A note on plane trees

Reprinted from JOURNAL OF COMBINATORIAL THEORY Vol. 2, No. 1, January 1967

A Note on Plane Trees

N. G. DE BRUIJN AND B. J. M. MORSELT

Technological University,

Eindhoven, Netherlands

ABSTRACT

The paper gives three different ways of producing the one-to-one correspondence between planted plane trees and trivalent planted plane trees discovered by Harary, Prins, and Tutte.

In a recent paper [2], Harary, Prins, and Tutte observed two classes of trees with the same enumeration function, and indeed they succeeded in establishing a one-to-one correspondence between those two classes. The description of this correspondence, however, was quite complicated. In this note we give three different descriptions of the same correspondence. It will be comparatively easy to check that these are mutually equivalent. It is somewhat harder to check that our correspondence is (apart from a simple transformation) identical to the one described in [2], but we shall not try to do this in this paper.

The classes of trees are (1) *planted plane trees* and (2) *trivalent planted plane trees*. (A plane tree is called *planted* if a point of degree 1 is indicated as its root; it is called *trivalent* if every point has degree 1 or 3.) If n is an integer ≥ 2, the number of planted plane trees with n points is equal to the number of trivalent planted plane trees having n points of degree 1 (and, consequently, $n - 2$ points of degree 3). Between these classes we wish to establish a one-to-one correspondence.

1. Our first description of the correspondence is intuitively the simplest of the three. It is purely geometric. We start from a given trivalent

planted plane tree T with root A. We represent it in a plane such that at every point of degree 3 the lines are pointing upward, downward, and to the right, respectively. The line pointing downward indicates the direction to the root, which is the lowest point. See the heavy lines in Figure 1. In this picture we take a new point B as the root of the new tree T', which has been drawn in broken lines. The points of T' are A, B, and all trivalent points of T. The tree T' is drawn anew in Figure 2.

The principle is so simple that it seems to be a pity to obscure it by giving a formal description or a formal proof of the one-to-one correspondence.

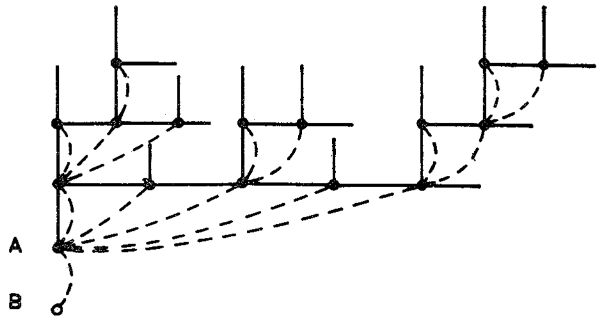

FIGURE 1.

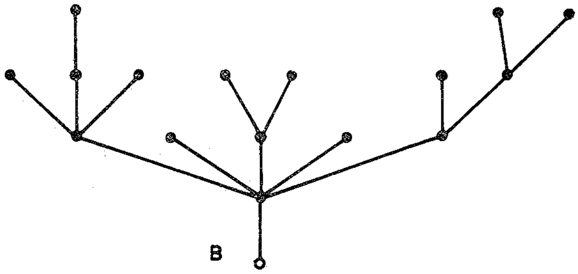

FIGURE 2.

2. Our second method concerns the code words of the trees. It is more complicated than the first one, but it contains aspects which are adequate for handling trees by computers.

To any planted plane tree we can assign a code word consisting of a string of symbols U and D (standing for "up" and "down"). It is obtained by walking in the plane around the tree, as indicated in Figure 3, starting at the root on the left bank of the line leaving it. During this process we follow each line twice, once in the "up" direction (away from the root), once in the "down" direction (leading to the root).

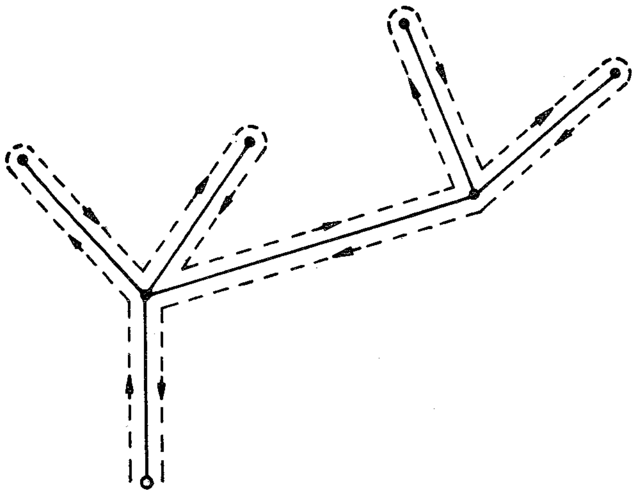

FIGURE 3. Code UUDUDUUDUDDD.

There is a simple rule for deciding whether a given string of U's and D's is or is not the code of a planted plane tree. We evaluate *levels*, starting with level 0. Any U increases the level by 1, any D decreases the level by 1. Indicating the levels as indices we obtain in the case of the code word UUDUDUUDUDDD:

$$ {}_0U_1U_2D_1U_2D_1U_2U_3D_2U_3D_2D_1D_0 $$

In this case of a code word arising from a planted plane tree we observe that at each stage the level indicates the distance to the root. Hence we notice that the beginning and end level are 0 and that all other levels are positive. Moreover, it is easy to see that this condition (to be referred to as the *level condition*) is also sufficient, and hence we have a clear survey of all such code words.

The number of plane trees with m lines equals (see [2])

$$ \frac{1}{m}\binom{2m-2}{m-1}. \tag{1} $$

This expression occurs in many combinatorial problems: for a survey we refer to [1]. Our above description in terms of U's and D's shows that (1) also represents the number of sequences of symbols ± 1 with m times $+1$ and m times -1 such that all partial sums (except the full sum) are positive. The latter result was already shown in 1878 by Whitworth [3].

We can now describe the correspondence of Figure 1 in terms of code words. We get to the following recipe (applied to the trivalent tree of Figure 1):

(i) Write the code word of the tree:

UUUDUUUDUDDUUDUDDDUUDUUUDUUDUD
DDUUDUUUDUUUDUUDUDDDUDDDUDDDDDD

(ii) Add an extra U in front, and then write the word as product of expressions of the type U^aD^b:

$$(U^4D)\,(U^3D)\,(UD^2)\,(U^2D)\,(UD^4)\,(U^2D)\,(U^3D)\,(U^2D)\,(UD^3)\,(U^2D)$$
$$(U^2D)\,(U^3D)\,(U^2D)\,(UD^3)\,(UD^3)\,(UD^6).$$

(Notice that the fact that T is trivalent implies that only products U^kD^l with $k = 1$, $l = 1$ or with $k = 1$, $l > 1$ occur.)

(iii) Replace U^kD by $U^{k-1}D$, UD^l by D:

$$(U^3D)\,(U^2D)\,(D)\,(UD)\,(D)\,(UD)\,(U^2D)\,(UD)\,(D)\,(UD)$$
$$(U^3D)\,(U^2D)\,(UD)\,(D)\,(D)\,(D).$$

(iv) Omitting brackets, we obtain the code word of T':

$$U^3DU^2D^2UD^2UDU^2DUD^2UDU^2DU^2DUD^4.$$

It is easy to show that, if we start from the code word of a trivalent planted plane tree, the recipe (i)–(iv) leads to a word which satisfies the level condition, whence it is the code of a planted plane tree. It is slightly more difficult to see that every code satisfying the level condition can arise in this fashion from exactly one trivalent tree.

This amounts to the following: if we start from a code word satisfying the level condition, like

$$U^5DUD^4U^3D^4,$$

this can, by our recipe, arise from any word of the type

$$U^5D\ U^2D\ UD^{x_1}\ UD^{x_2}\ UD^{x_3}\ U^4D\ UD^{x_4}\ UD^{x_5}\ UD^{x_6},$$

where $x_1, ..., x_6$ are integers ≥ 2. The relevant fact is that exactly one of these words (here the one with $x_1 = 3, x_2 = 2, x_3 = 2, x_4 = 2, x_5 = 2, x_6 = 3$) is the code word of a trivalent planted plane tree. This can be proved by Theorem 1, to be explained presently.

For every D in a code word we can evaluate the level reached after this D. For example, in the word UUUDUDUDDD the levels following the D's are 2, 2, 2, 1, 0, respectively, since the word can be indexed as

$$ {}_0\mathrm{U}_1\mathrm{U}_2\mathrm{U}_3\mathrm{D}_2\mathrm{U}_3\mathrm{D}_2\mathrm{U}_3\mathrm{D}_2\mathrm{D}_1\mathrm{D}_0 \, . $$

We shall say that a certain D in the word is *even* if the level reached after this D was obtained an odd number of times after previous D's. Otherwise D is called *odd*. So in our example of D's with levels 2, 2, 2, 1, 0 these D's are odd, even, odd, odd, odd, respectively. We can now formulate

THEOREM 1. *A planted plane tree is trivalent if and only if in its code word every even* D *is followed by a* U, *and every odd* D (*except the last letter of the word*) *is followed by a* D.

For example, in the word (1) the levels obtained after the D's are (in this order)

2, 4, *4*, 3, 4, *4*, *3*, *2*, 1, 2, 4, 5, *5*, *4*, 3, 4, 6,

8, 9, *9*, *8*, 7, *7*, *6*, 5, *5*, *4*, *3*, *2*, *1*, 0.

The level is printed in italics if it is reached for the k-th time with k even. Precisely the D's corresponding to a level printed in italics are followed by a D, and indeed, the word (1) is the code word of a trivalent tree.

We omit the proof of the above theorem. It is easy to convince one's self of its truth, and a formal proof is necessarily tedious.

In section 4 the reader will find two ALGOL 60 procedures. The first one produces the code of the trivalent planted plane tree if the code of the corresponding planted plane tree is given; the second one describes the inverse of this mapping.

3. In this section we shall prefer to discuss a mapping slightly different from that of the previous sections. Actually it is the mapping of this section 3 that is equivalent to the one given in [2]. The difference is merely a matter of inflection. Every planted plane tree T can be inflected

in the plane (with respect to some line). The effect of the inflection on the code word is reading it backward and interchanging U's and D's.

If T^* denotes the image of T by inflection, and T' the result of the mapping of section 1 applied to a trivalent tree T, then the mapping to be described in this section is the one that takes T^* into $(T')^*$.

Again we shall formulate the mapping in terms of a recipe, explained for an example. Take

$$T = \text{UUUDUDDUUUDUDDUUDUDDDD},$$

and, by section 2,

$$T' = \text{UUUDDUUDDUDD}.$$

Hence

$$T^* = \text{UUUUDUDDUUDUDDDUUDUDDD},$$
$$(T')^* = \text{UUDUUDDUUDDD}.$$

The recipe for obtaining $(T')^*$ from T^* is connected with the reverse polish notation for algebraic expressions. First replace in T^* every U by open parenthesis, every D by close parenthesis:

$$(((\ ) \ (\ )) \ ((\ ) \ (\ ))) \ ((\ ) \ (\ ))). \tag{2}$$

Insert between every pair ( ) a letter and between every pair)(a sign, for simplicity writing + for all signs:

$$(((a) + (b)) + ((c) + (d))) + ((e) + (f))). \tag{3}$$

We write this expression in reverse polish notation:

$$a\,b + c\,d + + e\,f + +. \tag{4}$$

Next we replace every letter by U, every sign by D:

$$\text{UUDUUDDUUDD}.$$

Finally we add an extra D at the end. This gives $(T')^*$.

The above recipe is essentially the same as that of section 2, since the rule for obtaining (4) from (2) can be expressed in a similar manner. We can split (2) in groups, each starting with a number of opening parentheses and ending with a number of closing parentheses. The transition rule, derived from section 2, is: replace a string of one opening paranthesis and k closing parentheses by a string of one letter and $k - 1$ signs; replace a string of l opening parentheses and one closing parenthesis by a single letter. And indeed, this can be shown to be the rule for obtaining the reverse polish notation (4) from the algebraic expression (2).

4. This section contains the ALGOL 60 procedures announced at the end of section 2. These procedures were run and tested for $m \leq 11$, $n \leq 6$ on an Electrologica ELX8 computer, using the EL-Algol translator.

```
procedure triv planted plane tree to planted plane tree (tppt, ppt, m);
          integer m; integer array tppt, ppt;
          begin comment m is the number of lines of the trivalent tree,
                so m is odd.
                The bounds are 1 : 2 × m for tppt and 1 : m + 1 for
                ppt.
                The input for tppt and the output for ppt are sequences
                of + 1 and − 1, where + 1 stands for Up, − 1 for
                Down;
                integer q, r; q: = r : = ppt[1]: = 1;
          here: q: = q + 1; if q > 2 × m then goto ready; r: = r + 1;
                if tppt[q] = 1 then begin ppt[r]: = 1; goto here end;
                ppt[r]: = − 1;
          there: q: = q + 1; if q > 2 × m then goto ready;
                if tppt[q] = 1 then goto here else goto there;
          ready:
          end;
procedure planted plane tree to triv planted plane tree (ppt, tppt, n);
          integer n; integer array ppt, tppt;
          begin comment n is the number of lines of the planted
                plane tree .
                The bounds are 1 : 2 × n for ppt and 1 : 4 × n − 2
                for tppt. The input for ppt and the output for tppt are
                sequences of + 1 and − 1, where + 1 stands for Up,
                − 1 for Down;
```

```
        integer q, r, level; boolean prevup; boolean array
        mult [1:n];
        comment mult[i] = false means that thus far in tppt
        we had an even number of downward arrivals at level i,
        prevup = true means that in ppt the previous step
        was an Up;
        for q: = 1 step 1 until n do mult[q]: = true;
        q: = r: = level: = 0; prevup: = false;
here:   q: = q + 1; r: = r + 1;
        if ppt[r] = 1 then
        begin prevup: = true; tppt[q]: = 1; level: = level + 1;
        goto here end;
there:  tppt[q]: = − 1; level: = level − 1;
        if level = 0 then goto ready;
        mult[level]: = ¬ mult[level]; q: = q + 1;
        if prevup ∨ ¬ mult[level] then
        begin prevup: = false; tppt[q]: = 1; level: = level + 1;
        goto here end;
        goto there;
ready:
end;
```

References

1. H. W. Becker, Discussion of Problem 4277, *Amer. Math. Monthly* **56** (1949), 697-699.

2. F. Harary, G. Prins, and W. T. Tutte, The Number of Plane Trees, *Nederl. Akad. Wetensch. Proc. Ser. A* **67** [*Indag. Math.* **26** (1964), 319–329].

3. W. A. Whitworth, Arrangements of *m* Things of One Sort and *m* Things of Another Sort under Certain Conditions of Priority, *Messenger of Math.* **8** (1878), 105–114.

Printed in Italy by I.T.E.C. - Milan