UDC:003.29
Professional paper

# AN APPLICATION OF PLANAR BINARY BITREES TO PREFIX AND HUFFMAN PREFIX CODE

**Zlatko Erjavec**
Faculty of Organization and Informatics, Varaždin, CROATIA
*zlatko.erjavec@foi.hr*

---

**Abstract:** *In this paper we construct prefix code in which the use of planar binary trees is replaced by the use of the planar binary bitrees. In addition, we apply the planar binary bitrees to the Huffman prefix code. Finally, we code English alphabet in such a way that characters have codewords different from already established ones.*

**Keywords:** *planar binary bitree, Huffman prefix code.*

---

## 1. INTRODUCTION

Although the concept of planar binary trees originated in discrete mathematics, it has numerous applications in other fields of mathematics, computer science and physics. In particular it is extensively used in algebraic structures (see [3], [8]) and quantum physics (see [1], [4]). The generalization of planar binary trees is the concept of planar binary bitrees first induced in [2]. In principle, it is possible to use planar binary bitree wherever planar binary tree is used. However this topics has not been investigated extensively.

The main idea in this paper is to apply a planar binary bitrees to the prefix code, the Huflman prefix code and to the coding of English alphabet. The construction of mentioned codes using planar binary trees can be found in [5, p. 99-103.], [7, vol. 1, p. 402-404.] and originally in [6]. Roughly speaking, we can say that our approach is dual to approach by planar binary trees, because a role of edges is played by internal vertices.

**Definition 1.1.** A *planar binary bitree* (p. b. bitree) is an oriented planar graph which contains the upper and lower tree whose roots are connected by the edge. This edge is called the root of the planar binary bitree. The ordered pair of numbers of internal vertices from the upper and lower trees is called the bidegree of the p. b. bitree.

In every planar binary tree, particularly in every upper or lower tree of the planar binary bitree, each internal vertex has two leaves and one root. The depth of a vertex $v$ is its distance from the nearest vertex of the root. Thus, the root has two vertex of depth 0. Also, the number of leaves of the p. b. bitree with bidegree $(n,m)$ (n,m) is $n+m+2$ (for details see [2]).

---

1991 *Mathematics Subject Classification.* Primary 68R10; Secondary 05C05.

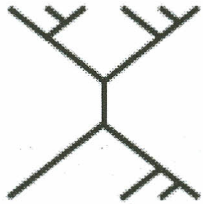

**Figure 1:** An example of a planar binary bitree

**Definition 1.2.** Let $Y$ be a binary bitree with leaves $s_1, \cdots, s_n$, such that each leaf $s_i$ is assigned a weight $w_i$. Then the *average weighted depth* of the planar binary bitree *Y*, denoted *wbt(Y)*, is given by

$$wbt(Y) = \sum_{i=1}^{n} depth(s_i) \cdot w_i$$

## 2. CONSTRUCTING PREFIX CODES FOR PLANAR BINARY BITREES

**Definition 2.1.** A prefix code is an assignment of symbols or other meanings to a set of bitstrings (so called a *codeword*), with the property that no codeword is an initial substring of any other codeword.

It is known for planar binary trees, if all symbols are located at leaves of planar binary trees, then all assigned codewords have prefix property (see [7, Vol 3, p. 452-453.]).

Let $S = \{s_1, \cdots s_k\}$ be a set of symbols. First, we draw an arbitrary planar binary bitree, bidegrees $(n, m)$ where $n + m > k - 2$, whose leaves are bijectively labeled by the symbols (i. e. each leaf gets a different symbol). Second, we label every vertex that goes to a left from the root with a zero, and to a right with a one. Then each symbol corresponds to the codeword formed by the sequence of vertex labels on the path from the root to the leaf labeled by that symbol. As we know, a bitree consists of upper and lower trees, and for which of these trees an associated set of codewords satisfies the prefix property.

*Remark 2.2.* Different kinds of bitrees can generate different sets of codewords.

**Example 2.3.** Suppose *that each relevant message can be expressed as a string of letters (repetitions allowed) drawn from the restricted alphabet {a, b, c, d, e, f, g}. The planar binary bitree shown in* ***Figure2*** *represents the prefix code whose seven codewords correspond to the unique paths from the root to each of seven leaves. The resulting encoding scheme is shown below.*

| letter | *a* | *b* | *c* | *d* | *e* | *f* | *g* |
|---|---|---|---|---|---|---|---|
| codeword | 000 | 0010 | 0011 | 0101 | 011 | 100 | 101 |

Let us recall some facts. One measure of a codes efficiency can be the average weighted length of its codewords, where the length of each codeword is multiplied by the frequency of the symbol it encodes. Suppose that the frequency for each letter of the restricted alphabet of the previous example is given by the following table.

| letter | *a* | *b* | *c* | *d* | *e* | *f* | *g* |
|---|---|---|---|---|---|---|---|
| frequency | 0.2 | 0.05 | 0.1 | 0.1 | 0.25 | 0.15 | 0.15 |

Then the average length of a codeword for the prefix code in Figure 2. is

$$3 \times 0.2 + 4 \times 0.5 + 4 \times 0.1 + 4 \times 0.1 + 3 \times 0.25 + 3 \times 0.15 + 3 \times 0.15 = 3.25$$

For a prefix code constructed from a planar binary bitree, as described above, the length of each codeword is simply the depth of its corresponding leaf.

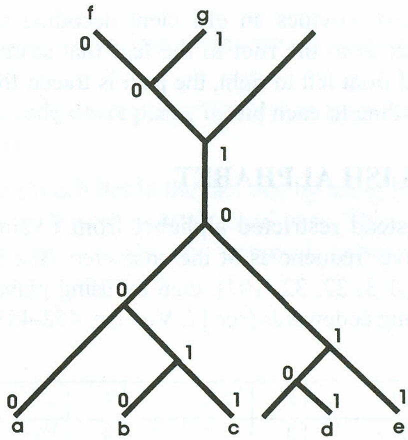


**Figure 2:** Constructing prefix code

## 3. HUFFMAN CODES

The Huflman codes is a prefix code which uses shorter codewords to encode the more frequently occurring symbols. For instance, in a prefix code for "The Master of the Rings", it would make sense to use a short codeword to represent the letter " *a* " and longer codeword to represent " *x* ".

Let us recall some facts. One measure of a codes efficiency can be the *average weighted length* of its codewords, where the length of each codeword is multiplied by the frequency of the symbol it encodes. Suppose that the frequency for each letter of the restricted alphabet of the previous example is given on this way.

**The Huffinan algorithm.** The following algorithm, developed by David Huflman in [6], constructs a prefix code whose codewords have the smallest possible average weighted length.

Let $S = \{s_1, \cdots, s_k\}$ be a set of symbols and $W = w_1, \cdots, w_k$ list of weights associated with symbols Sj. Let *F* be a forest of isolated vertices, labeled with $s_1, \cdots, s_k$, with respective weights . First, we choose two trees of a smallest weights from forest *F*, $T_1$ and $T_2$. Then we create a new planar binary tree which has $T_1$ and $T_2$ as its left and right subtrees, respectively. The new tree has the weight *w(Ti) + w(T2)*. We repeat this procedure until only two trees have left and then amalgamate their roots. The one tree become lower tree and we label its root vertex with 0, and other tree become upper tree and its root vertex label with 1.

**Definition 3.1.** The planar binary bitree which is result of Huffman algorithm is called *Huffman bitree*

The resulting Huffinan code has the following encoding scheme.

| letter | *a* | *b* | *c* | *d* | *e* | *f* | *g* |
|---|---|---|---|---|---|---|---|
| codeword | 00 | 0100 | 0101 | 011 | 10 | 110 | 111 |

The average length of a codeword for this prefix code is

$$2 \times 0.2 + 4 \times 0.5 + 4 \times 0.1 + 3 \times 0.1 + 2 \times 0.25 + 3 \times 0.15 + 3 \times 0.15 = 2.7.$$

The Huffinan bitree also provides an effi cient decoding scheme. A given codeword determines the unique path from the root to the leaf that stores the corresponding symbol. As the codeword is scanned from left to right, the path is traced from the root by traversing 0-vertices or 1-vertices, according to each bit.

## 4. CODING OF ENGLISH ALPHABET

If we imagine that instead restricted alphabet from Example 1. have whole English alphabet with all the relative frequencies of the characters (the first weights of characters _, A, B, C, D, E are 186, 64,1 3, 22, 32, 103), then by using planar binary trees and Huffinan codes we obtain the following codewords (see [7, Vol 3, p. 452-453.]):

| | | | | | |
|---|---|---|---|---|---|
| | 00 | I | 1000 | R | 11001 |
| A | 0100 | J | 1001000 | S | 1101 |
| B | 010100 | K | 1001001 | T | 1110 |
| C | 010101 | L | 100101 | U | 111100 |
| D | 01011 | M | 10011 | V | 111101 |
| E | 0110 | N | 1010 | W | 111110 |
| F | 011100 | O | 1011 | X | 11111100 |
| G | 011101 | P | 110000 | Y | 11111101 |
| H | 01111 | Q | 110001 | Z | 1111111 |

Thus a message like "RIGHT ON" would be encoded by string

11001100001110101111111000101111010.

If we aplly the Huffinan bitree for coding alphabet, then we obtain the following codewords:

| | | | | | |
|---|---|---|---|---|---|
| | 00 | I | mi | R | 10110 |
| A | 0100 | J | 1110111 | S | 1010 |
| B | 010100 | K | 1110110 | T | 1001 |
| C | 010101 | L | 111010 | U | 100011 |
| D | 01011 | M | 11100 | V | 100010 |
| E | 0110 | N | 1101 | W | 100001 |
| F | 011100 | O | 1100 | X | 10000011 |
| G | 011101 | P | 101111 | Y | 10000010 |
| H | 01111 | Q | 101110 | Z | 1000000 |

Thus a message "RIGHT ON" would be encoded by string

1011011110111010111110010011001101.

It is clear that this is a different coding of alphabet than coding by planar binary trees. Some characters have the same codewords as with the coding by trees but in most cases they have different codewords. However, the length of codewords is the same as with planar binary trees.

## SUMMARY

In this paper we have constructed:

- The prefix code where planar binary trees are replaced with planar binary bitrees.
- The Huffman prefix code where planar binary trees are replaced with planar binary bitrees.

The contribution of our approach lies in the fact that by using Huffman bitrees we obtain a coding of alphabet different from already established ones. This coding of alphabet does not change the length of codewords and also our approach leaves a possibility of eventually applying of planar binary bitrees to searching.

## REFERENCES

[1] C. Brouder, On the Trees of Quantum Fields, *Eur. Phys. J. C* 12 (2000), 535-549.

[2] Z. Erjavec, Arithmetic on Planar Binary Bitrees, to appear.

[3] A. Frabetti, Simplicial Properties of the Set of Planar Binary Trees, *J. Alg. Combin.* 13 (2001), no. 1, 41-65. [4] A. Frabetti and C. Brouder, Renormalization of QED with Planar Binary Trees, *Eur. Phys. J. C* **19** (2001), 715-741; hep-th/0003202.

[5] J. Gross and J. Yellen, Graph Theory and Its Applications, CRC Press, London, 1999.

[6] D. Huffinan, *Proc. IRE* 40 (1952), 1098-1101.

[7] D. E. Knuth, *The AH of Computer Programming,* Vol 1.,2.,3.,4., Addison Wesley, 2000.

[8] J.-L. Loday and M. O. Ronco, Hopf Algebra of the Planar Binary Trees, *Adv. in Maths* **139** (1998), 293-309.

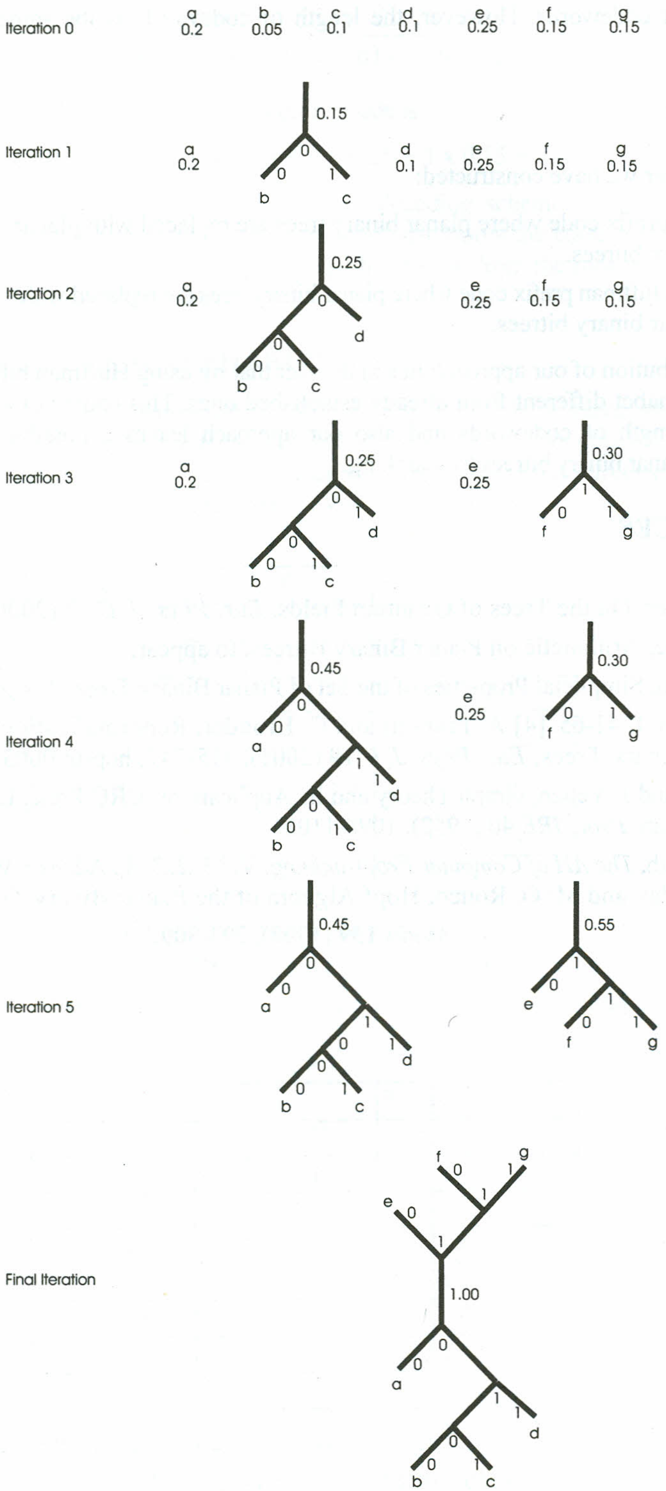


**Figure 3:** The Huffman bitree

**Received:** 05 May 2003
**Accepted:** 01 December 2004