# NON-AMBIGUOUS TREES: NEW RESULTS AND GENERALISATION

JEAN-CHRISTOPHE AVAL, ADRIEN BOUSSICAULT, BÉRÉNICE DELCROIX-OGER, FLORENT HIVERT, AND PATXI LABORDE-ZUBIETA

Abstract. We present a new definition of non-ambiguous trees (NATs) as labelled binary trees. We thus get a differential equation whose solution can be described combinatorially. This yield a new formula for the number of NATs. We also obtain $q$-versions of our formula. And we generalize NATs to higher dimension.

## Introduction

Non-ambiguous trees (NATs for short) were introduced in a previous paper [ABBS14]. We propose in the present article a sequel to this work.

Tree-like tableaux [ABN13] are certain fillings of Ferrers diagram, in simple bijection with permutations or alternative tableaux [Pos07, Vie08]. They are the subject of an intense research activity in combinatorics, mainly because they appear as the key tools in the combinatorial interpretation of the well-studied model of statistical mechanics called PASEP: they naturally encode the states of the PASEP, together with the transition probabilities through simple statistics [CW07].

Among tree-like tableaux, NATs were defined as rectangular-shaped objects in [ABBS14]. In this way, they are in bijection with permutation $\sigma = \sigma_1 \sigma_2 \dots \sigma_n$ such that the excedences ($\sigma_i > i$) are placed at the beginning of the word $\sigma$. Such permutations were studied by Ehrenborg and Steingrimsson [ES00], who obtained an explicit enumeration formula. Thanks to NATs, a bijective proof of this formula was described in [ABBS14].

In the present work, we define NATs as labelled binary trees (see Definition 1.1, which is equivalent to the original definition). This new presentation allows us to obtain many new results about these objects. The plan of the article is the following.
In Section 1, we (re-)define NATs as binary trees whose right and left

*Date*: December 10, 2015.

*Key words and phrases.* Non-ambiguous trees, binary trees, ordered trees, q-analogues, permutations, hook-length formulas.

children are respectively labelled with two sets of labels. We show how the generating series for these objects satisfies differential equations (Prop. 1.8), whose solution is quite simple and explicit (Prop. 1.9). A combinatorial interpretation of this expression involves the (new) notion of hooks in binary trees, linked to the notion of leaves in ordered trees. Moreover this expression yields a new formula for the number of NATs as a positive sum (see Theorem 1.19), where Ehrenborg-Steingrimsson's formula is alternating. To conclude with Section 1, we obtain $q$-analogues of our formula, which are similar to those obtained for binary trees in [BW89, HNT08] (see Theorem 1.21, the relevant statistics are either the number of inversions or the inverse major index).

Section 2 presents a generalisation of NATs in higher dimension. For any $k \leq d$, we consider NATs of dimension $(d, k)$, embedded in $\mathbb{Z}^d$, and with edges of dimension $k$ [1]. The original case corresponds to dimension $(2, 1)$. Our main result on this question is a differential equation satisfied by the generating series of these new objects. Finally, we study the (new) notion of hooks on binary trees in Section 3. We prove (through the use of generating series, and bijectively) that the number of hooks is distributed on binary trees as another statistics: the childleaf statistic, defined as the number of vertices who has at least one leaf as a child.

## 1. Non-ambiguous trees

1.1. **Definitions.** We recall that a *binary tree* is a rooted tree whose vertices may have no child, or one left child, or one right child or both of them. The size of a binary tree is its number of vertices. The empty binary tree, denoted by $\emptyset$, is the unique binary tree with no vertices. Having no child in one direction (left or right) is the same as having an empty subtree in this direction. We denote by $\mathcal{BT}$ the set of binary trees and by $\mathcal{BT}^*$ the set $\mathcal{BT} \setminus \{\emptyset\}$. Given a binary tree $B$, we denote by $\mathcal{V}_L(B)$ and $\mathcal{V}_R(B)$ the set of left children (also called left vertices) and the set of right children (also called right vertices). We shall extend this notation to NATs.

We now define the notion of non-ambiguous trees:

**Definition 1.1.** *A* non-ambiguous tree *(NAT) $T$ is a labelling of a binary tree $B$ such that :*

- *the left (resp. right) children are labelled from* 1 *to* $|\mathcal{V}_L(B)|$ *(resp.* $|\mathcal{V}_R(B)|$*), such that different left (resp. right) vertices*

[1] A definition in terms of labelled trees is given in Subsection 2.1.

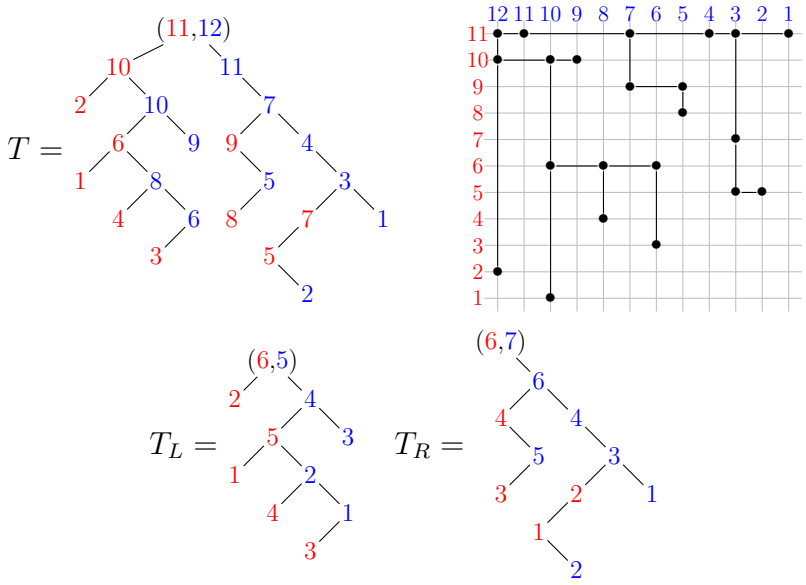



FIGURE 1. A non-ambiguous tree, its geometrical presentation, and its left and right subtrees

*have different labels. In other words, each left (right) label appears only once.*

- *if $U$ and $V$ are two left (resp. right) children in the tree, such that $U$ is an ancestor of $V$, then the label of $U$ in $T$ is strictly greater than the label of $V$.*

The underlying tree of a non-ambiguous tree is called its *shape*. The size $n(T)$ of a NAT $T$ is its number of vertices. Clearly $n(T) = 1 + |\mathcal{V}_L(T)| + |\mathcal{V}_R(T)|$. It is sometimes useful to label the root as well. In this case, it is considered as both a left and right child so that it carries a pairs of labels, namely $(|\mathcal{V}_L(T)|+1, |\mathcal{V}_R(T)|+1)$. On pictures, to ease the reading, we color the labels of left and right vertices in red and blue respectively. Figure 1 shows an example of a NAT, and illustrates the correspondence between the geometrical presentation of [ABBS14] and Definition 1.1. The rectangle which contains the non-ambiguous tree $T$ is of dimension $(w_L(T), w_R(T)) = (|\mathcal{V}_L(T)| + 1, |\mathcal{V}_R(T)| + 1)$.

1.2. **Differential equations on non-ambiguous trees.** The goal of this section is to get (new) formulas for the number of NATs with prescribed shape. The crucial argument is the following remark: Let $T$ be a NAT of shape a non empty binary tree $B = \overset{L}{\bullet}\!\!\diagdown\!\!\diagup\overset{R}{\bullet}$. Restricting

the labellings of the left and right children of $T$ to $L$ and $R$ gives non-decreasing labelling of their respective left and right children. Note that the root of $L$ (resp. $R$) is a left (resp. right) child in $T$. By renumbering the labels so that they are consecutive numbers starting from 1, we get two non-ambiguous labellings for $L$ and $R$, that is two non-ambiguous trees $T_L$ and $T_R$. See Figure 1 for an example.

Conversely, knowing the labelling of $L$ and $R$, to recover the labelling of $T$, one has to choose which labels among $1 \dots \mathcal{V}_L(T)$ will be used for $L$ (including its root) and the same for right labels. As a consequence:

$$\left|\mathcal{NAT}\left(\begin{smallmatrix} & \bullet & \\ L & & R \end{smallmatrix}\right)\right| = \binom{|\mathcal{V}_L(T)|}{|\mathcal{V}_L(R)|}\binom{|\mathcal{V}_R(T)|}{|\mathcal{V}_R(L)|}\,|\mathcal{NAT}(L)|\,|\mathcal{NAT}(R)|. \quad (1)$$

Our first step is to recover hook-length formula for the number of NATs of fixed shape ([ABBS14]). We use the method from [HNT08], namely, applying recursively a bilinear integro-differential operator called here a *pumping function* along a binary tree.

First of all, we consider the space $\mathbb{Q}\mathcal{NAT}$ of formal sums of non-ambiguous trees and identifies $\mathcal{NAT}(B)$ with the formal sum of its elements. We consider the map $\mathbb{M} : \mathcal{NAT} \times \mathcal{NAT} \mapsto \mathbb{Q}\mathcal{NAT}$ sending $(T_1, T_2)$ to the formal sum of NATs $T$ such that $T_L = T_1$ and $T_R = T_2$. By linearity, we extend $\mathbb{M}$ to a bilinear map $\mathbb{Q}\mathcal{NAT} \times \mathbb{Q}\mathcal{NAT} \mapsto \mathbb{Q}\mathcal{NAT}$. The main remark is that $\mathcal{NAT}(B)$ can be computed by a simple recursion using $\mathbb{M}$:

**Lemma 1.2.** *The set $\mathcal{NAT}(B)$ of non-ambiguous tree of shape $B$ satisfies the following recursion:*

$$\mathcal{NAT}(\emptyset) = \emptyset \qquad \textit{and} \qquad \mathcal{NAT}\left(\begin{smallmatrix} & \bullet & \\ L & & R \end{smallmatrix}\right) = \mathbb{M}\left(\mathcal{NAT}(L), \mathcal{NAT}(R)\right). \quad (2)$$

To count non-ambiguous trees, and as suggested by the binomial coefficients in (1), we shall use *doubly exponential generating functions* in two variables $x$ and $y$ where $x$ and $y$ count the size of the rectangle in which the NAT is embedded: the weight of the NAT $T$ is $\Phi(T) := \frac{x^{w_L(T)}}{w_L(T)!}\frac{x^{w_R(T)}}{w_R(T)!}$. We extend $\Phi(T)$ by linearity to a map $\mathbb{Q}\mathcal{NAT} \mapsto \mathbb{Q}[[x, y]]$. Consequently, $\Phi(\mathcal{NAT}(B))$ is the generating series of the non-ambiguous trees of shape $B$. Thanks to (1) the image in $\mathbb{Q}[[x, y]]$ of the bilinear map $\mathbb{M}$ under the map $\Phi$ is a simple differential operator:

**Definition 1.3.** *The* pumping function $\mathbb{B}$ *is the bilinear map* $\mathbb{Q}[[x,y]]\times\mathbb{Q}[[x,y]]\mapsto\mathbb{Q}[[x,y]]$ *defined by*

$$\mathbb{B}(u,v)=\int_x\int_y\partial_x(u)\cdot\partial_y(v). \tag{3}$$

*We further define recursively, for any binary tree* $B$ *an element* $\mathbb{B}(B)\in\mathbb{Q}[[x,y]]$ *by*

$$\mathbb{B}(\emptyset)=x+y \qquad \textit{and} \qquad \mathbb{B}\left(\begin{smallmatrix} & \bullet & \\ L & & R\end{smallmatrix}\right)=\mathbb{B}\left(\mathbb{B}(L),\mathbb{B}(R)\right). \tag{4}$$

Now (1) rewrites as

**Proposition 1.4.** *For* $T_1,T_2\in\mathbb{Q}\mathcal{NAT}$*, one as* $\Phi(\mathbb{M}(T_1,T_2))=\mathbb{B}(\Phi(T_1),\Phi(T_2))$*. As a consequence, for any non empty binary tree* $B$*,* $\Phi(\mathcal{NAT}(B))=\mathbb{B}(B)$*.*

By de-recursiving the expression for $\mathbb{B}(B)$, we recover the hook-length formula of [ABBS14] for non-ambiguous trees of a given shape:

**Proposition 1.5.** *Let* $B$ *be a binary tree. For each left (resp. right) vertex* $U$*, we denote* $\mathcal{E}_L(U)$ *(resp.* $\mathcal{E}_R(U)$*) the number of left (resp. right) vertices of the subtree with root* $U$ *(itself included in the count). Then*

$$|\mathcal{NAT}(B)|=\frac{|\mathcal{V}_L(B)|!\cdot|\mathcal{V}_R(B)|!}{\prod\limits_{U:\textit{left child}}\mathcal{E}_L(U)\cdot\prod\limits_{U:\textit{right child}}\mathcal{E}_R(U)}. \tag{5}$$

We consider now the *exponential generating function of non-ambiguous trees* with weight $\Phi$:

$$\mathfrak{H}:=\sum_{T\in\mathcal{NAT}}\Phi(T)=\sum_{T\in\mathcal{NAT}}\frac{x^{w_L(T)}}{w_L(T)!}\frac{x^{w_R(T)}}{w_R(T)!}. \tag{6}$$

It turns out that we need to consider the two following slight modifications to get nice algebraic properties (because of the empty NAT).

$$\mathfrak{G}=\sum_{B\in\mathcal{BT}}\mathbb{B}(B) \quad \text{and} \quad \mathfrak{N}=\sum_{T\in\mathcal{NAT}^*}\frac{x^{|\mathcal{V}_L(T)|}\cdot y^{|\mathcal{V}_R(T)|}}{|\mathcal{V}_L(T)|!\cdot|\mathcal{V}_R(T)|!}. \tag{7}$$

The function $\mathfrak{H}$, $\mathfrak{N}$, $\mathfrak{G}$ are closely related. Each function is used in different situation. The first one is the natural definition we want to give. The second one is convenient from a bijective point of view. The last one is convenient from the algebraic and analytic point of view.

They differ by their constant term and shift in the degree. Precisely, $\mathfrak{N} = \partial_x\partial_y\mathfrak{H}$ so that

$$\mathfrak{H} = 1+\int_x\int_y \mathfrak{N} \quad \text{and} \quad \mathfrak{G} = x+y+\int_x\int_y \mathfrak{N} \quad \text{and} \quad \mathfrak{G} = \mathfrak{H}+x+y-1 \tag{8}$$

The two last relations are consequences of Proposition 1.4.

**Proposition 1.6.** *The generating function $\mathfrak{N}$ and $\mathfrak{G}$ can be computed by the following fixed point differential equations:*

$$\mathfrak{G} = x+y+\int_x\int_y \partial_x\mathfrak{G}\cdot\partial_y\mathfrak{G} \quad \textit{and} \quad \mathfrak{N} = \left(1+\int_x \mathfrak{N}\right)\cdot\left(1+\int_y \mathfrak{N}\right) \tag{9}$$

*Proof.* The first equation is a just a consequence of the definition of the bilinear map $\mathbb{B}$:

$$\mathfrak{G} = x+y+\sum_{L,R\in\mathcal{BT}} \mathbb{B}\left(L \diagup\diagdown R\right) = x+y+\sum_{L,R\in\mathcal{BT}} \mathbb{B}(\mathbb{B}(L),\mathbb{B}(R)) = x+y+\mathbb{B}(\mathfrak{G},\mathfrak{G}).$$

To prove the second equation, remark that the first can be rewritten as $\partial_x\partial_y\mathfrak{G} = \partial_x\mathfrak{G}.\partial_y\mathfrak{G}$. So that, $\mathfrak{N} = \partial_x\partial_y\mathfrak{H} = \partial_x\partial_y\mathfrak{G}$. To conclude, it suffices to remark that $\partial_x\mathfrak{G} = 1+\int_y \mathfrak{N}$ □

Now, a closed formula can be computed for $\mathfrak{N}$ and $\mathfrak{H}$.

**Proposition 1.7.** *The exponential generating function for non-ambiguous trees are given by*

$$\mathfrak{N} = \frac{e^{x+y}}{(1-(e^x-1)(e^y-1))^2}, \quad \textit{and} \quad \mathfrak{H} = -\log(1-(e^x-1)(e^y-1)).$$

A VERIFIER

*Proof.*

(1) We know that

$$\partial_x\partial_y\mathfrak{G} = \partial_x\mathfrak{G}\times\partial_y\mathfrak{G}, \tag{10}$$

If we have a particular solution $s(x,y)$ for $\mathfrak{G}$, then, for every function $s_1(x)$ and $s_2(y)$, the function $s(s_1(x),s_2(y))$ is a solution of Equation 10.

(2) A particular solution can be obtained for $\mathfrak{G}$ by studying Equation 9 and by setting $\mathfrak{N}_x = \left(1+\int_x \mathfrak{N}\right)$ and $\mathfrak{N}_y = \left(1+\int_y \mathfrak{N}\right)$. We obtain the following relation : $\partial_x\mathfrak{N}_x\cdot\partial_y\mathfrak{N}_y = \mathfrak{N}_x^2\cdot\mathfrak{N}_y^2$. As we just want to compute one solution, we can suppose that $\partial_x\mathfrak{N}_x = \mathfrak{N}_x^2$ and $\mathfrak{N}_y = \mathfrak{N}_x(y,x)$ . Hence, for each constant $c$, $\frac{1}{(c+x+y)^2}$ is a particular solution for Equation 9. So $s(x,y) = -\ln(c+x+y)+x+y$ is a particular solution for

Equation 10 and $-\partial_x\partial_y \ln(c + s_1(x) + s_2(y))$ is a family of solution for Equation 9.

(3) To find $c$, $s_1$ and $s_2$, we just need to consider the initial conditions : $\mathfrak{N}(x, 0) = e^x$, $\mathfrak{N}(0, y) = e^y$.

□

Now, we will introduce two statistics : the number of right (resp. left) vertices in the rightmost (resp. leftmost) branch of the root of a tree. For a binary tree $B$, we will denote by $\mathcal{R}_0(B)$ (resp. $\mathcal{L}_0(B)$) the two previous statistics. We define now an $(\alpha, \beta)$-generating function for non-ambiguous trees:

$$\mathfrak{N}^{(\alpha,\beta)} = \sum_{T\in\mathcal{NAT}} \frac{x^{|\mathcal{V}_L(T)|} \cdot y^{|\mathcal{V}_R(T)|} \cdot \alpha^{\mathcal{R}_0(T)} \cdot \beta^{\mathcal{L}_0(T)}}{|\mathcal{V}_L(T)|! \cdot |\mathcal{V}_R(T)|!}.$$

**Proposition 1.8.** *A differential equation for* $\mathfrak{N}^{(\alpha,\beta)}$ *is*

$$\mathfrak{N}^{(\alpha,\beta)} = \left(1 + \alpha \int_x \mathfrak{N}^{(\alpha,1)}\right) \cdot \left(1 + \beta \int_y \mathfrak{N}^{(1,\beta)}\right),$$

*Proof.* We just have to define a new pumping function by setting $\mathbb{B}^{(\alpha,\beta)}(B) = \alpha^{\mathcal{R}_0(B)}\beta^{\mathcal{L}_0(B)}\,\mathbb{B}(B)$ and deduce the expected differential equation. □

The solution of the new differential equation is given by Proposition 1.9.

**Proposition 1.9.** *The* $(\alpha, \beta)$*-exponential generating function for non-ambiguous trees is equal to*

$$\mathfrak{N}^{(\alpha,\beta)} = \frac{e^{\alpha x+\beta y}}{(1 - (e^x - 1)(e^y - 1))^{\alpha+\beta}}.$$

1.3. **Bijection with some labelled ordered trees.** In what follows, we will use *rooted ordered trees*. These are trees such that each node has an ordered (possibly empty) list of children. We draw the children from left to right on the pictures.

Note that the solution of Proposition 1.9 can be rewritten as :

$$\mathfrak{N}^{(\alpha,\beta)} = e^{\alpha x}e^{\beta y}e^{-\alpha \ln(1-(e^x-1)(e^y-1))}e^{-\beta \ln(1-(e^x-1)(e^y-1))}. \qquad (11)$$

The purpose of this subsection is to explain this expression combinatorially. Let us first describe objects "naturally" enumerated by the RHS of (11). We recall that $e^x$ is the exponential generating series of sets and $-\ln(1-x)$ is the exponential generating series of cycles. The objects can be described as 4-tuples consisting of two sets of elements and two sets of cycles whose elements are pairs of non empty sets. The

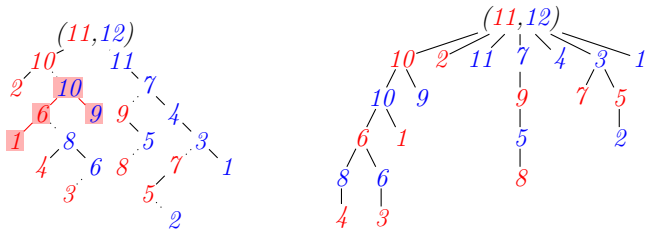



FIGURE 2. Hooks on a non-ambiguous tree and associated ordered tree

Utilité de cette def ?

elements considered will be of type $X$ or of type $Y$ and of colour $a$ or $b$. The objects can be described as 4-tuples consisting of :

- a set of elements of type $X$, each element coloured with colour $a$,
- a set of elements of type $Y$, each element coloured with colour $b$,
- a set of cycles whose elements are pairs, consisting of a non empty set of elements of type $X$ and a non empty set of elements of type $Y$, each cycle coloured with colour $a$
- and a set of cycles whose elements are pairs, consisting of a non empty set of elements of type $X$ and a non empty set of elements of type $Y$, each cycle coloured with colour $b$.

Let us denote by $\mathcal{T}_4$ the set of such 4-tuples.

We first link non-ambiguous trees with ordered trees. We need the following definition:

**Definition 1.10.** *Let $B$ be a binary tree and $v$ one of his node. The* hook *of a vertex $v$ is the union of $\{v\}$, its leftmost branch and its rightmost branch. There is a unique way to partition the vertices in hooks. The number of hooks in such a partition is the* hook number of the tree.

**Remark 1.11.** *We can obtain recursively the unique partition of the preceding definition by extracting the root's hook and iterating the process on each tree of the remaining forest.*

**Example 1.12.** *On the left part of Figure 2, we represented in red the hook of 10. The partition of vertices in hooks is obtained by removing the dotted edges. The hook number of the tree is* 8.

We denote by $\mathcal{NOT}$ the set of ordered trees $O$ such that:

- Each vertex, except the root, is labelled and coloured (in red or blue). The root is labelled by a red label and a blue label, both maximal.

- The root has red and blue children, the red children being on the left side of blue children. Blue (resp. red) vertices have only red (resp. blue) children.
- The labels of red (resp. blue) descendants or right siblings of a red (resp. blue) vertex $v$ are smaller than the label of $v$.

**Proposition 1.13.** *The set of non-ambiguous trees $\mathcal{NAT}$ on $n$ nodes is in bijection with the set of trees of $\mathcal{NOT}$ on $n$ nodes. This bijection is denoted by $\xi$.*

*Proof.* Let us consider a non-ambiguous tree $T$ and construct an ordered tree $\xi(T) = O$. The root of $T$ will be associated to the root of $O$. Starting from the root $r$ of the ordered tree, the red (resp. blue) children of $r$ are the set of left (resp. right) descendants of the root of $T$. The expected ordered tree is then obtained recursively by the following rule : if a node $v$ in the ordered tree is a left (resp. right) child in $T$, then its children in the ordered tree is the set of right (resp. left) descendants of $v$ in $T$, with every right (resp. left) child on the right side of its parent.

We can reconstruct recursively the non-ambiguous tree associated to such an ordered tree, by reversing the process from the children of the root to the leaves in the ordered tree. □

**Remark 1.14.** *Let us remark that the hook of a vertex $v$, different from the root in the non-ambiguous tree, can be read off from the ordered trees : it consists in the children of $v$ in the ordered tree and the siblings of $v$ on the right side of $v$ in the ordered tree.*

**Example 1.15.** *The ordered tree associated to the non-ambiguous tree on the left part of Figure 2 is represented on the right part of the same figure.*

**Proposition 1.16.** *The set of non-ambiguous trees $\mathcal{NAT}$ is in bijection with pairs of 2-coloured words, with blue letters on $\{1, \dots, |\mathcal{V}_R|\}$ and red letters on $\{1, \dots, |\mathcal{V}_L|\}$, where each letter appear exactly once (in the first word or in the second word), letters in blocks of the same colors are decreasing, the first (resp. second) word ends by a red (resp. blue) letter and $\mathcal{V}_R$ (resp. $\mathcal{V}_L$) is the set of right (resp. left) children in the non-ambiguous tree. This bijection is denoted by $\xi \circ \Omega$. Moreover, the pairs of 2-coloured words are exactly described by the previous 4-tuples.*

*Proof.* We describe here the bijection $\Omega$ between the ordered trees obtain in Proposition and such words. We can obtain such a pair of words

from this ordered tree by a post-order traversal visit of the descendant of the red children of the root for the first word and of the blue children of the root for the second one. Indeed, the obtained pair of words satisfies the expected properties because if two letters of the same colours are adjacent in one of the words, then the second letter is the label of a descendant of a right sibling of the node labelled by the first letter: it is then smaller.

Let us show the injectivity of $\Omega$. Suppose that there are two ordered trees $O_1$ and $O_2$ having the same image by $\Omega$. We compare these trees from the root to the leaves and from left to right. Suppose that these trees are different, then in $O_1$ there exists a vertex $v$ of parent $p$ such that the ancestors of $p$ and branches on the left of $p$ and $v$ are the same in both trees, but $v$ is a child of $p$ in $O_1$ and not in $O_2$. Then, in the pair of words associated to the trees, $v$ is before $p$. If $v$ is adjacent to $p$, then $v$ must be in a branch on the left of $p$, which contradicts the hypothesis. Otherwise, it means that $v$ has right siblings in $O_1$. Let us denote by $w$ the last descendant of the leftmost one, denoted by $b$. Then $v$ is before $w$ in the word so $v$ must be a child of $w$ as it cannot be on the left of it by hypothesis. However, $v$ is one the left of $b$ in $O_1$, which is ordered with a decreasing property, so $v > b$. Moreover, $v$ is a descendant of $b$ in $O_2$, therefore $v < b$ : this is impossible so $O_1 = O_2$ and $\Omega$ is injective.

From such a word, we can build back recursively the associated ordered trees by reading each word from right to left. We begin from the root of the ordered tree. Then, if the last added vertex is labelled by $l$ and the letter on the left of $l$ in the considered word is $p$, we add a vertex labelled $p$ to the left of the closest ancestor of $l$ whose label is of the same colour as $p$ and smaller than $p$. This enables us to preserve the expected order of elements in the obtained ordered tree.

The consecutive maximal red (rep. blue) elements from right to left in the first (resp. second) word correspond to the children of the root in the ordered tree. The descendant of these vertices are elements in front of them and before the next child of the root. We can defined the first (resp. second) set of the 4-tuple to be the set of blue (resp. red) children of the root in the ordered tree, or, translated in term of non ambiguous trees, the set of right (resp. left) descendant of the root with no child. Hence, each remaining subword, corresponding to one child of the root and its descendants in the ordered tree, contains both blue and red elements, the rightmost letter corresponding to the child of the root. Then, given a blue (resp. red) child of the root $f$, the subword associated to it can be viewed as a blue (resp. red) cycle, as $f$ is the biggest blue element in the subword and can be found again. This cycle

is made of alternating sets of blue and red elements, corresponding to right and left vertices in the non-ambiguous tree, which can be joined in pairs of non empty sets, giving the two set of cycles of the 4-tuple. □

**Example 1.17.** *The pair of words associated with trees of Figure 2 is (4 8 3 6 6 1 10 9 10 2, 11 8 5 9 7 4 7 2 5 3 1). The associated 4-tuple is: ({2}, {1,4,11}, {(\{10 4 8\}\{36\}\{6110 9\})}, {(\{85\}\{97\}),(\{72\}\{53\}))}.*

**Remark 1.18.** *The bijection $\Omega$ is similar to the "zigzag" bijection of* [SW07].

We may derive from our construction a bijective proof of the following enumeration formula. detailler !

**Theorem 1.19.** *The $(\alpha, \beta)$-analogue of the number of non empty non-ambiguous trees with $w$ left vertices and $h$ right vertices is given by:*

$$\mathcal{NAT}_{w,h} = \sum_{p\geq 1}(p-1)! \cdot (p-1)^{(\alpha+\beta)} \cdot S_{2,\alpha}(w+1,p)S_{2,\beta}(h+1,p) \quad (12)$$

*where $p^{(q)}$ is the rising factorial, and $S_{2,q}$ denotes the $q$-analogue of the Stirling numbers of the second kind such that, if we consider a set partition, $q$ counts the number of elements different from 1 in the subset containing 1. In this positive summation expression, each summand corresponds to the number of NATs with prescribed size, and whose number of hooks equals $p$.*

1.4. **q-analogs of the hook formula.** As for binary trees, there exists $q$-analogues of the hook formula for NATs of a given shape associated to either the number of inversions or the major index. There are two ingredients: first we need to associate two permutations to a non-ambiguous tree, and second we need to give a $q$-analogue of the bilinear map $\mathbb{B}$. It turns out that it is possible to use two different $q$ namely $q_R$ and $q_L$ for the derivative and integral in $x$ and $y$.

The first step to formulate a $q$-hook formula is to associate to any non empty non-ambiguous tree $T$ a pair of permutations $\sigma(T) = (\sigma_L(T), \sigma_R(T)) \in \mathfrak{S}_{\mathcal{V}_L(T)} \times \mathfrak{S}_{\mathcal{V}_R(T)}$.

**Definition 1.20.** *Let $T$ be a non-ambiguous tree. Then $\sigma_L(T)$ is obtained by performing a left postfix reading of the left labels: precisely we recursively read trees*  *by reading the left labels of $L$, then the left labels of $R$ and finally the label of the root if it is a left child. The permutation $\sigma_R(T)$ is defined similarly reading right labels, starting from the right subtree, then the left subtree and finally the root.*

If we take back the example of Figure 1 we get the two permutations $\sigma_L(T) = (2, 1, 4, 3, 6, 10, 8, 9, 5, 7)$ and $\sigma_R(T) = (1, 2, 3, 4, 5, 7, 11, 9, 6, 8, 10)$.

Recall that the *number of inversions* of a permutation $\sigma \in \mathfrak{S}_n$ is the number of $i < j <= n$ such that $\sigma(i) > \sigma(j)$. A descent of $\sigma$ is a $i < n$ such that $\sigma(i) > \sigma(i+1)$ and the *inverse major index* of $\sigma$ is the sum of the descents of $\sigma^{-1}$. Finally for a repetition free word $w$ of length $l$ we write $\mathrm{Std}(w)$ the permutations in $\mathfrak{S}_l$ obtained by renumbering $w$ keeping the order of the letters. For example $\mathrm{Std}(36482) = 24351$. We define as usual the $q$-integer $[n]_q := \frac{1-q^n}{1-q}$, and the $q$-factorial $[n]_q! := \prod_{i=1}^{n} [i]_q$.

**Theorem 1.21.** *For a non-ambiguous tree $T$ and a statistic $S \in \{\mathrm{Inv}, \mathrm{iMaj}\}$, define*

$$w_S(T) := q_L^{S(\sigma_L(T))} q_R^{S(\sigma_R(T))}. \tag{13}$$

*Then, for any non empty binary tree $B$*

$$\sum_{T \in \mathcal{NAT}(B)} w_{\mathrm{Inv}}(T) = \sum_{T \in \mathcal{NAT}(B)} w_{\mathrm{iMaj}}(T) = \frac{|\mathcal{V}_L(B)|_{q_L}! \cdot |\mathcal{V}_R(B)|_{q_R}!}{\prod_{U:\textit{left child}} [\mathcal{E}_L(U)]_{q_L} \cdot \prod_{U:\textit{right child}} [\mathcal{E}_R(U)]_{q_R}}. \tag{14}$$

Going back to the non-ambiguous tree of Figure 1, the inversions numbers are $\mathrm{Inv}(\sigma_L(T)) = 11$ and, $\mathrm{Inv}(\sigma_R(T)) = 7$ so that $w_{\mathrm{Inv}}(T) = q_L^{11} q_R^7$. For the inverse major index, we get the permutations $\sigma_L(T)^{-1} = (2, 1, 4, 3, 9, 5, 10, 7, 8, 6)$ and $\sigma_R(T)^{-1} = (1, 2, 3, 4, 5, 9, 6, 10, 8, 11, 7)$. Consequently, $\mathrm{iMaj}(\sigma_L(T)) = 1+3+5+7+9 = 25$ and $\mathrm{iMaj}(\sigma_R(T)) = 6 + 8 + 10 = 24$ so that $w_{\mathrm{iMaj}}(T) = q_L^{25} q_R^{24}$.

Note that it is possible to read directly $w_S(T)$ on $T$. We do not give the precise statement here to keep the presentation short.

The argument of the proof follows the same path as for the hook formula, using pumping functions: recall that the $q$-derivative and $q$-integral are defined as $\partial_{x,q} x^n := [n]_q x^{n-1}$ and $\int_{x,q} x^n := \frac{x^{n+1}}{[n+1]_q}$. Then the $(q_L, q_R)$-analogue of the pumping function is given by

$$\mathbb{B}_q(u, v) = \int_{x,q_L} \int_{y,q_R} \partial_{x,q_L}(u) \cdot \partial_{y,q_R}(v). \tag{15}$$

We also define recursively $\mathbb{B}_q(B)$ by $\mathbb{B}_q(\emptyset) := x + y$ and $\mathbb{B}_q\left(\begin{smallmatrix} & \bullet & \\ L & & R \end{smallmatrix}\right) = \mathbb{B}_q\left(\mathbb{B}_q(L), \mathbb{B}_q(R)\right)$. Then the main idea is to go through a pumping function on pairs of permutations. We write $\mathbb{Q}\,\mathfrak{S}$ the vector space of formal sums of permutations. For any permutation $\sigma \in \mathfrak{S}_n$ we write $\int \sigma = \sigma[n+1]$ the permutation in $\mathfrak{S}_{n+1}$ obtained by adding $n+1$ at the end. Again we extend $\int$ by linearity.

**Definition 1.22.** *The* pumping function on permutation *is the bilinear map* $\mathbb{B}\mathfrak{S} : \mathbb{Q}\mathfrak{S} \times \mathbb{Q}\mathfrak{S} \mapsto \mathbb{Q}\mathfrak{S}$ *defined for* $\sigma \in \mathfrak{S}_m$ *and* $\mu \in \mathfrak{S}_n$ *by* $\mathbb{B}\mathfrak{S}(\sigma, \mu) = \sum_{\substack{uv \in \mathfrak{S}_{m+n+1} \\ \mathrm{Std}(u)=\int \sigma \\ \mathrm{Std}(v)=\mu}} uv\,.$

*We define also a pumping function on pairs of permutations*

$$\mathbb{B}\mathfrak{S}^2\left((\sigma_L, \sigma_R), (\mu_L, \mu_R)\right) := \left(\mathbb{B}\mathfrak{S}(\sigma_L, \mu_L), \mathbb{B}\mathfrak{S}(\mu_R, \sigma_R)\right)$$

For example $\mathbb{B}\mathfrak{S}(21, 12) = 21345 + 21435 + 21534 + 31425 + 31524 + 41523 + 32415 + 32514 + 42513 + 43512$. Note that for two non empty non-ambiguous tree $C, D$.

$$\sum_{T \in \mathbb{M}(C,D)} \sigma_L(T) = \mathbb{B}\mathfrak{S}(\sigma_L(C), \sigma_L(D)) \qquad \text{and} \qquad \sum_{T \in \mathbb{M}(C,D)} \sigma_R(T) = \mathbb{B}\mathfrak{S}(\sigma_R(D), \sigma_R(C))$$

The central argument is the following commutation property:

**Proposition 1.23.** *For a statistic* $S \in \{\mathrm{Inv}, \mathrm{iMaj}\}$, *and* $(\sigma_L, \sigma_R) \in \mathfrak{S}_m \times \mathfrak{S}_n$, *define*

$$\Psi_S((\sigma_L, \sigma_R)) := q_L^{S(\sigma_L)} \frac{x^{m+1}}{[m+1]_{q_L}!}\, q_R^{S(\sigma_R)} \frac{y^{n+1}}{[n+1]_{q_L}!}\,. \tag{16}$$

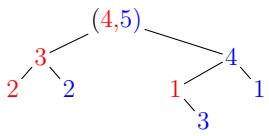
*Then for any pairs* $\sigma = (\sigma_L, \sigma_R)$ *and* $\mu = (\mu_L, \mu_R)$, *one has* $\Psi_S(\mathbb{B}\mathfrak{S}^2(\sigma, \mu)) = \mathbb{B}_q(\Psi_S(\sigma), \Psi_S(\mu))$

As a consequence, noting that $w_S(T) = \Phi_S(\sigma(T))$, one finds that for any non empty non-ambiguous trees $C$ and $D$,

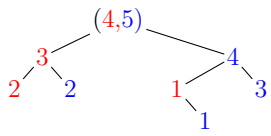
$$\sum_{T \in \mathbb{M}(C,D)} w_S(T) = \Phi_S\left(\mathbb{B}\mathfrak{S}^2(\sigma(C), \sigma(D)\right) = \mathbb{B}_q(w_S(C), w_S(D))\,.$$

Applying this recursively on the structure of a binary tree $B$, we have that $\sum_{T \in \mathcal{NAT}(B)} w_S(T) = \mathbb{B}_q(B)$. Unfolding the recursion for $\mathbb{B}_q(B)$, gives finally Theorem 1.21.

We conclude this section by an example. Let $B =$  . Then one finds that the $q$- hook formula gives $(qx^3 + qx^2 + qx + 1)(qy^2 + qy + 1)(qx + 1)$. Expanding this expression, one finds that the coefficient of $qx^2qy$ is 2. For the iMaj statistic it corresponds to the two following non-ambiguous trees which are shown with their associated left and right permutations:




$((2, 3, 1), (1, 3, 4, 2))\,,$




$((2, 3, 1), (3, 1, 4, 2))$

## 2. Non-ambiguous trees in higher dimension

In this section we give a generalisation of NATs to higher dimensions. NATs are defined as binary trees whose vertices are embedded in $\mathbb{Z}^2$, and edges are objects of dimension 1 (segments). Let $d \geq k \geq 1$ be two integers. In higher dimension, binary trees are replaced by $\binom{d}{k}$-ary trees embedded in $\mathbb{Z}^d$ and edges are objects of dimension $k$. As in Section 1.2 we obtain differential equations for these objects.

2.1. **Definitions.** We call $(d,k)$-*direction* a subset of cardinality $k$ of $\{1,\dots,d\}$. The set of $(d,k)$-directions is denoted by $\Pi_{d,k}$. A $(d,k)$-tuple is a $d$-tuple of $(\mathbb{N} \cup \{\bullet\})^d$, in which $k$ entries are integers and $d-k$ are $\bullet$. For instance, $(\bullet,\ 1,\ \bullet,\ 5,\ 2,\ \bullet,\ \bullet,\ 3,\ \bullet)$ is a $(9,4)$-tuple. The direction of a $(d,k)$-tuple $U$ is the set indices of $U$ corresponding to entries different from $\bullet$. For instance, the direction of our preceding example is $\{2,4,5,8\}$.

**Definition 2.1.** *A* $\binom{d}{k}$-ary *tree $M$ is a tree whose children of given vertex are indexed by a $(d,k)$-direction.*

A $(d,k)$-ary tree has at most $\binom{d}{k}$ children. A $\binom{d}{k}$-ary tree will be represented as an ordered tree where the children of a vertex $S$ are drawn from left to right with respect to the lexicographic order of their indices. If a vertex $S$ has no child associated to an index $\pi$, we draw an half edge in this direction. An example is drawn on Figure 3.

**Definition 2.2.** *A* non-ambiguous tree of dimension $(d,k)$ *is a labelled* $\binom{d}{k}$-ary *tree such that:*

1. *a child of index $\pi$ is labelled with a $(d,k)$-tuple of direction $\pi$ and the root is labelled with a $(d,d)$-tuple;*
2. *for any descendant $U$ of $V$, if the $i$-th component of $U$ and $V$ are different from $\bullet$, then the $i$-th component of $V$ is strictly greater than the $i$-th component of $U$;*
3. *for each $i \in [\![1,d]\!]$, all the $i$th components, different from $\bullet$, are pairwise distinct and the set of $i$th components, different from $\bullet$, of every vertices in the tree, is an interval, whose minimum is* 1.

*The set of non-ambiguous trees of dimensions $(d,k)$ is denoted by $\mathcal{NAT}_{d,k}$.*

We write $\mathrm{NAT}_{d,k}$ for a non-ambiguous tree (of dimensions $(d,k)$). Figure 3 gives an example of a $\mathrm{NAT}_{3,1}$ and a $\mathrm{NAT}_{3,2}$.

**Definition 2.3.** *The* geometric size *of a $NAT_{d,k}$ is the $d$-tuple of integers $(w_1,\dots,w_d)$ which labels the root of the $NAT_{d,k}$, it is denoted by*

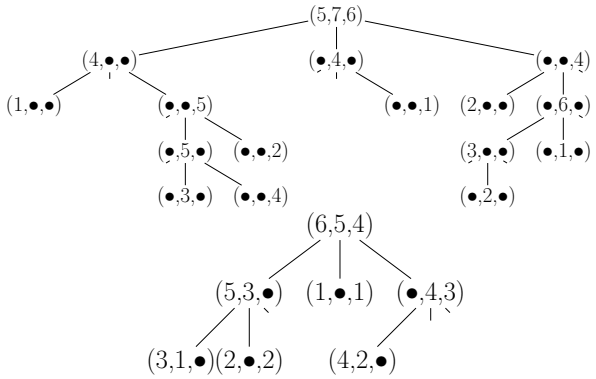



FIGURE 3. A NAT of dimension $(3,1)$ and a NAT of dimension $(3,2)$.

$w_1 \times \cdots \times w_d$. *The* $\pi$-size *of a* $NAT_{d,k}$ *is the number of vertices in the tree of direction* $\pi$*, the set of such vertices is denoted by* $\mathcal{V}_\pi$.

Proposition 2.4 gives the relation between the geometric size and the $\pi$-size of a non-ambiguous trees.

**Proposition 2.4.** *Let* $M$ *be a* $\binom{d}{k}$*-ary tree, the root label is constant on* $\mathcal{NAT}_{d,k}$*s of shape* $M$ *(*$\mathcal{NAT}_{d,k}(M)$*):*

$$w_i(M) := w_i = \sum_{\pi \in \Pi_{d,k} \ | \ i \in \pi} |\mathcal{V}_\pi(M)| + 1.$$

2.2. **Associated differential equations.** In this section, we denote by $x_{\{i_1,\ldots,i_k\}}$ the product $x_{i_1} \times \ldots \times x_{i_k}$, by $\partial_{\{i_1,\ldots,i_k\}}$ the operator $\partial_{x_{i_1}} \partial_{x_{i_2}} \ldots \partial_{x_{i_k}}$ and by $\int_{\{i_1,\ldots,i_k\}}$ the operator $\int_{x_{i_1}} \int_{x_{i_2}} \ldots \int_{x_{i_k}}$. As for non-ambiguous trees (Proposition 1.5), there is a hook formula for the number of non-ambiguous trees with fixed underlying tree. Let $M$ be a $\binom{d}{k}$-ary tree, for each vertex $U$ we denote by $\mathcal{E}_i(U)$ the number of vertices, of the subtree whose root is $U$ (itself included in the count), whose direction contains $i$.

$$|\mathcal{NAT}_{d,k}(M)| = \prod_{i=1}^{d} (w_i(M)-1)! \left( \prod_{U\text{: child of direction containing } i} \mathcal{E}_i(U) \right)^{-1}. \tag{17}$$

There is a $(d,k)$-dimensional analogue of the fixed point differential Equation 9:

**Proposition 2.5.** *The exponential generating function $\mathfrak{N}_{d,k}$ of generalized non-ambiguous trees satisfies the following differential equation*

$$\mathfrak{N}_{d,k} := \sum_{T \in \mathcal{NAT}^*_{d,k}} \prod_{i=1}^{d} \frac{x_i^{w_i(T)-1}}{(w_i(T)-1)!} = \prod_{\pi \in \Pi_{d,k}} \left(1 + \int_\pi \mathfrak{N}_{d,k}\right) \tag{18}$$

*Proof.* The method is analogue to the method of Section 1.2, and goes through the use of a $\binom{d}{k}$-linear map and a pumping function for $\binom{d}{k}$-ary trees. □

The family of differential equations defined by Equation 18 can be rewritten using differential operators instead of primitives. We need to introduce the function $\mathfrak{G}_{d,k} = \int_{\{1,\dots,d\}} \mathfrak{N}_{d,k} + \sum_{\pi \in \Pi_{d,d-k}} x_\pi$. Then, we show that $\mathfrak{G}_{d,k}$ satisfies the following differential equations:

**Proposition 2.6.** *The differential equation satisfied by $\mathfrak{G}_{d,k}$ is $\partial_1 \dots \partial_d \mathfrak{G}_{d,k} = \prod_{\pi \in \Pi_{d,d-k}} \partial_\pi \mathfrak{G}_{d,k}$.*

In the generic case, we are not able to solve those differential equations. We know that setting a variable $x_d$ to 0 gives the generating function of NATs of lower dimension.

**Proposition 2.7.** *Let $d > k \geq 1$, then $\mathfrak{N}_{d,k}|_{x_d=0} = \mathfrak{N}_{d-1,k}$.*

For some specific values of $d$ and $k$ we have (at least partial) results.

**Proposition 2.8.** *Let $k = d-1$, if we know a particular solution $s(x_1, \dots, x_d)$ for*

$$\partial_1 \dots \partial_d \mathfrak{G}_{d,d-1} = \partial_1 \mathfrak{G}_{d,d-1} \times \dots \times \partial_d \mathfrak{G}_{d,d-1}$$

*then, for any function $s_1(x_1), \dots, s_d(x_d)$, the function $s(s_1(x_1), \dots, s_d(x_d))$ is also a solution.*

**Proposition 2.9.** *Some non trivial rational functions are solutions of $\partial_1 \dots \partial_d \mathfrak{G}_{d,1} = \prod_{\pi \in \Pi_{d,d-1}} \partial_\pi \mathfrak{G}_{d,1}$.*

*(sketch).* We define $\mathfrak{G}_{(i)} = \partial_\pi \mathfrak{G}_{d,1}$ where $i \in [\![1,d]\!]$ and $\pi = [\![1,d]\!] \setminus \{i\}$. We get the relation $\partial_i \mathfrak{G}_{(i)} = \prod_{j=1}^d \mathfrak{G}_{(j)}$ and then $\prod_{i=1}^d \partial_i \mathfrak{G}_{(i)} = \prod_{i=1}^d \mathfrak{G}^d_{(i)}$. To obtain a particular solution, we just need to identify, in the previous equation, the term $\partial_i \mathfrak{G}_{(i)}$ to the term $\mathfrak{G}^d_{(i)}$. We thus obtain some non trivial solutions for our equation, which are rational functions. □

Since dimension $(2,1)$ is the unique case where Proposition 2.8 and Proposition 2.9 can be applied at the same time, and the computation of $\mathfrak{N}_{d,d}$ is straightforward, we have the following proposition.

**Proposition 2.10.** *We have the closed formulas:* $\mathfrak{N}_{2,1} = \mathfrak{N}$ *and* $\mathfrak{N}_{d,d} = \sum_{n\geq 0} \frac{(x_1\cdot\ldots\cdot x_d)^n}{(n!)^d}$.

We see $\mathfrak{N}_{d,d}$ as is a kind of generalized Bessel function because $\mathfrak{N}_{2,2}(x/2, -x/2) = J_0(x)$ where $J_\alpha$ is the classical Bessel function. This supports our feeling that the general case leads to serious difficulties.

2.3. **Geometric interpretation.** As for non-ambiguous trees, we can give a geometric definition of non-ambiguous trees of dimensions $(d, k)$ as follows. We denote by $(e_1, \ldots, e_d)$ the canonical basis of $\mathbb{R}^d$ and $(X_1, \ldots, X_d)$ its dual basis, i.e. $X_i$ is $\mathbb{R}$-linear $X_i(e_i) = \delta_{i,j}$. Let $P \in \mathbb{R}^d$ and $\pi = \{i_1, \ldots, i_k\}$ a $(d, k)$-direction, we call *cone of origin* $P$ *and direction* $\pi$ the set of points $C(P, \pi) := \{P + a_1 e_{i_1} + \cdots + a_k e_{i_k} \mid (a_1, \ldots, a_k) \in \mathbb{N}^k\}$.

**Definition 2.11.** *A* geometric non-ambiguous tree *of dimension* $(d, k)$ *and box* $w_1 \times \cdots \times w_d$ *is a non empty set* $\mathcal{V}$ *of points of* $\mathbb{N}^d$ *such that:*

*(1)* $[\![1, w_1]\!] \times \cdots \times [\![1, w_d]\!]$ *is the smallest box containing* $\mathcal{V}$,
*(2)* $\mathcal{V}$ *contains the point* $(w_1, \ldots, w_d)$*, which is called the* root,
*(3) For* $P \in \mathcal{V}$ *different from the root, there exists a unique* $(d, k)$*-direction* $\pi = \{i_1, \ldots, i_k\}$ *such that the cone* $c(P, \pi)$ *contains at least one point different from* $P$*. We say that* $P$ *is of type* $\pi$*.*
*(4) For* $P$ *and* $P'$ *two points of* $\mathcal{V}$ *belonging to a same affine space of direction* $\mathrm{Vect}(e_{i_1}, \ldots, e_{i_k})$*, then, either* $\forall j \in [\![1, k]\!]$, $X_{i_j}(P) > X_{i_j}(P')$*, or* $\forall j \in [\![1, k]\!]$, $X_{i_j}(P') > X_{i_j}(P)$.
*(5) For each* $i \in [\![1, d]\!]$*, for all* $l \in [\![1, w_i - 1]\!]$*, the affine hyperplane* $\{x_i = l\}$ *contains exactly one point of type* $\pi$ *such that* $i \in \pi$*.*

**Proposition 2.12.** *There is a simple bijection between the set of geometric non-ambiguous tree of box* $w_1 \times \cdots \times w_d$ *and the set of non-ambiguous tree of geometric size* $w_1 \times \cdots \times w_d$*.*

*Proof.* If $k = d$, $\mathcal{V}$ is of the form $\{(w, \ldots, w), (w-1, \ldots, w-1), \ldots, (1, \ldots,$ a verifier which corresponds to a exactly to non-ambiguous trees of dimensions $(d, d)$ defined in 2.2.

Let us now suppose that $k < d$.

**2.2 implies 2.11:**

Let $T$ be a non-ambiguous tree of dimension $(d, k)$-defined with Definition 2.2and let $w_1 \times \cdots \times w_d$ be its geometric size. The first step is to define the completed label of a vertex $U$ by replacing the $\bullet$ by integers in the vertices labels, we do it as follows. Let $U$ be a vertex of $T$ such that its $i$th component is a $\bullet$ and let $V$ be his parent. If the $i$th component of $V$ is not a $\bullet$, then replace the $i$th component of $U$ by the $i$th component of $V$. Else replace recursively the $i$th component of

$V$ and then do the replacement. It is equivalent to say that we replace the $i$th component of $U$ with the $i$th component of the first ancestor of $U$ with a $i$th component different from $\bullet$. Such an ancestor exists since the root has no $\bullet$ component. As a consequence, using 2 we deduce that for a vertex $V$ of completed label $(v_1, \dots, v_d)$, if $V$ has a child $U$ indexed by a $(d,k)$-direction $\pi$ and of completed label $(u_1, \dots, u_d)$, then for $i \in \pi$, $v_i > u_i$ and for $i \notin \pi$, $v_i = u_i$.

Let $V$ be a vertex of $T$ of completed label $(u_1, \dots, u_d)$. We denote by $P_V \in \mathbb{N}^d$ the point $(u_1, \dots, u_d)$. Let $\mathcal{V}$ be the set of points $\{P_V \mid V \text{ vertex of } T\}$. Let us prove that $\mathcal{V}$ satisfies the conditions of 2.11.

(1) It is a consequence of 2, 3 of Definition 2.2, and of the definition of the geometric size (2.3).
(2) $\mathcal{V}$ contains $(w_1, \dots, w_d)$, since $(w_1, \dots, w_d)$ is $T$'s root's label.
(3) Let $P_U$ be a point of $\mathcal{V}$ different from $(w_1, \dots, w_d)$. Since $U$ is not the root, it is the child indexed by a $(d,k)$-direction $\pi$, of a vertex $V$. Hence for $i \notin \pi$, $X_i(P_V) = X_i(P_U)$ and for $i \in \pi$, $X_i(P_V) > X_i(P_U)$. So $P_V$ is in the cone of origin $P_U$ and direction $\pi$, in particular, $P_U$ is of type $\pi$. Suppose there is another $(d,k)$-direction $\pi'$ such that the cone of origin $P_U$ and direction $\pi'$ contains a point $P_{V'}$ different from $P_U$. By definition of $V'$ $\forall i \in [\![1, d]\!]$ $X_i(P_{V'}) \geqslant X_i(P_U)$ which is in contradiction with $V'$ being $U$'s descendant.
(4) Let us show this condition by contradiction. Suppose there are two points $P_U$ and $P_{U'}$ contradicting 4. We denote by $\pi = \{i_1, \dots, i_k\}$ the $(d,k)$-direction corresponding to the direction of the affine space. Let $(V_0, V_1, \dots, V_m)$ be the sequence of ancestors of $U$, i.e $V_0 = U$ for all $j$, $V_j$ is a child of $V_{j+1}$ and $V_m$ is the root of $T$. If $m = 0$, then $l = 0$, else let $l$ the index such that $V_{l-1}$ is a child of $V_l$ not indexed by $\pi$ and $\forall j \in [\![0, l-1]\!]$, $V_j$ is the child indexed by $\pi$ of $\mathcal{V}_{j+1}$. We denote $V_l$ by $V$ and in the same way, we define $V'$. $U$ cannot be an ancestor of $U'$ and vice versa, hence, $V$ and $V'$ are different, furthermore, $V$ cannot be an ancestor of $V'$ and vice versa. Since $V$ is not a child indexed by $\pi$, there exists $i$ such that the $i$th component $v_i$ of $V$ is not $\bullet$ and $i$ is in $\pi$. As a result, by Condition 3 of Definition 2.2 and the construction of completed labels, $X_i(P_{V'}) \neq v_i = X_i(P_V)$, yet, by construction, as $i \in \pi$, $X_i(V) = X_i(U) = X_i(U') = X_i(V')$.
(5) The uniqueness comes from the fact that if the type $\pi$ of a point $P_U$ contains $i$, since the index of $U$ is $\pi$, the $i$th component of

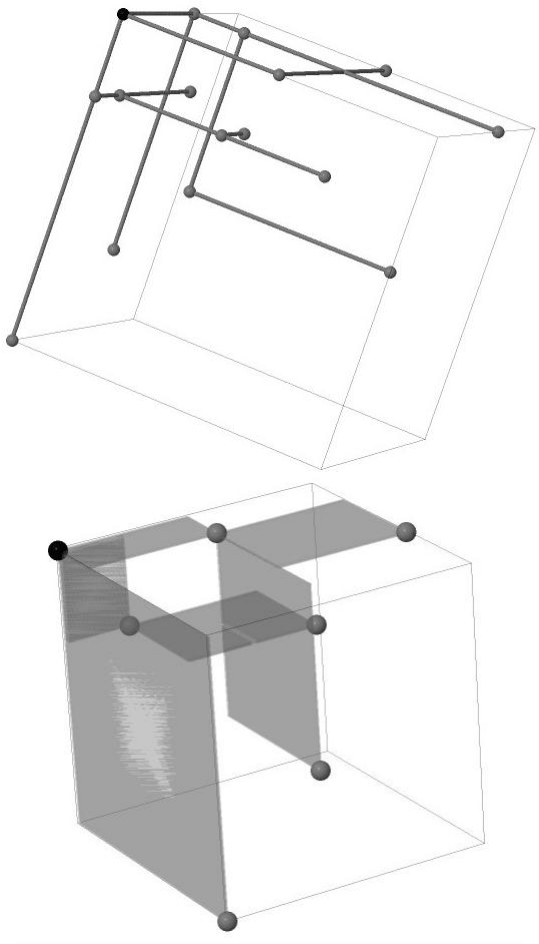


FIGURE 4. Geometric representation of the NATs of Figure 3.

$U$ is not $\bullet$. Hence, using 3 and 3, we proved that this condition is satisfied.

**2.11 implies 2.2:**

Let $T$ be a non-ambiguous tree defined with Definition 2.11. We start by constructing the underlying $\binom{d}{k}$-ary tree $M$ of $T$. The vertices of $M$ correspond to the points of $\mathcal{V}$, in particular, the root of $M$ corresponds to the root of $T$. Let $P$ be a point of $\mathcal{V}$ different from the root, we denote by $V_P$ the corresponding vertex of $M$. Consider the cone defined by Condition 3 in Definition 2.11. By Condition 4, we can define the point $P'$ of the cone which is the closest to $P$, we set $V_{P'}$ to be the parent of $V_P$. We index the child $V_P$ by the $(d,k)$-direction $\pi$ corresponding

to the cone. By 4, a point cannot have more than one child of the same direction. The labelling is done as follows. We start by labelling the vertices $V_P$ by the coordinates of $P$. Then for each vertex $V$ and each $(d,k)$-direction $\pi$, for all $i \notin \pi$, we replace the $i$th component of the child of $V$ indexed by $\pi$, by $\bullet$. Let us prove that the condition of Definition 2.2 are satisfied. Thus, if the $i$th component of a vertex $V_P$ is equal to $l$ then for a descendant $V_{P'}$ of $V_P$, if $X_i(P') = l$, then the $i$th component of $V_{P'}$ is $\bullet$.

(1) By construction of the labels.
(2) It is a consequence of Condition 2 of Definition 2.11.
(3) If $V_P$ is the child of $V_{P'}$ of direction $\pi$. For $i \in \pi$, $X_i(P') > X_i(P)$ and for $i \notin \pi$, $X_i(P') = X_i(P)$. Hence, for a vertex $V_P$ of $i$th component different from $\bullet$, the $i$th component of one of his descendant is either smaller or $\bullet$.
(4) Let $V_P$ be a vertex of $M$ such that its $i$th component is different from $\bullet$. If $V_P$ is the root, then all the vertices of $M$ are its descendants, hence its $i$th label appears only once. Else, $V_P$ is the child indexed by $\pi$ of a vertex $V_{P'}$. In particular, $\pi$ contains $i$ since the $i$th component of $V_P$ is not $\bullet$. Hence, by Condition 5 of Definition 2.11, the $i$th component of $V_P$ is unique.
(5) Condition 5 implies that for each $i \in [\![1, d]\!]$, for all $l \in [\![1, w_i - 1]\!]$ there is a point $P$ of $\mathcal{V}$ such that $X_i(P) = l$. Moreover, the coordinates of the root are $(w_1, \dots, w_d)$ and $T$ is contained in the box $[\![1, w_1]\!] \times \cdots \times [\![1, w_d]\!]$. Therefore, the set of $i$th components, different from $\bullet$, is the interval $[\![1, w_i]\!]$.

□

## 3. A new statistic on binary trees : the hook statistic

We present in this section a bijection between binary trees and ordered trees, sending the vertices to edges and the hook statistic defined in Definition 1.10 to the number of vertices having at least a child which is a leaf, what we will call the *child-leaf statistic*. The corresponding integer series appears as [Slo, A127157] in OEIS.

We denote by $\mathcal{B}_p$ and $\mathcal{O}_p$ respectively the exponential generating series of these trees, with these statistics, the variable $x$ indexing the number of vertices and $t$, the statistic.

Then, these generating series satisfy :

**Proposition 3.1.** *The generating series of binary trees with hook statistic and ordered trees with the child-leaf statistic are given by the following functional equations:*

$$\mathcal{B}_p = 1 + xt \times \left(\frac{1}{1 - x\mathcal{B}_p}\right)^2$$

$$\mathcal{O}_p = \frac{1}{1 - x(\mathcal{O}_p - 1))} \times \left(1 + xt \times \frac{1}{1 - x\mathcal{O}_p}\right)$$

*These generating series are equal.*

*Proof.* The first functional equation is obtained by considering the vertices in the hook of the root : there can be none or there is a root, a list of left descendant (whose right child is a binary tree) and a list of right descendant (whose left child is a binary tree).

The second functional equation is obtained by considering, if the ordered tree is not reduced to a vertex, the first leaf of the root from left to right, if it exists. Then, on the left side of this leaf, there is a list of non empty ordered tree and on the right side also, if there is a leaf.

Then, by multiplying the preceeding equations by $1 - x\mathcal{B}_p$ and $1 - x(\mathcal{O}_p - 1))$ respectively, they are equivalent to:

$$\mathcal{B}_p - x\mathcal{B}_p^2 = 1 - x\mathcal{B}_p + xt\frac{1}{1 - x\mathcal{B}_p}$$

$$\mathcal{O}_p - x\mathcal{O}_p^2 + x\mathcal{O}_p = 1 + xt\frac{1}{1 - x\mathcal{O}_p}.$$

□

Let us now exhibit a bijection between these two objects. This bijection comes from the following equation:

$$(\mathcal{B}_p - 1) - x(\mathcal{B}_p - 1) - x(\mathcal{B}_p - 1)^2 = xt\frac{1}{1 - x\mathcal{B}_p}.$$

This equation can be viewed as considering only binary trees whose root has no left descendants or ordered trees such that the leftmost child of the root is a leaf.

We obtain the following bijection:

**Proposition 3.2.** *The map $\zeta$ sends a binary tree $B$ to an ordered tree $O$ by mapping:*

- *the leftmost descendant of the root, if it is a leaf, to an edge between the root and its only child*

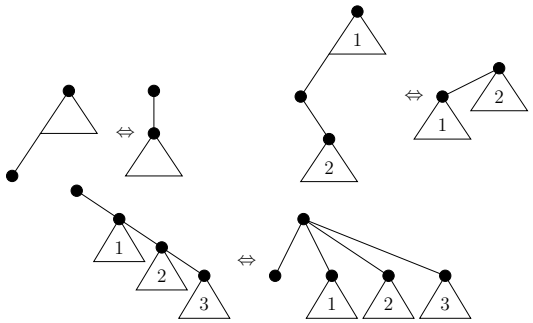



FIGURE 5. Bijection $\zeta$

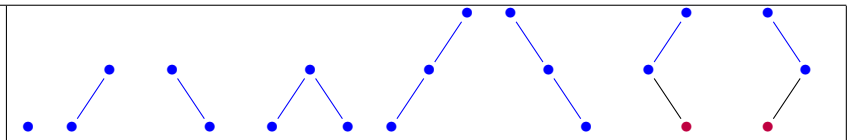
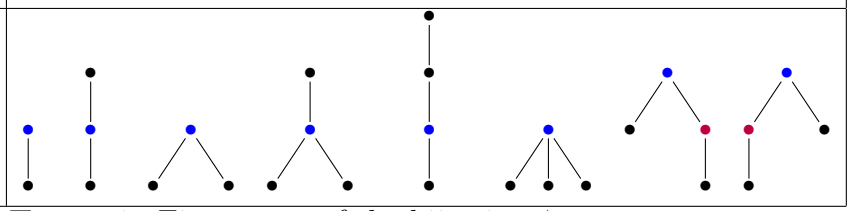
| | |
|---|---|
| binary trees |  |
| ordered trees |  |

TABLE 1. First terms of the bijection $\zeta$

- *the leftmost descendant of the root $v$ to an edge between the root of the tree associated with the descendants of $v$ and the root of the tree obtained from what is left*
- *the set of right descendants of the root to the set of children of the root.*

*It is a bijection between binary trees and ordered trees, sending the vertices to edges and the hook statistic to the child-leaf statistic.*

We sum up this bijection on Figure 4.
We present in Table 1 the first terms in the bijection.

## CONCLUSION

a faire

**Acknowledgement.** This research was driven by computer exploration using the open-source software `Sage` [S+15] and its algebraic combinatorics features developed by the `Sage-Combinat` community [SCc08].

## References


[ABBS14] J.C. Aval, A. Boussicault, M. Bouvel, and M. Silimbani. Combinatorics of non-ambiguous trees. *Advances in Applied Mathematics*, 56:78–108, May 2014.

[ABN13] Jean-Christophe Aval, Adrien Boussicault, and Philippe Nadeau. Tree-like tableaux. *Electron. J. Combin.*, 20(4):Paper 34, 24, 2013.

[BW89] A. Björner and M. L. Wachs. q-hook length formulas for forests. *J. Comb. Theory Ser. A*, 52(2):165–187, November 1989.

[CW07] Sylvie Corteel and Lauren K. Williams. Tableaux combinatorics for the asymmetric exclusion process. *Adv. in Appl. Math.*, 39(3):293–310, 2007.

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

[HNT08] F. Hivert, J.C. Novelli, and J.Y. Thibon. Trees, functional equations, and combinatorial Hopf algebras. *European Journal of Combinatorics*, 29(7):1682–1695, 2008.

[Pos07] Alexander Postnikov. Total positivity, grassmannians, and networks, 2007.

[S+15] W. A. Stein et al. *Sage Mathematics Software (Version 6.10.beta1)*. The Sage Development Team, 2015. `http://www.sagemath.org`.

[SCc08] The Sage-Combinat community. Sage-Combinat: enhancing Sage as a toolbox for computer exploration in algebraic combinatorics, 2008. `http://combinat.sagemath.org`.

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

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

[Vie08] Xavier Viennot. Alternative tableaux, permutations and partially asymmetric exclusion process. Slides of a talk at the Isaac Newton Institute in Cambridge, 2008.



Laboratoire Bordelais de Recherche en Informatique (UMR CNRS 5800), Université de Bordeaux, 33405 TALENCE

Laboratoire Bordelais de Recherche en Informatique (UMR CNRS 5800), Université de Bordeaux, 33405 TALENCE

Institut de Mathématiques de Toulouse (UMR CNRS 5219), Université Paul Sabatier, 31062 TOULOUSE

Laboratoire de Recherche en Informatique (UMR CNRS 8623), Bâtiment 650, Université Paris Sud 11, 91405 ORSAY CEDEX

Laboratoire Bordelais de Recherche en Informatique (UMR CNRS 5800), Université de Bordeaux, 33405 TALENCE