FPSAC 2013 Paris, France DMTCS proc. **AS**, 2013, 49–60

Combinatorics of non-ambiguous trees‡

Jean-Christophe Aval Adrien Boussicault
Mathilde Bouvel Matteo Silimbani

LaBRI - CNRS, Université de Bordeaux, 351 Cours de la Libération, 33405 Talence, France.

Abstract. This article investigates combinatorial properties of non-ambiguous trees. These objects we define may be seen either as binary trees drawn on a grid with some constraints, or as a subset of the tree-like tableaux previously defined by Aval, Boussicault and Nadeau. The enumeration of non-ambiguous trees satisfying some additional constraints allows us to give elegant combinatorial proofs of identities due to Carlitz, and to Ehrenborg and Steingrímsson. We also provide a hook formula to count the number of non-ambiguous trees with a given underlying tree. Finally, we use non-ambiguous trees to describe a very natural bijection between parallelogram polyominoes and binary trees.

Résumé. Cet article s'intéresse aux propriétés combinatoires des arbres non-ambigus. Ces objets, que nous définissons, peuvent être vus soit comme des arbres dessinés sur une grille sous certaines contraintes, soit comme un sous-ensemble des tableaux boisés précédemment définis par Aval, Boussicault et Nadeau. L'énumération des arbres non-ambigus satisfaisant des contraintes supplémentaires nous permet de donner des preuves combinatoires élégantes d'identités dues à Carlitz, et à Ehrenborg et Steingrímsson. Nous donnons aussi une formule des équerres pour le comptage des arbres non-ambigus dont l'arbre sous-jacent est fixé. Enfin, nous utilisons les arbres non-ambigus pour décrire une bijection très naturelle entre polyominos parallélogrammes et arbres binaires.

Keywords: tree, polyomino, non-ambiguous tree, tree-like tableau, hook formula, Bessel function

1 Introduction

It is well known that Catalan numbers $C_n = \frac{1}{n+1}\binom{2n}{n}$ enumerate many combinatorial objects, such as binary trees and parallelogram polyominoes. Several bijective proofs in the literature show that parallelogram polyominoes are enumerated by Catalan numbers, the two most classical being Delest-Viennot's bijection with Dyck paths [DV84] and Viennot's bijection with bicolored Motzkin paths [DV84].

In this paper we demonstrate a bijection –which we believe is more natural– between binary trees and parallelogram polyominoes. In some sense, we show that parallelogram polyominoes may be seen as two-dimensional drawings of binary trees. This point of view gives rise to a new family of objects – we call them *non-ambiguous trees* – which are particular compact embeddings of binary trees in a grid.

†This short paper is an extended abstract of [ABBS], where details of the proofs are provided.

‡All authors are supported by ANR – PSYCO project (ANR-11-JS02-001).

The tree structure of these objects leads to a hook formula for the number of non-ambiguous trees with a given underlying tree. Unlike the classical hook formula for trees due to Knuth (see [Knu98], §5.1.4, Exercise 20), this one is defined on the edges of the tree.

Non-ambiguous trees are in bijection with permutations such that all their (strict) excedances stand at the beginning of the permutation word. Ehrenborg and Steingrímsson in [ES00] give a closed formula (involving Stirling numbers of the second kind) for the number of such permutations. We show that this formula can be easily proved using non-ambiguous trees and a variation of the insertion algorithm for tree-like tableaux introduced in [ABN11]. Indeed, non-ambiguous trees can also be seen as a subclass of tree-like tableaux, objects defined in [ABN11], that are in bijection with permutation tableaux [SW07] or alternative tableaux [Nad11, Vie07].

A particular subclass of non-ambiguous trees leads to unexpected combinatorial interpretations. We study complete non-ambiguous trees, defined as non-ambiguous trees such that their underlying binary tree is complete, and show that their enumerating sequence is related to the formal power series of the logarithm of the Bessel function of order 0. This gives rise to new combinatorial interpretations of some identities due to Carlitz [Car63].

The paper is organized as follows: in Section 2 we define non-ambiguous trees. Then, in Section 3 we give the enumeration of non-ambiguous trees satisfying certain constraints: those contained into a given rectangular box, and those with a fixed underlying tree. Section 4 introduces the family of complete non-ambiguous trees, and studies the relations between this family and the Bessel function. Finally, in Section 5 we describe our new bijection between binary trees and parallelogram polyominoes.

2 Definitions and notations

In this paper, trees are embedded in a bidimensional grid $\mathbb{N} \times \mathbb{N}$. The grid is not oriented as usual: the x-axis has south-west orientation, and the y-axis has south-east orientation, as shown on Figure 1.

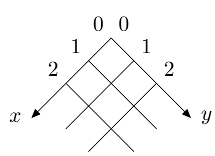


Fig. 1: The underlying grid for non-ambiguous trees

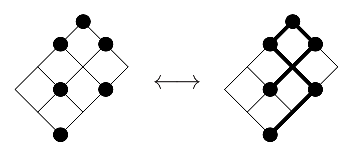
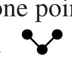


Fig. 2: The edges of a non-ambiguous tree are not necessary

Every x-oriented (resp. y-oriented) line will be called column (resp. row). Each column (resp. row) on this grid is numbered with an integer corresponding to its y (resp. x) coordinate. A vertex v located on the intersection of two lines has the coordinate representation: $(X(v), Y(v))$.

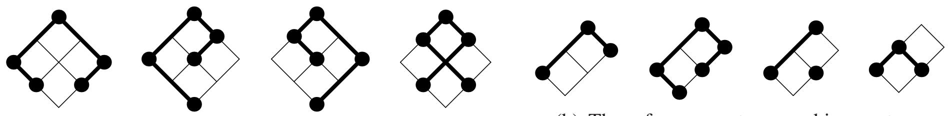
A non-ambiguous tree may be seen as a binary tree embedded in the grid in such a way that the embedding of its vertices in the grid determines the tree completely (i.e. determines its edges – see Figure 2).

Formally, a *non-ambiguous tree* of size n is a set A of n points $(x, y) \in \mathbb{N} \times \mathbb{N}$ such that:

1. $(0, 0) \in A$; we call this point the *root* of A;
2. given a non-root point $p \in A$, there exists one point $q \in A$ such that $Y(q) < Y(p)$ and $X(q) = X(p)$, or one point $r \in A$ such that $X(r) < X(p)$, $Y(r) = Y(p)$, but not both (which means that the pattern  is avoided);

3. there is no empty line between two given points: if there exists a point $p \in A$ such that $X(p) = x$ (resp. $Y(p) = y$), then for every $x' < x$ (resp. $y' < y$) there exists $q \in A$ such that $X(q) = x'$ (resp. $Y(q) = y'$).

Figure 3 shows some examples and counterexamples of non-ambiguous trees.



(a) Four non-ambiguous trees (b) These four are not non-ambiguous trees

Fig. 3: Some examples and counterexamples of non-ambiguous trees

It is straightforward that a non-ambiguous tree A has a tree structure: except for the root, every point $p \in A$ has a unique parent, which is the nearest point q preceding p in the same row (resp. column). In this case, we will say that p is the *right child* (resp. *left child*) of q. In this paper, we orient every edge of a tree from the root to the leaves. We shall denote by $\mathcal{T}(A)$ the underlying binary tree associated to A.

Figure 4 shows all the non-ambiguous trees of size 4, grouping inside a rectangle those having the same underlying binary tree.

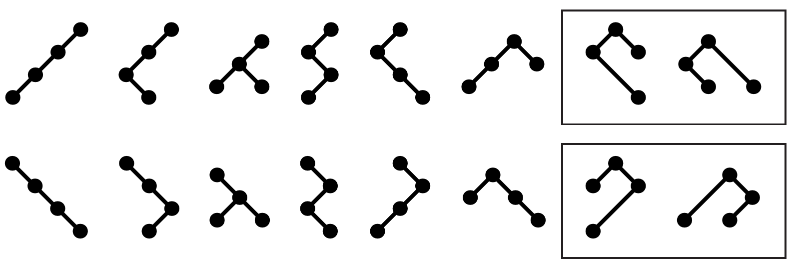


Fig. 4: The 16 non-ambiguous trees of size 4

Remark 1 *A tree-like tableau [ABN11] of size n is a set of n points placed in the boxes of a Ferrers diagram such that conditions 1, 2, 3 defining non-ambiguous trees are satisfied. Figure 5 shows an example of a tree-like tableau of size 7. It should be clear that non-ambiguous trees are in bijection with tree-like tableaux with rectangular shape.*

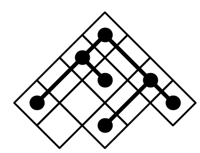


Fig. 5: A tree-like tableau

3 Enumeration of non-ambiguous trees

Non ambiguous trees of size n are in bijection with permutations of size n with all their strict excedances at the beginning. This fact is a consequence of Lemma 5 in [SW07] and of results proved in [ABN12]. The sequence $(a_n)_{n\geq 1}$ counting the number of non-ambiguous tree of size n is referenced in [Slo] as `A136127` $= [1, 2, 5, 16, 63, 294, 1585, 9692, \ldots]$, but no simple formula is known.

3.1 Non-ambiguous trees inside a fixed box

Given a non-ambiguous tree, its x-size (resp. y-size) may be defined as the maximum of the x-coordinate (resp. y-coordinate) of its points. The aim of this subsection is to give a formula for the number $A(k, \ell)$

of non-ambiguous trees with x-size equal to k and y-size equal to ℓ. We denote by $c(n, j)$ the unsigned Stirling numbers of the first kind, *i.e.* the number of permutations of size n with exactly j disjoint cycles.

Proposition 2 *For every integers n, ℓ, one has:*

$$\sum_{k=1}^{n} c(n,k)\, A(k,\ell) = n^{\ell-1}\, n!\,. \tag{1}$$

We may inverse (1) to get:

$$A(k,\ell) = \sum_{i=1}^{k} (-1)^{k-i}\, S(k,i)\, i!\, i^{\ell-1}, \tag{2}$$

where $S(k, i)$ denotes the Stirling numbers of the second kind, *i.e.* the number of partitions of a set of k elements into i non-empty parts. Since (from [SW07, ABN12]) $A(k, \ell)$ is equal to the number of permutations of size $k+\ell$ with exactly k strict excedances in position $1, 2, \ldots, k$, Equation (1) is equivalent to Corollary 6.6 in [ES00]. In that paper, (2) is obtained through an inclusion-exclusion argument, and (1) is deduced by inversion.

In our setting, we may interpret $c(n, k)$ through tree-like tableaux. We refer to [ABN11] for definitions, and basic properties. As mentioned in Remark 1, non-ambiguous trees are nothing but tree-like tableaux with a rectangular shape. Permutations of size n with exactly j disjoint cycles are in bijection with tree-like tableaux of size n with exactly j points in their first row (a consequence of Theorem 4.2 in [Bur07] and of the results contained in [ABN12]). We are thus able to interpret (1) with unified objects: tree-like tableaux and non-ambiguous trees.

With these tools, the proof of Proposition 2 is a *simple* use of a variation of the insertion algorithm defined on tree-like tableaux in [ABN11], but we cannot give the details in this extended abstract.

3.2 Non-ambiguous trees with a fixed underlying tree: a new hook formula

Let T be a binary tree. We define $NA(T)$ as the number of non-ambiguous trees A such that their underlying binary tree $\mathcal{T}(A)$ is T. The aim of this section is to get a formula for $NA(T)$: this will be done by Proposition 6, which shows that $NA(T)$ may be expressed by a new and elegant hook formula on the edges of T. To do this, we encode any non-ambiguous tree A by a triple $\Phi(A) = (T, \alpha_L, \alpha_R)$ where T is a binary tree, and α_L (resp. α_R) is a word called the left (resp. right) code of A. To distinguish the vertices of A, we label them by integers from 1 to the size of A, as shown on Figure 6.

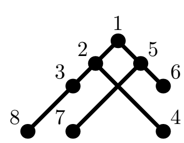
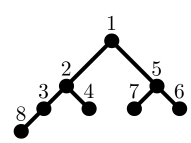
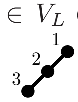
$A =$  , $T =$  .

Fig. 6: A non-ambiguous tree A with labeled vertices, and the associated binary tree T

The first entry in $\Phi(A)$ is the underlying binary tree T associated to A. Observe that we keep the labels on vertices when we extract the underlying binary tree. Now we denote by V_L (resp. V_R) the set of the end points of the left (resp. right) edges of A, which gives $V_L = \{2, 3, 8, 7\}$ and $V_R = \{5, 6, 4\}$ on the example in Figure 6. The definition of non-ambiguous trees ensures that the set $\{X(v), v \in V_L\}$ is the interval $\{1, \ldots, |V_L|\}$. Thus for $i = 1, \ldots, |V_L|$, we may set $\alpha_L(i)$ as the unique label $v \in V_L$ such that $X(v) = i$, and we proceed symmetrically for α_R. On the example of Figure 6, we have: $\alpha_L = 2378$ and $\alpha_R = 564$. Our starting point is the following lemma.

Lemma 3 *The application* Φ *which sends* A *to the triple* (T, α_L, α_R) *is injective.*

Proof: The proof shall not be detailed here: it is elementary to check that, given the tree T, the left and right codes uniquely determine the coordinates of every point in A. $\square$

Lemma 3 allows us to encode a non-ambiguous tree A by a triple (T, α_L, α_R), where T is a binary tree, and α_L (resp. α_R) is a word in which every label $v \in V_L$ (resp. V_R) appears exactly once. Of course, Φ is not surjective on such triples. If we take $T =$ 

, it should be clear that α_L is forced to be 23. Consequently, our next task is to characterize the pairs of codes (α_L, α_R) which are compatible with a given binary tree T, *i.e.* such that (T, α_L, α_R) is in the image of Φ. In order to describe this characterization, we need to define partial orders on the sets V_L and V_R. The pairs (α_L, α_R) of compatible codes will be seen to correspond to pairs of linear extensions of the posets V_L and V_R. The posets are defined as follows: given $a, b \in V_L$ (resp. V_R), we say that $a \leq b$ if and only if there exists a path in the oriented tree starting from a and ending at b. Figure 7 and Figure 9 (with minima at the top) illustrate this notion.

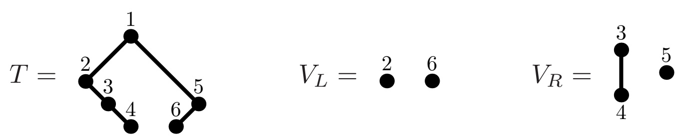


Fig. 7: The posets V_L and V_R of a tree T

The next lemma is the crucial step to prove Proposition 6.

Lemma 4 *Given a binary tree* T*, the pairs of codes compatible with* T *are exactly the pairs* (α_L, α_R) *where* α_L *is a linear extension of* V_L *and* α_R *is a linear extension of* V_R*.*

Figure 8 gives these compatible codes, together with the corresponding non-ambiguous trees, in the case of the tree T of Figure 7.

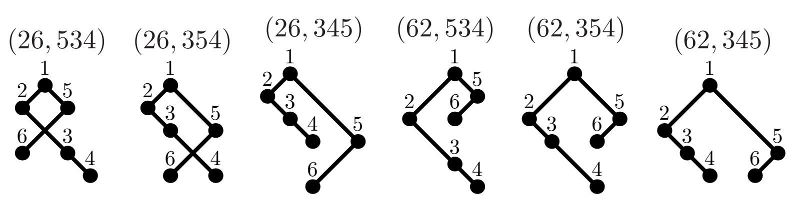


Fig. 8: Non-ambiguous trees of the tree T of Figure 7

Proof: We shall only give the main arguments of the proof.

Given a tree T, consider the map Φ_T defined on the set of non-ambiguous trees with underlying tree T as follows:

$$\Phi_T(A) := (\alpha_L, \alpha_R),$$

where $\Phi(A) = (T, \alpha_L, \alpha_R)$. Since Φ is injective (by Lemma 3), so is Φ_T. It remains to prove that the image of Φ_T is $\mathcal{L}(V_L) \times \mathcal{L}(V_R)$, where we denote by $\mathcal{L}(P)$ the set of linear extensions of a poset P.

First, we prove that $Im\,\Phi_T \subseteq \mathcal{L}(V_L) \times \mathcal{L}(V_R)$. Without loss of generality, we will prove that $\alpha_L \in \mathcal{L}(V_L)$. We need to prove that, if $s <_{V_L} t$, then s precedes t in α_L, which we shall write $s <_{\alpha_L} t$. If $s <_{V_L} t$, there exists a path in T starting from s and ending at t. When we go through the path, the X-coordinates of the vertices remain unchanged along right edges, while they increase along left edges. Since $s \neq t$, we have $X(s) < X(t)$, which is equivalent to $s <_{\alpha_L} t$.

Now the hard part is to prove that $\mathcal{L}(V_L) \times \mathcal{L}(V_R) \subseteq Im\,\Phi_T$. Let $(\alpha_L, \alpha_R) \in \mathcal{L}(V_L) \times \mathcal{L}(V_R)$. It is always possible to use the triple (T, α_L, α_R) to build a set of points in the grid which we may denote by A: we just have to place the root at position $(0, 0)$ and every other vertex v of T at the position

$$\begin{cases} X(v) = i \ \text{ with } \ \alpha_L(i) = v \quad \text{and} \quad Y(v) = Y(\text{parent}(v)) & \text{if } v \in V_L; \\ X(v) = X(\text{parent}(v)) \quad \text{and} \quad Y(v) = j \ \text{ with } \ \alpha_R(j) = v & \text{if } v \in V_R. \end{cases}$$

The goal is to prove that A is a non-ambiguous tree, which is quite technical. The main steps are:

1. check that for every left (resp. right) edge (s, t) of T, we have $X(s) < X(t)$ (resp. $Y(s) < Y(t)$) in A;
2. prove that A avoids the pattern  ;
3. check that two different vertices in T occupy different positions in A. □

Now we come to the final step toward proving Proposition 6.

Lemma 5 *The Hasse diagrams of V_L and V_R are forests.*

Figure 9 shows an example of the forests obtained by computing the Hasse diagrams of V_L and V_R.

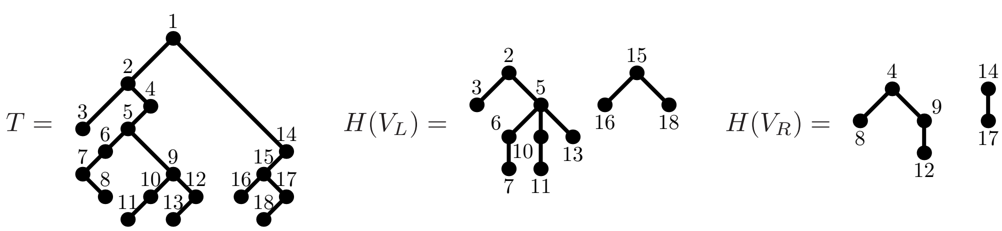


Fig. 9: The Hasse diagrams $H(V_L)$ and $H(V_R)$ of V_L and V_R are forests

Proof: We prove this proposition by contradiction. Suppose that there is a cycle in the Hasse diagram of V_R (the case of V_L is analogous). We can deduce from the poset structure that there are two paths in V_R starting from an element v and ending at w. This would imply that in the tree there are two different paths from v to w, and hence there would be a cycle in the tree. □

As a consequence the number of non-ambiguous trees with underlying tree T is given by the product of the results of Knuth's hook formula [Knu98] applied to the Hasse diagram of V_L and to the Hasse diagram of V_R. We can make this more precise. To do so, we associate to each edge an integer n_e. Given a left edge e (resp. right edge) the integer n_e is the number of left edges (resp. right edges) contained in the subtree whose root is the ending point of e, plus 1.

Proposition 6 *The number of non-ambiguous trees with underlying tree T is given by*

$$NA(T) = \frac{\#\{\textit{left edges}\}!\ \#\{\textit{right edges}\}!}{\displaystyle\prod_{e \in V_L} n_e \prod_{e \in V_R} n_e}. \tag{3}$$

This new hook formula is illustrated by Figure 10.

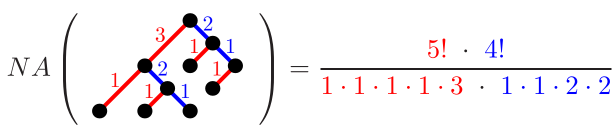


Fig. 10: A hook formula for non-ambiguous trees

Remark 7 *Equation (3) gives a way to compute the number of permutations of size n with all their strict excedances at the beginning, by summing over all binary trees T with n vertices.*

4 Complete non-ambiguous trees and Bessel function

4.1 Definition and enumeration of complete non-ambiguous trees

A non-ambiguous tree is *complete* whenever its vertices have either 0 or 2 children. An example of complete non-ambiguous tree can be found in Figure 11. A complete non-ambiguous tree has always an odd number of vertices. Moreover, as in complete binary trees, a complete non-ambiguous tree with $2k+1$ vertices has exactly k internal vertices, $k+1$ leaves, k right edges and k left edges. Denote by b_k the number of complete non-ambiguous trees with k internal vertices. The sequence $(b_k)_{k \geq 0}$ is known in [Slo] as `A002190` $= [1, 1, 4, 33, 456, 9460, \dots]$, and two remarkable identities satisfied by this sequence are given by Carlitz [Car63]. Propositions 8 and 10 give combinatorial interpretations for these identities.

Denote by C_n the number of complete binary trees with n internal vertices. It is well-known that $C_n = \frac{1}{n+1}\binom{2n}{n}$ is the n-th Catalan number, and that, for every $n \geq 0$, we have the identity:

$$C_{n+1} = \sum_{i+j=n} C_i C_j. \tag{4}$$

Proposition 8 gives a variant of this identity for complete non-ambiguous trees:

Proposition 8 *For every $n \geq 0$, we have:*

$$b_{n+1} = \sum_{i+j=n} \binom{n+1}{i}\binom{n+1}{j} b_i\, b_j. \tag{5}$$

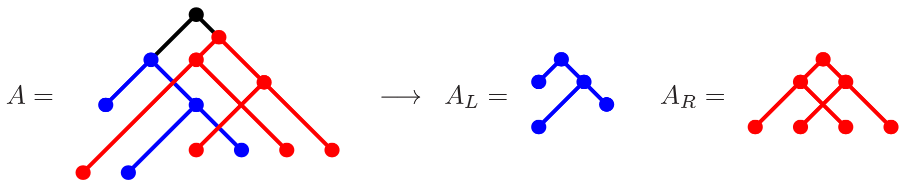


Fig. 11: The root suppression in a complete non-ambiguous tree

Proof: The proof of this proposition is similar to the classical proof of (4): the left (resp. right) subtree A_L (resp. A_R) of a complete non-ambiguous tree A with $n+1$ internal vertices is a complete non-ambiguous tree with i (resp. j) internal vertices, where $i + j = n$.

Figure 11 shows an example of left and right subtree of a complete non-ambiguous tree.

Hence, in order to construct an arbitrary complete non-ambiguous tree A with $n + 1$ internal vertices, we need to choose:

- the number i of internal vertices contained in A_L (i may range between 0 and n, the number j is equal to $n - i$);
- the complete non-ambiguous tree structure of A_L (resp. A_R) – we have b_i (resp. b_j) choices;
- the way of interlacing the right (resp. left) edges of A_L and A_R.

We denote by $u_1, u_2, \dots, u_i$ (resp. $v_1, v_2, \dots, v_j$) the end points of the right edges in A_L (resp. A_R) such that if $k < l$, then $Y(u_k) < Y(u_l)$ (resp. $Y(v_k) < Y(v_l)$), and by u_0 and v_0 the roots of A_L and A_R. Now, if we want to interlace the right edges in A_L with those in A_R, we need to decide at what positions we want to insert the vertices $u_1, u_2, \dots, u_i$ with respect to $v_0, v_1, v_2, \dots, v_j$, saving the relative order among $u_0, u_1, u_2, \dots, u_i$ and $v_0, v_1, v_2, \dots, v_j$. A vertex u_k can be placed either to the left of v_0, or between v_t and v_{t+1} ($0 \leq t \leq j-1$), or to the right of v_j.

Hence, we must choose the i positions of $u_1, u_2, \dots, u_i$ (multiple choices of the same position are allowed) among $j + 2$ possible ones. This shows that there are $\left(\binom{j+2}{i}\right) = \binom{i+j+1}{i} = \binom{n+1}{i}$ ways of interlacing the right edges of the subtrees A_L and A_R, where $\left(\binom{a}{b}\right)$ denotes the number of way of choosing b objects within a, with possible repetitions.

Analogous arguments apply to left edges. In this case, we have $\left(\binom{i+2}{j}\right) = \binom{n+1}{j}$ different interlacements. This ends the proof. $\square$

Corollary 9 *The sequence b_k satisfies the following identity*

$$\sum_{k\geq 0} b_k \frac{x^{2(k+1)}}{((k+1)!2^{k+1})^2} = -\ln\left(J_0(x)\right). \tag{6}$$

Proof: It is well known (see, e.g., [AS64]) that the Bessel function $J_0(x) = \sum_{k\geq 0} j_k x^k$ satisfies the differential equation

$$\frac{d^2 y}{dx^2} + \frac{1}{x}\frac{d\,y}{dx} + y = 0, \tag{7}$$

The first coefficients in its series expansion are $j_0 = 1$ and $j_1 = 0$.

Consider now the function $B(x) = exp\left(-\sum_{k\geq 0} b_k \frac{x^{2(k+1)}}{((k+1)!2^{k+1})^2}\right) = \sum_{k\geq 0} \beta_k x^k$. Equation (5) ensures that $B(x)$ satisfies Equation (7), *i.e.* the same second order differential equation as $J_0(x)$.

Setting $x = 0$, we have $\beta_0 = B(0) = 1 = j_0$. Moreover, in $Z(x) = -\sum_{k\geq 0} b_k \frac{x^{2(k+1)}}{((k+1)!2^{k+1})^2}$ only the even powers of x have non-zero coefficients. Hence, since $B(x) = exp\,(Z(x)) = \sum_{k\geq 0} \frac{Z(x)^k}{k!}$, we have $\beta_{2i+1} = 0$ for every $i \geq 0$. In particular, $\beta_1 = 0 = j_1$. These arguments imply that $B(x) = J_0(x)$. □

4.2 Proving identities combinatorially

Corollary 9 shows that non-ambiguous trees provide a combinatorial interpretation –and to our knowledge, the first one– of sequence `A002190` [Slo].
In [Car63], the author shows analytically that identities (5) and (8) below are equivalent. We give a combinatorial proof of this fact.

Proposition 10 *For every $n \geq 1$, we have:*

$$\sum_{k=0}^{n-1} (-1)^k \binom{n}{k+1}\binom{n-1}{k} b_k = 1. \tag{8}$$

Proof: We fix an integer n and we take $0 \leq k \leq n-1$. We define a gridded tree of size (k, n) to be a set of $2k+1$ points placed in a $n \times n$ grid, such that Condition 2 defining non-ambiguous trees is satisfied (which means we consider a non-ambiguous tree of size $2k+1$ embedded in a $n \times n$ grid) and such that the underlying tree is complete and that its root belongs to the first column. This implies that there are $n-k-1$ empty columns and $n-k-1$ empty rows, and that the first column is not empty. Figure 12 shows an example of a gridded tree of size $(2, 6)$.

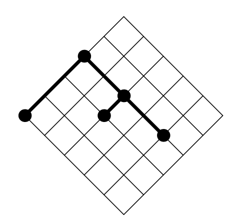


Fig. 12: An example of gridded tree with 2 internal vertices drawn on a 6×6 grid

It is easy to verify that there are $\binom{n}{k+1}\binom{n-1}{k} b_k$ gridded tree of size (k, n). We call trivial gridded tree the tree of size $(0, n)$ consisting of a single vertex in $(0, 0)$. Now, for every integer n, we define an involution on the set of non trivial gridded trees. This involution associates a gridded tree of size (k, n) with a gridded tree either of size $(k-1, n)$ or $(k+1, n)$.
To define this involution, consider a gridded tree of size (k, n) and add a virtual root at position $(-1, 0)$; the previous root becomes the left child of the virtual root. Now consider the path starting from the virtual root, going down through the tree, turning at each internal vertex, and ending at a leaf. This path is unique. There are two cases:

1. the path does not cross an empty row, nor an empty column: we erase the leaf and its parent from the tree, getting a new gridded tree of size $(k-1, n)$. We can always erase the leaf and its parent, except if the parent were the virtual root. This happens only if the tree is the trivial gridded tree. As we restricted to non trivial gridded trees, this case never happens.

2. the path crosses an empty row or an empty column: we choose the first empty row or column met while visiting the path. Without loss of generality, we suppose that it is a column, say c. Then, we add a new vertex v at the position where c crosses the path, and we add in the same column a new leaf (whose parent is v) in the topmost empty row. While visiting the path, we did not meet an empty row. Since there are as many empty rows as empty columns, there is always an empty row below v. This operation gives rise to a new gridded tree of size $(k+1, n)$.

Figure 13 shows how the involution acts on two examples.

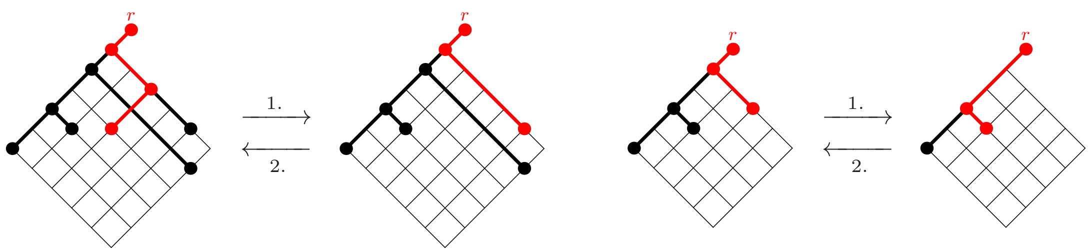


Fig. 13: The involution acting on two examples of non trivial gridded trees

Remark that adding (resp. removing) a leaf and its parent p in (resp. from) a gridded tree following the previous algorithm does not remove (resp. add) any empty row or column that crosses the path from the virtual root to p. For this reason, this operation is an involution. □

In a similar fashion to the proof of Proposition 10, it is possible to prove that Catalan numbers satisfy $\sum_{k=0}^{n}(-1)^{n+k}\binom{n+k}{n-k}C_k = 0$, for any $n \geq 1$. This identity and Proposition 10 allow us to prove a further identity involving the sequence b_n. Our proof uses the methodology described in [PWZ96] and settles a conjecture of P. Hanna (see [Slo] sequence `A002190`):

Proposition 11 *For every $n \geq 1$, we have* $\displaystyle\sum_{k=0}^{n}(-1)^k b_k C_k \binom{n+k}{n-k}^2 = 0.$

5 A new bijection between trees and parallelogram polyominoes

We recall that a parallelogram polyomino of size n is a pair of lattice paths of length $n+1$ with south-west and south-east steps starting at the same point, ending at the same point, and never meeting each other. Figure 14 shows some examples of parallelogram polyominoes of size 4. The two paths defining a given parallelogram polyomino delimit a connected set of boxes. We will consider the parallelogram polyomino from this point of view.

We now describe a bijection between parallelogram polyominoes of size n and binary trees with n vertices by showing that a parallelogram polyomino hides a non-ambiguous tree.

Given a parallelogram polyomino P, consider the set S_P of dots defined as follows:

- we enlighten P from north-west to south-east and from north-east to south-west;
- we put a dot in the enlightened boxes.

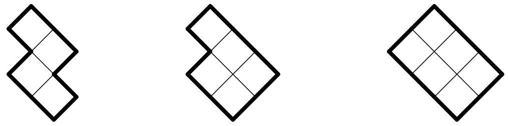


Fig. 14: Example of parallelogram polyominoes of size 4

It is easy to verify that S_P is a non-ambiguous tree. Indeed, it is impossible that all three points in the pattern  are enlightened. Moreover, only the northernmost box in the parallelogram polyomino can be enlightened twice. This implies that every dot (except for the one in the northernmost box) has a parent. Let Ψ be the application that associates to a parallelogram polyomino the underlying binary tree of S_P. An example of this application is shown in Figure 15.

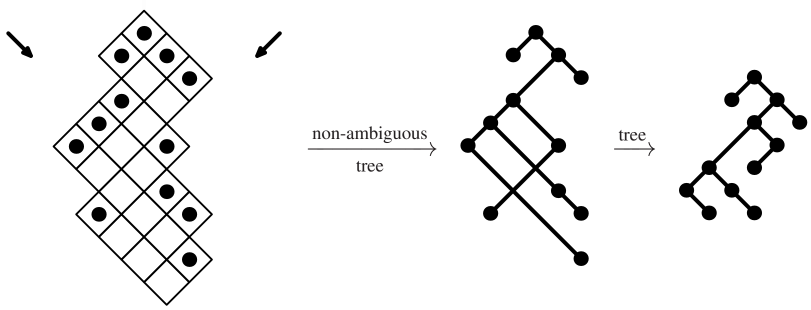


Fig. 15: Parallelogram polyominoes are just a way of drawing a binary tree in the plane

Proposition 12 *The map Ψ is a bijection between the set of parallelogram polyominoes of size n and the set of binary trees with n vertices.*

Proof: We are able in [ABBS] to describe explicitly the inverse of Ψ. But in this extended abstract, we shall only prove that Ψ is injective. Since the two considered sets have the same cardinality, this is enough to prove Proposition 12. In order to do that, we will construct a parallelogram polyomino P and the associated tree $\Psi(P)$ (actually, a non-ambiguous tree of shape $\Psi(P)$) at the same time. More precisely, when creating the parallelogram polyomino, we start from the origin of the two paths, and we add:

- one step to each of the two paths at a time in the parallelogram polyomino;
- the enlightened dot(s) corresponding to the inserted steps, when needed.

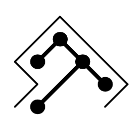


Fig. 16: An example of parallelogram polyomino with its tree under construction

Figure 16 shows an example of this construction.
Consider two different parallelogram polyominoes and construct them simultaneously, together with the associated non-ambiguous trees. While the beginning of the paths are the same, the associated trees are also the same. Consider the first time where one path in the first parallelogram polyomino differs from its homologous in the other parallelogram polyomino. One of the two added steps will be SW-oriented, and the other will be SE-oriented. This means that,

only one of these steps is associated with a new dot, connected to its parent v. The dot v exists in both trees, but it does not have the same number of children in both trees. □

Acknowledgements

The authors are grateful to Philippe Nadeau for helpful discussions around rectangular alternative tableaux.

This research was driven by computer exploration using the open-source mathematical software `Sage` [S+12] and its algebraic combinatorics features developed by the `Sage-Combinat` community [SCc12].

References

[ABBS] J.-C. Aval, A. Boussicault, M. Bouvel, and M. Silimbani. Combinatorics of non-ambiguous trees. Full version in preparation.

[ABN11] J.-C. Aval, A. Boussicault, and P. Nadeau. Tree-like tableaux. In *DMTCS Proceedings 23rd International Conference on Formal Power Series and Algebraic Combinatorics (FPSAC 2011)*, pages 63–74, Islande, 2011. DMTCS.

[ABN12] J.-C. Aval, A. Boussicault, and P. Nadeau. Tree-like tableaux. Full version, in preparation. 2012.

[AS64] M. Abramowitz and I. A. Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55 of *National Bureau of Standards Applied Mathematics Series*. U.S. Government Printing Office, Washington, D.C., 1964.

[Bur07] A. Burstein. On some properties of permutation tableaux. *Ann. Comb.*, 11(3-4):355–368, 2007.

[Car63] L. Carlitz. A sequence of integers related to the Bessel functions. *Proc. Amer. Math. Soc.*, 14:1–9, 1963.

[DV84] M.-P. Delest and G. Viennot. Algebraic languages and polyominoes enumeration. *Theoretical Computer Science*, 34(1-2):169 – 206, 1984.

[ES00] R. Ehrenborg and E. Steingrímsson. The excedance set of a permutation. *Advances in Applied Mathematics*, 24(3):284 – 299, 2000.

[Knu98] D. E. Knuth. *The art of computer programming, volume 3: (2nd ed.) sorting and searching*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1998.

[Nad11] P. Nadeau. The structure of alternative tableaux. *J. Combin. Theory Ser. A*, 118(5):1638–1660, 2011.

[PWZ96] M. Petkovšek, H. S. Wilf, and D. Zeilberger. *A = B*. Ak Peters Series. Peters, 1996.

[S+12] W. A. Stein et al. *Sage Mathematics Software*. The Sage Development Team, 2012. `http://www.sagemath.org`.

[SCc12] The Sage-Combinat community. Sage-Combinat: enhancing Sage as a toolbox for computer exploration in algebraic combinatorics, 2012. `http://combinat.sagemath.org`.

[Slo] N. J. A. Sloane. The On-Line Encyclopedia of Integer Sequences. `http://oeis.org`.

[SW07] E. Steingrímsson and L. K. Williams. Permutation tableaux and permutation patterns. *J. Combin. Theory Ser. A*, 114(2):211–234, 2007.

[Vie07] X. Viennot. Alternative tableaux, permutations and partially asymmetric exclusion process. *Talk at Isaac Newton institute*, April 2007. `http://www.newton.ac.uk/webseminars/pg+ws/2008/csm/csmw04/0423/viennot/`.