A Generating Tree for Permutations Avoiding the Pattern $122^{+}3$

E. Duchi[1], V. Guerrini[2], and S. Rinaldi[2]

[1] IRIF, Université Paris Diderot, France,
Enrica.Duchi@irit.univ-paris-diderot.fr,
[2] DIISM, Università di Siena, Italy,
guerrini55@student.unisi.it, rinaldi@unisi.it

Abstract. In this paper we study the family of permutations avoiding the pattern $122^{+}3$ (trivially equivalent to those avoiding $1\,\underline{23}\,4$), which extend the popular 123-avoiding permutations. In particular we provide an algorithmic description of a generating tree for these permutations, that is a way to build every object of a given size $n+1$ in a unique way by performing local modifications on an object of size n. Our algorithm leads to a direct bijection between $1\,\underline{23}\,4$-avoiding permutations and valley-marked Dyck paths. It extends a known bijection between 123-avoiding permutations and Dyck paths, and makes explicit the connection between these objects that was earlier obtained by Callan through a series of non-trivial bijective steps. In particular our construction is simple enough to allow for efficient exhaustive generation.

1 The notion of pattern avoidance and its mathematical framework

Patterns in permutations have been occasionally studied for over a century, but in the last two decades this area has grown, with several published papers. The study of permutations which are constrained by not having one or more subsequences ordered in various prescribed ways has been motivated both by its combinatorial difficulty and by its appearance in some data structuring problems in Computer Science.

Indeed, the study of permutation patterns started with Knuth's consideration of stack-sorting [21]. Knuth showed that a permutation π can be sorted by a stack if and only if π avoids 231, and that stack-sortable permutations are enumerated by the Catalan numbers. Knuth also raised questions about sorting with deques. Later, Tarjan investigated sorting by networks of stacks, while Pratt showed that the permutation π can be sorted by a deque if and only if for all k, π avoids

$$\sigma = 5\,2\,7\,4\,\ldots\,4k+1\,4k-2\,3\,4k\,1 \qquad \text{and} \qquad \delta = 5\,2\,7\,4\,\ldots\,4k+3\,4k\,1\,4k+2\,3\,,$$

and every permutation that can be obtained from either of these by interchanging the last two elements or the 1 and the 2 [23]. This set of permutations is infinite, and it is one of the first examples of an infinite antichain of permutations. So, it is not immediately clear how long it takes to decide if a permutation can be sorted by a deque. Rosenstiehl and Tarjan (1984) later presented a linear (in the length of π) time algorithm to establish if π can be sorted by a deque [24].

It is well known that the number of permutations of length n avoiding any one classical pattern of length 3 is the nth Catalan number, which counts a large amount of different combinatorial objects [27]. There are many other results in this direction, relating pattern avoiding permutations to various other combinatorial structures, either via bijections, or by analytic approaches. It now seems clear that this field of research will continue growing for a long time to come, due to the several problems that are related to other branches of combinatorics, other fields of mathematics, and to other disciplines such as computer science, physics, computational biology and theoretical physics.

Recently, the notion of generalized pattern in permutations has been introduced and considered. Whereas an occurrence of a classical pattern p in a permutation π is simply a subsequence of π whose letters are in the same relative order (of size) as those in p, in an occurrence of a generalized pattern, some letters of that subsequence may be required to be adjacent in the permutation. For example, the classical pattern 1234 simply corresponds to an increasing subsequence of length four, whereas an occurrence of the generalized pattern $1\,\underline{23}\,4$ would require the middle two letters of that sequence to be adjacent in π. Thus, the permutation 23145 contains 1234 but not $1\,\underline{23}\,4$. We point out that our notation differs from the usual one (which uses dashes) since we use the symbol $\underline{\ \ }$ to indicate the elements of the pattern that are required to be adjacent in an occurrence.

Generalized patterns provide a significant addition to classical patterns and their connections to other combinatorial structures. In fact the non-classical generalized patterns are likely to provide richer connections to other combinatorial structures than the classical ones do. Other than combinatorics, generalized patterns find applications in other scientific areas such as, for instance, in the genome rearrangement problem, which is one of the major trends in bioinformatics and biomathematics [3,30], and discrete tomography [13,14,17].

It is also worth mentioning that the concept of *substructure* (or *pattern*) within a *combinatorial structure* is now an essential notion in combinatorics, whose study has had many developments in various branches of discrete mathematics. Nowadays, the research on permutation patterns is being developed in several directions. One of them is to define and study analogues of the concept of pattern in permutations in other combinatorial objects such as set partitions [18,19,26], words [6,9], trees [12,25], matrices and polyominoes [16].

In this paper we will study permutations avoiding the generalized pattern $122^{+}3$, which is equivalent to the pattern $1\,\underline{23}\,4$. These patterns are of great importance since they constutute a neat but non trivial generalization of the popular pattern 123, and they have been considered in some previous papers [10,15] leaving several open problems.

We will provide an algorithmic description of a generating tree for these permutations, that is a way to build every object of a given size $n+1$ in a unique way by performing local modifications on an object of size n. Our algorithm leads to a direct bijection between $1\,\underline{23}\,4$-avoiding permutations and valley-marked Dyck paths. It extends a known bijection between 123-avoiding permutations and Dyck paths, and makes explicit the connection between these objects that was earlier obtained by Callan [10] through a series of non-trivial bijective steps. In particular our construction is simple enough to allow for efficient exhaustive generation [3,30].

2 Getting started

We start recalling some basic facts on pattern avoiding permutations. For more details we address the reader to [7]. A permutation $\sigma = \sigma_1\sigma_2\ldots\sigma_n$ contains $\tau = \tau_1\tau_2\ldots\tau_k$ if there exists $i_1 < i_2 < \ldots < i_k$ such that $\sigma_{i_a} < \sigma_{i_b}$ if and only if $\tau_a < \tau_b$. Otherwise, σ avoids τ, and we denote $AV_n(\tau)$ the set of permutations σ of $\{1,\ldots,n\}$ that avoids a fixed permutation τ.

Among the several existing notions which extend the classical pattern avoidance, here we are interested in the so-called *bivincular patterns* [1,11], and more precisely we consider the patterns $122^{+}3$ and $1\,\underline{23}\,4$:

i) a permutation π contains the pattern $122^{+}3$ if there are four indices $i < j < k < t$ such that $\pi_i < \pi_j < \pi_k = \pi_j + 1 < \pi_t$;
ii) a permutation π contains the pattern $1\,\underline{23}\,4$ if there are three indices $i < j < k$ such that $\pi_i < \pi_j < \pi_{j+1} < \pi_k$.

For instance, the permutation 4 1 7 3 5 2 6 contains the pattern $1\,\underline{23}\,4$ because of the entries $\pi_2 = 1$, $\pi_4 = 3$, $\pi_5 = 5$ $\pi_7 = 6$, whereas π does not contain the pattern $122^{+}3$. We immediately point out that $122^{+}3$-avoiding permutations are in one to one correspondances with $1\,\underline{23}\,4$-avoiding ones via group inversion, i.e. $\pi \in AV(122^{+}3)$ if and only if $\pi^{-1} \in AV(1\,\underline{23}\,4)$, so we can study one or the other set equivalently.

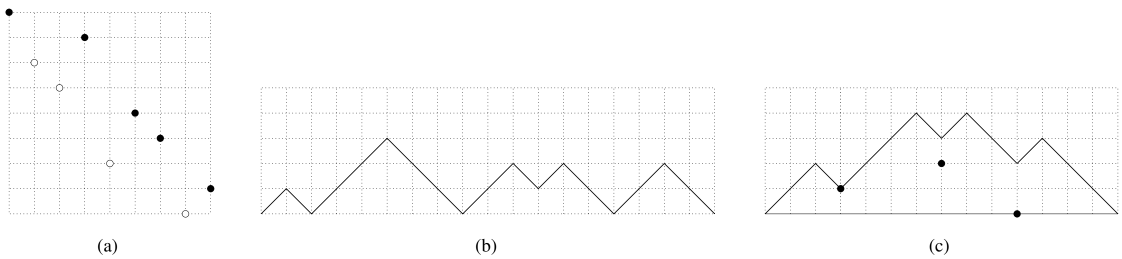

Fig. 1. (a) The 123-avoiding permutation 9 7 6 8 3 5 4 1 2; (b) The corresponding Dyck path; (c) A Dyck path with marked valleys.

Permutations avoiding 123 and permutations avoiding 122^+3. Our interest in permutations avoiding 122^+3 is motivated by the fact that they form a natural extension of the class of 123-avoiding permutations, which on their turn are perhaps the simplest and widely studied class of permutations counted by the famous *Catalan numbers*, enumerating more than 200 combinatorial structures, listed by Stanley in [28]. We recall that 123-avoiding permutations have a simple characterization in terms of left-to-right minima and right-to-left maxima, where as usual, π_i is a left-to-right minimum (resp. right-to-left maxima) of π if $\pi_i = \min\{\pi_j : j \leq i\}$ (resp. $\pi_i = \max\{\pi_j : j \geq i\}$), and $AV_n(123)$ (resp. $AV_n(122^+3)$) denotes the set of 123- (resp. 122^+3-) avoiding permutations of length n.

Proposition 1. *If a permutation π of length n belongs to $AV_n(123)$ then, for every index $1 \leq i < n$ we have that:*
if π_i is on the left of π_i+1, then π_i is a left-to-right minimum and π_i+1 is a right-to-left maximum.

Observe that the previous characterization can be extended to $AV(122^+3)$ as follows.

Proposition 2. *A permutation π of length n belongs to $AV_n(122^+3)$ if and only if for every index $1 \leq i < n$ we have that:*
if π_i is on the left of π_i+1, then π_i is a left-to-right minimum or π_i+1 is a right-to-left maximum.

Krattenthaler's bijection. Another popular structure counted by Catalan numbers is that of *Dyck paths*, i.e. lattice paths in the first quadrant which begin at the origin, end on the x-axis, and consist of up steps $U = (1, 1)$, down steps $D = (1, -1)$ (see Figure 1 (b)).

In [20] Krattenthaler shows the following bijection between 123-avoiding permutations and Dyck paths. Let π be a 123-avoiding permutation, decomposed as $\pi = w_r m_r w_{r-1} m_{r-1} \dots w_1 m_1$, where $m_1, \dots, m_s$ are the right-to-left maxima of π from right to left. By Proposition 1, for all i the entries in w_i are left-to-right minima, so they are in decreasing order and smaller than all the entries of w_{i+1}. Krattenthaler's bijection consists in reading π from right to left and building the corresponding Dyck path form right to left as follows: any right-to-left maximum m_i is translated into $m_i - m_{i-1}$ down steps (with $m_0 = 0$), whereas any factor w_i is translated into $|w_i| + 1$ up steps. Figure 1 (a) shows 123-avoiding permutation 9 7 6 8 3 5 4 1 2 with its right-to-left maxima depicted in black, which is mapped in the Dyck path in Figure 1 (b).

The sequence A113227 and valley-marked Dyck paths. As already mentioned, the sequence counting $1\underline{23}4$ avoiding permutations is not new in combinatorics: in [15] Elizalde obtained asymptotic bounds for these numbers. Moreover in [10] Callan proved that the family of $AV(1\underline{23}4)$-avoiding permutations is counted by the sequence A113227 in [22], whose first terms are:

$$1, 2, 6, 23, 105, 549, 3207, 20577, 143239, 1071704, 8555388, 72442465, \dots$$

More precisely, according to [10] the nth term p_n of the sequence above can be expressed as $p_n = \sum_{j=0}^{n} p_{n,j}$, where the terms $p_{n,j}$ satisfy the recurrence relation

$$\begin{cases} p_{0,0} = 1 \\ p_{n,0} = 0 \text{ if } n \geq 1, \\ p_{n,j} = p_{n-1,j-1} + j \sum_{i=j}^{n-1} p_{n-1,i} \ \text{ for } n \geq j \geq 1. \end{cases} \tag{1}$$

On the one hand Callan points out that there are some "simple" families of objects counted by sequence A113227, for which it is easy to prove Recurrence (1): in particular Callan considers increasing ordered trees with increasing leaves and valley-marked Dyck paths [10]. Let us briefly illustrate Callan's simple proof on the latter structures: *valley-marked Dyck paths* are Dyck paths where every valley (i.e. factor DU) at level h is marked at a given level between 0 to h (see Figure 1 (c)). One can easily prove that the number of valley-marked Dyck paths (briefly, VM-Dyck paths) of semi-length n is p_n, and more precisely, the number $p_{n,j}$ of VM-Dyck paths of length n and with first ascent of length j satisfies recurrence (1). Indeed a VM-Dyck path of semi-length $n > 1$ and first ascent of length $j \geq 1$ can be uniquely obtained as follows:

i) adding a peak UD on the top of the first ascent of any VM-Dyck path of semi-length $n-1$ and first ascent of length $j-1$, thus giving the first term of (1);
ii) adding a peak UD at level j of the first ascent of any VM-Dyck path of semi-length $n-1$ and first ascent of length $i \geq j$; in this case, we produce a new valley at level $j-1$, and it can be marked in j different ways, thus giving the second term of (1).

On the other hand Recurrence (1) appears to be difficult to understand directly on $AV_n(1\underline{23}4)$. Therefore, in order to enumerate these permutations, in [10] Callan presents a chain of several non trivial bijections going from $AV_n(1\underline{23}4)$ to increasing ordered trees with increasing leaves. In view of the large amount of intermediary combinatorial structures involved, a natural question is to find a direct bijection between $1\underline{23}4$-avoiding permutations or 122^+3-avoiding permutations, and one of the "simple" families satisfying recurrence A113227.

Our main result is a recursive bijection between 122^+3-avoiding permutations and valley-marked Dyck paths. Our construction is simple enough to yield for instance an efficient exhaustive generation algorithm for these permutations.

Organization of the paper. In Section 2 we present a generation algorithm `INSERTPOINT`(π, `i`, `j`) for 122^+3-avoiding permutations, which receives as input a permutation $\pi \in AV_n(122^+3)$ and two indices i, j and returns a permutation in $AV_{n+1}(122^+3)$. We prove in particular that, for every 122^+3-avoiding permutation π' of size $n+1$, there are exactly a permutation π of size n and two indices i, j such that π' is the output of `INSERTPOINT`(π, `i`, `j`). In Section 3 we present an analogous algorithm `INSERTPEAK`($\mathcal{P}$, `i`, `j`) which inserts a peak UD in a specified position of a valley-marked Dyck path of semi-length n, and marks the possibly obtained valley, producing in a unique way a valley-marked Dyck path of semi-length $n+1$. In the above two sections, we also show that the recursive constructions of `INSERTPOINT` and `INSERTPEAK` can be both described by means of the same generating tree, described by the succession rule with two labels Ω_T:

$$\Omega_T = \begin{cases} (1,0) \\ \\ (h,s) \rightarrow (h+1,s)(0,1)\ldots(0,h)(0,h+1)^2\ldots(0,h+s)^{s+1} \,. \end{cases} \tag{2}$$

This fact immediately yields a recursive bijection between 122^+3-avoiding permutations and VM-Dyck paths. Basics about generating trees and succession rules are recalled in Section 3, while for further details we address the reader to [4,5,8]. Moreover we prove that, restricting to 123-avoiding permutations, our bijection reduces to Krattenthaler's bijection between 123-avoiding permutations and Dyck paths explained before, and we observe that our bijection can be performed in an iterative (non-recursive) way.

3 Generation of $AV(122^+3)$

In this section we write down an algorithm INSERTPOINT that takes as input a permutation π of $AV_n(122^+3)$ and two positive integers i, j, and inserts a point in π in a position that depends on i and j, and returns as output a permutation of $AV_{n+1}(122^+3)$.

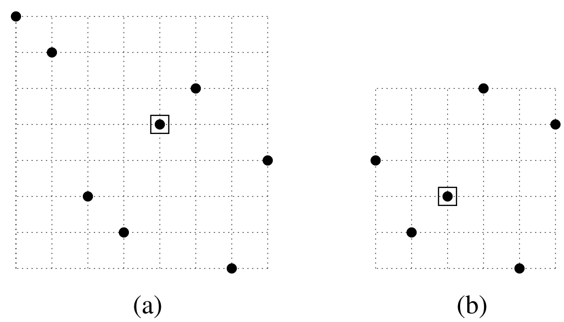

Fig. 2. (a) The permutation $\pi = 8\,7\,3\,2\,5\,6\,1\,4$, where $h(\pi) = 2$ and $t(\pi) = 5$; (b) The permutation $\sigma = 4\,2\,3\,6\,1\,5$, where $h(\sigma) = 0$ and $t(\sigma) = 3$. The threshold of both permutations is highlighted.

Before describing the algorithm INSERTPOINT we need to define the following parameters on π:

- the length $h(\pi)$ of the (possible empty) maximal initial subsequence $n\,(n-1)\dots(n-h+1)$ of consecutive points of π, called *double points*. More precisely $h(\pi) = h$ if and only if $\pi = n\,n-1\dots n-h+1\,\pi_{h+1}\,\dots\,\pi_n$ and $\pi_{h+1} \neq n-h$.
- the *threshold index* of π, denoted by $t(\pi) = t$, where t is the smallest index among $\{1,\dots,n+1\}$ that satisfies the two conditions:
  (1) π_t is not a left-to-right minimum;
  (2) $\pi_t = n$ OR $\pi_t + 1$ is on the left of π_t OR $\pi_t + 1 = n - h(\pi)$.
  If there is no point π_t in π satisfying these two properties, then $t(\pi)$ is set to be equal to $n+1$ (i.e. the threshold index is out of the permutation).
  Figure 2 shows the threshold of two permutations. Observe that the condition $\pi_t+1 = n-h(\pi)$ simply means that, if we remove the double points from π, then π_t is placed just below the maximal element (as in Figure 2 (b)). Note also that the threshold index $t(\pi)$ is larger than the parameter $h(\pi)$, because any double point is a left-to-right minimum. If $1 \leq t \leq n$, then point π_t is called the *threshold* of π.
- the parameter $s(\pi)$ such that $s(\pi) = t(\pi) - h(\pi) - 1$.

Moreover, we define the set of *admissible values* for π, as the set of pairs (i, j) such that $1 \leq i \leq t(\pi)$ and ($j = i$ or $h(\pi) < j < i$). This is the set of values of i and j for which INSERTPOINT(π, i, j) returns an output. Finally, we define the *label* of π as the pair $(h(\pi), s(\pi))$. Let LENGTH (resp. THRESHOLD, DOUBLE) be a function that computes the size n (resp. $t(\pi)$, $h(\pi)$) of $\pi \in AV_n(122^+3)$. The algorithm INSERTPOINT is given as Algorithm 1 below.

Proposition 3. *Let π be a permutation of $AV_n(122^+3)$ with label (h, s), where $h = h(\pi)$ and $s = s(\pi)$. The application of* INSERTPOINT(π, i, j)*, where i, j run over all admissible values, produces a set of $h + \frac{(s+1)(s+2)}{2}$ permutations of length $n+1$. More precisely, the multiset of the labels of all permutations produced from π is*

$$(h+1, s)(0, 1)\dots(0, h)(0, h+1)^2\dots(0, h+s)^{s+1}\,. \quad (3)$$

(the notation $(x, y)^g$ means that g permutations are being produced with label (x, y)).

Algorithm 1: Insertion of a point in a $122^{+}3$-avoiding permutation.

Input : A permutation π and two positive integers i and j.
Output: A permutation τ.

INSERTPOINT (π, i, j);
n=LENGTH(π);
t=THRESHOLD(π);
h=DOUBLE(π);
if $1 \leq i \leq t$ *and* $1 \leq j \leq i$ **then**
  if $h = 0$ **then**
    Order the array $[n+1, n, \pi_1, \dots, \pi_{i-1}]$ decreasingly;
    Remove from it the last entry and call S the resulting array of length i;
    Set x be equal to the $(i+1-j)$th element of S;
    Build $\tau = \pi'_1 \dots \pi'_{i-1}\, x\, \pi'_i \dots \pi'_n$, where $\pi'_k = \pi_k$ if $\pi_k < x$, and $\pi'_k = \pi_k + 1$ otherwise, for $k \in \{1, \dots, n\}$;
    return τ;
  else
    if $j = i$ **then**
      Build $\tau = \pi_1 \dots \pi_{i-1}\, (n+1)\, \pi_i \dots \pi_n$;
      return τ;
    end
    if $h < j < i$ **then**
      Build $\tau = \pi'_2 \dots \pi'_h\, \pi'_{h+1} \dots \pi'_j\, (n-h)\, \pi'_{j+1} \dots\, \pi'_{i-1}\, (n+1)\, \pi'_i \dots \pi'_n$, where $\pi'_k = \pi_k$ if $\pi_k < n-h$, and $\pi'_k = \pi_k + 1$ otherwise, for $k \in \{1, \dots, n\}$;
      return τ;
    else
      return *error*;
    end
  end
else
  return *error*;
end

Proof. It is immediate to see that $\pi = 1$ has label $(1, 0)$, since $h(\pi) = 1$ and $s(\pi) = 0$. Let us consider a permutation π labeled by (h, s), then we have to distinguish two cases:

- $h = 0$;

  CASE 1: If $i = 1$, then $j = 1$ and the previous algorithm inserts the element $n+1$ at the beginning of π, producing an output permutation τ labeled by $(1, s)$.

  CASE 2: Otherwise, for any $i > 1$ and every $1 \leq j \leq i$, consider the set $A = \{n + 1, n, \pi_1, \dots, \pi_{i-1}\}$. The algorithm inserts in position i the j-th element of the set $A \backslash \{\min A\}$ increasingly ordered and, after normalizing, it obtains τ. Note that if x is the point added in position $i > 1$, then x is the threshold of the output permutation τ. Indeed x is not a left-to-right minimum of τ and $x+1$ is not on its right or it is equal to the maximum of τ. So, the threshold index of τ is i and τ has label $(0, i-1)$. Since $1 \leq j \leq i$ it means that we obtain i permutations labeled by $(0, i-1)$ for $i \in \{2, \dots, t = s+1\}$.

- $h > 0$;

  CASE 1: $i \leq h$, i.e. the insertion is performed just before a double point: then j must be equal to i and the permutation τ is obtained by adding to π the element $n+1$ in position i. Then,

  - if $i = 1$, τ has label $(h+1, s)$; the number of double points increases by one, the threshold index of the obtained permutation is $t+1$, so $s(\tau) = s$.
  - otherwise τ has no double points and the threshold $t(\tau) = i$, i.e. $n+1$ is the new threshold point of τ, which then has label $(0, i-1)$, for $i \in \{2, \dots, h\}$.

$8\,7\,3\,2\,\underline{5}\,6\,1\,4$ (2,2)	$\longrightarrow$	$\underline{9}\,8\,7\,3\,2\,5\,6\,1\,4$ (3,2) $i{=}j{=}1$	$8\,\underline{9}\,7\,3\,2\,5\,6\,1\,4$ (0,1) $i{=}j{=}2$	$8\,7\,\underline{9}\,3\,2\,5\,6\,1\,4$ (0,2) $i{=}j{=}3$
		$8\,7\,3\,\underline{9}\,2\,5\,6\,1\,4$ (0,3) $i{=}4,j{=}4$	$8\,3\,6\,\underline{9}\,2\,5\,7\,1\,4$ (0,3) $i{=}4,j{=}3$	
		$8\,7\,3\,2\,\underline{9}\,5\,6\,1\,4$ (0,4) $i{=}5,j{=}5$	$8\,3\,2\,6\,\underline{9}\,5\,7\,1\,4$ (0,4) $i{=}5,j{=}4$	$8\,3\,6\,2\,\underline{9}\,5\,7\,1\,4$ (0,4) $i{=}5,j{=}3$
$4\,2\,\underline{3}\,6\,1\,5$ (0,2)	$\longrightarrow$	$\underline{7}\,4\,2\,3\,6\,1\,5$ (1,2) $i{=}j{=}1$		
		$4\,\underline{7}\,2\,3\,6\,1\,5$ (0,1) $i{=}2,j{=}2$	$4\,\underline{6}\,2\,3\,7\,1\,5$ (0,1) $i{=}2,j{=}1$	
		$4\,2\,\underline{7}\,3\,6\,1\,5$ (0,2) $i{=}3,j{=}3$	$4\,2\,\underline{6}\,3\,7\,1\,5$ (0,2) $i{=}3,j{=}2$	$5\,2\,\underline{4}\,3\,7\,1\,6$ (0,2) $i{=}3,j{=}1$

Fig. 3. An example of the outputs of algorithm `INSERTPOINT` on a given permutation π and all the admissible values i, j. For each permutation the threshold is underlined.

CASE 2: $i > h$. We have to split the two cases:

- $j = i$. If $j = i$ the permutation τ is obtained by adding to π the point $n+1$ in position i, therefore τ has no double points and the threshold $t(\tau) = i$, i.e. $n+1$ is the new threshold point of τ, so it has label $(0, i-1)$.
- If $h < j < i$, the algorithm adds in position i the element $n+1$, then it removes the leftmost double point n, inserts in position j the value $n-h$ and, after normalizing, it obtains τ. Note that also in this case the threshold index is i, because the point $n-h+1$ is on the right of $n-h$ and then the point $n-h$, that is added in position $j \leq i$, is not a possible threshold index. Therefore the label of τ is $(0, i-1)$. Since $h < j < i$ it means that we obtain $i-h-1$ permutations labeled by $(0, i-1)$ for $i \in \{h+1, \ldots, t = s+h+1\}$.

Observe finally that the resulting multisets of labels in both cases can be written uniformly as (3).

Proposition 4. *Any permutation τ obtained as output of* `INSERTPOINT` *avoids* 122^+3.

Proof. Let π be a 122^+3-avoiding permutation, and i, j admissible values. First note that on the left of the index threshold $t(\pi)$ there is no occurrence of the pattern 122^+. This means that the new element x, inserted in π with `INSERTPOINT`(π, i, j), cannot create the pattern 122^+3 by playing the role of the 3. Moreover note that, by construction, x cannot be a left-to-right minimum. This means that x cannot create the pattern 122^+3 by playing the role of the 1. Therefore the new point x inserted in π can only create an occurrence of 122^+3 by playing the role of 2 or 2^+.

Let us examine all the possible cases:

– $h(\pi) = 0$; Let us recall that the element x is inserted in position i of the permutation π, with $i \leq t(\pi)$. If $x = n+1$ no occurrences of 122^+3 can be created. If $x = n$ then, by construction, $n+1$ is on the right of n, that is n cannot create the pattern 122^+3 by playing the role of the 2. It could create it by playing the role of 2^+ but this is impossible since $n-1$ is on the right of n. Indeed, if it was on its left, then $n-1$ would satisfy the conditions of threshold of π and, at the same time, it would be on the left of $t(\pi)$ (since $i \leq t(\pi)$), thus leading to a contradiction. Otherwise if $x = \pi_s$, with $s \leq i$, then by construction $x+1$ lies on the left of x. This means that x cannot create the pattern 122^+3 by playing the role of the 2. It could create it by playing the role of 2^+ but we are going to show that this is impossible. In order to do it we need to take account of the position of the element $x-1$. If $x-1$ is on the right of x then x cannot create the pattern 122^+3. Instead, if $x-1$ is on the left of x and it is not

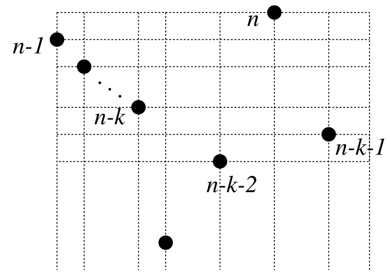

Fig. 4. The plough configuration in a 122^+3-avoiding permutation of length n.

a left-to-right minimum, then $x-1$ must also be on the left of $x+1$, since otherwise $x-1$ would satisfy the conditions of threshold in π and, at the same time, it would be on the left of $t(\pi)$. Suppose now that x creates the pattern 122^+3 in τ by playing the role of 2^+, since we showed that $x-1$ is also on the left of $x+1$, then $x+1$ plays the role of 2^+, together with the 1, 2, and 3 that are in the same pattern created by x. This means that removing x from τ and coming back to π, an occurrence of the pattern is obtained, thus yielding to a contradiction.
- If $h(\pi) \neq 0$ and $j = i$, with $i \leq t(\pi)$, then $n+1$ is inserted in position i and no occurrences of 122^+3 can be produced. If $h < j < i$, then $n+1$ is inserted in position i, and the leftmost double point n is removed and the point $y = n-h$ is inserted in position $j < i$. After normalizing this array, τ is obtained. We need to check that y, which is not a left-to-right minimum, does not form any 122^+3 occurrence. Since $y+1 = n-h+1$ is a right-to-left maximum, then y cannot create an occurrence of 122^+3 by playing the role of 2. It could create an occurrence of 122^+3, but we have that this is impossible, since if $y-1 = n-h-1$ is on the left of y then it must be a left-to-right minimum. Indeed, suppose that $y-1$ is on the left of y and it is not a left-to-right minimum. Since $y-1$ is on the left of y, then it is also on the left of $y+1 = n-h+1$, then it satisfies the conditions of threshold of π and, at the same time, it would be on the left of $t(\pi)$, thus giving a contradiction.

Figure 3 shows the applications of the algorithm `INSERTPOINT` to the permutation $8\,7\,3\,2\,\underline{5}\,6\,1\,4$, whose threshold is underlined, for all the admissible values of i and j.

Our aim is now to prove that each permutation of $AV_n(122^+3)$ is produced exactly once by our algorithm `INSERTPOINT`. To do this, we define a second algorithm `REMOVEPOINT` that deletes a point from a permutation $\tau \in AV_n(122^+3)$ giving as output a triple (π, i, j), where $\pi \in AV_{n-1}(122^+3)$ and i, j depend on the deletion and we will prove that these two algorithms are inverse one of the other.

The algorithm `REMOVEPOINT` is given as Algorithm 2 below: it takes as input a permutation $\tau \in AV(122^+3)$ of length n and returns a permutation π of length $n-1$ and two parameters i, j. A crucial step in this algorithm is to check whether π contains a special configuration of points, called *plough configuration*, graphically represented in Figure 4. Formally, $\tau \in AV_n(122^+3)$ contains the plough configuration if the point $\tau_t = n$, where t is the threshold index, and the index of the point $n-k-1$ (resp. $n-k-2$) is greater than t (resp. smaller than t but greater than $k+1$), where k is the cardinality of the starting (possibly empty) sequence having the form $n-1\,n-2\ldots n-k$ of τ.

Proposition 5. *Any permutation π obtained as output of* `REMOVEPOINT` *avoids* 122^+3.

Proof. It is clear that if $\tau_t = n$ and τ does not contain the plough configuration, then the permutation π, obtained by removing n still avoids 122^+3. Then we still have to consider two cases:

- $h(\tau) = 0$ and $t(\tau) \neq n$; If $t(\tau) = n-1$ removing $n-1$ could not create a 122^+3 pattern since n is a left-to-right maximum of τ. If $t(\tau) = x$ with $x+1$ on its left, then removing x could create a 122^+3 pattern depending on the position of $x-1$. If $x-1$ is on the right of $x+1$, then removing x does not create any occurrence of 122^+3 in π. If $x-1$ is on the left of $x+1$

Algorithm 2: Removing a point from a 122^+3-avoiding permutation.

Input : A permutation τ.
Output: A permutation π and two positive integers i and j.

```
REMOVEPOINT (τ);
n=LENGTH(τ);
t=THRESHOLD(τ);
h=DOUBLE(τ);
if h = 0 then
    if τ_t ≠ n then
        Set j be equal to the number of point of τ smaller than π_t having index smaller than t;
        Build π = τ'_1 ... τ'_{t−1} τ'_{t+1} ... τ'_n where τ'_i = τ_i − 1 if τ_i > τ_t and τ'_i = τ_i otherwise; return
        (π, t, j);
    end
    if τ_t = n and there exist some points forming the plough configuration then
        Set j be equal to the index of point n − k − 2;
        Build π = n − 1 τ'_1 ... τ'_{j−1} τ'_{j+1} ... τ'_{t−1} τ'_{t+1} ... τ'_n where τ'_i = τ_i − 1 if τ_i > n − k + 2 and
        τ'_i = τ_i otherwise ;
        return (π, t, j);
    else
        Build π = τ_1 ... τ_{t−1} τ_{t+1} ... τ_n;
        return (π, t, t);
    end
else
    Build π = τ_2 ... τ_n;
    return (π, 1, 1);
end
```

then $x-1$ is also on the left of x (since, by construction, $x+1$ is on the left of x) and, by construction, n is on the right of x. Then, since τ avoids 122^+3, $x-1$ must be a left-to-right minimum and no occurrence of 122^+3 can be generated removing it.

– $t(\tau)=n$ and there exist points forming the plough configuration; It is clear that since $n-k-1$ is a right-to-left maximum, removing point $n-k-2$ no 122^+3 occurrences are formed.

Observation 1 *Let π be a permutation of $AV_n(122^+3)$. The application of* $\texttt{INSERTPOINT}(\pi,\texttt{i},\texttt{j})$, *where i,j run over all admissible values, produces a set of permutations of length $n+1$ which can be described by means of the following* succession rule*:*

$$\Omega_T = \begin{cases} (1,0) \\ \\ (r,s) \rightarrow (r+1,s)(0,1)\dots(0,r)(0,r+1)^2\dots(0,r+s)^{s+1}\,. \end{cases}$$

Indeed, as pointed out in the description of INSERTPOINT*, $\pi=1$ has label $(1,0)$ and for a permutation π with label (r,s), the admissible values of i,j are listed in table in Figure 1, where it is also indicated the label of* $\texttt{INSERTPOINT}(\pi,\texttt{i},\texttt{j})$*, and the notation $(e,f)^g$ means that there are g permutations being produced with label (e,f):*

We recall that Ω_T is an object known in the literature under the same of *succession rule*. It defines a generating tree, i.e. a labelled planar tree whose root has label $(1,0)$ and such that the sons of a node with label (r,s) have labels according to Ω_T. Due to our observation, every node at level n of Ω_T corresponds to a 122^+3-avoiding permutation of n. Generating trees have wide applications to problems concerned with the enumeration and the generation of combinatorial objects [2,4,5,8].

Proposition 6. *For any permutation $\tau \in AV_{n+1}(122^+3)$,* $\texttt{INSERTPOINT}\,(\,\texttt{REMOVEPOINT}(\tau)) = \tau$*.*

Conversely for any permutation $\pi \in AV_{n+1}(122^+3)$ and admissible pair (i,j) for π, we have $\texttt{REMOVEPOINT}\,(\,\texttt{INSERTPOINT}(\pi,i,j)) = (\pi,i,j)$*.*

Admissible values	Output label
$i = j = 1$	$(r+1, s)$
$i = j = 2$	$(0, 1)$
$\vdots$	$\vdots$
$i = j = r+1$	$(0, r)$
$i = r+2, r+1 \leq j \leq r+2$	$(0, r+1)^2$
$\vdots$	$\vdots$
$i = r+s+1, r+1 \leq j \leq r+s+1$	$(0, r+s)^{s+1}$

Fig. 5. All the admissible values of i, j for a permutation π and the label corresponding to the output of INSERTPOINT(π, i, j).

Proof. Let us start with the first point. The algorithm distinguishes two cases, depending whether $h = 0$ and $h > 0$:

- $h = 0$:
  - $\tau_t \neq n$: REMOVEPOINT(τ) removes from τ the point in position t and normalizes the obtained array. The output is a triple (π, i, j), where π is the normalized permutation, $i = t$ and j is given by the number of points on the left of τ_t and smaller than τ_t. Clearly the threshold index of the resulting permutation is at least t so that (i, j) is an admissible pair for π and INSERTPOINT correctly reconstruct τ using it.
  - $\tau_t = n$:
    - if τ contains the plough configuration REMOVEPOINT(τ) removes from τ both points $\tau_t = n$ and $n - k - 2$, then adds at the beginning the point $n - 1$ and normalizes the obtained array. The output in this case is the triple (π, i, j), where π is the permutation of length $n - 1$, $i = t$ and j is the index of $n - k - 2$. Observe that π does not contain the plough configuration anymore and starts with a sequence of double points so that the last case of INSERTPOINT applies to reconstruct the plough configuration.
    - otherwise, the algorithm simply removes the point $\tau_t = n$ and the output is (π, i, j), where $i = j = t$ and the permutation π has threshold index at least i; the inverse operation is clearly given by case $i = j$ of INSERTPOINT (with or without $h = 0$).
- $h > 0$; The output in this case is the triple (π, i, j), where π is the permutation obtained removing the leftmost element, and $i = j = 1$. Clearly the inverse operation is to insert the maximum at the leftmost position, as done by INSERTPOINT for $i = j = 1$.

The proof of the second assertion of the theorem follows a similar easy case analysis.

Recall here from [4,5,8] that the *generating tree* defined by the succession rule Ω_T of Formula 2 is a labeled planar tree whose root has label $(1, 0)$ and such that the sons of a node with label (h, s) have labels according to Ω_T. The previous proposition immediately yields the following theorem:

Theorem 1. *Algorithm* INSERTPOINT *determines a generating tree for* $AV(122^+3)$ *that can be described by succession rule* Ω_T. *In particular, the nodes at level* n *of the generating tree of* Ω_T *are in one-to-one correspondence with* 122^+3*-avoiding permutations of size* n.

4 Generation of valley-marked Dyck paths

In this section we deal with the class of VM-Dyck paths. We are going to describe two algorithms INSERTPEAK and REMOVEPEAK, whose composition results to be the identity, providing the exhaustive generation of this family of objects. Moreover, assigning to each VM-Dyck path $\mathcal{P}$ a label

Algorithm 3: Insertion of a peak in a VM-Dyck path.

Input : A VM-Dyck path $\mathcal{P}$ and two positive integers i and j.
Output: A VM-Dyck path $\mathcal{T}$.

```
INSERTPEAK (𝒫, i, j);
Write 𝒫 = (U D 0)^h U^k α according to (4);
if 1 ≤ i ≤ h + k and j ≤ i then
    if h = 0 then
        Build the VM-Dyck path 𝒯 = U^i D j−1 U^{k+1−i} α;
        return 𝒯;
    else
        if j = i then
            Let q = min{i, h + 1};
            Build the VM-Dyck path 𝒯 = U^i D^q j−q (U D 0)^{h+1−q} U^{k+q−i} α;
            return 𝒯;
        end
        if h < j < i then
            Build the VM-Dyck path 𝒯 = U^i D^{h+1} j−h−1 U^{k+h+1−i} α;
            return 𝒯;
        else
            return error;
        end
    end
else
    return error;
end
```

(h, s) and considering all the admissible output of $\texttt{INSERTPEAK}(\mathcal{P}, \mathtt{i}, \mathtt{j})$, for any i, j, we retrieve the succession rule Ω_T. In this section we use a formal representation of a VM-Dyck path $\mathcal{P}$ of semi-length as a pair (P, v) where P is Dyck path of semi-length n (the *underlying Dyck path* of $\mathcal{P}$), m is the number of valleys of P, and $v = (v_1, \dots, v_m)$ is an array whose ith entry indicates the level where the ith valley of P is marked. We often use a compact representation of $\mathcal{P}$, obtained by replacing the occurrence of the ith factor DU in P (i.e., the ith valley of P) by the factor $D\,\mathbf{v}_i\,U$. For example the VM-Dyck path in Figure 1 (c) is represented as $UUD\,\mathbf{1}\,UUUD\,\mathbf{2}\,UDD\,\mathbf{0}\,UDDD$.

Let $\mathcal{P}$ be a VM-Dyck path. We represent $\mathcal{P}$ in a compact way using two indices h and k, as follows:

$$\mathcal{P} = (UD\,\mathbf{0})^h U^k \alpha \tag{4}$$

where $h \geq 0$ and $k > 0$ are chosen to be maximal. Observe that such a representation is not allowed if the underlying Dyck path is a sequence of peaks, and in this case we write $\mathcal{P} = (UD\,\mathbf{0})^h\, UD$, and set $k = 0$.

Moreover, we introduce the parameter $s = k - 1$ if $k \neq 0$, or $s = 0$ otherwise.

Proposition 7. *Let $\mathcal{P}$ be a VM-Dyck path of semi-length n with label (h, s). The application of* $\texttt{INSERTPEAK}(\mathcal{P}, \mathtt{i}, \mathtt{j})$*, where i, j run over all admissible values, produces a set of $h + \frac{(s+1)(s+2)}{2}$ VM-Dyck paths of semi-length $n + 1$. More precisely, the multiset of the labels of all VM-Dyck paths produced from $\mathcal{P}$ is*

$$(h + 1, s)(0, 1) \dots (0, h)(0, h + 1)^2 \dots (0, h + s)^{s+1}\,. \tag{5}$$

Proof. It is immediate to see that the VM-Dyck path $U\,D$ has label $(1, 0)$. Let us consider a VM-Dyck path $\mathcal{P}$ with indices h and k defined above and label (h, s), then we have to distinguish two cases :

- $h = 0$; Consider the line $y = i \leq k$, $\texttt{INSERTPEAK}$ adds a peak to $\mathcal{P}$ at height i, which means that the peak point lies on the line $y = i$. Then, a VM-Dyck path $\mathcal{T}$ is obtained by marking

Algorithm 4: Removing a peak from a VM-Dyck path.

Input : A VM- Dyck path $\mathcal{T}$.
Output: A VM- Dyck path $\mathcal{P}$ and two positive integers i and j.
`REMOVEPEAK` $(\mathcal{T})$;
Write $\mathcal{T} = (U\,D\,\mathbf{0})^h\,U^k\,\alpha$ such as h, k are maximal;
if $h = 0$ **then**
  Write $\alpha = D^{\ell+1}\,\mathbf{a}\,U\,\beta$, with $\ell < k$. Build the path $\mathcal{P} = (U\,D\,\mathbf{0})^{\ell}\,U^{k-\ell}\,\beta$;
  return $(\mathcal{P}, k, a + \ell + 1)$;
else
  Build the path $\mathcal{P} = (U\,D\,\mathbf{0})^{h-1}\,U^k\,\alpha$;
  return $(\mathcal{P}, 1, 1)$;
end

the produced valley at level $j-1$. If $i = j = 1$, we insert at the beginning of $\mathcal{P}$ a new UD occurrence and we mark the valley originated at level $y = 0$. In this case, the output $\mathcal{T}$ has label $(1, s)$. Whereas, for any $i > 1$ and every $1 \leq j \leq i$, we produce a new VM-path obtained by inserting the peak UD at height i in a point of the first ascent of $\mathcal{P}$ and marking the originated valley at level $j-1$. In this case, the output $\mathcal{T}$ has label $(0, i-1)$.

– $h > 0$; we distinguish two cases:
  - If $i \leq h$ then j must be equal to i and the path of $\mathcal{T}$ is obtained by replacing the prefix $(UD)^{i-1}$ of $\mathcal{P}$ with U^iD^i and by marking the originated valley at level 0. Then, if $i = 1$, $\mathcal{T}$ has label $(h+1, s)$, otherwise it has label $(0, i-1)$.
  - If $i > h$, the path of $\mathcal{T}$ is obtained by deleting the prefix $(UD)^h$ of $\mathcal{P}$, by adding a $U^{h+1}D^{h+1}$ occurrence at level i, and by marking the originated valley at level $j-h-1$. Then for any $i > h$, and any $j \leq i$, $\mathcal{T}$ has label $(0, i-1)$.

In Figure 6 is shown an example of the application of `INSERTPEAK` to a given VM-Dyck path, for any admissible values i, j. By construction we have that:

Proposition 8. *The output $\mathcal{T}$ of* `INSERTPEAK` *is a VM-Dyck path.*

Our algorithm `INSERTPEAK` is given as Algorithm 3, where the two cases described above are treated as one by the introduction of the value q which takes into account the minimum among i and $k+1$.

The inverse algorithm `REMOVEPEAK` is given as Algorithm 4. It decreases by one the semi-length of a VM-Dyck path $\mathcal{T}$ by removing a peak and gives as output a triple $(\mathcal{P}, i, j)$, where i, j keep track of the removed peak. The new path $\mathcal{P}$ obtained with `REMOVEPOINT`$(\mathcal{T})$ is clearly a VM-Dyck path. The following statement, analogous to that of Proposition 6, is straightforward.

Proposition 9. *For any VM-Dyck path $\mathcal{P}$, we have*

$$\texttt{INSERTPEAK}\ (\texttt{REMOVEPEAK}(\mathcal{P})) = \mathcal{P}.$$

Conversely for any VM-Dyck path $\mathcal{P}$ and admissible pair (i, j) for $\mathcal{P}$, we have

$$\texttt{REMOVEPEAK}\ (\texttt{INSERTPEAK}(\mathcal{P}, i, j)) = (\mathcal{P}, i, j).$$

These propositions prove that `INSERTPEAK` generates VM-Dyck paths and its generation can be described by means of the generating tree Ω_T.

5 A bijection between $AV(122^+3)$ and valley-marked Dyck paths

We recursively define a function φ which maps a permutation in $AV_n(122^+3)$ onto a VM-Dyck path of size n:

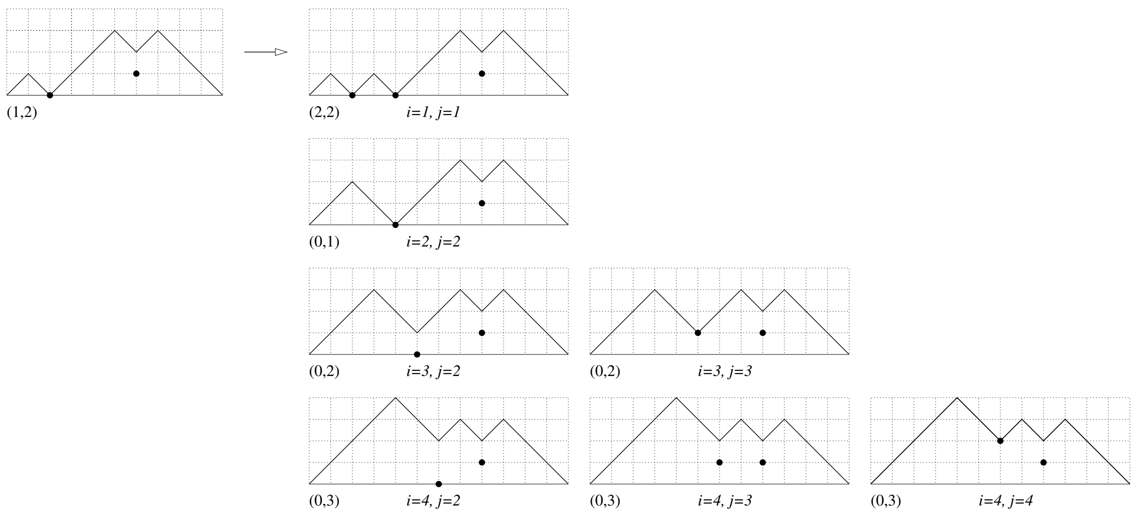

Fig. 6. All the VM-Dyck paths produced by the VM-Dyck path $\mathcal{P}$ on the left through the application of $\texttt{INSERTPEAK}(\mathcal{P}, \texttt{i}, \texttt{j})$. For each VM-Dyck path the corresponding label in Ω_T is reported.

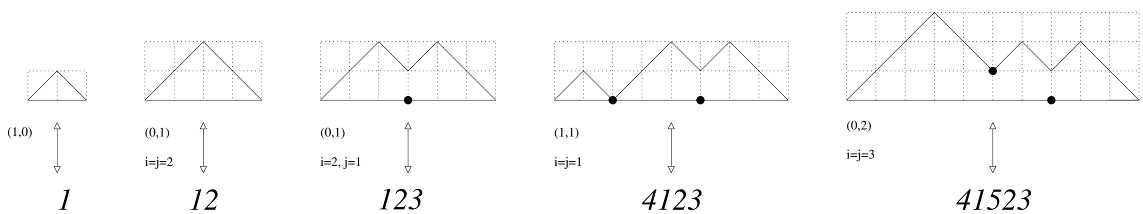

Fig. 7. The growth of a VM-Dyck path starting from the path UD and the corresponding 122^+3-avoiding permutations via the bijection φ. At each step the values i, j and the label of the object in Ω_T is reported.

$$\begin{cases} \varphi(1) = UD & \text{if } n = 1 \\ \\ \varphi(\texttt{INSERTPOINT}(\pi, \texttt{i}, \texttt{j})) = \texttt{INSERTPEAK}(\varphi(\pi), \texttt{i}, \texttt{j}) & \text{if } n > 1. \end{cases}$$

The fact that φ is a bijection follows from the fact that in the generating tree of Ω_T each 122^+3-avoiding permutation of size n (resp. VM-Dyck path of semi-length n) is uniquely identified by a path from the root to a node at level n. Thus φ maps a permutation in $AV_n(122^+3)$ onto the VM-Dyck path which corresponds to the same path in Ω_T (see Figure 7).

When restricting to 123-avoiding permutations, we have the following remarkable result.

Corollary 1. *The restriction of φ to 123-avoiding permutations determines a bijection between 123-avoiding permutations of length n and Dyck paths of semi-length n which is precisely Krattenthaler's bijection (explained in Section 1)*

Proof. On the one hand, Dyck paths can be naturally represented as VM-Dyck paths where every valley is marked at the maximal level. By construction, every Dyck path of semi-length $n+1$ is obtained from a Dyck path $\mathcal{P}$ of semi-length n through the application of $\texttt{INSERTPEAK}(\mathcal{P}, i, i)$, for any admissible i. On the other hand, any 123-avoiding permutation π of length n is a 122^+3-avoiding permutation whose threshold is the point $n - h(\pi)$ and any permutation in $AV_{n+1}(123)$ is obtained by inserting $n+1$ in any point on the left of the threshold of π, namely by applying $\texttt{INSERTPOINT}(\pi, i, i)$, for any admissible i. It is easy to verify that restricting the previous bijection to these two subclasses of objects give us the same one-to-one correspondence as Krattenthaler's bijection. □

The succession rule describing the growth of 123-avoiding permutations presented in Corollary 1, denoted by Ω_R, is a neat restriction of Ω_T:

$$\Omega_R = \begin{cases} (1,0) \\ \\ (h,s) \rightarrow (h+1,s)(0,1)\ldots(0,h+s)\,. \end{cases}$$

Finally, adapting the general strategy of [2], the fact that we have a generating tree for $AV(122^+3)$ with finitely many labels (two labels in our case) and such that each node produces at least two valid children implies an amortized constant time generation algorithm (CAT) for codes of permutations of $AV_n(122^+3)$.

References

1. E. Babson, E. Steingrìmsson, Generalized permutation patterns and a classification of the Mahonian statistics, *Sem. Lothar. Combin.* 44, Art. B44b (2000).
2. S. Bacchelli, E. Barcucci, E.Grazzini, E. Pergola, Exhaustive generation of combinatorial objects by ECO, *Acta Informatica* 40, 585–602, 2004.
3. V. Bafna, P. A. Pevzner, Genome rearrangements and sorting by reversals, *34th Annual Symposium on Foundations of Computer Science* (Palo Alto, CA, 1993), IEEE Comput. Soc. Press, Los Alamitos, CA, 148–157 (1993).
4. C. Banderier, M. Bousquet-Mélou, A. Denise, P. Flajolet, D. Gardy, D. Gouyou-Beauchamps, Generating functions for generating trees, *Disc. Math.* 246, 29–55, 2002.
5. E. Barcucci, A. Del Lungo, E. Pergola, R. Pinzani, ECO: a methodology for the Enumeration of Combinatorial Objects, *J. Diff. Eq. and App.* 5, 435–490, 1999.
6. A. Björner, The Möbius function of subword order, *Invariant theory and tableaux* (Minneapolis, MN, 1988), 118–124, IMA, *Math. Appl.*, vol. 19, Springer, New York, (1990).
7. M. Bona, *Combinatorics of Permutations*, Second Edition (2012), Chapman and Hall/CRC.
8. M. Bousquet-Mélou, Four classes of pattern-avoiding permutations under one roof: generating trees with two labels, *Electronic J. Combinatorics* 9(2), paper R19, 2003.
9. A. Burstein, Enumeration of Words with Forbidden Patterns, *Ph.D. Thesis, University of Pennsylvania*, (1998).
10. D. Callan, A bijection to count $(1 - 23 - 4)$-avoiding permutations, `arxiv:1008.2375.v1`.
11. A. Claesson, Generalized Pattern Avoidance, *European J. Combinatorics* 22(7), 961–971, 2001.
12. M. Dairyko, L. Pudwell, S. Tyner, C. Wynn, Non-contiguous pattern avoidance in binary trees, *Electr. J. Combin.*, vol.19, no. 3, (2012), 22.
13. P.Dulio, S.M.C.Pagani, A. Frosini, A geometrical characterization of regions of uniqueness and applications to discrete tomography, *Inverse Problems*, 31 (12) (2015) DOI 10.1088/0266-5611/31/12/125011.
14. P. Dulio, S. M. C. Pagani, A. Frosini, Regions of uniqueness quickly reconstructed by three directions in discrete tomography, *Fundamenta Informaticae*, 155 (4) 407-423 (2017).
15. S. Elizalde, *Asymptotic enumeration of permutations avoiding generalized patterns*, *Advances in Applied Mathematics* 36, 138–155 (2006).
16. A.Frosini, V.Guerrini, S.Rinaldi, Geometric properties of matrices induced by pattern avoidance, *Theoretical Computer Science* 624 (2016) 109-120.
17. A. Frosini, C. Picoleau, How to decompose a binary matrix into three hv-convex polyominoes, R. Gonzales-Diaz et al. (Eds.): *DGCI 2013, LNCS* 7749, 311–323. Springer, Heidelberg (2013).
18. A. Goyt, Avoidance of partitions of a three element set, *Adv. Appl. Math.*, vol. 41, (2008), 95–114.
19. M. Klazar, On abab-free and abba-free sets partitions, *European J. Combin.*, vol. 17, (1996), 53–68.
20. C. Krattenthaler, Permutations with Restricted Patterns and Dyck Paths, *Advances in Applied Mathematics* 27(2-3), 510–530 (2000).
21. D. E. Knuth, *The Art Of Computer Programming* Vol. 1, (1968) Boston: Addison-Wesley.
22. OEIS Foundation Inc., *The On-line Encyclopedia of Integer Sequences*, `http://oeis.org` (2011).
23. V. R. Pratt, Computing permutations with double-ended queues. Parallel stacks and parallel queues, *Proc. Fifth Annual ACM Symposium on Theory of Computing* (Austin, Tex., 1973), 268–277 (1973).
24. P. Rosenstiehl, R. Tarjan, Gauss codes, planar Hamiltonian graphs, and stack-sortable permutations, *Journal of Algorithms*, 5 (3) 375–390 (1984).
25. E. Rowland, Pattern avoidance in binary trees, *J. Combin. Theory Ser. A*, vol. 117, (2010), 741–758.
26. B. E. Sagan, Pattern avoidance in set partitions, *Ars Combin.*, vol. 94, (2010), 79–96.
27. R. Simion, F. Schmidt, Restricted permutations, *European Journal of Combinatorics*, 6 383–406 (1985).
28. R. P. Stanley, *Enumerative Combinatorics*, Vol. 2, Cambridge University Press, Cambridge (1999).
29. R. Tarjan Sorting using networks of queues and stacks, *Journal of the ACM* 19 (2), 341–346 (1972).
30. G. A. Watterson, W. J. Ewens, T. Hall, A. Morgan, The chromosome inversion problem, *J. Theor. Biol.* (99) 1–7 (1982).