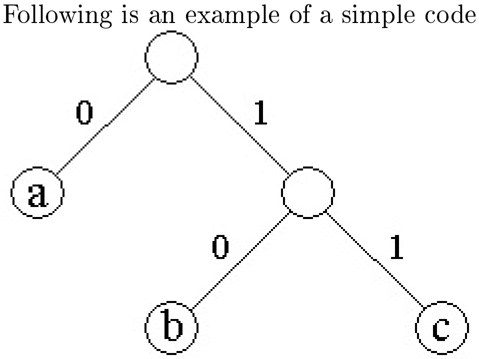
Code Tree, and Kraft's Inequality

September 11, 2013

In coding theory, *Code Tree* is used to encode some symbol

Following is an example of a simple code tree



The code tree means

symbol	code
a	0
b	10
c	11

From the code tree , notice that

$$\text{code length} = \text{level of the end node in the tree}$$

Symbol c is in level 2, is same as the code length of the symbol length$(11) = 2$

Review on Binary Tree Theorem

A binary tree is $full$ if all node is either a leaf or has exactly 2 child nodes.

A binary tree is *complete* if all nodes (except last level) are full

Theorem on number of leaves

A binary tree with k levels has at most 2^{k-1} number of leaves

Proof (By mathematical induction)

When $k = 1$ (base case) , the tree has only 1 level, then only one node with no child node. Thus , the base case is true.

Assume for some integer $K \geq 1$, the $K-$level binary tree has at most 2^{K-1} leaf nodes

When $k = K + 1$, the $K + 1$ level binart tree is actually compose of two $K-$level subtree.

Since each subtree with $K-$level can has at most 2^{K-1}leaf, thus the upper bound of the total number of leaves of these 2 subtree is $2^{K-1}+2^{K-1}=2^K=2^{(k+1)-1}$. Thus the $K+1$ case is also true.

By mathematical induction , k-level binary tree has at most 2^{k-1} leaves.

Now consider the Kraft's Inequality

For a $n-$level full binary tree , the number of leaves are 2^{n-1}

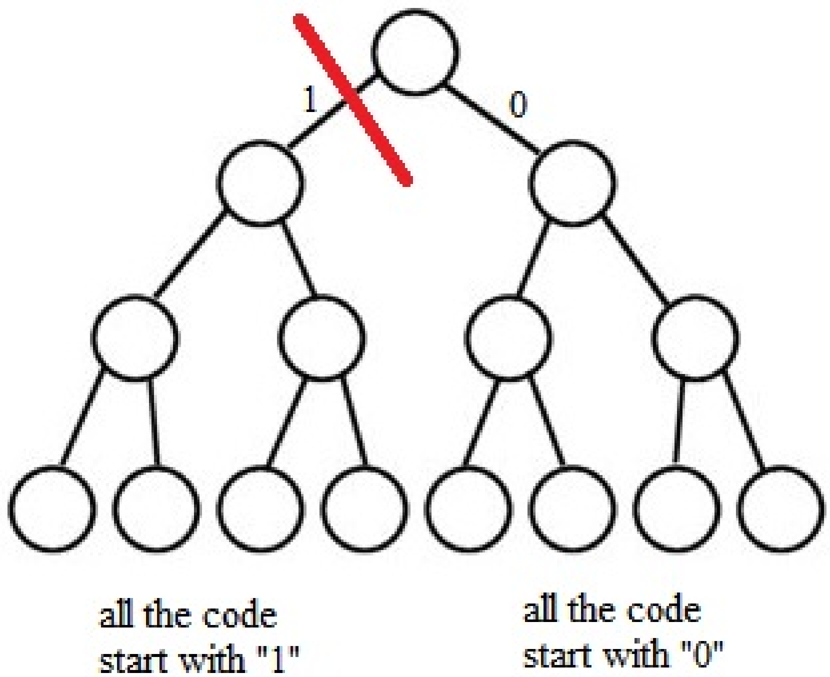
Kraft's Inequality

$$\sum_{i=1}^{m} 2^{-l_i} \leq 1$$

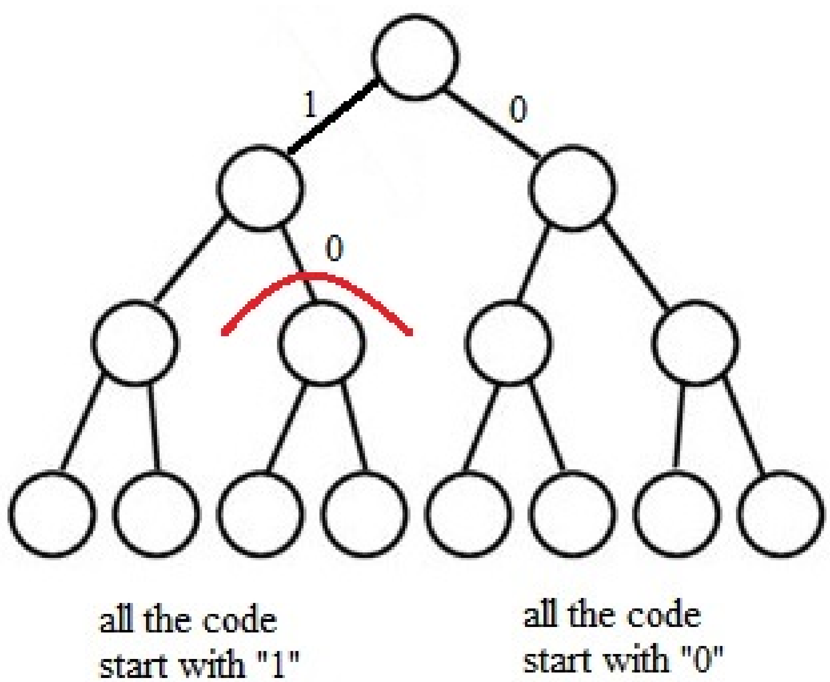
Illustration (not a proof !)

If we "*cut*" a leaf on level k from the tree , that means we extract a code with code length k (explained above)

If we extract a part of code with code length 1 , it is same as cut the tree at level 1 , 2^{-1} (50%) of leaves are being cut



If we extrac a part of code with code length 2 , it is same as cut the tree at level 2 , it is same as cut the tree at level 2, 2^{-2} (25%) of leaves are being cut



If extrac a part of code with code length k , it is same as cut the tree at level k , it is same as cut the tree at level k , 2^{-k} % of leaves are being cut

For a tree with n -level, it has at most 2^{n-1} leaves (the top one also being cut) , thus cutting at level k with resulted in $2^n \cdot 2^{-k} = 2^{n-k}$ leaves being cut

If there are m codes with code length l_1, l_2, ... , l_m , they are being cut from the tree , then

$$\sum_{i=1}^{m} 2^{n-l_i} = \text{total number of leaves to be cut}$$

The tree has to support enough leaves for cutting, thus for a $n-$level tree (including the top node as a leaf)

$$\sum_{i=1}^{m} 2^{n-l_i} \leq 2^n$$

Rearrange, and thus

$$\sum_{i=1}^{m} 2^{-l_i} \leq 1$$

$-END-$