BIJECTIONS FOR TERNARY TREES AND NON-CROSSING TREES*

ALOIS PANHOLZER† AND HELMUT PRODINGER‡

ABSTRACT. The number N_n of non-crossing trees of size n satisfies $N_{n+1} = T_n$ where T_n enumerates ternary trees of size n. We construct a new bijection to establish that fact. Since $T_n = \frac{1}{2n+1}\binom{3n}{n}$, it follows that $3(3n-1)(3n-2)T_{n-1} = 2n(2n+1)T_n$. We construct two bijections "explaining" this recursion; one of them easily extends to the case of t-ary trees.

1. Introduction

Let $P_n = \{v_1, v_2, \ldots, v_n\}$ be a fixed set of points, conventionally ordered counter-clockwise, forming the vertices of a convex polygon; for convenience we think about them as the vertices of a regular n-gon. Define a non-crossing tree as a connected acyclic graph with vertex set P_n whose edges are straight line segments that do not cross. In the following we consider the vertices labelled from 1 to n and the root of the tree is the vertex 1.

When considering for each vertex v_i in a non-crossing tree the ordered set of sons with smaller label than v_i, let us call them left sons, and the ordered set of sons with larger label than v_i, let us call them right sons, we get an equivalent description of non-crossing trees in the following way: A non-crossing tree is a tree, where each node v_i apart from the root is associated to a (possibly empty) ordered set of left sons $(l_{v_i,1}, \ldots, l_{v_i,L(v_i)})$ and a (possibly empty) ordered set of right sons $(r_{v_i,1}, \ldots, r_{v_i,R(v_i)})$. The root is associated to an ordered set of right sons. By an extended version of the recursive inorder traversing algorithm for binary trees $(l_{v_i,1}, \ldots, l_{v_i,L(v_i)}, v_i, r_{v_i,1}, \ldots, r_{v_i,R(v_i)})$, one gets back the original representation of the non-crossing trees.

Another equivalent description of non-crossing trees is to imagine them as planted plane trees (ordered trees), where each node v_i apart from the root gets a mark from $\{0, 1, \ldots, |v_i|\}$ and the root gets mark 0. Here the mark of a node counts the number of left sons of this node. (As usual $|v_i|$ counts the number of sons of v_i.)

An example for a non-crossing tree using these three representations is given in figure 1.

*This research was partially done while the first author was a research visitor of the Centre for Applicable Analysis and Number Theory at the University of the Witwatersrand, Johannesburg, South Africa.

Date: February 26, 2001.

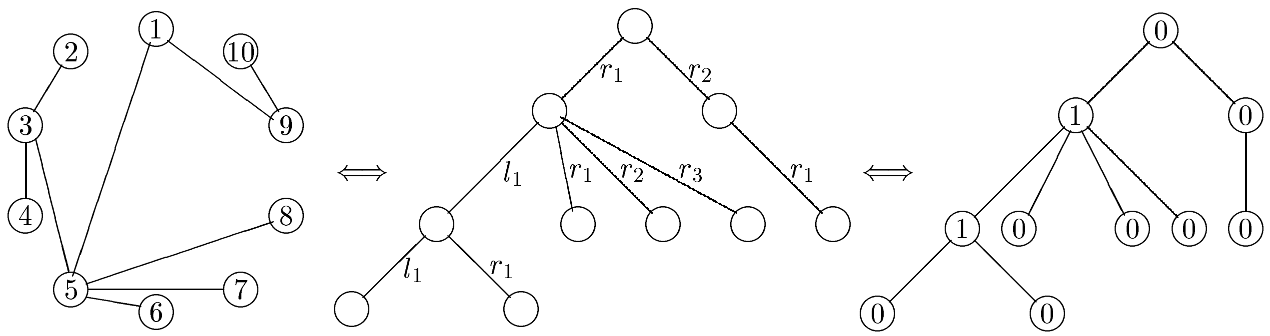


FIGURE 1. Representations of a non-crossing tree.

As far as we know, the enumeration problem for non-crossing trees was proposed first as Problem E 3170 in the American Mathematical Monthly; the published solutions are due to Lossers and Peck, [6]. Further papers on the subject are [2, 3, 5]. It turns out, that the number N_{n+1} of non-crossing trees on $n+1$ points is equal to the number T_n of ternary trees with n internal vertices.

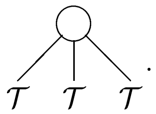
Therefore we have for $n \geq 0$

$$N_{n+1} = \frac{1}{2n+1}\binom{3n}{n}, \tag{1}$$

a generalized Catalan number.

A ternary tree is either a single leaf (or external node) or it is an internal node associated with three ternary trees, which can be described in the following formal equation:

$$\mathcal{T} \quad = \quad \square \quad + \tag{2}$$



2. A rotation correspondence between non-crossing trees and ternary trees

Since there is the strong connection $N_{n+1} = T_n$ between the number of non-crossing trees and the number of ternary trees, there should be some bijections between these families of trees.

Such a bijection, which uses configurations of chords on a circle is given in [2], another one is given in [8, p. 138] (suggested independently by R. Simion and A. Postnikov, as mentioned by R. Stanley) and still another one is presented here. It uses the idea of the well known rotation correspondence between binary trees and planted plane trees (e. g. [9, p. 60]). This classical correspondence proves that the number P_{n+1} of planted plane trees with $n+1$ nodes equals the number B_n of binary trees with n internal vertices, given by Catalan numbers: $P_{n+1} = B_n = \frac{1}{n+1}\binom{2n}{n}$.

Formally a bijection φ from ternary trees to non-crossing trees can be described in the following way, where T is a ternary tree and we construct a non-crossing tree $N = \varphi(T)$:

(1) The vertices of T are the vertices of N with the root deleted.
(2) The root of T is the first right son of the root of N.
(3) Vertex v is a left son of vertex w in T if and only if v is the first left son of w in N.
(4) Vertex v is a middle son of vertex w in T if and only if v is the first right son of w in N.
(5) Vertex v is a right son of vertex w in T if and only if v is the brother to the right of w in N.

The operations (3), (4), and (5) of this bijection can actually be interpreted as "rotations", as it is illustrated by an example (figure 2).

To verify that this correspondence is actually a bijection, one can use the fact, that the points (3), (4), and (5) are if and only if conditions. So when starting with a non-crossing tree N, we can read above operations the other way around, where point (2) means, that the first son of the root of N will be the root of the constructed ternary tree T^*.

The structure of the rotation correspondence between binary trees and planted plane trees gives a bijection involving the height of these trees: The number of planted plane trees with $n+1$ nodes and height $m+1$ is equal to the number of binary trees with n internal nodes and "left-height" m. The height of a node is the number of edges on the direct path from the root to this node and the height of a tree is then the maximum of all heights of the nodes in this tree. The "left-height" of a binary tree only counts the left edges on the path. In computer science the "left-height" of a binary tree is strongly related to the required stack-size when traversing a binary tree.

As an immediate consequence of the rotation correspondence between ternary trees and non-crossing trees we get the following result in a bijective way: The number of non-crossing trees with $n+1$ nodes and height $m+1$ is equal to the number of ternary trees with n internal nodes and "left-middle-height" m. The "left-middle-height" of a node in a ternary tree only counts the left and middle edges on the path.

*As one referee remarked, in the work of Dulucq and Penaud [2] this inverse bijection was described there in an inductive way.

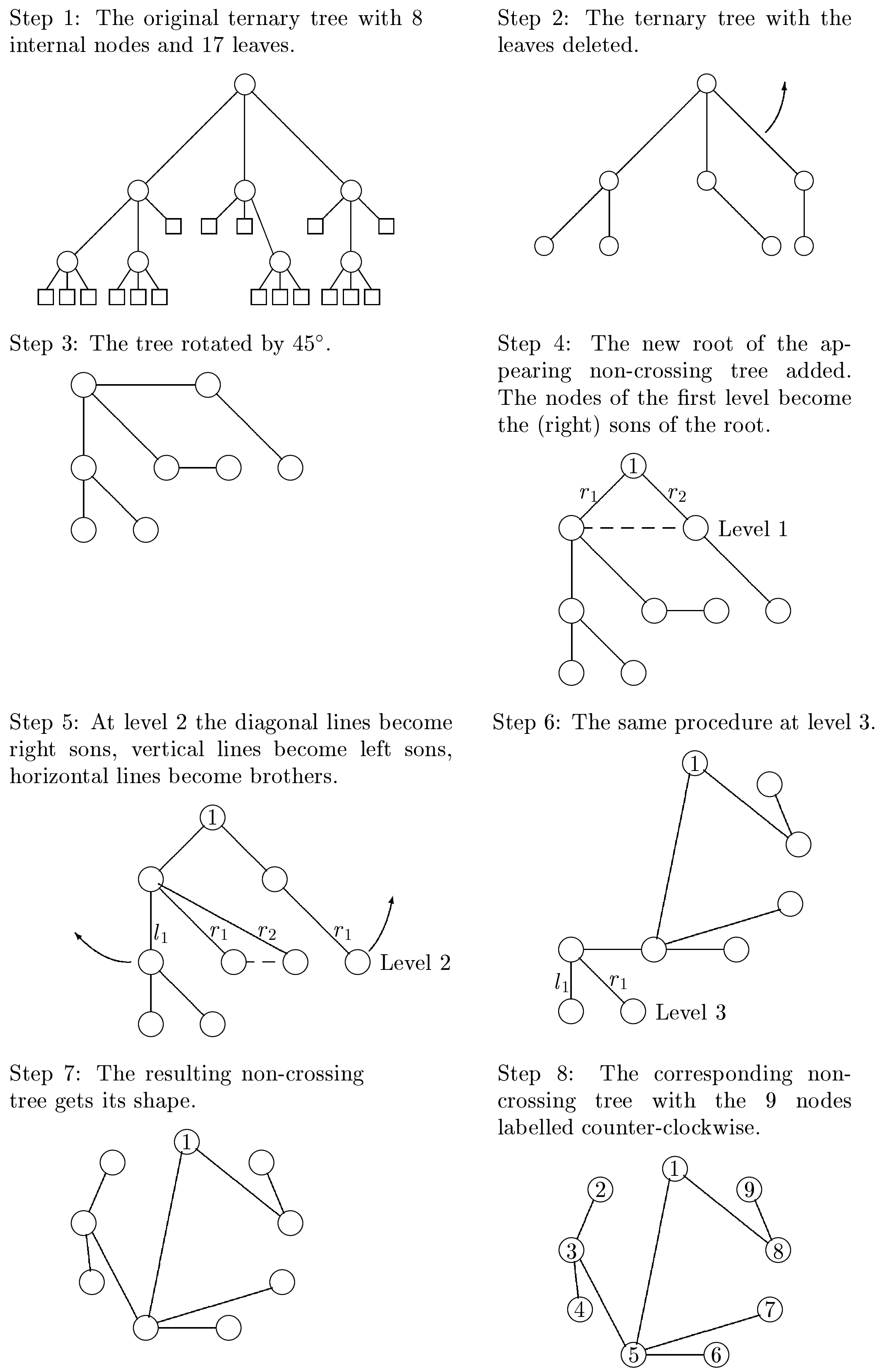


FIGURE 2. The rotation correspondence between non-crossing trees and ternary trees.

The expected height of non-crossing trees (all trees of the same size are considered to be equally likely) was recently studied by Deutsch and Noy in [1][†]. We can apply theorem 5 of [1] and get, that the expected "left-middle-height" of ternary trees with n nodes is asymptotically equivalent to $\frac{2}{3}\sqrt{3\pi n}$.

3. Two bijective proofs of a recurrence for ternary trees

Due to $T_n = \frac{1}{2n+1}\binom{3n}{n}$ the numbers T_n of ternary trees with n internal nodes satisfy the recurrence

$$3(3n-1)(3n-2)T_{n-1} = 2n(2n+1)T_n \,. \tag{3}$$

In the following, we will interpret this equation by a one-to-one correspondence between (somehow) marked objects of the families $\mathcal{T}_n$ resp. $\mathcal{T}_{n-1}$ of ternary trees with n resp. $n-1$ internal nodes.

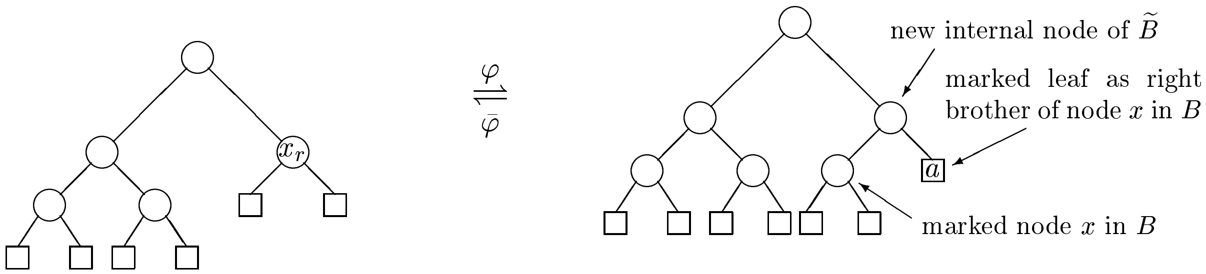
The first correspondence is a generalization of the well known bijective proof of the recurrence

$$2(2n-1)B_{n-1} = (n+1)B_n \tag{4}$$

for binary trees [7] (compare also [4]). A sketch of this proof is given in the sequel.

The binary tree B with $n = 5$ internal nodes. One node is marked with label x and coloured with colour r.

The binary tree $\widetilde{B}$ with $n = 6$ internal nodes. One leaf is marked with label a.



The binary tree B with $n = 5$ internal nodes. One node is marked with label x and coloured with colour l.

The binary tree $\widetilde{B}$ with $n = 6$ internal nodes. One leaf is marked with label a.

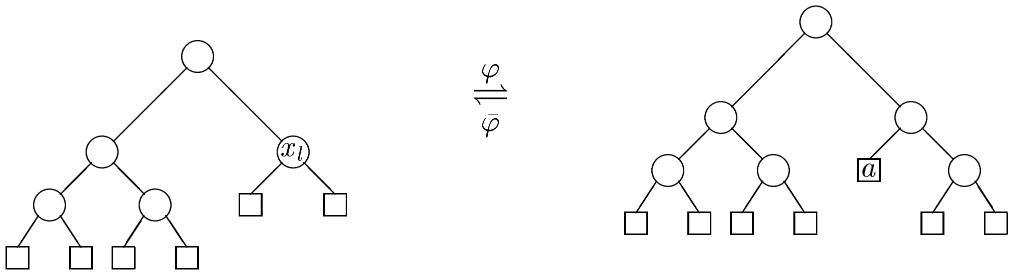


FIGURE 3. A bijective proof of the recurrence for binary trees.

Here the right side of this equation can be interpreted as the binary trees with n internal nodes, where one of the $n+1$ leaves is marked (let us say with a). Also the left side of this equation can be interpreted as the binary trees with $n-1$ internal nodes, where one of the $2n-1$ nodes is marked (with x) and coloured with one of the two colours $\{l, r\}$.

[†]This paper actually analyses a lot of parameters for non-crossing trees like number of end-points, number of vertices of a given degree, pathlength etc.

We start with a binary tree B with $n-1$ internal nodes, where one of the (internal or external) nodes is marked with x_l or x_r). To construct in a bijective way a binary tree $\widetilde{B}$ with n internal nodes, where one of the $n+1$ leaves is marked with a, we define the following map φ:

- Insert a new node as father of x.
-
  - If x is coloured with l then let the subtree x of B be the right son of the new node and add a marked leaf a as the left brother of x.
  - If x is coloured with r then append the subtree x of B as the left son of the inserted node and let a marked leaf a be its right brother.

It is not hard to see, that this construction is a bijection, when considering the inverse map $\bar{\varphi}$. We are starting with a binary tree $\widetilde{B}$ with n internal nodes, where one of its leaves is marked with a:

- The brother of the leaf a will be the marked node x in the binary tree B with $n-1$ internal nodes.
-
  - If a is the left brother of x, then x is coloured with l.
  - If a is the right brother of x, the x is coloured with r.
- Deleting the leaf a and substituting the father of x by the subtree with x as its root gives the tree B.

With this construction we get $\bar{\varphi}(\varphi(B)) = B$ and $\varphi(\bar{\varphi}(\widetilde{B})) = \widetilde{B}$. It is illustrated by an example (figure 3).

3.1. **First correspondence.** In order to avoid clumsy notations we use the following abbreviations (see figure 4):

- $f(x)$: The father of the node x.
- $l(x)$: The left son of x.
- $m(x)$: The middle son of x.
- $r(x)$: The right son of x.
- $cr(x)$: The "cyclic" right brother of x. (The next element in the cycle left son-middle son-right son-left son of $f(x)$, also iterations like $cr^2(x) = cr(cr(x))$ are used.)
- $cl(x)$: The "cyclic" left brother of x. (The next element in the cycle right son-middle son-left son-right son of $f(x)$.)

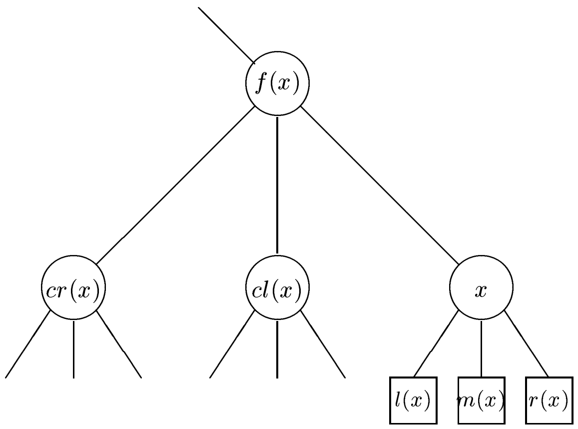


FIGURE 4. The node x and its "neighbours."

We will also use the following basic tree operations:

- Cut t from T (in symbols $C(t)$): The subtree with root t will be cut from the whole tree T and so we get two trees.
- Substitute t by s (in symbols $s \to t$): The subtree with root t in the tree T will be replaced by a second tree with root s.

- Two rotations:
  - Rotate s, t and w (in symbols $s \to t \to w$): This operation is only defined if t is not a descendant of s and s is not a descendant of w. It is the composition of three operations: $C(s)$, $s \to t$, $t \to w$, which means that first the node t will be replaced by the subtree with root s and then w will be replaced by t.
  - Rotate s, t and w and add r (in symbols $r \to s \to t \to w$): With the same conditions on s, t and w as before and r being the root of a second tree. It is the composition of three operations: $r \to s$, $s \to t$, $t \to w$.

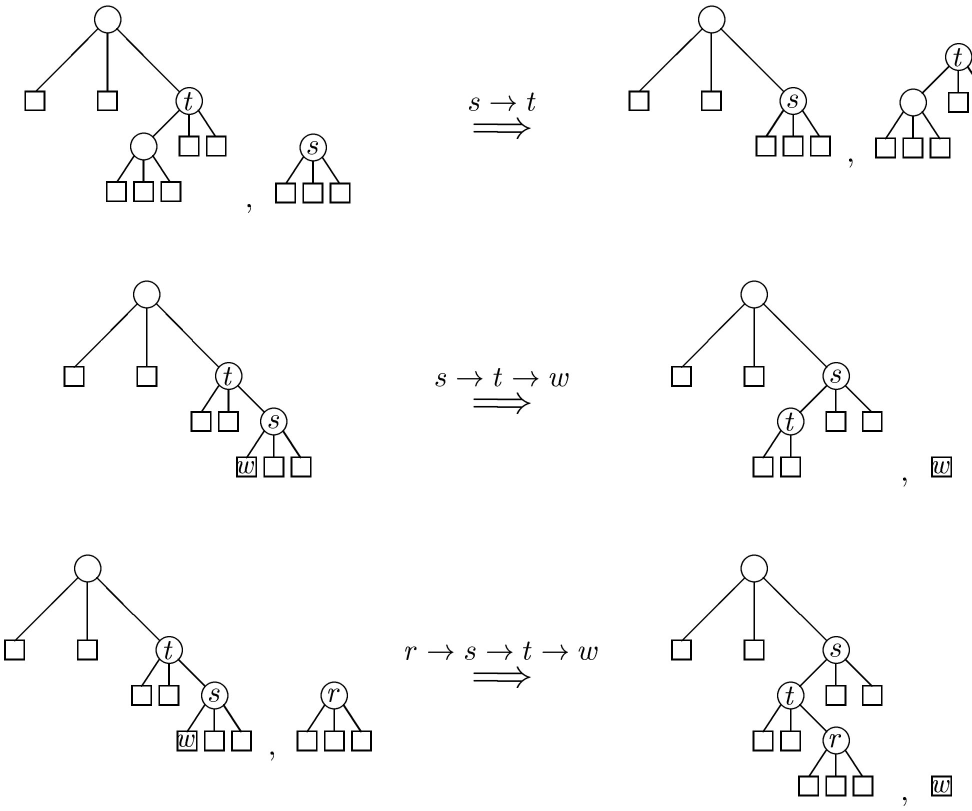


FIGURE 5. Tree operations

Now we are ready to describe equation (3) in a bijective combinatorial way.

Using the fact, that a ternary tree with n internal nodes has $2n+1$ leaves, we interpret the right side of equation (3) as the combinatorial objects of ternary trees with n internal nodes, where two *different* leaves are marked with the labels a resp. b.

To interpret the left side of this equation, we use the following easy construction with the combinatorial objects of ternary trees with $n-1$ internal nodes. In a first step, one of the $3n-2$ nodes of the ternary tree is marked with x and coloured with one of the three colours $\{l, m, r\}$. In a second step we add one extra leaf and mark one of these (together with the original tree) $3n-1$ nodes with label y.

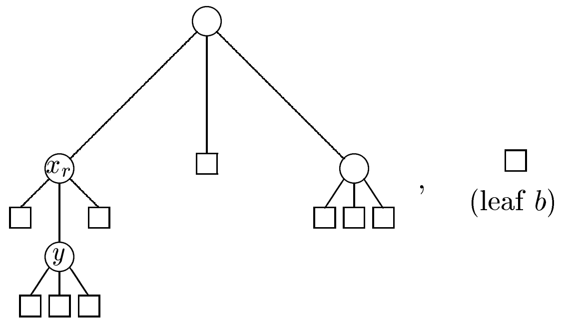
In the following we give an algorithm, that transforms objects of the left side to objects of the right side of the equation. With the above tree operations the algorithm can be described in the following way:

- Input: A ternary tree T with $n-1$ internal nodes where one of its (internal or external) nodes is marked with label x and coloured by l, m or r. Moreover one of these nodes or an extra leaf is marked with label y.

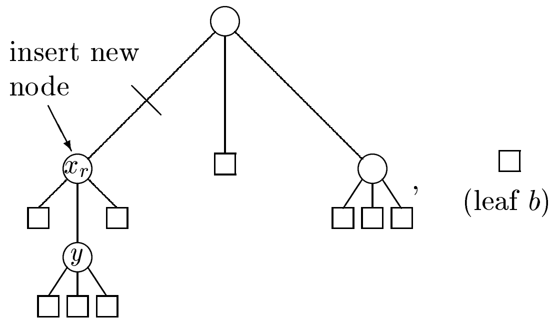
- A new node (let us call it p) and a marked leaf a will be inserted in the tree in the following way: p replaces x and according to its colours l, m or r, leaf a will be the left, middle or right son of p. Then x will be the cyclic right brother of leaf a.
- Next the extra leaf will be marked with label b.
-
  - If $p = f(a)$ is not a descendant of y: Marked leaf b substitutes y and then y will be the cyclic left brother of a, in symbols $b \to y$, $y \to cl(a)$.
  - If $p = f(a)$ is a descendant of y a rotation will be performed: $b \to f(a) \to y \to cl(a)$.
- Output: A ternary tree $\widetilde{T}$ with n internal nodes, where two different leaves are marked with a resp. b.

The algorithm is illustrated by a few examples (figure 6, figure 7, figure 8, and figure 9).

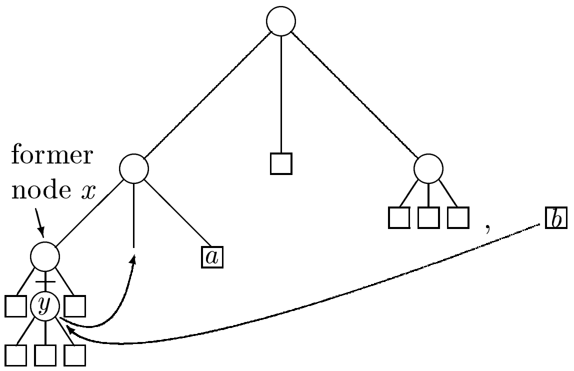
Step 1: The original ternary tree T with $n-1 = 4$ internal nodes. One node is marked with label x and coloured with r. One node is marked with label y.



Step 2: Insert a new node as father of x and a new leaf marked with a as its right son. x_r will be the cyclic right brother of a.



Step 3: Node y will be the cyclic left brother of a. Extra leaf b marks the tear off of y.



Step 4: The resulting tree $\widetilde{T}$ with $n = 5$ internal nodes. One leaf is marked with label a, another one with b.

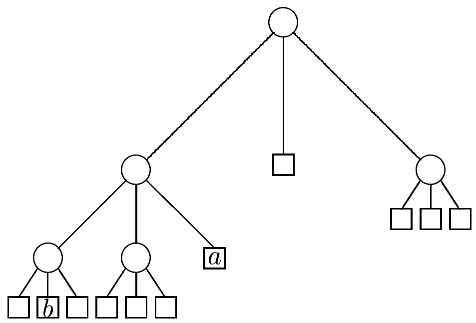
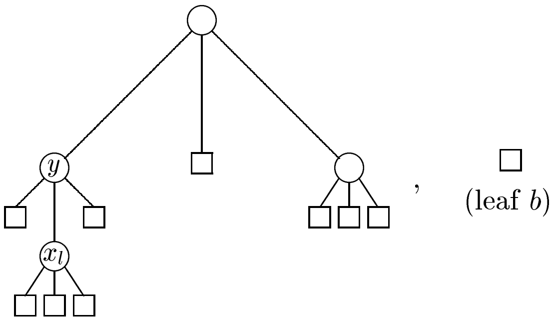
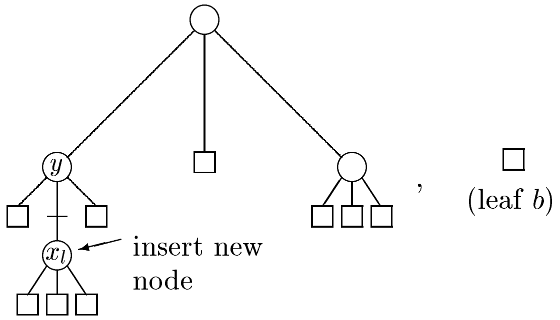


FIGURE 6. First Correspondence: The map from T to $\widetilde{T}$ (easy case).

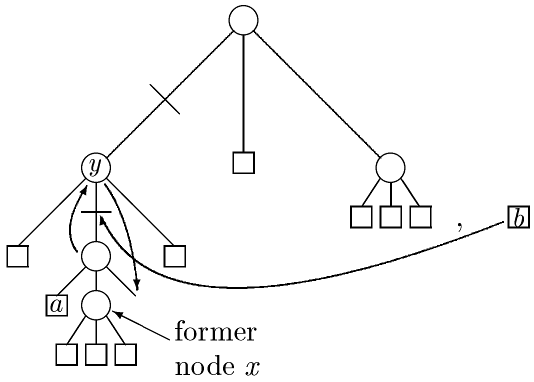
Step 1: The original ternary tree T with $n-1=4$ internal nodes. One node is marked with label x and coloured with l. One node is marked with label y.



Step 2: Insert a new node as father of x and a new leaf marked with a as its left son. x_l will be the cyclic right brother of a.



Step 3: Node y will be the cyclic left brother of a. Because a is a descendant of y, a rotation will be done. Extra leaf b marks here the tear off of $f(x)$.



Step 4: The resulting tree $\widetilde{T}$ with $n=5$ internal nodes. One leaf is marked with label a, another one with b.

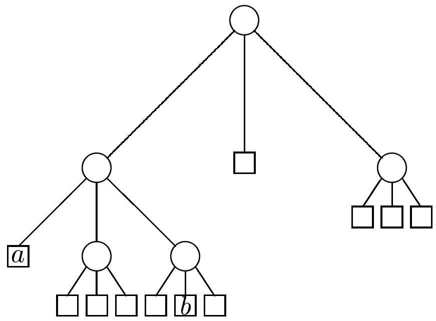
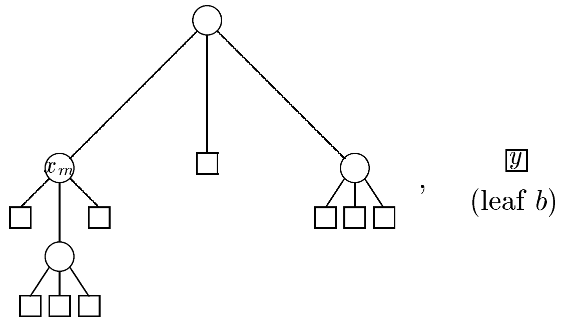
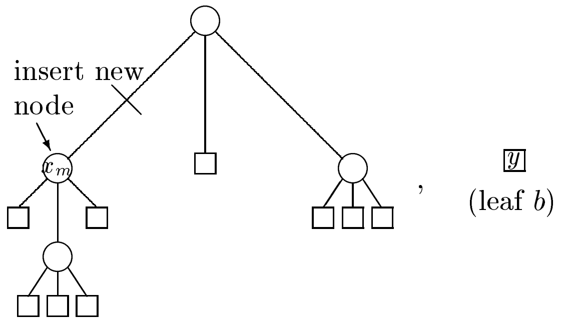


FIGURE 7. First Correspondence: The map from T to $\widetilde{T}$ (a rotation is done).

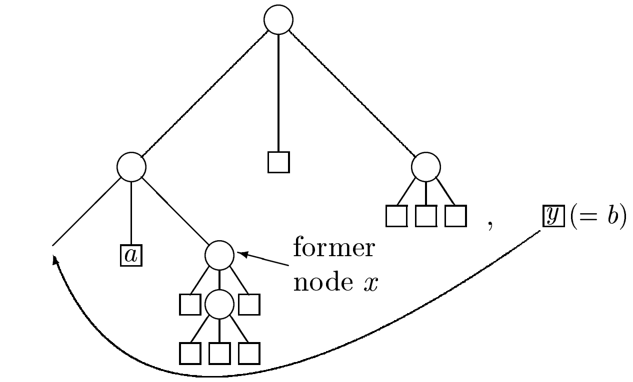
Step 1: The original ternary tree T with $n-1=4$ internal nodes. One node is marked with label x and coloured with m. One node (the extra leaf) is marked with label y.



Step 2: Insert a new node as father of x and a new leaf marked with a as its middle son. x_m will be the cyclic right brother of a.



Step 3: Node y (the extra leaf) will be the cyclic left brother of a and marked with b.



Step 4: The resulting tree $\widetilde{T}$ with $n=5$ internal nodes. One leaf is marked with label a, another one with b.

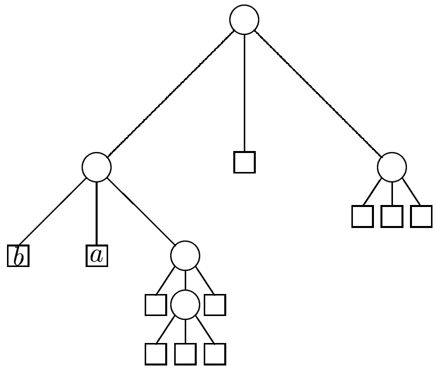
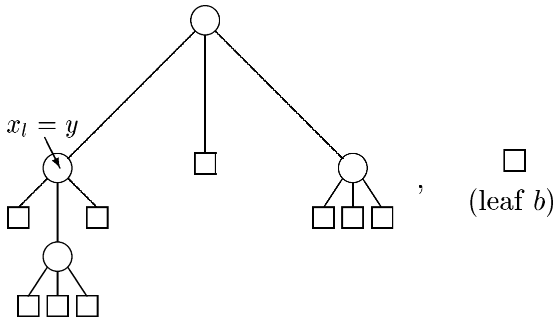
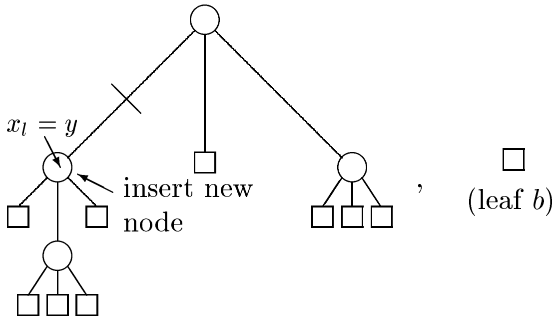


FIGURE 8. First Correspondence: The map from T to $\widetilde{T}$ (the extra leaf is marked).

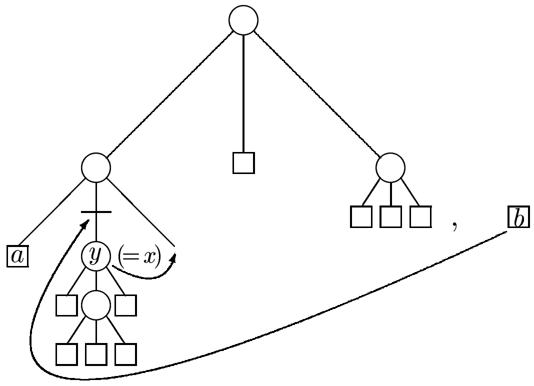
Step 1: The original ternary tree T with $n-1=4$ internal nodes. One node is marked with label x and coloured with l. The same node is marked with label y.



Step 2: Insert a new node as father of x and a new leaf marked with a as its left son. x_l will be the cyclic right brother of a.



Step 3: Node y $(= x)$ will be the cyclic left brother of a. Extra leaf b marks the tear off of y.



Step 4: The resulting tree $\widetilde{T}$ with $n=5$ internal nodes. One leaf is marked with label a, another one with b.

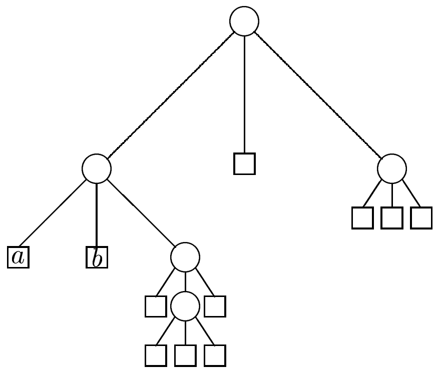


FIGURE 9. First Correspondence: The map from T to $\widetilde{T}$ (one node marked with x and y).

To show that this algorithm is constructing a one-to-one correspondence, we give the inverse map:

- Input: A ternary tree $\widetilde{T}$ with n internal nodes, where two different leaves are marked with a resp. b.
- The cyclic left brother of leaf a will be marked with y.
-
  - If leaf b is not a descendant of y, we will do the following: Cut y and substitute leaf b. ($C(y)$, $y \to b$).
  - If leaf b is a descendant of y make the rotation $y \to f(a) \to b$.
- The cyclic right brother of a will be marked with label x.
-
  - If a is the left son of its father then colour x with l.
  - If a is the middle son of its father then colour x with m.
  - If a is the right son of its father then colour x with r.
- Cut x and substitute $f(a)$, in symbols $C(x)$, $x \to f(a)$.
- Output: A ternary tree T with $n-1$ internal nodes where one of its (internal or external) nodes is marked with label x and coloured by l, m or r. Moreover one of these nodes or an extra leaf (the leaf marked with b in $\widetilde{T}$) is marked with label y.

It is not hard to see, that this algorithm can be extended to prove the recurrence

$$t(tn-1)(tn-2)\dots(tn-t+1)T_{n-1}^{[t]} = \big((t-1)n+1\big)\big((t-1)n\big)\dots\big((t-1)n-t+3\big)T_n^{[t]}$$

for general t-ary trees, where $T_n^{[t]}$ denotes the number of t-ary trees with n internal nodes.

The right side of this equation can be interpreted combinatorially as t-ary trees with n internal nodes, where $t-1$ *different* leaves are marked with labels a, resp. $b_1, b_2, \dots, b_{t-2}$. To interpret the left side of this equation, we iterate the previously described construction for ternary trees with $n-1$ internal nodes. In a first step, one of the $tn-t+1$ nodes of the t-ary tree is marked with x and coloured with one of the t colours $\{1, 2, \dots, t\}$. In a second step we add one extra leaf and mark one of these (together with the original tree) $tn-t+2$ nodes with label y_1. Iterating the second step of adding a leaf and marking $t-2$ times we add in the last step a leaf and mark one of the resulting $tn-1$ nodes with label y_{t-2}. Then the presented algorithm for ternary trees can be adapted to prove the above recurrence for t-ary trees:

- Input: A t-ary tree T with $n-1$ internal nodes, where one of its (internal or external) nodes is marked with label x and coloured with a colour from $1, 2, \dots, t$. Moreover for all $k = 1, 2, \dots, t-2$ one of these nodes or one of the first k extra leaves is marked with label y_k.
- A new node (let us call it p) and a marked leaf a will be inserted in the tree in the following way: p replaces x and according to its colour $1, 2, \dots, t$ leaf a will be the first, second, …, t-th son of p. Then x will be the cyclic right brother of leaf a.
- Iterate the following steps for all $k = 1, 2, \dots, t-2$.
-
  - The k-th extra leaf will be marked with label b_k.
  -
    - If $p = f(a)$ is not a descendant of y_k: Marked leaf b_k substitutes y_k and then y_k will be the $k+1$-th cyclic right brother of a, in symbols $b_k \to y_k$, $y_k \to cr^{k+1}(a)$.
    - If $p = f(a)$ is a descendant of y_k a rotation will be done: $b_k \to f(a) \to y_k \to cr^{k+1}(a)$.
- Output: A t-ary tree $\widetilde{T}$ with n internal nodes, where $t-1$ different leaves are marked with labels a, resp. $b_1, b_2, \dots, b_{t-2}$.

3.2. **Second correspondence.** For the second bijection we extend the recurrence (3) by a factor n:

$$(3n)(3n-1)(3n-2)T_{n-1} = n(2n)(2n+1)T_n\,. \tag{5}$$

Using the fact, that a ternary tree with n internal nodes has $2n+1$ leaves, we interpret the right side of (5) as the combinatorial objects of ternary trees with n internal nodes, where one internal node is marked with label a and two *different* leaves are marked with labels b_1 resp. b_2.

To interpret the left side of above equation, we use the following construction for ternary trees with $n-1$ internal nodes. In a first step, one of the $3n-2$ nodes of the ternary tree is marked with x. In a second step we add one extra leaf and mark one of these (together with the original tree) $3n-1$ nodes with label y_1. In a third step we add one further extra leaf and mark one of the resulting $3n$ nodes with label y_2.

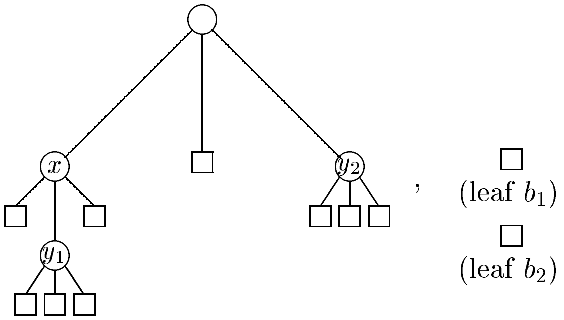
The following algorithm transforms objects of the left side to objects of the right side of equation (5):

- Input: A ternary tree T with $n-1$ internal nodes, where one of its (internal or external) nodes is marked with label x. Further one of the nodes of T or the first extra leaf is marked with label y_1, and one of the nodes of T or one of the two extra leaves is marked with label y_2.
- A new node, marked with label a will be inserted in the tree in the following way: a replaces x and x will be the left son of a.
- The first extra leaf will be marked with label b_1.
-
  - If a is not a descendant of y_1: Marked leaf b_1 substitutes y_1 and then y_1 will be the middle son of a, in symbols $b_1 \to y_1$, $y_1 \to m(a)$.
  - If a is a descendant of y_1 a rotation will be done: $b_1 \to a \to y_1 \to m(a)$.
- The second extra leaf will be marked with label b_2.
-
  - If a is not a descendant of y_2: Marked leaf b_2 substitutes y_2 and then y_2 will be the right son of a, in symbols $b_2 \to y_2$, $y_2 \to r(a)$.
  - If a is a descendant of y_2 a rotation will be done: $b_2 \to a \to y_2 \to r(a)$.

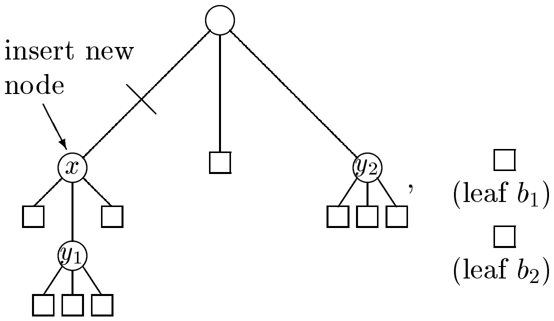
- Output: A ternary tree $\widetilde{T}$ with n internal nodes, where one internal node is marked with a and two different leaves are marked with labels b_1 resp. b_2.

One could illustrate all appearing cases by drawings, but it seems sufficient to give as an example the main case, where no rotation is done and no extra leaf is marked (figure 10):

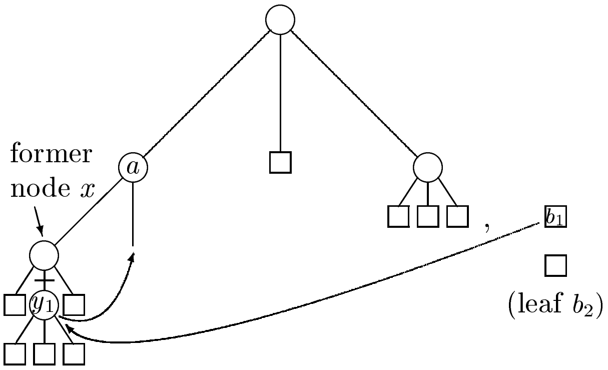
Step 1: The original ternary tree T with $n-1=4$ internal nodes. One node is marked with label x, one is marked with label y_1, and one is marked with label y_2.



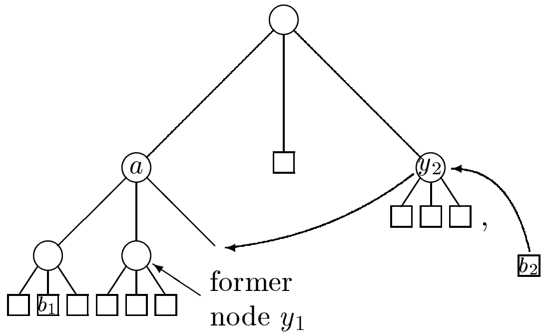
Step 2: Insert a new node marked with a as father of x. x will be the left son of a.



Step 3: Node y_1 will be the middle son of a. Extra leaf b_1 marks the tear off of y_1.



Step 4: Node y_2 will be the right son of a. Extra leaf b_2 marks the tear off of y_2.



Step 5: The resulting tree $\widetilde{T}$ with $n=5$ internal nodes. One internal node is marked with label a and two different leaves are marked with labels b_1 resp. b_2.

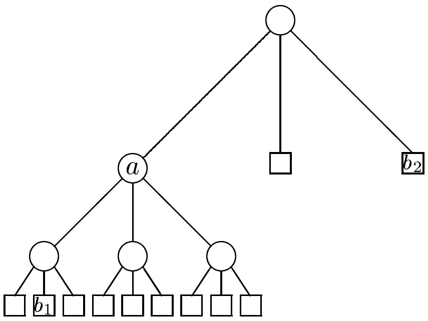


FIGURE 10. Second Correspondence: The map from T to $\widetilde{T}$ (easy case).

For the sake of completeness we give also the inverse map:

- Input: A ternary tree $\widetilde{T}$ with n internal nodes, where one internal node is marked with a and two different leaves are marked with labels b_1 resp. b_2.
- The right son of leaf a will be marked with y_2.
- – If leaf b_2 is not a descendant of y_2, we will do the following: Cut y_2 and substitute leaf b_2. $(C(y_2), y_2 \to b_2)$.
  – If leaf b_2 is a descendant of y_2 make the rotation $y_2 \to a \to b_2$.
- The middle son of a will be marked with label y_1.
- – If leaf b_1 is not a descendant of y_1, we will do the following: Cut y_1 and substitute leaf b_1. $(C(y_1), y_1 \to b_1)$.
  – If leaf b_1 is a descendant of y_1 make the rotation $y_1 \to a \to b_1$.
- The left son of a will be marked with label x.
- Cut x and substitute a, in symbols $C(x)$, $x \to a$.
- Output: A ternary tree T with $n-1$ internal nodes where one of its (internal or external) nodes is marked with label x. Further one of the nodes of T or the first extra leaf (the leaf marked with b_1 in $\widetilde{T}$) is marked with label y_1, and one of the nodes of T or one of the two extra leaves is marked with label y_2.

References

[1] E. Deutsch and M. Noy. Statistics on Non-crossing Trees. *to appear*, 12 pages.
[2] S. Dulucq and J.-G. Penaud. Cordes, arbres et permutations. *Discrete Mathematics*, 117:89–105, 1993.
[3] P. Flajolet and M. Noy. Analytic Combinatorics of Non-crossing Configurations. *Discrete Mathematics*, 204:203–229, 1999.
[4] D. Foata and D. Zeilberger. A classic proof of a recurrence for a very classical sequence. *Journal of Combinatorial Theory, Series A*, 80:380–384, 1997.
[5] M. Noy. Enumeration of non-crossing trees on a circle. *Discrete Mathematics*, 180:301–313, 1998.
[6] Problem E 3170. Noncrossing trees. *American Mathematical Monthly*, 96:359–361, 1989.
[7] J.-L. Rémy. Un procédé iteratif de denombrement d'arbres binaires et son application à leur generation aléatoire. *RAIRO Informatique théorique*, 19:179–195, 1985.
[8] R. Stanley. *Enumerative Combinatorics, Vol. 2.* Cambridge University Press, 1999.
[9] D. Stanton and D. White. *Constructive Combinatorics.* Springer, 1986.

†Institut für Algebra und Computermathematik, Technische Universität Wien, Wiedner Hauptstrasse 8–10, A-1040 Wien, Austria.
E-mail address: Alois.Panholzer@tuwien.ac.at

‡The John Knopfmacher Centre for Applicable Analysis and Number Theory, Mathematics Department, University of the Witwatersrand, P.O. Wits, 2050 Johannesburg, South Africa.
E-mail address: helmut@gauss.cam.wits.ac.za
WWW-address: http://www.wits.ac.za/helmut/index.htm