On the number of Huffman Codes

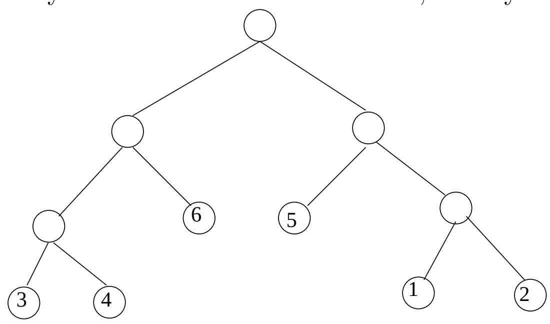
Prof. Dr. Ulrich Tipp

2009-12-01

1 The Problem

In [1] the author discusses the question: How many different Huffman Codes are there? He gave to answers to the question and illustrates it with the following example:

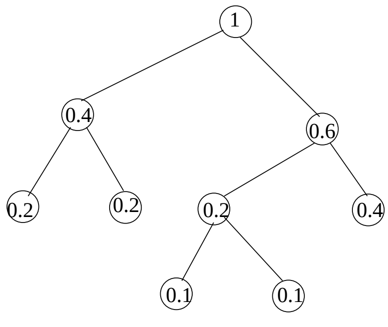
The Probability distribution is: .11, .12, .13, .14, .24, .26 There is exactly one way to construct the Huffman tree, namely



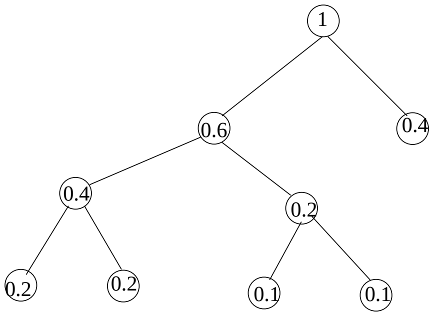
The number of different Huffman Codes results by counting the different choices of Labeling with 0 or 1. The number in this example is 32.

However, it is possible that a probability distribution allows different constrictions of Huffman trees

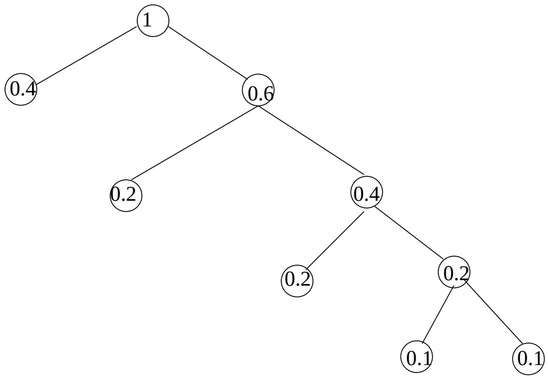
Example: 0.1 0.1 0.2 0.2 0.4 There are three unisomrphic graphs allowed in the Huffman tree construction.



Case 1



Case 2



Case 3

The question now is, what are exactly the distributions with the possibility of unisomorphic Huffman trees? Is there a formula to calculate the number of different Huffamn Codes in that case?

2 Towards a solution

Definition 1. *A Huffman Constellation is a k-tupel of 3-tuples* (p_i, G_i, l_i), $i = 1, \ldots, k$ *consisting of numbers* $p_i (0 < p_i \leq 1)$*, an Isomorphism class* G_i *of a binary tree and a natural number* l_i*. Furthermore* $(p_i, G_i) \neq (p_j, G_j)$ *for all* $i \neq j$ *and* p_i *are in ascending order and*

$$\sum_{i=1}^{k} p_i \cdot l_i = 1$$

The Meaning of this Definition is the representation of a state in the construction of the Huffman Code. That means the p_i represent the probability of an already constructed node together with the tree hanging at this node considered as a root. Nodes with the same probability and the same isomorphism class of the tree hanging at that node are collected in one 3-tupel. The number of such nodes is l_i.

Example: The construction of Huffman tree in Case 2 can be described by the following sequence of Huffman-Constellations

$$\begin{aligned}
&(0.1, [], 2), (0.2, [], 2), (0.4, [], 1) \\
\longrightarrow&(0.2, [[][]], 1), (0.2, [], 2), (0.4, [], 1) \\
\longrightarrow&(0.2, [[][]], 1), (0.4, [[][]], 1), (0.4, [], 1) \\
\longrightarrow&(0.6, [[[][]][[][]]], 1), (0.4, [], 1) \\
\longrightarrow&(1, [[[[][]][[][]]][]], 1)
\end{aligned}$$

We distinguish three different cases:

C0: $p_1 < p_2 < p_3$ In this case we have to put two nodes with p_1 together, if $l_i > 1$ or we have to put a node with p_1 and a node with p_2 together. In any case there is only one choice (up to isomorphism of the graph) in the Huffman Code construction.

C1: $p_1 = p_2 = \dots p_r (r \geq 2)$ In this case we can put nodes with G_1 together (if $l_1 > 1$) or nodes of G_2 together (if $l_2 > 1$) or one with G_1 and one with G_2 resulting in unisomorphic graphs. The number of different possibilities is

$$\frac{s \cdot r}{2} + \binom{r-s}{2}$$

where s is the number of Isomorphism Classes with $l_i > 1$.

C2: $p_1 < p_2 = p_3 = \ldots = p_r (r \geq 3)$. In this case we have to take a node with (p_1, G_1) but we can choose the other node to be (p_2, G_2). In this case we have $r - 1$ different possibilities corresponding to the pairwise unisomorphic trees $G_2, \ldots, G_r$.

After the calculation of different ways to proceed in the Huffman construction we have to look at the different transitions between the cases. This is a complex task. If for example we are in case C0. What will be the next state? If $l_1 = 1$ we know that p_1 and p_2 are the lowest values. In the next step we will have a node with $p_1 + p_2$ instead. Now if $p_1 + p_2 < p_3 < p_4$ the new state will be C0. But if for example $p_1 + p_2 = p_3$ we have to distinguish whether the corresponding Graphs are unisomorphic or not. One could write down a complete list of possible transitions which would be quite lengthy.

References

[1] David Salomon: A Concise Introduction to Data Compression, UTiCS Springer 2008