Sorted and/or sortable permutations

(Extended abstract - Proofs omitted)

Mireille Bousquet-Mélou*

Abstract

In his Ph.D. thesis [20], Julian West studied in depth a map Π that acts on permutations of the symmetric group $\mathcal{S}_n$ by partially sorting them through a stack. The main motivation of this paper is to characterize and count the permutations of $\Pi(\mathcal{S}_n)$, which we call sorted permutations. This is equivalent to counting preorders of binary decreasing trees. Using an extension of Zeilberger's factorization of two-stack sortable permutations [22], we obtain for the generating function of sorted permutations a strange functional equation.

Out of curiosity, we apply the same treatment to four other families of permutations (general permutations, one-stack sortable permutations, two-stack sortable permutations, sorted and sortable permutations) and compare the functional equations we obtain. All of them have similar features, involving a divided difference. Moreover, most of them have interesting q-analogues obtained by counting inversions. We solve (some of) our equations.

Résumé

Julian West a étudié dans sa thèse [20] une transformation Π qui agit sur les permutations du groupe symétrique $\mathcal{S}_n$ en les triant partiellement par passage dans une pile. L'objectif principal de cet article est de caractériser et compter les permutations de $\Pi(\mathcal{S}_n)$, que nous appelons permutations triées. Nous montrons que ceci équivaut à compter les différents ordres préfixes des arbres binaires décroissants. En utilisant une extension de la factorisation des permutations triables par deux piles décrite par Zeilberger [22], nous obtenons pour la série génératrice des permutations triées une curieuse équation fonctionnelle.

Nous appliquons la même approche à quatre autres familles de permutations (permutations génerales, triables par une et deux piles, triées et triables par une pile), et comparons les équations fonctionnelles ainsi obtenues : toutes font intervenir des différences divisées, et la plupart admettent un q-analogue simple, obtenu en comptant les inversions (ou une variante des inversions). Nous résolvons certaines de ces équations.

1 Introduction

1.1 The sorting procedure

In his Ph.D. thesis [20], Julian West studied a procedure Π that permutes the letters of a word σ having distinct letters in the alphabet $\{1, 2, 3, \ldots\}$. The procedure uses a stack and works as follows (Fig.1). At the beginning, the word $\sigma = \sigma^{(0)}$ lies to the right of the stack, which is empty. If σ has n letters, the procedure will have $2n$ steps. After the ith step, for $i \geq 0$, a word $\sigma^{(i)}$ lies to the right of the stack, while a word $\tau^{(i)}$ lies to the left of the stack. If $\sigma^{(i)}$ is not empty, and if its first letter, say a, is smaller than the top letter of the stack (or if the stack is empty), we add a to the top of the stack. Otherwise, we remove the top letter from the stack and add it at the end of $\tau^{(i)}$. In other words, we add letters to the stack as long as it remains a "Hanoï tower", and otherwise remove letters from the stack. The word $\tau^{(2n)}$ has n letters, and we define it to be $\Pi(\sigma)$, the word obtained by *sorting σ through a stack.*

This procedure extends a procedure described by Knuth [15, p. 238] (although Knuth's procedure, nicely described in terms of railway switching networks, goes somehow backwards). As observed by West [21], the map Π can alternatively be described recursively by

$$\Pi(\sigma^L m \sigma^R) = \Pi(\sigma^L)\Pi(\sigma^R)m, \tag{1}$$

where m is the largest letter of the word $\sigma = \sigma^L m \sigma^R$. We observe that, if σ has n letters, then $\Pi^{n-1}(\sigma)$ is an increasing word; this shows that Π really *sorts* the letters of σ (although not very fast!).

*LaBRI, Université Bordeaux 1, 351 cours de la Libération, 33405 Talence Cedex, France — E-mail: bousquet@labri.u-bordeaux.fr
Partially supported by the Conseil Régional d'Aquitaine.

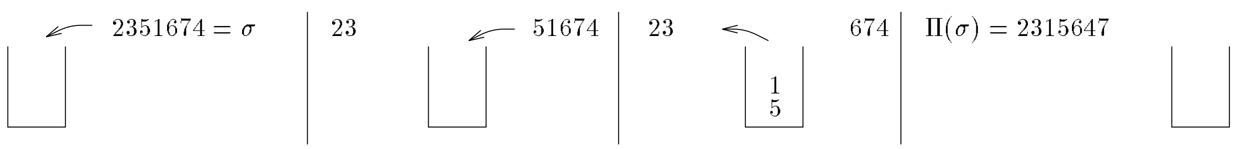


Figure 1: The sorting algorithm applied to $\sigma = 2351674$.

Clearly, we can restrict our attention to the action of Π on permutations. Let $\mathcal{S}_n$ be the set of permutations of length n. Following West [20], we represent the action of Π on $\mathcal{S}_n$ by a *sorting tree*: the nodes of this tree are the elements of $\mathcal{S}_n$, and an edge connects σ to $\Pi(\sigma)$ for all $\sigma \in \mathcal{S}_n$ (Fig.2).

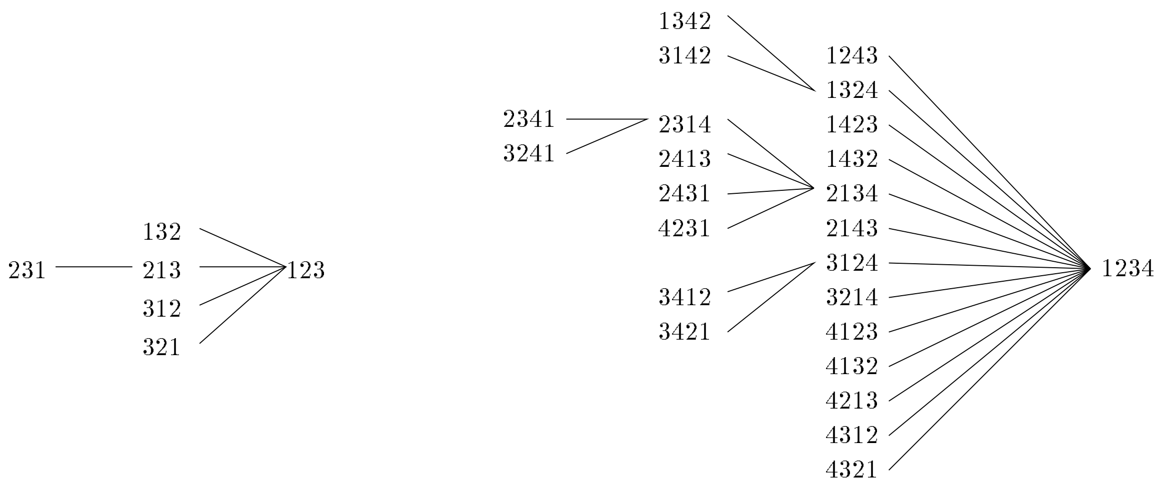


Figure 2: The sorting trees for $\mathcal{S}_3$ and $\mathcal{S}_4$.

We can visualize on this tree the four classes of permutations we will consider in this paper.

• One-stack sortable permutations

A permutation $\sigma \in \mathcal{S}_n$ is *one-stack sortable* if $\Pi(\sigma) = 12\ldots n$, i.e., if it occurs in the last two columns of the sorting tree. It is known [15] that the number of such permutations is the Catalan number $C_n = \binom{2n}{n}/(n+1)$, and that these permutations are exactly the permutations avoiding the pattern 231: there exist no triple (i,j,k) with $1 \le i < j < k \le n$ such that $\sigma(k) < \sigma(i) < \sigma(j)$.

• Two-stack sortable permutations

A permutation $\sigma \in \mathcal{S}_n$ is *two-stack sortable* if $\Pi(\Pi(\sigma)) = 12\ldots n$, i.e., if it occurs in the last three columns of the sorting tree. West characterized these permutations in terms of forbidden patterns [21] and conjectured that their number is $c_n = 2(3n)!/[(2n+1)!(n+1)!]$. This conjecture was first proved by Zeilberger [22]. Two bijective proofs [10, 14] were found later, based on the fact that c_n is the number of non-separable planar maps [4, 7]. Note that the corresponding generating function $\sum c_n t^n$ is cubic over $\mathbb{R}(x)$.

• Sorted permutations

A permutation $\tau \in \mathcal{S}_n$ is *sorted* if it belongs to $\Pi(\mathcal{S}_n)$. In other words, the sorted permutations are the inner nodes of the sorting tree, or, using West's terminology [20], the nodes of *positive fertility*.

Characterizing and counting these permutations is the main motivation of this paper. Let us make three simple remarks about them. First of all, if $\tau \in \mathcal{S}_n$ is sorted, then it ends with n. But this is not a sufficient condition, as shown by $\tau = 3214$, which is not sorted (see Fig.2). Then, if $\tau = \tau^L(n-1)\tau^R n$, where τ^R is not empty, then (1) implies that τ is sorted if and only if $\tau^L(n-1)$ and τ^R are sorted words. But if $\tau = \tau^L(n-1)n$, there is no obvious way of deciding whether τ is sorted or not. Finally, we can see by induction of the length of permutations that any suffix of a sorted permutation is a sorted word. Of course, no similar property holds for prefixes.

Concerning the characterization of sorted permutations, we shall first obtain a negative result: sorted permutations cannot be described in terms of forbidden patterns. More precisely, any pattern occurs as a factor in some sorted permutation. We shall also give a positive result, which takes the form of a linear algorithm that decides whether a permutation is sorted, and, in this case, exhibits one of its pre-images.

Concerning the enumeration of sorted permutations, we shall give a functional equation (of a strange sort) satisfied by their generating function. So far, we have not been able to say whether this generating function is *D-finite* [18], or at least *differentiably algebraic* [9].

• **Sorted and (one-stack) sortable permutations**

We can describe these permutations by any of the three equivalent conditions:

– $\tau \in \Pi(\mathcal{S}_n)$ and $\Pi(\tau) = 12\ldots n$,

– τ is the image by Π of a two-stack sortable permutation,

– τ is an inner node of one of the last two columns of the sorting tree.

We will characterize these permutations and show that their generating function is algebraic of degree 4.

One of the main tools of this paper is a factorization of permutations that stabilizes the four classes of permutations described above: essentially, a permutation will be one-stack sortable (resp. two-stack sortable, sorted, sorted and sortable) if and only if its factors are one-stack sortable (resp. two-stack sortable, sorted, sorted and sortable). This property will enable us to write, for each of these four classes, a functional equation defining their generating function.

1.2 Functional equations

A lot of work has already been done on one- and two-stack sortable permutations [10, 11, 14, 15, 21, 22]. Hence, enumerating them once again might seem slightly uninteresting. But we are actually going to be even more naïve: to the list of our four functional equations, we will add a fifth equation corresponding to the enumeration of... all permutations.

The reason for this stubbornness is that we are very much interested in the equations themselves (Section 4). Each of them defines implicitely a bivariate power series $F(x,y)$ and involves a divided difference

$$\Delta F(x,y) \stackrel{\text{def}}{=} \frac{F(x,y) - F(x,0)}{y}.$$

In all cases, we are mostly interested in $F(x,0)$; but the form of the equation that defines $F(x,y)$ makes the substitution $y = 0$ impossible (or tautological), and there is no obvious way of deriving information on $F(x,0)$ only (not talking of *solving* the entire equation).

Such equations are quite frequent in enumerative combinatorics. Examples can be found in the enumeration of permutations [3, 22, 15, p.532–534], of polygons [2, 12], and are ubiquitous in the enumeration of maps [4, 7, 8, 19]. To our knowledge, all examples that have been solved so far were polynomial in $F(x,y)$ and $F(x,0)$, of degree at most 3 in $F(x,y)$ [6]. Their solution, inspired by the remarkable *quadratic method* of Brown [5, 13, section 2.9.1], provides an *algebraic* equation satisfied by $F(x,0)$.

Three out of our five equations are polynomial in $F(x,y)$ and $F(x,0)$, and can be solved using previously known tools. The last two involve a partial derivative $\partial F/\partial x(x,y)$. They look very much like each other, but one of them is relative to general permutations and has an extremely simple solution, while the other is relative to sorted permutations and will remain quite mysterious. However, we have found a method to derive, from the functional-differential equation satisfied by $F(x,y)$, a (strange) equation satisfied by $F(x,0)$. This method appears as a "differential analogue" of the quadratic method.

Finally, we will enrich our collection of equations with some q-analogues. They will be obtained by enumerating our classes of permutations either by the inversion number or, using Rawlings' notations [16], by the number of $2\underline{3}1$ patterns: this is the number of pairs (i,k) where $i < k$ such that there exists $j \in [i,k]$ such that $\sigma(k) < \sigma(i) < \sigma(j)$. Note that the number of $2\underline{3}1$ patterns in a permutation σ, denoted below $\underline{\text{inv}}(\sigma)$, is the number of inversions of $\Pi(\sigma)$ (see [21, Lemma 4]). For instance, the permutation σ of Fig.1 has four $2\underline{3}1$ patterns (corresponding to the pairs of letters $(2,1)$, $(3,1)$, $(5,4)$ and $(6,4)$) and $\Pi(\sigma)$ has four inversions (given by the same pairs of letters).

The structure of the paper is the following. In Section 2, we describe our factorization of permutations, first as a factorization of words, and then as a factorization of decreasing binary trees, which are one of the many classical representations of permutations. In Section 3, we describe in terms of trees the sorting procedure Π and the four families of permutations we are interested in. In Section 4, we establish and compare our $4+1$ functional equations. Section 5 is devoted to the solution of (some of) them.

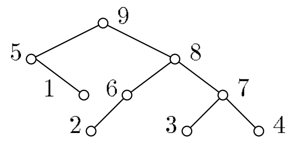
2 Permutations and trees

2.1 Terminology on trees

A *decreasing binary tree* is a binary tree whose nodes are labelled by distinct positive integers in such a way that each node has a larger label than its children. The tree is said to be *normalized* if the number of its nodes coincides with the label of the root. The set of normalized trees having n nodes is denoted $\mathcal{T}_n$.

Reading a decreasing binary tree in *symmetric order* establishes a one-to-one correspondence with words on the alphabet $\{1, 2, \ldots\}$ having all their letters distinct. The symmetric order $S(t)$ of a tree t is defined recursively by reading first the left subtree of t, then its root, and finally its right subtree. In particular, S induces a well-known bijection between normalized trees and permutations. The reverse bijection of S is denoted T (Fig.3).

$\sigma = 519268374 \longrightarrow$



$= T(\sigma)$

Figure 3: The bijection between permutations and normalized trees.

Let t be a decreasing binary tree having n nodes, and let L be the set of its labels. Let f be the unique order preserving bijection from L to $\{1, 2, \ldots, n\}$. *Normalizing* the tree t means replacing the label i by $f(i)$, for all $i \in L$. We define similarly the normalization of words having distinct letters.

We define recursively the *leftmost branch* and the *leftmost path* of a tree. If t^L (resp. t^R) is the left (resp. right) subtree of t, then the leftmost branch of t consists of the root of t and the leftmost branch of t^L. The leftmost path of t consists of the root of t and the leftmost path of t^L *if* t^L *is not empty*; otherwise, it consists of the root of t and the leftmost path of t^R. Hence the leftmost path joins the root to the "leftmost" leaf: for the tree of Fig.3, it consists of the nodes labelled $1, 5, 9$. We define symmetrically the rightmost branch and path.

2.2 A factorization of permutations and trees

We now describe a factorization of permutations and translate it in terms of trees. This factorization is not really new: it is the extension to all permutations of the factorization of two-stack sortable permutations used by Zeilberger in [22]. It requires the introduction of a new statistic on permutations. For $\sigma \in \mathcal{S}_n$, we define $z(\sigma)$ by:

$$z(\sigma) = \max\{\ell : \sigma^{-1}(n) > \sigma^{-1}(n-1) > \cdots > \sigma^{-1}(n-\ell+1)\}.$$

For instance, $z(\sigma) = 3$ for the permutation σ of Fig.3. If σ is the empty permutation, of length 0, we set $z(\sigma) = 0$. For $m, n \geq 0$, we define the sets $\mathcal{S}_{m,n}$ and $\overline{\mathcal{S}}_{m,n}$ by

$$\mathcal{S}_{m,n} = \{\sigma \in \mathcal{S}_{m+n} : z(\sigma) \geq n\} \qquad \text{and} \qquad \overline{\mathcal{S}}_{m,n} = \{\sigma \in \mathcal{S}_{m+n} : z(\sigma) = n\}.$$

Note that $\mathcal{S}_{m,0} = \mathcal{S}_m$ and that for $m \geq 1$, $\mathcal{S}_{m,n}$ is the disjoint union of $\mathcal{S}_{m-1,n+1}$ and $\overline{\mathcal{S}}_{m,n}$.

The principle of the factorization is very simple: it splits a permutation into two factors, a prefix and a suffix. Let $m, n \geq 1$ and take $\sigma \in \overline{\mathcal{S}}_{m,n}$. This means that σ has length $m+n$, that the numbers $m+n, m+n-1, \ldots, m+1$ appear in this order in σ, and that m lies to the left of $m+1$. Let $j \in \{0, 1, \ldots, n-1\}$ be the largest number such that m lies to the left of $m+j+1$. We have

$$\sigma = \ldots(m+n)\ldots(m+n-1)\ldots\ldots(m+j+2)\ldots(m)\ldots(m+j+1)\ldots(m+j)\ldots\ldots(m+1)\ldots$$

Let us write $\sigma = \sigma^{(1)}(m+j+1)\sigma^{(2)}$. The length of $\sigma^{(2)}$ is $i+j$ for some $i \in \{0, 1, \ldots, m-1\}$. Let L be the set of numbers smaller than m occurring in $\sigma^{(2)}$. Then L has cardinality i. Finally, let $\tau^{(1)}$ (resp. $\tau^{(2)}$) be obtained by normalizing $\sigma^{(1)}$ (resp. $\sigma^{(2)}$). Note that $\tau^{(1)} \in \mathcal{S}_{m-i-1,n-j}$ and $\tau^{(2)} \in \mathcal{S}_{i,j}$. Let us denote $\Phi(\sigma) = (i, j, L, \tau^{(1)}, \tau^{(2)})$.

Example. Let $m = 6$ and $n = 3$. For $\sigma = 519268374 \in \mathcal{S}_{6,3}$ we find $j = 1$, $\sigma^{(1)} = 51926$ and $\sigma^{(2)} = 374$. We have $i = 2$ and $L = \{3, 4\}$. Normalizing the permutations gives $\tau^{(1)} = 31524$ and $\tau^{(2)} = 132$, and finally $\Phi(\sigma) = (2, 1, \{3, 4\}, 31524, 132)$.

Proposition 2.1 *For $m, n \geq 1$, the map Φ establishes a one-to-one correspondence between $\overline{\mathcal{S}}_{m,n}$ and the five-tuples $(i, j, L, \tau^{(1)}, \tau^{(2)})$ such that*

$$0 \leq i < m, \quad 0 \leq j < n, \quad L \subset \{1, \ldots, m-1\}, \quad |L| = i, \quad \tau^{(1)} \in \mathcal{S}_{m-i-1,n-j} \quad \text{and} \quad \tau^{(2)} \in \mathcal{S}_{i,j}.$$

Moreover, if $\Phi(\sigma) = (i, j, L, \tau^{(1)}, \tau^{(2)})$, then

$$\underline{\mathrm{inv}}(\sigma) = \underline{\mathrm{inv}}(\tau^{(1)}) + \underline{\mathrm{inv}}(\tau^{(2)}) + \mathrm{inv}(m-1, L) + i$$

where $\mathrm{inv}(m-1, L) = |\{(a, b) : m-1 \geq a > b \geq 1, a \notin L, b \in L\}|$.

Example. For the permutation σ above, we have $\underline{\mathrm{inv}}(\sigma) = 5$, $\underline{\mathrm{inv}}(\tau^{(1)}) = 1$, $\underline{\mathrm{inv}}(\tau^{(2)}) = 0$, $\mathrm{inv}(5, L) = 2$ and we check that $\underline{\mathrm{inv}}(\tau^{(1)}) + \underline{\mathrm{inv}}(\tau^{(2)}) + \mathrm{inv}(m-1, L) + i = 1 + 0 + 2 + 2 = \underline{\mathrm{inv}}(\sigma) = 5$.

Let us now describe the factorization in terms of trees. First of all, we note that the statistic $z(\sigma)$ is easily read on the tree $t = T(\sigma)$: if t has n nodes, $z(\sigma)$ is the largest ℓ such that $n, n-1, \ldots, n-\ell+1$ lie on the rightmost branch of t. When we do not want to make the underlying permutation explicit, we will use the notation $z(t)$ instead of $z(\sigma)$. By analogy with $\mathcal{S}_{m,n}$ and $\overline{\mathcal{S}}_{m,n}$, we define, for $m, n \geq 0$, the sets $\mathcal{T}_{m,n}$ and $\overline{\mathcal{T}}_{m,n}$ by

$$\mathcal{T}_{m,n} = \{t \in \mathcal{T}_{m+n} : z(t) \geq n\} \qquad \text{and} \qquad \overline{\mathcal{T}}_{m,n} = \{t \in \mathcal{T}_{m+n} : z(t) = n\}.$$

Let $m, n \geq 1$ and take $t \in \overline{\mathcal{T}}_{m,n}$. This means that the nodes $m+n, m+n-1, \ldots, m+1$ lie on the rightmost branch of t, and that m is the left child of one of them – say, of $m+j+1$, with $0 \leq j < n$. Let $t^{(2)}$ be the right subtree of the node $m+j+1$. Let $i+j$ be the number of its nodes, and L the set of its labels smaller than m. Then $|L| = i$. Let $t^{(1)}$ be obtained from t by replacing the subtree of root $m+j+1$ by the subtree of root m. Let $u^{(1)}$ (resp. $u^{(2)}$) be obtained by normalizing $t^{(1)}$ (resp. $t^{(2)}$). Define $\Phi(t) = (i, j, L, u^{(1)}, u^{(2)})$. Then Φ establishes a one-to-one correspondence between $\overline{\mathcal{T}}_{m,n}$ and the five-tuples $(i, j, L, u^{(1)}, u^{(2)})$ such that

$$0 \leq i < m, \quad 0 \leq j < n, \quad L \subset \{1, \ldots, m-1\}, \quad |L| = i, \quad u^{(1)} \in \mathcal{T}_{m-i-1,n-j} \quad \text{and} \quad u^{(2)} \in \mathcal{T}_{i,j}.$$

The factorization of trees is schematized in Fig.4.

3 Let us sort trees instead of permutations

3.1 The postorder of a tree

Recall that the *postorder* $P(t)$ of a tree t is recursively defined by reading first the left subtree of t, then its right subtree, and finally its root.

Proposition 3.1 *Let σ be a permutation and $t = T(\sigma)$ the corresponding tree. Then the permutation $\Pi(\sigma)$ obtained by sorting σ through a stack is exactly the word $P(t)$ obtained by reading t in postorder. In other words, $\Pi = P \circ T$.*

This extremely simple proposition enables us to reformulate in terms of trees all questions relative to the sorting procedure. In particular, it gives what is probably the simplest way of counting one-stack sortable permutations.

Corollary 3.2 1) *A permutation $\sigma \in \mathcal{S}_n$ is one-stack sortable if and only if the associated tree $T(\sigma)$ has postorder $12\ldots n$. Consequently, the number of one-stack sortable permutations is the Catalan number $C_n = \binom{2n}{n}/(n+1)$.*
2) *A permutation $\sigma \in \mathcal{S}_n$ is two-stack sortable if and only if the postorder of $T(\sigma)$ avoids the pattern* 231.

We shall say that a tree t is one-stack sortable (resp. two-stack sortable, sorted) if the permutation $\sigma = S(t)$ is one-stack sortable (resp. two-stack sortable, sorted).

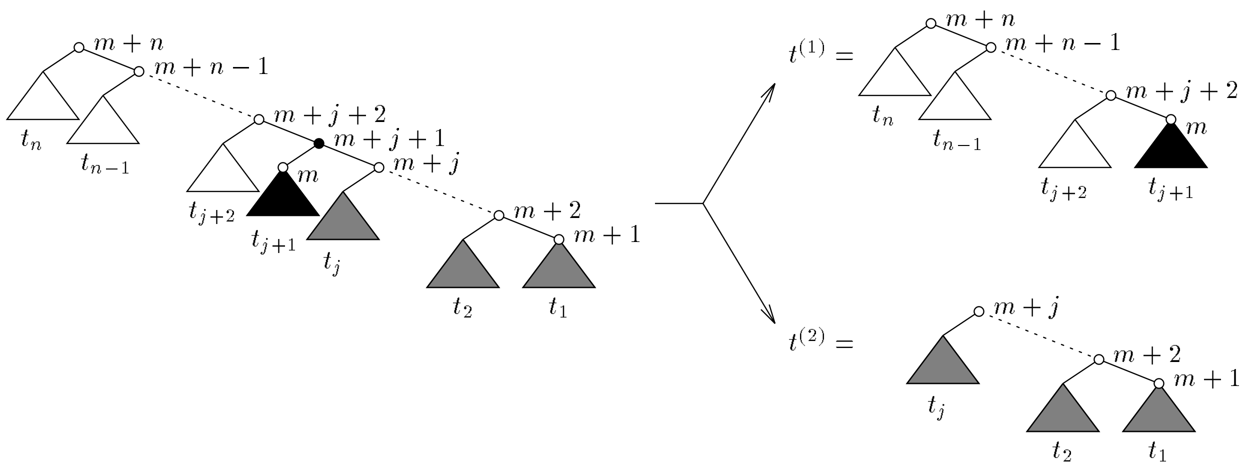
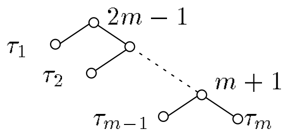


Figure 4: The factorization of trees (the set of labels of the grey trees is L).

3.2 Sorted permutations

We now come to the main motivation of this paper: sorted permutations. According to Proposition 3.1, the sorted permutations are the postorders of binary decreasing trees. An immediate consequence is that sorted permutations cannot be described by forbidding a set of patterns.

Corollary 3.3 *Any pattern occurs as a factor in some sorted permutation. More precisely, if $\tau = \tau_1 \dots \tau_m \in \mathcal{S}_m$, then the permutation $\tau_1 \dots \tau_m (m+1)(m+2) \dots (2m-1)$ is sorted (see the figure below).*



Clearly, different trees might have the same postorder (Fig.5). In order to characterize sorted permutations, we are going to describe a *canonical representative* of the pre-images of a sorted permutation.

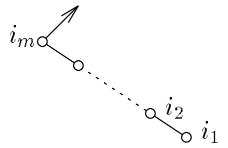
Definition 3.4 *A normalized tree is said to be* canonical *if it satisfies the following property. Each node that has a left child x has a non empty right subtree t^R. Moreover, the first node of t^R (for the symmetric order) has a label y smaller than x. A permutation σ is* canonical *if the tree $T(\sigma)$ is canonical.*

Examples. The first tree of Fig.5 is not canonical because the left child of the node 5 has label $x = 1$, whereas the first node of its right subtree has label $y = 3 > 1$. The second tree of the figure is canonical.

Figure 5: Two trees having postorder 13245.

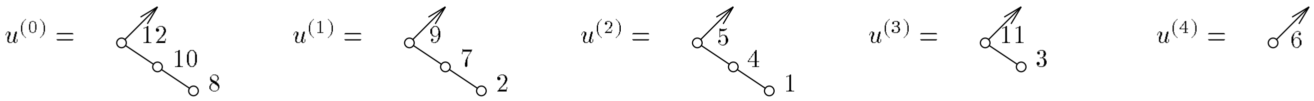
Proposition 3.5 *Every sorted permutation τ is the image by Π of a unique canonical permutation σ. Moreover, σ is the pre-image of τ having the largest number of inversions.*

An algorithm that decides whether a permutation is sorted
Assume that τ has k descents and write $\tau = \tau^{(k)}\tau^{(k-1)}\dots\tau^{(0)}$ where the $\tau^{(j)}$ are the maximal increasing factors of τ. For $0 \le j \le k$, if $\tau^{(j)} = i_1 \dots i_m$, with $i_1 < i_2 < \dots < i_m$, let $u^{(j)}$ be the (linear) tree $T(i_m i_{m-1} \dots i_1)$:

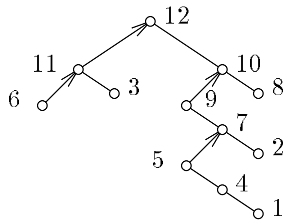
$u^{(j)} =$ 

Observe the arrow attached to the root of $u^{(j)}$. We now build canonical trees $t^{(0)}, t^{(1)}, \dots$ as follows.
Step 0. Start from the tree $t^{(0)} = u^{(0)}$.
Step i, $i = 1, \dots, k$. If all nodes of the leftmost path of the tree $t^{(i-1)}$ that have no left child are smaller than the root of $u^{(i)}$, then τ is not sorted and we stop. Otherwise, let $t^{(i)}$ be obtained by attaching $u^{(i)}$ to the smallest node in the leftmost path of $t^{(i-1)}$ that is larger than the root of $u^{(i)}$ and has no left child.
The tree $t^{(k)}$ (when we can construct it) is the canonical pre-image of τ.

Example. Let $\tau = 6.3.11.1.4.5.2.7.9.8.10.12 \in \mathcal{S}_{12}$. This permutation has $k = 4$ descents, and we obtain the following elementary trees:



We can attach them to each other, step by step; we finally obtain the canonical pre-image of τ:



Hence $\tau = \Pi(6.11.3.12.9.5.4.1.7.2.10.8)$ is sorted.

Proposition 3.6 *The number of pre-images of a sorted permutation only depends on the* shape *of its canonical tree, i.e., of the binary tree obtained by removing the labels.*

Remarks. 1) West proved [20] that the permutations $\mu_{n,k} = 23\dots k1(k+1)(k+2)\dots n$ and $\nu_{n,k} = 12\dots(k+1)k(k-1)(k+2)\dots n$ have the same number of pre-images. This result is a consequence of the above proposition, as the corresponding canonical trees have the same shape.
2) It would be interesting to determine, other than recursively, the number of pre-images of a sorted permutation from the shape of its canonical tree.

3.3 Recursive characterizations

We have obtained characterizations of one-stack sortable, two-stack sortable and sorted permutations in terms of trees. We now combine them with the factorization of Section 2 to obtain recursive characterizations, from which we shall obtain, in the next section, our functional equations.

Proposition 3.7 *Let σ be a permutation of $\overline{\mathcal{S}}_{m,n}$, where $m, n \ge 1$, and let $t = T(\sigma)$ be the corresponding normalized tree. Let $(i, j, L, u^{(1)}, u^{(2)})$ be the five-tuple obtained by factoring t.*
1) *σ is one-stack sortable if and only if $L = \emptyset$ and $u^{(1)}$ and $u^{(2)}$ are one-stack sortable.*
2) *σ is two-stack sortable if and only if $L = \{m-i, m-i+1, \dots, m-1\}$ and $u^{(1)}$ and $u^{(2)}$ are two-stack sortable.*
3) *σ is canonical if and only if $u^{(1)}$ and $u^{(2)}$ are canonical and*
- *either $j = 0$ and $m+1$ has a non-empty right subtree,*
- *or $j > 0$ and $m+j$ has a non-empty left subtree.*

4 Functional equations

In this section, we establish and compare five functional equations that define implicitely the generating functions for the following five families of permutations: general permutations, one-stack sortable permutations, two-stack sortable permutations, sorted permutations and sorted and sortable permutations. These functional equations are derived using the factorization of permutations described in Section 2 and Proposition 3.7.

Notations. We shall use the following standard definitions and notations. For $n \geq 1$, the q-analogue of n is

$$[n] = 1 + q + \cdots + q^{n-1} = \frac{1-q^n}{1-q}.$$

The q-analogue of $n!$ is $[n]! = [1][2]\dots[n]$. By convention, $[0]! = 1$.
Let $\mathcal{C}$ be a set of permutations. By the *ordinary* (resp. *exponential*) *generating function of* $\mathcal{C}$ we mean the series

$$C(x,y) = \sum_{m,n\geq 0} c_{m,n}\ x^m y^n \qquad \left(\text{resp.}\quad C(x,y) = \sum_{m,n\geq 0} c_{m,n}\ \frac{x^m}{m!}\ y^n\right),$$

where $c_{m,n} = |\mathcal{C} \cap \mathcal{S}_{m,n}|$ is the number of permutations σ of $\mathcal{C}$ of length $m+n$ such that $z(\sigma) \geq n$. Similarly, let $\mathtt{inv}$ denote the statistics inv or $\underline{\text{inv}}$. The *ordinary* (resp. *eulerian*) $\mathtt{inv}$-*generating function of* $\mathcal{C}$ is

$$\bar{C}(x,y;q) = \sum_{m,n\geq 0} \bar{c}_{m,n}\ x^m y^n \qquad \left(\text{resp.}\quad \bar{C}(x,y) = \sum_{m,n\geq 0} \bar{c}_{m,n}\ \frac{x^m}{[m]!}\ y^n\right),$$

where $\bar{c}_{m,n} = \sum_{\sigma\in\mathcal{C}\cap\mathcal{S}_{m,n}} q^{\mathtt{inv}(\sigma)}$.

4.1 General permutations

Let $a_{m,n}$ be the number of permutations $\sigma \in \mathcal{S}_{m+n}$ such that $z(\sigma) \geq n$, i.e., $a_{m,n} = |\mathcal{S}_{m,n}|$. Clearly, $a_{m,n} = (m+n)!/n!$ (shuffle the word $(m+n)(m+n-1)\dots(m+1)$ with any permutation of $\mathcal{S}_m$). Consequently, the exponential generating function for general permutations is $1/(1-x-y)$. Although this result is very simple, it is interesting to see the functional equation satisfied by $A(x,y)$ provided by the factorization of permutations described in Section 2. The following proposition actually gives the functional equation satisfied by the $\underline{\text{inv}}$-generating function.

Proposition 4.1 *The Eulerian* $\underline{\text{inv}}$*-generating function* $\bar{A}(x,y)$ *for general permutations is completely characterized by the initial condition* $\bar{A}(0,y) = 1/(1-y)$ *and the equation*

$$\frac{\bar{A}(x,y) - \bar{A}(xq,y)}{x(1-q)} = \left[1 + y\bar{A}(xq,y)\right]\ \frac{\bar{A}(x,y) - \bar{A}(x,0)}{y}. \tag{2}$$

In the limit $q \to 1$*, we find, for the series* $A(x,y)$*, the initial condition* $A(0,y) = 1/(1-y)$ *and the equation*

$$\frac{\partial A}{\partial x}(x,y) = [1 + yA(x,y)]\ \frac{A(x,y) - A(x,0)}{y}. \tag{3}$$

Remark. We obtain a different information on the series $\bar{A}(x,y)$ if we use the standard factorization of trees into their left and right subtrees. We find:

$$\frac{\bar{A}(x,0) - \bar{A}(xq,0)}{x(1-q)} = \bar{A}(x,0)^2 \qquad \text{and} \qquad \bar{A}(x,y) = \frac{\bar{A}(x,0)}{1 - y\bar{A}(x,0)}. \tag{4}$$

One checks easily that Eq. (4) implies Eq. (2). But conversely, deriving (4) from (2) does not seem so simple. Note that Rawlings [16, 17] has studied a close relative to the statistics $\underline{\text{inv}}$, and essentially obtained Eq. (4).

4.2 One-stack sortable permutations

Proposition 4.2 *The ordinary generating function* $B(x,y)$ *for one-stack sortable permutations is completely characterized by the equation*

$$B(x,y) = \frac{1}{1-y} + \frac{x}{1-y}\ \frac{B(x,y) - B(x,0)}{y}.$$

Enumerating one-stack sortable permutations according to the statistic $\underline{\text{inv}}$ is irrelevant, as these permutations avoid the pattern 231. For their enumeration according to the number of inversions, see [1].

4.3 Two-stack sortable permutations

Proposition 4.3 *The ordinary* inv*-generating function $C(x,y)$ for two-stack sortable permutations is completely characterized by the equation*

$$\bar{C}(x,y)=\frac{1}{1-y}+x\left[1+y\bar{C}(xq,y)\right]\ \frac{\bar{C}(x,y)-\bar{C}(x,0)}{y}.$$

When $q=1$, we recover (a version of) Zeilberger's equation [22].

4.4 Sorted permutations

For sorted permutations, we obtain a functional equation that looks very much like the one we got for general permutations (Proposition 4.1), but is actually much more mysterious.

Proposition 4.4 *The Eulerian* inv*-generating function $\bar{D}(x,y)$ for sorted permutations is completely characterized by the initial condition $\bar{D}(0,y)=1/(1-y)$ and the equation*

$$\frac{\bar{D}(x,y)-\bar{D}(xq,y)}{x(1-q)}=(1-y)\left[1+y\bar{D}(xq,y)\right]\ \frac{\bar{D}(x,y)-\bar{D}(x,0)}{y}. \tag{5}$$

In the limit $q\to 1$, we obtain for the exponential generating function $D(x,y)$ the initial condition $D(0,y)=1/(1-y)$ and the equation

$$\frac{\partial D}{\partial x}(x,y)=(1-y)\left[1+yD(x,y)\right]\ \frac{D(x,y)-D(x,0)}{y}.$$

4.5 Sorted and sortable permutations

The functional equation for this class of permutations looks very much like the equation obtained for two-stack sortable permutations (Proposition 4.3).

Proposition 4.5 *The ordinary* inv*-generating function $\bar{E}(x,y)$ for sorted and sortable permutations is completely characterized by the equation*

$$\bar{E}(x,y)=\frac{1}{1-y}+x(1-y)\left[1+y\bar{E}(xq,y)\right]\ \frac{\bar{E}(x,y)-\bar{E}(x,0)}{y}.$$

5 Solving the functional equations

The five functional equations we have obtained are of three different sorts. The simplest one is relative to one-stack sortable permutations (Proposition 4.2). It is linear in $B(x,y)$. Two others (Propositions 4.3 and 4.5) are q-quadratic in the unknown series. They are relative to two-stack sortable permutations and sorted and sortable permutations respectively. The last two equations (Propositions 4.1 and 4.4) involve a q-derivative with respect to x.

Notations. Given a ring $\mathbb{L}$ and n indeterminates $x_1,\dots,x_n$, we denote by

- $\mathbb{L}[x_1,\dots,x_n]$ the ring of polynomials in $x_1,\dots,x_n$ with coefficients in $\mathbb{L}$,
- $\mathbb{L}[[x_1,\dots,x_n]]$ the ring of formal power series in $x_1,\dots,x_n$ with coefficients in $\mathbb{L}$,

and if $\mathbb{L}$ is a field, we denote by

- $\mathbb{L}(x_1,\dots,x_n)$ the field of rational functions in $x_1,\dots,x_n$ with coefficients in $\mathbb{L}$.

5.1 Linear equations

Proposition 5.1 ([15]) *The ordinary length generating function $B(x,0)$ for one-stack sortable permutations is:*

$$B(x,0)=\frac{1-\sqrt{1-4x}}{2x}=\sum_{n\geq 0}\frac{1}{n+1}\binom{2n}{n}x^n.$$

Proof. The equation of Proposition 4.2 can be rewritten as

$$[y(1-y)-x]B(x,y)=y-xB(x,0).$$

Let $Y=(1-\sqrt{1-4x})/2=x+O(x^2)$. Then $Y(1-Y)=x$. Substituting Y for y in the above equation shows that $B(x,0)=Y/x$.

5.2 Quadratic equations

The equations of Propositions 4.3 (two-stack sortable permutations) and 4.5 (sorted and sortable permutations) can be solved when $q = 1$ via the so-called *quadratic method*, which is due to Brown [5, 13, section 2.9.1].

Proposition 5.2 ([10, 14, 22]) *The ordinary length generating function $C(x, 0)$ for two-stack sortable permutations is cubic on the field $\mathbb{R}(x)$:*

$$x^2 C(x,0)^3 + x(2+3x)C(x,0)^2 + (1-14x+3x^2)C(x,0) + x^2 + 11x - 1 = 0.$$

This implies that

$$C(x,0) = 1 + 2\sum_{n\geq 1} \frac{(3n)!}{(2n+1)!(n+1)!} x^n.$$

Proposition 5.3 *The ordinary length generating function $E(x, 0)$ for sorted and sortable permutations is algebraic of degree 4 on the field $\mathbb{R}(x)$:*

$$x^3 E(x,0)^4 + x^2(3+4x)E(x,0)^3 + x(3-29x+6x^2)E(x,0)^2 + (1-7x+29x^2+4x^3)E(x,0) - (1-x)^3 = 0.$$

So far, we have found no q-analogue of the quadratic method that would enable us to solve the equations of Propositions 4.3 and 4.5 for $q \neq 1$.

5.3 Differential equations

We finally come to the functional-differential equation that defines the generating function for sorted permutations (Proposition 4.4). Even when $q = 1$, it is (and will remain...) quite intriguing. It is very similar to the equation obtained for general permutations (see (3)), whose solution is extremely simple: $A(x, y) = 1/(1 - x - y)$. Both equations are of the following form:

$$\frac{\partial F}{\partial x}(x,y) = c(y)\,[1 + yF(x,y)]\ \frac{F(x,y) - F(x,0)}{y}, \tag{6}$$

where $c(y) = 1$ for general permutations and $c(y) = 1 - y$ for sorted permutations. Eq. (6), together with the initial condition $F(0, y) = 1/(1-y)$, defines $F(x, y)$ as a formal power series in x with rational coefficients in y. Expressing $F(x, y)$ in terms of $F(x, 0)$ is not difficult: Eq. (6) is essentially a Bernoulli equation and can be linearized, as stated in the following lemma.

Lemma 5.4 *Let $F(x, y) \in \mathbb{R}(y)[[x]]$ be defined by the initial condition $F(0, y) = 1/(1-y)$ and Eq. (6), with $c(y) = 1$ or $c(y) = 1 - y$. Let $G(x, y)$ be the following series of $\mathbb{R}(y)[[x]]$:*

$$G(x,y) = \frac{1}{c(y)\,[1 + yF(x,y)]}. \tag{7}$$

Then $G(0, y) = (1 - y)/c(y)$ and

$$y\frac{\partial G}{\partial x}(x,y) - c(y)\,[1 + yF(x,0)]\,G(x,y) + 1 = 0. \tag{8}$$

Most importantly, $G(x, y)$ has polynomial coefficients *in y, i.e., $G(x, y) \in \mathbb{R}[y][[x]]$.*

The above lemma justifies our interest for the following proposition.

Proposition 5.5 *Let $h(x, y) \in \mathbb{R}[y][[x]]$ be a formal series in x with polynomial coefficients in y. Let $G(x, y)$ be the series of $\mathbb{R}(y)[[x]]$ defined by an initial condition $G(0, y) \in \mathbb{R}(y)$ and the differential equation:*

$$y\frac{\partial G}{\partial x}(x,y) - [1 + yh(x,y)]\,G(x,y) + 1 = 0. \tag{9}$$

Let

$$H(x,y) = \exp\left[-\int_0^x h(u,y)du\right] = \sum_{i\geq 0} H_i(y)\ \frac{x^i}{i!}.$$

Then the coefficients of $G(x,y)$ are polynomials in y if and only if $G(0,y) \in \mathbb{R}[y]$ and

$$\sum_{i\geq 0} H_i(y)\ y^i = G(0,y).$$

In other words, the Laplace transform of $H(x,y)$ with respect to x is exactly $G(0,y)$ when evaluated at $x=y$:

$$\frac{1}{y}\int_0^\infty e^{-u/y} H(u,y)du = G(0,y).$$

Let us now apply Proposition 5.5 to the equations of Lemma 5.4.

• **General permutations.** When $c(y)=1$, the series $F(x,y)$ is the exponential generating function $A(x,y)$ for general permutations. The series $G(x,y)=1/(1+yA(x,y))$ satisfies (8) with $h(x,y)=A(x,0)$. Moreover, $G(0,y)=1-y$. With the notations of Proposition 5.5, we have

$$H(x,y) = \exp\left[-\int_0^x A(u,0)du\right] = \sum_{i\geq 0} H_i\ \frac{x^i}{i!} = H(x). \tag{10}$$

Proposition 5.5 gives the Laplace transform of $H(x)$:

$$\sum_{i\geq 0} H_i\ y^i = G(0,y) = 1-y.$$

Consequently, $H(x)=1-x$. Using (10), we obtain $A(x,0)=1/(1-x)$. This is exactly (fortunately!) the exponential generating function for general permutations. Then, we integrate (9) and find $G(x,y)=(1-x-y)/(1-x)$. Finally, using (7), we find

$$A(x,y) = \frac{1}{1-x-y} = \sum_{m,n\geq 0} \frac{(m+n)!}{n!}\frac{x^m}{m!}y^n.$$

Hence, our – admittedly complicated – method is at least able to recover the expected result: the number of permutations σ of length $m+n$ such that $z(\sigma)\geq n$ is $(m+n)!/n!$.

• **Sorted permutations.** The success of our method on a problem we knew how to solve encourages us to apply the same method to the equation of Lemma 5.4 in the case $c(y)=1-y$. The series $F(x,y)$ is the exponential generating function $D(x,y)$ for sorted permutations. The series $G(x,y)=1/\left[(1-y)(1+yD(x,y))\right]$ satisfies (8) with $h(x,y)=(1-y)D(x,0)-1$. Moreover, $G(0,y)=1$. With the notations of Proposition 5.5, we have:

$$H(x,y) = \exp(x+(y-1)\mathcal{D}(x))$$

where

$$\mathcal{D}(x) = \int_0^x D(u,0)du.$$

Proposition 5.5 gives the following result.

Proposition 5.6 *Let*

$$\mathcal{D}(x) = \sum_{m\geq 0} d_{m,0}\frac{x^{m+1}}{(m+1)!}$$

where $d_{m,0}$ is the number of sorted permutations of length m. Then the series $\mathcal{D}(x)$ is completely characterized by the following equation:

$$\frac{1-y}{y}\int_0^\infty e^{-u(1-y)/y}\exp\left[(y-1)\mathcal{D}(u)\right]du = 1-y.$$

In other words, let $K(x,y)=\exp\left[(y-1)\mathcal{D}(x)\right]=\sum_{i\geq 0}K_i(y)\ x^i/i!$, and let $\hat{K}(x,y)=\sum_{i\geq 0}K_i(y)\ x^i$ be its Laplace transform with respect to x. Then

$$\hat{K}\left(\frac{y}{1-y},y\right) = 1-y, \quad \textit{or, equivalently} \quad \hat{K}\left(u,\frac{u}{1+u}\right) = \frac{1}{1+u}.$$

Remark. The approach we have presented for "solving" our functional-differential equations admits a q-analogue that enables us to "solve" Eqs. (2) and (5). It will be exposed in the complete version of this paper.

References

[1] E. Barcucci, A. Del Lungo, S. Lanini, M. Macrì and R. Pinzani, The inversion number of some permutations with forbidden subsequences, *Proceedings of SOCA'96*, Tianjin (1996) 21–32.

[2] M. Bousquet-Mélou, A method for the enumeration of various classes of column-convex polygons, *Discrete Math.* **154** (1996) 1–25.

[3] M. Bousquet-Mélou, Multi-statistic enumeration of two-stack sortable permutations, to appear in *Electronic J. Combin.*

[4] W. G. Brown, Enumeration of non-separable planar maps, *Canad. J. Math.* **15** (1963) 526–545.

[5] W. G. Brown, On the existence of square roots in certain rings of power series, *Math. Annalen* **158** (1965) 82-89.

[6] W. G. Brown, Enumeration of quadrangular dissections of the disk, *Canad. J. Math.* **17** (1965) 302–317.

[7] W. G. Brown and W. T. Tutte, On the enumeration of rooted non-seperable planar maps, *Canad. J. Math.* **16** (1964) 572–577.

[8] R. Cori, B. Jacquard and G. Schaeffer, Description trees for some families of planar maps, Proceedings of the 9th Conference "Formal Power Series and Algebraic Combinatorics", Vienna, 1997, pp. 196–208.

[9] J. Denef and L. Lipshitz, Power series solutions of algebraic differential equations, *Math. Ann.* **267** (1984) 312-238.

[10] S. Dulucq, S. Gire and O. Guibert, A combinatorial proof of J. West's conjecture. To appear in *Discrete Math.*

[11] S. Dulucq, S. Gire and J. West, Permutations with forbidden subsequences and non-separable planar maps, *Discrete Math.* **153** (1996) 85–103.

[12] S. Feretić and D. Svrtan, On the number of column-convex polyominoes with given perimeter and number of columns, in *Proceedings of the* 5^{th} *conference Formal Power Series and Algebraic Combinatorics*, A. Barlotti, M. Delest, and R. Pinzani eds., pages 201–214, Florence, june 1993.

[13] I. P. Goulden and D. M. Jackson, *Combinatorial enumeration,* John Wiley and Sons, New-York, 1983.

[14] I. P. Goulden and J. West, Raney paths and a combinatorial relationship between rooted non-separable planar maps and two-stack-sortable permutations, *J. Combin. Theory Ser. A* **75** (1996) 220–242.

[15] D. E. Knuth, *The art of computer programming* Vol.1, *Fundamental algorithms*, Addison-Wesley, Reading, Massachusetts, 1973.

[16] D. Rawlings, The ABC's of classical enumeration, *Ann. Sci. Math. Québec* **10** No.2 (1986) 207–235.

[17] D. Rawlings, The Euler-Catalan identity, *Europ. J. Combinatorics* **9** (1988) 53–60.

[18] R. P. Stanley, Differentiably finite power series, *Europ. J. Combinatorics*, **1** (1980) 175–188.

[19] W. T. Tutte, On the enumeration of planar maps, *Bull. Amer. Math. Soc.* **74** (1968) 64–74.

[20] J. West, Permutations with restricted subsequences and stack-sortable permutations, Ph.D. thesis, MIT, 1990.

[21] J. West, Sorting twice through a stack, *Theoret. Comput. Sci.* **117** (1993) 303–313.

[22] D. Zeilberger, A proof of Julian West's conjecture that the number of two-stack-sortable permutations of length n is $2(3n)!/((n+1)!(2n+1)!)$, *Discrete Math.* **102** (1992) 85–93.