Non-ambiguous trees: new results and generalization

Jean-Christophe Aval [1] and Adrien Boussicault [1] and Bérénice Delcroix-Oger [2] and Florent Hivert [3] and Patxi Laborde-Zubieta [1]

[1] *LAboratoire Bordelais de Recherche en Informatique (UMR CNRS 5800), Université de Bordeaux, 33405 TALENCE*
[2] *Institut de Mathématiques de Toulouse (UMR CNRS 5219), Université Paul Sabatier, 31062 TOULOUSE*
[3] *Laboratoire de Recherche en Informatique (UMR CNRS 8623) Bâtiment 650, Université Paris Sud 11, 91405 ORSAY CEDEX*

received 16th November 2015,

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.

Nous introduisons une nouvelle définition des arbres non ambigus (NATs) en terme d'arbres binaires étiquetés. Nous en dÃ©duisons une équation différentielle, dont les solutions peuvent être décrites de manière combinatoire. Ceci conduit à une nouvelle formule pour le nombre de NATs. Nous démontrons aussi des q-versions des formules obtenues. Enfin, nous généralisons la notion de NAT en dimension supÃ©rieure.

Keywords: Non-ambiguous trees, binary trees, ordered trees, q-analogues, permutations

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 permutation 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 $\sigma = \sigma_1\,\sigma_2\,\ldots\,\sigma_n$ such that the excedences ($\sigma_i > i$) are placed at the beginning of the word σ. 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 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 [HNT08] (see Theorem 1.22, the relevant statistics are either the number of inversions or the inverse major index).
Section 2 presents a generalization 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 [(i)]. 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.

This version of our work is an *extended abstract*; most proofs are only sketched or purely omitted.

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.

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 have different labels. Otherwise said, 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 $(w_L(T), w_R(T)) = (|\mathcal{V}_L(T)| + 1, |\mathcal{V}_R(T)| + 1)$.

[(i)] A definition in terms of labelled trees is given in Subsection 2.1.

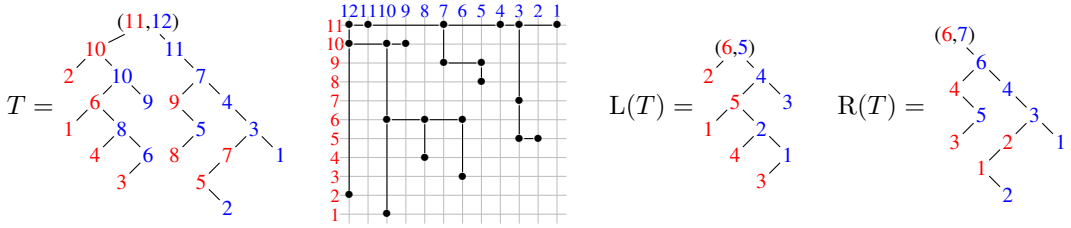


Fig. 1: A non-ambiguous tree and its left and right subtrees

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 $T = \begin{smallmatrix} & \bullet & \\ B_L & & B_R \end{smallmatrix}$ be a non empty binary tree. Restricting the labellings of the left and right children of T to B_L and B_R gives non-decreasing labelling of their respective left and right children. Note that the root of B_L (resp. B_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 B_L and B_R, that is two non-ambiguous trees $\mathrm{L}(T)$ and $\mathrm{R}(T)$. See Figure 1 for an example.

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

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

Our first step is to recover hoo-k 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. By linearity, we consider the map $\mathbb{M}$ as 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 $\mathcal{NAT}\,(T)$ of non-ambiguous tree of shape T satisfies the following recursion:*

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

To count non-ambiguous tree, 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 tree 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, $\Phi(\mathcal{NAT}\,(T))$ is the generating series of the non-ambiguous trees of shape T. Thanks to (1) the image in $\mathbb{Q}[[x,y]]$ of the bilinear map $\mathbb{M}$ under the map Φ 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 T *an element* $\mathbb{B}(T) \in \mathbb{Q}[[x,y]]$ *by*

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

Now (1) rewrites as

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

By de-recursiving the expression for $\mathbb{B}(T)$, 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*

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

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

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

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

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

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

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

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:*

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

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

$$\mathfrak{G} = x + y + \sum_{B_1, B_2 \in \mathcal{BT}} \mathbb{B}\left(\begin{array}{ccc} & \bullet & \\ B_1 & & B_2 \end{array}\right) = x + y + \sum_{B_1, B_2 \in \mathcal{BT}} \mathbb{B}(\mathbb{B}(B_1), \mathbb{B}(B_2)) = x + y + \mathbb{B}(\mathfrak{G}, \mathfrak{G}).$$

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 $\mathfrak{N}$.

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

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

Now, we will introduce two statistics : the number of right (resp. left) vertices in the right (resp. left) 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 (α, β)-generating function for non-ambiguous trees:

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

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 (α, β)-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, the trees will be represented with parents upper than children. Therefore, it will make sense to talk about vertices on the left and on the right of other vertices.

The solution of Proposition 1.9 can be rewritten as :

$$\mathfrak{N}^{(\alpha,\beta)} = e^{\alpha x} e^{\beta y} e^{-\alpha \ln(1-(e^x-1)(e^y-1))} e^{-\beta \ln(1-(e^x-1)(e^y-1))}. \tag{10}$$

The purpose of this subsection is to explain this expression combinatorially. Let us first describe objects "naturally" enumerated by the RHS of (10). 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. 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:

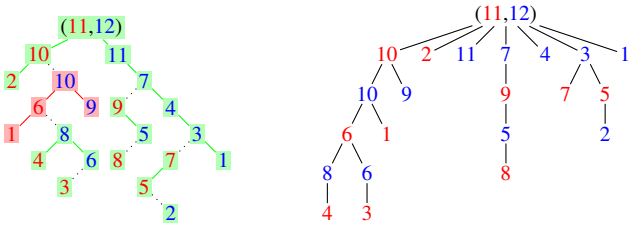


Fig. 2: Hooks on a non-ambiguous tree and associated ordered tree

Definition 1.10 *Let T 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 and in red and green the partition of the set of vertices in hooks. 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, bot 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 ξ.*

Proof: Let us consider a non-ambiguous tree N and construct an ordered tree $\xi(N) = O$. The root of N 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 N. 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 N, then its children in the ordered tree is the set of right (resp. left) descendants of v in N, 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 $\{1, \dots |\mathcal{V}_R| - 1\}$ and red letters on $\{1, \dots, |\mathcal{V}_L| - 1\}$, where 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 (sketch): From $T \in \mathcal{NOT}$, we obtain the two words $\Omega(T) = (w_1, w_2)$ by a post-order traversal visit of the descendant of the red (resp. blue) children of the root for w_1 (resp. w_2). The injectivity of Ω can be shown *ad absurdum*.

From such a word, we can build back recursively the associated ordered trees by reading each word from right to left and adding, for each new letter l, a node labelled by l to the left of the closest ancestor of the current position whose label is of the same colour as l and smaller than l.

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 first (resp. second) set of the 4-tuple can be defined as the set of blue (resp. red) children of the root in the ordered tree. Then, 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. Each of these subwords can be viewed as a blue (resp. red) cycle, as the child of the root is the biggest blue (resp. red) 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}{61109})}, {({85}{97}),({72}{53})).*

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

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

Theorem 1.19 *The number of non empty non-ambiguous trees with w left vertices and h right vertices is given by:*

$$\mathcal{NAT}_{w,h} = \sum_{p \geq 1} (p+1)! p! S_2(w+1, p+1) S_2(h+1, p+1), \tag{11}$$

where S_2 denotes Stirling numbers of the second kind. In this positive summation expression, each summand corresponds to the number of NATs with prescribed size, and whose number of hooks equals p.

We conclude this subsection with following result on binary trees. The corresponding integer series appears as A??? in OEIS.

Proposition 1.20 *The set of binary trees on n vertices with hook number p is in bijection with the number of ordered trees on $n+1$ vertices having p vertices being the parent of at least a leaf.*

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 1.21 *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 is a right 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 following two permutations:

$$\sigma_L(T) = (2,1,4,3,6,10,8,9,5,7), \quad \sigma_R(T) = (1,2,3,4,5,7,11,9,6,8,10)\,.$$

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 σ is a $i < n$ such that $\sigma(i) > \sigma(i+1)$ and the *inverse major index* of σ 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 1.22 *For a non-ambiguous tree N and a statistic $S \in \{\mathrm{Inv}, \mathrm{iMaj}\}$, we define*

$$w_S(T) := q_L^{S(\sigma_L(T))} q_R^{S(\sigma_R(T))}. \tag{12}$$

Then, for any non empty binary tree B

$$\sum_{T \in \mathcal{NAT}(B)} w_{\mathrm{Inv}}(T) \;=\; \sum_{T \in \mathcal{NAT}(B)} w_{\mathrm{iMaj}}(T) \;=\; \frac{|\mathcal{V}_L(B)|_{q_L}! \cdot |\mathcal{V}_R(B)|_{q_R}!}{\prod_{U:\textit{left child}} [\mathcal{E}_L(U)]_{q_L} \cdot \prod_{U:\textit{right child}} [\mathcal{E}_R(U)]_{q_R}}\,. \tag{13}$$

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:

$$\sigma_L(T)^{-1} = (2,1,4,3,9,5,10,7,8,6), \quad \sigma_R(T)^{-1} = (1,2,3,4,5,9,6,10,8,11,7)\,.$$

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

$$\partial_{x,q} x^n := [n]_q x^{n-1}\,, \qquad \text{and} \qquad \int_{x,q} x^n := \frac{x^{n+1}}{[n+1]_q}\,. \tag{14}$$

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 $\mathbb{B}_q(T)$ by $\mathbb{B}_q(\emptyset) := x + y$ and $\mathbb{B}_q(\,{}_L\!\!\diagup\!\!\diagdown_R\,) = \mathbb{B}_q\left(\mathbb{B}_q(L), \mathbb{B}_q(R)\right)$. 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 1.23 *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 1.24 *For a statistic* $S \in \{\mathrm{Inv}, \mathrm{iMaj}\}$*, and* $(\sigma_L, \sigma_R) \in \mathfrak{S}_m \times \mathfrak{S}_n$*, define*

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

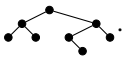
Then for any pairs $\sigma = (\sigma_L, \sigma_R)$ *and* $\mu = (\mu_L, \mu_R)$*, one has* $\Psi_S(\mathbb{B}\mathfrak{S}^2(\sigma,\mu)) = \mathbb{B}_q(\Psi_S(\sigma), \Psi_S(\mu))$

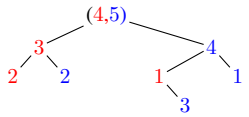
As a consequence, noting that $w_S(T) = \Phi_S(\sigma(T))$, one finds that for any non empty non-ambiguous trees N_L and N_R,

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

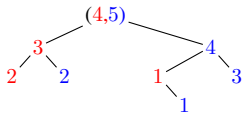
Applying this recursively on the structure of a binary tree B, we have that $\sum_{T \in \mathcal{NAT}(B)} w_S(T) = \mathbb{B}_q(B)$.

Unfolding the recursion for $\mathbb{B}_q(B)$, gives finally Theorem 1.22.

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 generalization of NATs to higher dimensions. In dimension two, 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 π, 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 π is labelled with a (d,k)-tuple of direction π;*
2. *the root is labelled with a (d,d)-tuple;*
3. *for any descendant U of V, if the i-th component of U and V are different from $\bullet$, then the ith component of V is strictly greater than the i-th component of U;*
4. *for each $i \in [\![1,d]\!]$, all the ith components, different from $\bullet$, are pairwise distinct;*
5. *for each $i \in [\![1,d]\!]$, the set of ith 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 $\mathrm{NAT}_{d,k}$ is the d-tuple of integers $(w_1,\dots,w_d)$ which label the root of the $\mathrm{NAT}_{d,k}$, it is denoted by $w_1\times\cdots\times w_d$. The* π-size *of a $\mathrm{NAT}_{d,k}$ is the number of vertices in the tree of direction π, the set of such vertices is denoted by $\mathcal{V}_\pi$.*

Property 2.4 gives the relation between the geometric size and the π-size of a non-ambiguous trees.

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

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

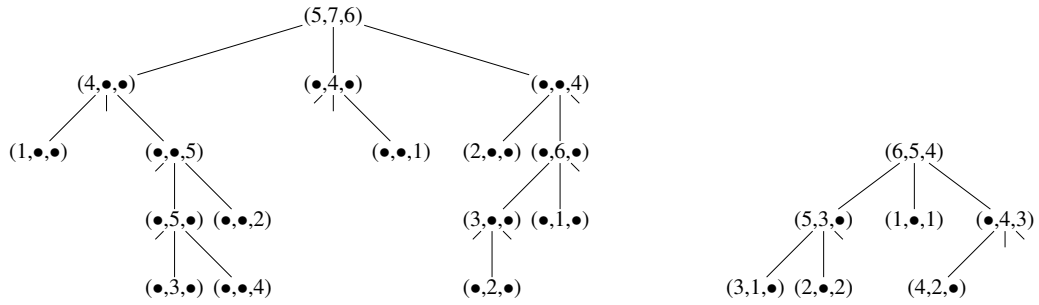


Fig. 3: A NAT of dimension $(3,1)$ and a NAT of dimension $(3,2)$.

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), we have a hook formula for the number of non-ambiguous trees with fixed underlying tree. Let M be a $\binom{d}{k}$-ary tree, we label each vertex V of direction π, by a (d,k)-tuple t of direction π, such that for $i\in\pi$, the ith component t_i of t, is the number of vertices of the subtree whose root is V, whose direction contains i. If we denote by $\mathcal{E}_\pi$ the multi-set of labels of π-vertices, then the hook formula of M is given by:

$$\prod_{i=1}^{d}\frac{(w_i(M)-1)!}{\prod\limits_{\substack{\pi\in\Pi_{d,k}\\ i\in\pi}}\prod\limits_{t\in\mathcal{E}_\pi(M)}t_i(t)}.$$

Let $\mathfrak{N}_{d,k}$ the exponential generating function of generalized non-ambiguous trees defined by :

$$\mathfrak{N}_{d,k}=\sum_{N\in\mathcal{NAT}^*_{d,k}}\prod_{i=1}^{d}\frac{x_i^{w_i(N)-1}}{(w_i(N)-1)!}.$$

As in section 1.2, the .

Proposition 2.5 *The function $\mathfrak{N}_{d,k}$ satisfies the differential equation (similar to Subsection 1.2):*

$$\mathfrak{N}_{d,k}=\prod_{\pi\in\Pi_{d,k}}\left(1+\int_\pi\mathfrak{N}_{d,k}\right)\tag{17}$$

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

The family of differential equations defined by Equation 17 can be rewritten using differential operators instead of primitives. We need to introduce the function $\mathfrak{G}_{d,k}=\int_{\{1,\ldots,d\}}\mathfrak{N}_{d,k}+\sum_{\pi\in\Pi_{d,d-k}}x_\pi$. 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* $\partial_1 \dots \partial_d \mathfrak{G}_{d,1} = \prod_{\pi \in \Pi_{d,d-1}} \partial_\pi \mathfrak{G}_{d,1}$.

Proof (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 $\prod_{i=1}^d \partial_i \mathfrak{G}_{(i)} = \prod_{i=1}^d \mathfrak{G}_{(i)}^d$. To obtain a particular solution, we just need to identify, in the previous equation, the term $\partial_i \mathfrak{G}_{(i)}$ to the term $\mathfrak{G}_{(i)}^d$. We thus obtain some non trivial solutions for our equation, which are rational functions. □

For $d = 2$ and $k = 1$ we can find a closed formula because it is the unique case where Property 2.8 and Property 2.11 can be applied at the same time.

Proposition 2.10 *Let* $d = 2$ *and* $k = 1$*, then* $\mathfrak{N}_{2,1} = \mathfrak{N}$.

The case $d = k$, is easy to solve

Proposition 2.11 *The function* $\mathfrak{N}_{d,d}$ *is equal to* $\sum_{n \geq 0} \frac{(x_1 \cdot \dots \cdot x_d)^n}{(n!)^d}$.

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

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.

[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.

[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.