Maximal decreasing runs in permutations

Bachelor of Science, Mathematics Thesis

March 14, 2016

Jón Steinn Elíasson
jonsteinn@gmail.com

Supervisor: Henning Arnar Úlfarsson

Abstract

Zeilberger introduced a restriction on the positions of the largest elements in a permutation in the process of verifying the conjecture on the number of West-2-stack-sortable permutations (West in [1] and Zeilberger in [2]). In this paper we look at Zeilberger's restriction combined with the avoidance of classical patterns of length 3. We provide enumeration results as well as bijections to two different types of Dyck paths.

Acknowledgements

I would like to thank Einar Steingrímsson who recommended the subject of this thesis, Anders Claesson for helping at one point and Henning Arnór Úlfarsson for his guidance. All of them I would also like to thank for their invaluable teachings.

Contents

Tables

Figures

Listings

1 Introduction

Definition 1.1. A *permutation* π is a one-to-one correspondence between a set and itself.

In this paper, we will denote the set $\{1, 2, \dots, n\}$ with $[n]$ and use the one line notation for permutations, that is $\pi = \pi_1\pi_2 \cdots \pi_n$ where $\pi_j = i$ if $\pi(j) = i$. Furthermore, we say i is in position j if $\pi_j = i$. Let $\mathcal{S}_n$ be the set of all permutations on $[n]$.

Definition 1.2. For a subset $A = \{a_1, a_2, \dots, a_k\} \subseteq [n]$ where $a_1 < a_2 < \cdots < a_k$ and a permutation π on $[n]$ we say that A is *decreasing* in π if

$$\pi^{-1}(a_1) > \pi^{-1}(a_2) > \cdots > \pi^{-1}(a_k).$$

Example 1.1. The set $\{1, 3, 5\} \subseteq [6]$ is decreasing in the permutation 563412.

There are n different subsets of $[n]$ containing only the k largest elements for $k = 1, 2, \dots, n$, namely $\{n\}, \{n-1, n\}, \dots, \{1, 2, \dots, n\}$. The following definition was first provided by Zeilberger in [2].

Definition 1.3. For a permutation π of length n, we say that it has a *maximal decreasing run of length k* if $\{n-k+1, n-k+2, \dots, n\}$ is decreasing in π and $\{n-k, n-k+1, \dots, n\}$ is not. If $n-k > 0$, we say that a *cut point* $n-k$ *cuts* the maximal decreasing run in π.

We will use the notation $\Lambda(\pi) = k$ for a permutation π that has a maximal decreasing run of length k.

Example 1.2. In the following examples we underline the maximal decreasing run.

$$\begin{aligned}
\Lambda(3124\underline{5}) &= 1 \text{ where 4 is a cut point,}\\
\Lambda(\underline{5}3\underline{4}21) &= 2 \text{ where 3 is a cut point,}\\
\Lambda(12\underline{543}) &= 3 \text{ where 2 is a cut point,}\\
\Lambda(\underline{543}1\underline{2}) &= 4 \text{ where 1 is a cut point.}
\end{aligned}$$

Let $\mathcal{S}_{n,k} = \{\pi \in \mathcal{S}_n : \Lambda(\pi) = k\}$ and $M(n,k)$ the cardinality of $\mathcal{S}_{n,k}$.

Definition 1.4. For $k \le n$ and two permutations $\pi \in \mathcal{S}_n$ and $\sigma \in \mathcal{S}_k$, we say that π *contains* σ if there exists indices $i_1 < i_2 < \cdots < i_k$ such that $\pi_{i_1}\pi_{i_2}\cdots\pi_{i_k}$ is in the same relative order as $\sigma_1\sigma_2\cdots\sigma_k$. If such indices exist, we say that $\pi_{i_1}\pi_{i_2}\cdots\pi_{i_k}$ is an *occurrence* of σ in π and σ is a *pattern*.

Example 1.3. The permutation $84372156 \in \mathcal{S}_8$ has 837 as an occurrence of the pattern 312 which it therefore contains.

Definition 1.5. For permutations $\pi \in \mathcal{S}_n$ and $\sigma \in \mathcal{S}_k$, we say that π *avoids* the pattern σ if it does not contain σ.

Example 1.4. The permutation $32416758 \in \mathcal{S}_8$ has no three elements in the same relative order as 312 which it therefore avoids.

We will only concern ourselves with the avoidance of patterns in $\mathcal{S}_3$. It is well known that the number of permutations of length n that avoid a fixed $\sigma \in \mathcal{S}_3$ is the n-th Catalan number, $C_n = \frac{1}{n+1}\binom{2n}{n}$ (MacMahon in [3] and Knuth in [4]). We let $\mathcal{S}_n(\sigma)$ be the set of permutations of length n that avoid σ.

Let $\mathcal{S}_{n,k}(\sigma)$ be the set of permutations with a maximal decreasing run of length k that avoid σ and $M_\sigma(n,k)$ the cardinality of that set. For $k > 2$ we have $\mathcal{S}_{n,k}(321) = \emptyset$ since the maximal decreasing run is an occurrence of 321. We will therefore ignore $\sigma = 321$ when we look at $\mathcal{S}_{n,k}(\sigma)$ for $\sigma \in \mathcal{S}_3$.

In section 2 we find the cardinality of the set $\mathcal{S}_{n,k}$ in two different ways, providing a combinatorial identity. We additionally provide a combinatorial proof of a recursion.

In section 3 and 4 we look at these restrictions in permutations that avoid a single three letter pattern. We provide five bijections that classify the pattern avoiding permutations into two different types of restricted Dyck paths, one with k returns to the x-axis and the other with the first descent of length k. We also provide the cardinality and a recursion for each class. Figure 1 demonstrates the classification.

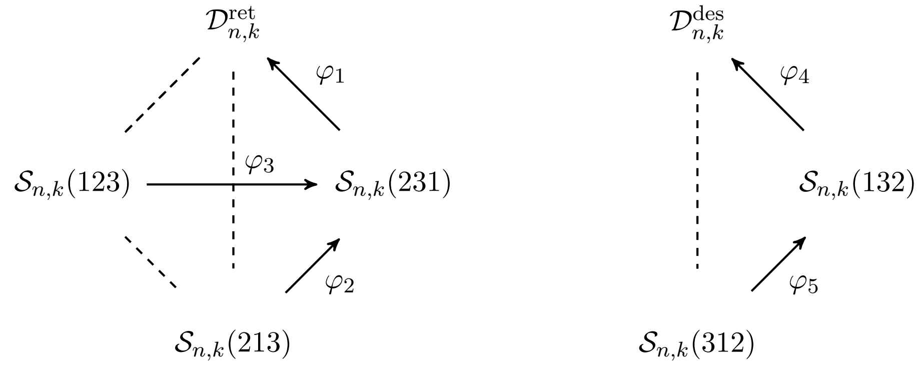

Figure 1: Classification of $\mathcal{S}_{n,k}(\sigma)$

2 Enumeration of $\mathcal{S}_{n,k}$

We will provide two ways of counting the elements in $\mathcal{S}_{n,k}$, both of which require k to be less than n. The results will give a combinatorial identity.

$n\backslash k$	1	2	3	4	5	6	7	8
1	1							
2	1	1						
3	3	2	1					
4	12	8	3	1				
5	60	40	15	4	1			
6	360	240	90	24	5	1		
7	2520	1680	630	168	35	6	1	
8	20160	13440	5040	1344	280	48	7	1

Table 1: Computer observation for $M(n,k)$

Proposition 2.1. For $k < n$ we have

$$M(n,k) = \sum_{i \leq n-1} i \binom{i-1}{k-1} (n-k-1)!.$$

Proof. Given a permutation π of length n, the element $n-k$ must be to the left of $n-k+1$. Let $\pi_{i+1} = n-k+1$,

$$\pi = \pi_1 \pi_2 \cdots \pi_i (n-k+1) \pi_{i+2} \pi_{i+3} \cdots \pi_n.$$

We can place the element $n-k$ in i possible positions to the left of $n-k+1$. The elements $n, n-1, \ldots, n-k+2$, in decreasing order, must all be to the left of $n-k+1$ and we can choose which of the remaining $i-1$ positions will contain them in $\binom{i-1}{k-1}$ ways. At this point, none of the remaining elements $1, 2, \ldots, n-k-1$ can affect $\Lambda(\pi)$ and can therefore be in any order, a total of $(n-k-1)!$ ways. For a given position $i+1$ of the element $n-k+1$, there are $i\binom{i-1}{k-1}(n-k-1)!$ permutations with a maximal decreasing run of length k. The sum of all possible positions of $n-k+1$ gives the number of permutations of length n with a maximal decreasing run of length k. □

Note that the binomial coefficient gives 0 whenever $n-k+1$ is placed in one of the first k positions and therefore the sum only needs an upper bound on i.

The only permutation in $\mathcal{S}_{n,n}$ is $n(n-1)\cdots 1$ and therefore $M(n,n) = 1$ for every $n \geq 1$. The empty permutation has no elements in decreasing order and is therefore an element of $\mathcal{S}_{0,0}$. For every $n, k > 1$ we have $\mathcal{S}_{n,0} = \mathcal{S}_{0,k} = \emptyset$ and for $k > n$ we have $\mathcal{S}_{n,k} = \emptyset$.

Proposition 2.2. For $k < n$ we have $M(n,k) = k\frac{n!}{(k+1)!}$.

Proof. For each positioning of $1, 2, \dots, n-k-1$ in a permutation π, there are k positions left that can contain $n-k$ since it can be anywhere but last (out of the remaining positions). The rest of the elements come in decreasing order. Now there are $\frac{n!}{(k+1)!}$ different positionings of $1, 2, \dots, n-k-1$ and therefore we have $M(n,k) = k\frac{n!}{(k+1)!}$. $\square$

From the two different ways of counting the elements of $\mathcal{S}_{n,k}$ we derive a combinatorial proof of the following identity.

Corollary 2.1. For $k < n$ we have

$$\sum_{i \leq n-1} i\binom{i-1}{k-1} = \frac{k \cdot n!}{(n-k-1)!(k+1)!}$$

We can derive a recursion from Proposition 2.1 or 2.2 but we will provide a combinatorial proof. Define γ to be a set-valued map from a permutation $\pi = \pi_1\pi_2\cdots\pi_{n-1}$ to a set of n permutations of length n such that

$$\gamma(\pi) = \{1(\pi_1+1)\cdots(\pi_{n-1}+1), (\pi_1+1)1\cdots(\pi_{n-1}+1), \dots, (\pi_1+1)\cdots(\pi_{n-1}+1)1\}$$

Lemma 2.1. For $k < n-1$ we have

i)

$$\bigcup_{\pi \in \mathcal{S}_{n-1,k}} \gamma(\pi) = \mathcal{S}_{n,k},$$

ii) For all $\pi_1, \pi_2 \in \mathcal{S}_{n-1,k}$ such that $\pi_1 \neq \pi_2$ we have $\gamma(\pi_1) \cap \gamma(\pi_2) = \emptyset$.

Proof.

i) Since $k < n-1$, there exists an element $n-k-1 > 0$ that cuts the maximal decreasing run for each permutation in $\mathcal{S}_{n,k}$. Raising the elements by one and then adding one somewhere in the permutation will not affect the maximal decreasing run since $n-k > 1$ cuts the maximal decreasing run after the elements have been raised. Therefore for every $\pi \in \mathcal{S}_{n-1,k}$ the set $\gamma(\pi)$ only contains permutations from $\mathcal{S}_{n,k}$, that is

$$\bigcup_{\pi \in \mathcal{S}_{n-1,k}} \gamma(\pi) \subseteq \mathcal{S}_{n,k}.$$

Now let $\pi \in \mathcal{S}_{n,k}$. Since $k < n-1$, then $n-k > 1$ and removing one from the permutation and then lowering the elements by one has no effect on the maximal decreasing run and the newly formed permutation $\pi' \in \mathcal{S}_{n-1,k}$ and $\pi \in \gamma(\pi')$, that is

$$\mathcal{S}_{n,k} \subseteq \bigcup_{\pi \in \mathcal{S}_{n-1,k}} \gamma(\pi).$$

ii) Let $\pi_1, \pi_2 \in \mathcal{S}_{n-1,k}$ such that $\pi_1 \neq \pi_2$. An element that would be in both $\gamma(\pi_1)$ and $\gamma(\pi_2)$ would have to be equal after removing 1 and being lowered by one, that is $\pi_1 = \pi_2$. Therefore the sets are disjoint. $\square$

Proposition 2.3. For $k < n-1$ we have $M(n,k) = nM(n-1,k)$.

Proof. By lemma 2.1 we get

$$M(n,k) = |\mathcal{S}_{n,k}| = \left|\bigcup_{\pi \in \mathcal{S}_{n-1,k}} \gamma(\pi)\right| = \sum_{\pi \in \mathcal{S}_{n-1,k}} |\gamma(\pi)| = n\,|\mathcal{S}_{n-1,k}| = nM(n-1,k)$$

since $|\gamma(\pi)| = n$ for all $\pi \in \mathcal{S}_{n-1,k}$. $\square$

3 Dyck paths with k returns

Definition 3.1. A lattice path in $\mathbb{Z}^2$ from $(0,0)$, with only $(1,1)$- and $(1,-1)$-steps, that never goes below the x-axis and ends with a return to the x-axis is called a *Dyck path*.

Let $\mathcal{D}_n$ be the set of Dyck paths from $(0,0)$ to $(2n,0)$ and say that $(1,1)$ is an up step, $(1,-1)$ a down step. We can view Dyck paths as a binary string on $\{u,d\}$ where u is an up step and d is a down step.

Example 3.1. The Dyck path $uduuuduuududdduddudd d \in \mathcal{D}_{10}$ is shown in the figure 2.

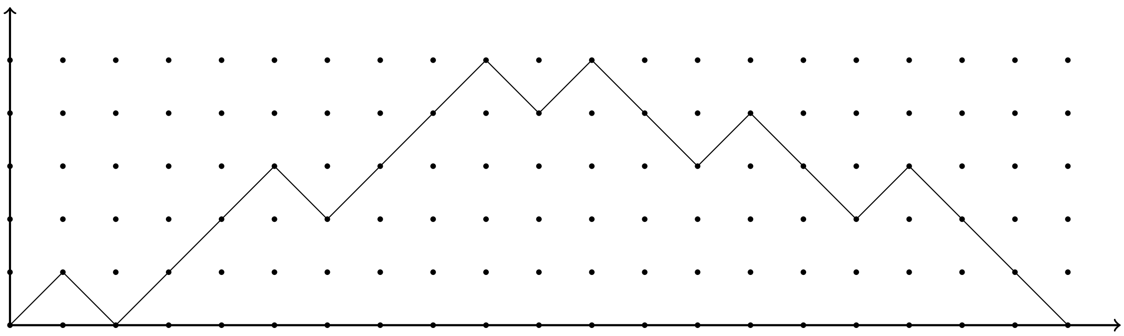

Figure 2: An example of a Dyck path

Definition 3.2. For a permutation π on A, we say that a permutation π' on B is a *sub-permutation* of π if $B \subseteq A$ and the elements of π' are in the same order as in π.

For example, the permutations 164 and 324 are sub-permutations of the permutation 165324.

For a fixed pattern σ of length 3 the pattern avoiding permutations $\mathcal{S}_n(\sigma)$ are equinumerous with $\mathcal{D}_n$. Simion and Schmidt provided the first bijection between the sets of 123- and 231-avoiding permutations in [5]. Trivial symmetry maps connect the others and Krattenthaler provided a bijection to Dyck paths in [6]. In this paper we will not use any particular bijection but will assume that for any sub-permutation of consecutive elements $a, a+1, \ldots, b-1, b$ where $1 < a \le b \le n$, the mapping will lower the elements, map them and then raise them again. We assume the same in other parts of our bijections. For example, the sub-permutation 43 will be lowered to 21, mapped and then raised by 2.

The symmetry maps are the *reverse*, *inverse* and *complement* maps. For a permutation $\pi = \pi_1\pi_2\cdots\pi_n$ we have

$$\operatorname{rev}(\pi) = \pi_n\pi_{n-1}\cdots\pi_1 \text{ and } \operatorname{com}(\pi) = (n+1-\pi_1)(n+1-\pi_2)\cdots(n+1-\pi_n).$$

Table 2 shows the symmetry map from A to B.

$A \backslash B$	$\mathcal{S}_n(123)$	$\mathcal{S}_n(132)$	$\mathcal{S}_n(213)$	$\mathcal{S}_n(231)$	$\mathcal{S}_n(312)$	$\mathcal{S}_n(321)$
$\mathcal{S}_n(123)$						rev
$\mathcal{S}_n(132)$			$\operatorname{rev}\circ\operatorname{com}$	rev	com	
$\mathcal{S}_n(213)$		$\operatorname{com}\circ\operatorname{rev}$		com	rev	
$\mathcal{S}_n(231)$		rev	com		$\operatorname{com}\circ\operatorname{rev}$	
$\mathcal{S}_n(312)$		com	rev	$\operatorname{rev}\circ\operatorname{com}$		
$\mathcal{S}_n(321)$	rev					

Table 2: Symmetry maps

Let $\mathcal{D}^{\text{ret}}_{n,k}$ be the set of Dyck paths of length $2n$ with k returns to the x-axis and $\mathcal{D}^{\text{des}}_{n,k}$ the set of Dyck paths of length $2n$ with the first descent of length k. We divide the three letter patterns (excluding 321) avoiding permutation into two classes depending on which set of Dyck paths they are in bijection with.

Every permutation in $\mathcal{S}_{n,k}(123)$ can be written as $\pi = A_1 n A_2 (n-1) \cdots A_k (n-k+1) A_{k+1}$ where A_i is a sub-permutation of a permutation on $[n-k]$. Every element of A_i must be larger then every element of A_{i+1} for $i \leq k-1$ or they would be an occurrence of 123 with $n-k+1$. Every A_i, $i \leq k$, needs to be in decreasing order or they would be an occurrence of 123 with $n-k+1$. None of the elements $1, 2, \ldots, n-k-1$ can be to the left of $n-k$ or they would be an occurrence of 123 with $n-k+1$, that is, $n-k$ is the leftmost element in the leftmost nonempty A_i, for some $i \leq k$.

Example 3.2. The permutation $75(10)3928164 \in \mathcal{S}_{10,3}(123)$ has $A_1 = 75$, $A_2 = 3$, $A_3 = 2$ and $A_4 = 164$.

Similarly every permutation in $\mathcal{S}_{n,k}(213)$ can be written as $\pi = A_1 n A_2 (n-1) \cdots A_k (n-k+1) A_{k+1}$ where A_i is a sub-permutation of a permutation on $[n-k]$. Every element of A_i must be smaller then every element of A_{i+1} for $i \leq k-1$ or they would be an occurrence of 213 with $n-k+1$. Every A_i, $i \leq k$, needs to be in increasing order or they would be an occurrence of 213 with $n-k+1$. None of the elements $1, 2, \ldots, n-k-1$ can be between $n-k$ and $n-k+1$ or they would be an occurrence of 213, that is, $n-k$ is the rightmost element in the rightmost nonempty A_i, for some $i \leq k$.

Example 3.3. The permutation $13(10)4697852 \in \mathcal{S}_{10,3}(213)$ has $A_1 = 13$, $A_2 = 46$, $A_3 = 7$ and $A_4 = 52$.

Similarly every permutation in $\mathcal{S}_{n,k}(231)$ can be written as $\pi = A_1 n A_2 (n-1) \cdots A_k (n-k+1) A_{k+1}$ where A_i is a sub-permutation of a permutation on $[n-k]$. If any element in A_i is larger then any element in A_{i+1} then those elements would be an occurrence of 231 with $n+1-i$. If A_{k+1} is not empty, it must contain $n-k$ or $\Lambda(\pi) > k$. Therefore $\pi = A_1 n A_2 (n-1) \cdots A_k (n-k+1)$ where every element of A_{i+1} is larger then all elements of A_i.

Example 3.4. The permutation $132(10)549678 \in \mathcal{S}_{10,3}(231)$ has $A_1 = 132$, $A_2 = 54$ and $A_3 = 67$.

$n\backslash k$	1	2	3	4	5	6	7	8
1	1							
2	1	1						
3	2	2	1					
4	5	5	3	1				
5	14	14	9	4	1			
6	42	42	28	14	5	1		
7	132	132	90	48	20	6	1	
8	429	429	297	165	75	27	7	1

Table 3: Computer observation for $M_{123}(n,k)$, $M_{213}(n,k)$, $M_{231}(n,k)$ and $\left|\mathcal{D}^{\mathrm{ret}}_{n,k}\right|$

3.1 Cardinality and recursion

Proposition 3.1. For all $n \geq 1$ we have the following recursion,

$$M_{231}(n,k) = M_{231}(n-1,k-1) + M_{231}(n,k+1)$$

Proof. Define $r_1 : \mathcal{S}_{n-1,k-1}(231) \cup \mathcal{S}_{n,k+1}(231) \to \mathcal{S}_{n,k}(231)$ in the following way. Let $\pi_1 = a_1 a_2 \cdots a_{n-1} \in \mathcal{S}_{n-1,k-1}(231)$, $\pi_2 = b_1 b_2 \cdots b_n = B_1 n B_2 (n-1) \cdots B_{k+1} (n-k) \in \mathcal{S}_{n,k+1}(231)$ and $b_i = n$. Then

$$\begin{aligned} r_1(\pi_1) &= n a_1 a_2 \cdots a_{n-1}, \\ r_1(\pi_2) &= i B_1 (b_{i+1}+1)(b_{i+2}+1) \cdots (b_n+1). \end{aligned}$$

Example 3.5. For $(n,k) = (8,3)$ and $\pi = 32184765 \in \mathcal{S}_{n,k+1}(231)$ we have

$$r_1(\pi) = 4321(4+1)(7+1)(6+1)(5+1) = 43215876.$$

The permutation $r_1(\pi_1)$ is in $\mathcal{S}_{n,k}(231)$. Now iB_1 is a 231-avoiding permutation on $\{1, 2, \ldots, i\}$ and $r_1(\pi_2)$ is therefore permutation in $\mathcal{S}_{n,k}(231)$. That is, the mapping maps the elements in $\mathcal{S}_{n-1,k-1}(231) \cup \mathcal{S}_{n,k+1}(231)$ to $\mathcal{S}_{n,k}(231)$. Let us consider three cases for the value of k to prove that the mapping is bijective.

i) Let $k = n$, then $\mathcal{S}_{n,k+1}(231)$ is empty. The only permutation in $\mathcal{S}_{n,n}(231)$ is $n(n-1)\cdots 1$ and the only permutation in $\mathcal{S}_{n-1,n-1}(231)$ is $(n-1)(n-2)\cdots 1$ which r_1 maps to the one in $\mathcal{S}_{n,n}(231)$.

ii) Let $k = 1$, then $\mathcal{S}_{n-1,k-1}(231)$ is empty. We will show that the inverse of r_1 is the mapping $r_1^{-1} : \mathcal{S}_{n,1}(231) \to \mathcal{S}_{n,2}(231)$ such that for $\pi' = \pi_1'\pi_2'\cdots\pi_n' \in \mathcal{S}_{n,2}(231)$ we have

$$r_1^{-1}(\pi') = \pi_2'\pi_3'\cdots\pi_{\pi_1'}'n(\pi_{\pi_1'+1}'-1)(\pi_{\pi_1'+2}'-1)\cdots(\pi_n'-1).$$

Let $\pi = \pi_1\pi_2\cdots\pi_n \in \mathcal{S}_{n,1}(231)$ where $\pi_i = n$, then

$$\begin{aligned} r_1(r_1^{-1}(\pi')) &= r_1(\pi_2'\pi_3'\cdots\pi_{\pi_1'}'n(\pi_{\pi_1'+1}'-1)(\pi_{\pi_1'+2}'-1)\cdots(\pi_n'-1)) \\ &= \pi_1'\pi_2'\cdots\pi_{\pi_1'}'\pi_{\pi_1'+1}'\cdots\pi_n' = \pi'. \end{aligned}$$

and

$$\begin{aligned} r_1^{-1}(r_1(\pi)) &= r_1^{-1}(i\pi_1\pi_2\cdots\pi_{i-1}(\pi_{i+1}+1)(\pi_{i+2}+1)\cdots(\pi_n+1)) \\ &= \pi_1\pi_2\cdots\pi_{i-1}n\pi_{i+1}\pi_{i+2}\cdots\pi_n = \pi \end{aligned}$$

Therefore r_1 is invertible for $k = 1$.

iii) Let $1 < k < n$. For every permutation in $\mathcal{S}_{n,k+1}(231)$, it must end with $n-k$ and the elements between $n-k$ and $n-k+1$ must be larger then the ones between $n-k+1$ and $n-k+2$ and so on and the smallest elements are to the left of n. If n is in position i, then i will be to the left of a 231-avoiding permutation of $\{1, 2, \ldots, i-1\}$ that avoids 231 with i when moved to the first position. The other elements are increased by one and that will not affect the 231-avoidance.

We will show that the inverse of r_1 is the mapping $r_1^{-1} : \mathcal{S}_{n,k}(231) \to \mathcal{S}_{n-1,k-1}(231) \cup \mathcal{S}_{n,k+1}(231)$ such that for $\pi' = \pi_1'\pi_2'\cdots\pi_n' \in \mathcal{S}_{n,k}(231)$ we have

$$r_1^{-1}(\pi') = \begin{cases} \pi_2'\pi_3'\cdots\pi_n' & \text{if } \pi_1' = n \\ \pi_2'\pi_3'\cdots\pi_{\pi_1'}'n(\pi_{\pi_1'+1}'-1)(\pi_{\pi_1'+2}'-1)\cdots(\pi_n'-1) & \text{if } \pi_1' \neq n. \end{cases}$$

Let $\pi = \pi_1\pi_2\cdots\pi_{n-1} \in \mathcal{S}_{n-1,k-1}(231)$ and $\pi_1' = n$, then

$$r_1(r_1^{-1}(\pi')) = r_1(\pi_2'\pi_3'\cdots\pi_n') = n\pi_2'\pi_3'\cdots\pi_n' = \pi'$$

and

$$r_1^{-1}(r_1(\pi)) = r_1^{-1}(n\pi_1\pi_2\cdots\pi_{n-1}) = \pi_1\pi_2\cdots\pi_{n-1} = \pi.$$

Let $\pi = \pi_1\pi_2\cdots\pi_n \in \mathcal{S}_{n,k+1}(231)$ where $\pi_i = n$ and $\pi_1' \neq n$, then

$$\begin{aligned} r_1(r_1^{-1}(\pi')) &= r_1(\pi_2'\pi_3'\cdots\pi_{\pi_1'}'n(\pi_{\pi_1'+1}'-1)(\pi_{\pi_1'+2}'-1)\cdots(\pi_n'-1)) \\ &= \pi_1'\pi_2'\cdots\pi_{\pi_1'}'\pi_{\pi_1+1}'\cdots\pi_n' = \pi' \end{aligned}$$

and

$$\begin{aligned} r_1^{-1}(r_1(\pi)) &= r_1^{-1}(i\pi_1\pi_2\cdots\pi_{i-1}(\pi_{i+1}+1)(\pi_{i+2}+1)\cdots(\pi_n+1)) \\ &= \pi_1\pi_2\cdots\pi_{i-1}n\pi_{i+1}\pi_{i+2}\cdots\pi_n = \pi. \end{aligned}$$

Therefore r_1 is invertible for $1 < k < n$.

Now the mapping is bijective for all cases of k and hence the recursion holds for all $n \geq 1$. $\square$

Proposition 3.2. For $n, k \geq 1$ we have

$$M_{231}(n,k) = \binom{2n-k-1}{n-k}\frac{k}{n}.$$

Proof. Let $f(x,y) = \binom{2x-y-1}{x-y}\frac{y}{x}$ and we have

$$
\begin{aligned}
f(n-1,k-1)+f(n,k+1) &= \binom{2n-k-2}{n-k}\frac{k-1}{n-1} + \binom{2n-k-2}{n-k-1}\frac{k+1}{n} \\
=&\frac{(2n-k-2)!}{(n-2)!(n-k)!}\frac{k-1}{n-1} + \frac{(2n-k-2)!}{(n-1)!(n-k-1)!}\frac{k+1}{n} = \frac{(2n-k-2)!}{n!(n-k-1)!}\left(\frac{n(k-1)}{n-k}+k+1\right) \\
=&\frac{(2n-k-2)!}{n!(n-k-1)!}\frac{n(k-1)+(n-k)(k+1)}{n-k} = \frac{(2n-k-2)!}{n!(n-k-1)!}\frac{k(2n-k-1)}{n-k} \\
=&\frac{(2n-k-1)!}{(n-1)!(n-k)!}\frac{k}{n} = \binom{2n-k-1}{n-k}\frac{k}{n} = f(n,k).
\end{aligned}
$$

Since $f(1,1)=1$ and $f(n,k)=0$ for $k<1$ and $k>n$, $f(n,k)$ satisfies the same recursion as $M_{231}(n,k)$ with the same initial value and are therefore equal. $\square$

3.2 Bijections

3.2.1 A bijection from $\mathcal{S}_{n,k}(231)$ to $\mathcal{D}^{\mathrm{ret}}_{n,k}$

Let ω_1 be any bijection from $\mathcal{S}_n(231)$ to $\mathcal{D}_n$ and define $\varphi_1 : \mathcal{S}_{n,k}(231) \to \mathcal{D}^{\mathrm{ret}}_{n,k}$ such that, for $\pi = A_1 n A_2 (n-1)\cdots A_k(n-k+1) \in \mathcal{S}_{n,k}(231)$, we have

$$\varphi_1(\pi) = u\omega_1(A_1)du\omega_1(A_2)d\cdots u\omega_1(A_k)d.$$

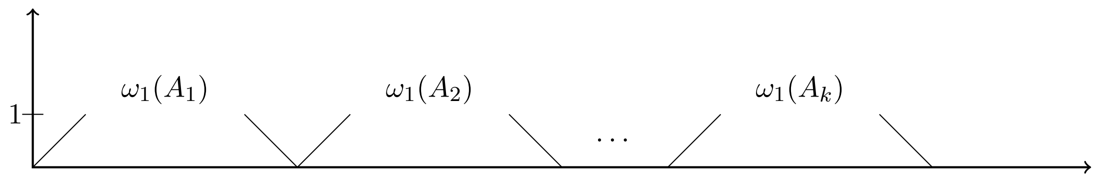

Figure 3: The mapping $\varphi_1(\pi)$

Example 3.6. For $21837645 \in \mathcal{S}_{8,4}(231)$ we have

$$\varphi_1(21837645) = u\omega_1(21)du\omega_1(3)du\omega_1(\epsilon)du\omega_1(4)d = u\omega_1(21)duudduduudd.$$

If ω_1 is the composition of Krattenthaler's map and the reverse map, then

$$\varphi_1(21837645) = u\omega_1(21)duudduduudd = uuuddduudduduudd$$

Since Dyck paths cannot go below the x-axis, then $\varphi_1(\pi)$ has exactly k returns to the x-axis and the mapping maps permutations from $\mathcal{S}_{n,k}(231)$ to Dyck paths from $\mathcal{D}^{\mathrm{ret}}_{n,k}$.

Proposition 3.3. The mapping φ_1 is bijective.

Proof. Let $D \in \mathcal{D}^{\mathrm{ret}}_{n,k}$, then D consists of k independent Dyck paths. By removing the first up step and last down step in each of them we still have k independent Dyck paths (some possibly empty). We will show that the inverse of φ_1 is the mapping $\varphi_1^{-1} : \mathcal{D}^{\mathrm{ret}}_{n,k} \to \mathcal{S}_{n,k}(231)$ such that for $D = uD_1duD_2d\cdots uD_kd \in \mathcal{D}^{\mathrm{ret}}_{n,k}$ we have

$$\varphi_1^{-1}(D) = \omega_1^{-1}(D_1)n\omega_1^{-1}(D_2)(n-1)\cdots\omega_1^{-1}(D_k)(n-k+1).$$

Let $\pi = A_1nA_2(n-1)\cdots A_k(n-k+1) \in \mathcal{S}_{n,k}(231)$, then

$$
\begin{aligned}
\varphi_1(\varphi_1^{-1}(D)) &= \varphi_1(\omega_1^{-1}(D_1)n\omega_1^{-1}(D_2)(n-1)\cdots\omega_1^{-1}(D_k)(n-k+1)) \\
&= u\omega_1(\omega_1^{-1}(D_1))du\omega_1(\omega_1^{-1}(D_2))d\cdots u\omega_1(\omega_1^{-1}(D_k))d \\
&= uD_1duD_2d\cdots uD_kd = D.
\end{aligned}
$$

and

$$
\begin{aligned}
\varphi_1^{-1}(\varphi_1(\pi)) &= \varphi_1^{-1}(u\omega_1(A_1)du\omega_1(A_2)d\cdots u\omega_1(A_k)d) \\
&= \omega_1^{-1}(\omega_1(A_1))n\omega_1^{-1}(\omega_1(A_2))(n-1)\cdots\omega_1^{-1}(\omega_1(A_k))(n-k+1) \\
&= A_1nA_2(n-1)\cdots A_k(n-k+1) = \pi.
\end{aligned}
$$

Therefore φ_1 is invertible and hence bijective. $\square$

3.2.2 A bijection from $\mathcal{S}_{n,k}(213)$ to $\mathcal{S}_{n,k}(231)$

Let $\pi = A_1 n A_2 (n-1) \cdots A_k (n-k+1) A_{k+1} \in \mathcal{S}_{n,k}(213)$ and let

$$\alpha_0 = 0, \alpha_1 = \max A_1, \alpha_2 = \max A_2, \ldots, \alpha_k = \max A_k,$$

where we let $\alpha_i = \alpha_{i-1}$ if $A_i = \epsilon$. Let $A^i = \delta_{i_1}\delta_{i_2}\cdots\delta_{i_r}$ be the substring of elements in A_{k+1}, in the same order as in A_{k+1}, such that $\delta_{i_j} \in [\alpha_{i-1}+1, \alpha_i]$ for $j \in \{1, 2, \ldots, r\}$. Note that every element of A^i must be to the right of every element of A^{i+1} in A_{k+1} or $\alpha_i A^i A^{i+1}$ would have an occurrence of 213. That is, A_{k+1} is in decreasing order of intervals and we can write $A_{k+1} = A^k A^{k-1} \cdots A^1$. Let

$$g : \mathcal{S}_{n,k}(213) \to \bigcup_{\substack{j_1+j_2+\cdots+j_k=n-k \\ j_1,j_2,\ldots,j_k \geq 0}} \mathcal{S}_{j_1}(213) \times \mathcal{S}_{j_2}(213) \times \cdots \times \mathcal{S}_{j_k}(213)$$

such that, for $\pi \in \mathcal{S}_{n,k}(213)$, we have

$$g(\pi) = (g_1(\pi), g_2(\pi), \ldots, g_k(\pi)) = (A_1A^1, A_2A^2, \ldots, A_kA^k).$$

Note that for $l > 1$ and $s = \sum_{i<l} j_i$, $\mathcal{S}_{j_l}(213)$ are permutations on $\{s+1, s+2, \ldots, s+j_i\}$.

Example 3.7. For $198367542 \in \mathcal{S}_{9,3}(213)$ where $\alpha_1 = 1$, $\alpha_2 = 1$ and $\alpha_3 = 6$, $A_1 = 1$, $A_2 = \epsilon$, $A_3 = 36$, $A^1 = \epsilon$, $A^2 = \epsilon$, $A^3 = 542$ and the intervals are $[1,1]$, $[2,1] = \emptyset$ and $[2,6]$ we have $g(198367542) = (1, \epsilon, 36542)$.

Example 3.8. For $125(12)7(10)(11)98634 \in \mathcal{S}_{12,2}(213)$ where $\alpha_1 = 5$, $\alpha_2 = 10$, $A_1 = 125$, $A_2 = 7(10)$, $A^1 = 34$, $A^2 = 986$ and the intervals are $[1,5]$ and $[6,10]$ we have $g(125(12)7(10)(11)98634) = (12534, 7(10)986)$.

Since the elements of A_i and A^i avoid 213 in π, they avoid 213 (amongst each other) when A^i is added to the right of A_i. Therefore the mapping maps a permutation from $\mathcal{S}_{n,k}(213)$ to $\bigcup \mathcal{S}_{j_1}(213) \times \mathcal{S}_{j_2}(213) \times \cdots \times \mathcal{S}_{j_k}(213)$.

Lemma 3.1. The mapping g is bijective.

Proof. We will show that the inverse of g is the mapping $g^{-1} : \bigcup \mathcal{S}_{j_1}(213) \times \mathcal{S}_{j_2}(213) \times \cdots \times \mathcal{S}_{j_k}(213) \to \mathcal{S}_{n,k}(213)$ such that for $P = (A'_1B'_1, A'_2B'_2, \ldots, A'_kB'_k) \in \bigcup \mathcal{S}_{j_1}(213) \times \mathcal{S}_{j_2}(213) \times \cdots \times \mathcal{S}_{j_k}(213)$ where A_i is the string from the first element to the largest and B_i the remaining elements for $i = 1, 2, \ldots, k$ we have

$$g^{-1}(P) = A'_1 n A'_2 (n-1) \cdots A'_k (n-k+1) B'_k B'_{k-1} \cdots B'_1.$$

Let $\pi = A_1 n A_2 (n-1) \cdots A_k (n-k+1) A^k A^{k-1} \cdots A^1 \in \mathcal{S}_{n,k}(213)$, then

$$\begin{aligned} g(g^{-1}(P)) &= g(A'_1 n A'_2 (n-1) \cdots A'_k (n-k+1) B'_k B'_{k-1} \cdots B'_1) \\ &= (A'_1B'_1, A'_2B'_2, \ldots, A'_kB'_k) = P \end{aligned}$$

and

$$\begin{aligned} g^{-1}(g(\pi)) &= g^{-1}((A_1A^1, A_2A^2, \ldots, A_kA^k)) \\ &= A_1 n A_2 (n-1) \cdots A_k (n-k+1) A^k A^{k-1} \cdots A^1 = \pi. \end{aligned}$$

Therefore g is invertible and hence bijective. □

Let ω_2 be any bijection from $\mathcal{S}_n(213)$ to $\mathcal{S}_n(231)$ and define $\varphi_2 : \mathcal{S}_{n,k}(213) \to \mathcal{S}_{n,k}(231)$ such that, for $\pi = A_1 n A_2 (n-1) \cdots A_k (n-k+1) A_{k+1} \in \mathcal{S}_{n,k}(213)$, we have

$$\varphi_2(\pi) = \omega_2(g_1(\pi)) n \omega_2(g_2(\pi))(n-1) \cdots \omega_2(g_k(\pi))(n-k+1).$$

Example 3.9. For the same permutation as in Example 3.7, we have

$$\varphi_2(\pi) = \omega_2(1)9\omega_2(\epsilon)8\omega_2(36542)7 = 198\omega_2(36542)7.$$

If $\omega_2 = \mathrm{rev} \circ (\mathrm{com} \circ \mathrm{rev})$ then

$$\varphi_2(\pi) = 198\omega_2(36542)7 = 198523467.$$

Since every element of $g_i(\pi)$ is smaller then every element of $g_{i+1}(\pi)$, the same goes for $\omega_2(g_i(\pi))$ and $\omega_2(g_{i+1}(\pi))$ and since every permutation $g_i(\pi)$ is 213-avoiding, every permutation $\omega_2(g_i(\pi))$ is 231-avoiding. That is, the mapping maps a permutation from $\mathcal{S}_{n,k}(213)$ to a permutation in $\mathcal{S}_{n,k}(231)$.

Proposition 3.4. The mapping φ_2 is bijective.

Proof. We will show that the inverse of φ_2 is the mapping $\varphi_2^{-1} : \mathcal{S}_{n,k}(231) \to \mathcal{S}_{n,k}(213)$ such that for $\pi' = A_1'nA_2'(n-1)\cdots A_k'(n-k+1) \in \mathcal{S}_{n,k}(231)$ we have

$$\varphi_2^{-1}(\pi') = g^{-1}(\omega_2^{-1}(A_1'), \omega_2^{-1}(A_2'), \ldots, \omega_2^{-1}(A_k')).$$

Let $\pi = A_1nA_2(n-1)\cdots A_k(n-k+1)A_{k+1} \in \mathcal{S}_{n,k}(213)$, then

$$\begin{aligned}
\varphi_2(\varphi_2^{-1}(\pi')) &= \varphi_2(g^{-1}(\omega_2^{-1}(A_1'), \omega_2^{-1}(A_2'), \ldots, \omega_2^{-1}(A_k'))) \\
&= \varphi_2(g^{-1}(\alpha_1'\beta_1', \alpha_2'\beta_2', \ldots, \alpha_k'\beta_k')) \\
&= \varphi_2(\alpha_1'n\alpha_2'(n-1)\cdots\alpha_k'(n-k+1)\beta_k'\beta_{k-1}'\cdots\beta_1') \\
&= \omega_2(\alpha_1'\beta_1')n\omega_2(\alpha_2'\beta_2')(n-1)\cdots\omega_2(\alpha_k'\beta_k')(n-k+1) \\
&= A_1'nA_2'(n-1)\cdots A_k'(n-k+1) = \pi'
\end{aligned}$$

and

$$\begin{aligned}
\varphi_2^{-1}(\varphi_2(\pi)) &= \varphi_2^{-1}(\omega_2(g_1(\pi))n\omega_2(g_2(\pi))(n-1)\cdots\omega_2(g_k(\pi))(n-k+1)) \\
&= g^{-1}(\omega_2^{-1}(\omega_2(g_1(\pi))), \omega_2^{-1}(\omega_2(g_2(\pi))), \ldots, \omega_2^{-1}(\omega_2(g_k(\pi)))) \\
&= g^{-1}(g_1(\pi), g_2(\pi), \ldots, g_k(\pi)) = g^{-1}(g(\pi)) = \pi
\end{aligned}$$

where α_i' is the string from the first element to the largest and β_i' the remaining elements in $\omega_2^{-1}(A_i)$ for $i = 1, 2, \ldots, k$. Therefore φ_2 is invertible and hence bijective. ☐

3.2.3 A bijection from $\mathcal{S}_{n,k}(123)$ to $\mathcal{S}_{n,k}(231)$

For this mapping we will first find a recursion within $\mathcal{S}_{n,k}(123)$ in a similar way we did for $\mathcal{S}_{n,k}(231)$ and then construct a recursive bijection between the two sets.

Define $r_2 : \mathcal{S}_{n-1,k-1}(123) \cup \mathcal{S}_{n,k+1}(123) \to \mathcal{S}_{n,k}(123)$ in the following way. Let $\pi_1 = a_1a_2\cdots a_{n-1} \in \mathcal{S}_{n-1,k-1}(123)$ and $\pi_2 = B_1nB_2(n-1)\cdots B_k(n-k+1)B_{k+1}(n-k)B_{k+2} \in \mathcal{S}_{n,k+1}(123)$. Then $r_2(\pi_1) = na_1a_2\cdots a_{n-1}$ and $r_2(\pi_2)$ is the permutation where the elements of $B_1B_2\cdots B_{k+1}$ have been rotated by one to the left amongst each other if B_{k+1} is not empty and then $n-k$ is moved to left end of the permutation.

Example 3.10. For $(n, k) = (8, 3)$ and $\pi = 84763251 \in \mathcal{S}_{n,k+1}(123)$ we have

$$84763251 \longrightarrow 83762451 \longrightarrow 58376241$$

and $r_2(\pi) = 58376241$

Example 3.11. For $(n, k) = (8, 3)$ and $\pi = 48731652 \in \mathcal{S}_{n,k+1}(123)$ we have $r_2(\pi) = 54873162$.

Proposition 3.5. The mapping r_2 is bijective.

Proof. The permutation $r_2(\pi_1)$ is a permutation of $\mathcal{S}_{n,k}(123)$. Since $n-k$ is larger then every element of B_i for $i \in [k+2]$, it avoids 123 with any of them and also with the k larger elements when moved to the left end of the permutation. Now the left most element of $B_1B_2\cdots B_{k+1}$ is $n-k-1$ (if it exists) and since $n-k$ did avoid 123 with any of the elements in $B_1B_2\cdots B_{k+1}$, then $n-k-1$ will too in the same place. Therefore $r_2(\pi_2)$ is an element of $\mathcal{S}_{n,k}(123)$.

i) Let $k = n$, then $\mathcal{S}_{n,k+1}(123)$ is empty. The only permutation in $\mathcal{S}_{n-1,k-1}(123)$ is $(n-1)(n-2)\cdots 1$ which r_2 maps to $n(n-1)\cdots 1$, the only permutation in $\mathcal{S}_{n,k}(123)$.

ii) Let $k = 1$, then $\mathcal{S}_{n-1,k-1}(123)$ is empty. Let

$$\pi_1 = A_1 n A_2 (n-1) A_3, \pi_2 = B_1 n B_2 (n-1) B_3 \in \mathcal{S}_{n,k+1}$$

such that $\pi_1 \neq \pi_2$. If n is in different positions in π_1 and π_2 then n remains in different positions after the mapping since the map always moves n one to the right. Lets assume that the position of n is shared in both. If $A_1A_2 \neq B_1B_2$, then they will remain different after the rotation so we can assume they are equal. Then A_3 and B_3 are of equal length and since $r_2(\pi_1)$ ends with A_3 and $r_2(\pi_2)$ with B_3, we have $r_2(\pi_1) \neq r_2(\pi_2)$ if $A_3 \neq B_3$. Let $\pi = \alpha_1\alpha_2\cdots\alpha_r n \beta_1\beta_2\cdots\beta_s \in \mathcal{S}_{n,1}(123)$. The element $n-1$ is always the left most of π'. If $n-2$ is to the left of n, then

$$\pi = \alpha_2\alpha_3\cdots\alpha_r n (n-1)\beta_1\beta_2\cdots\beta_s \in \mathcal{S}_{n,2}$$

and $r_2(\pi) = \pi'$. If $n-2$ is to the right of n and $\alpha_2\alpha_3\cdots\alpha_r \neq \epsilon$, then

$$\pi = \beta_i\alpha_2\alpha_3\cdots\alpha_{r-1} n \alpha_r \beta_1\beta_2\cdots\beta_{i-1}(n-1)\beta_{i+1}\beta_{i+2}\cdots\beta_s \in \mathcal{S}_{n,2}(123)$$

where $\beta_i = n-2$ and $r_2(\pi) = \pi'$. If $n-2$ is to the right of n and $\alpha_2\alpha_3\cdots\alpha_r = \epsilon$, then

$$\pi = n\beta_i\beta_1\beta_2\cdots\beta_{i-1}(n-1)\beta_{i+1}\beta_{i+2}\cdots\beta_s \in \mathcal{S}_{n,2}(123)$$

where $\beta_i = n-2$ and $r_2(\pi) = \pi'$.

iii) Let $1 < k < n$ and $\pi_1, \pi_2 \in \mathcal{S}_{n-1,k-1}(123) \cup \mathcal{S}_{n,k+1}(123)$ such that $\pi_1 \neq \pi_2$. If both permutations are from $\mathcal{S}_{n-1,k-1}(123)$ then $r_2(\pi_1)$ and $r_2(\pi_2)$ are different. If $\pi_1 \in \mathcal{S}_{n-1,k-1}(123)$ and $\pi_2 \in \mathcal{S}_{n,k+1}$ then the left most element of $r_2(\pi_1)$ is n while $n-k$ in $r_2(\pi_2)$. Lets assume that $\pi_1, \pi_2 \in \mathcal{S}_{n,k+1}(123)$ and let

$$\begin{aligned}\pi_1 &= A_1 n A_2 (n-1)\cdots A_k (n-k+1) A_{k+1} (n-k) A_{k+2},\\ \pi_2 &= B_1 n B_2 (n-1)\cdots B_k (n-k+1) B_{k+1} (n-k) B_{k+2}.\end{aligned}$$

If $n, n-1, \ldots, n-k+1$ are in different positions in π_1 and π_2, they will remains so after the mapping since the mapping increases their position by one. Therefore we assume that A_i and B_i are of equal length for $i = 1, 2, \ldots, n-k+1$. If $A_1A_2\cdots A_{k+1} \neq B_1B_2\cdots B_{k+1}$, then $r_2(\pi_1) \neq r_2(\pi_2)$ since $n-k-1$ is the left most element of both $A_1A_2\cdots A_{k+1}$ and $B_1B_2\cdots B_{k+1}$. Then A_{k+2} and B_{k+2} are of equal length and since the permutations $r_2(\pi_1)$ and $r_2(\pi_2)$ ends with A_{k+2} and B_{k+2} respectively, we have $r_2(\pi_1) \neq r_2(\pi_2)$ if $A_{k+2} \neq B_{k+2}$. Let $\pi' \in \mathcal{S}_{n,k}(123)$. All permutations in $\mathcal{S}_{n,k}(123)$ begin with either n or $n-k$. If $\pi' = na_1a_2\cdots a_{n-1}$, then $\pi = a_1a_2\cdots a_{n-1} \in \mathcal{S}_{n-1,k-1}(123)$ and $r_2(\pi) = \pi'$. If π begins with $n-k$, let $\pi' = A_1 n A_2 (n-1)\cdots A_k (n-k+1) A_{k+1}$. If $n-k-1$ is to the left of $n-k+1$ then

$$\pi = a_2a_3\cdots a_r n A_2 n-1\cdots A_k(n-k+1)(n-k)A_{k+1} \in \mathcal{S}_{n,k+1}(123)$$

where $A_1 = (n-k)a_2a_3\cdots a_r$ and $r_2(\pi) = \pi'$. If $n-k+1$ is to the right of $n-k+1$, let $A_{k+1} = \beta_1\beta_2\cdots\beta_s$ and $\beta_i = n-k-1$. Then

$$\pi = A_1' n A_2'(n-1)\cdots A_k'(n-k+1)\beta_1'\beta_2'\cdots\beta_i'(n-k)\beta_{i+1}\beta_{i+2}\cdots\beta_s \in \mathcal{S}_{n,k+1}(123)$$

where $A_1'A_2'\cdots A_k'\beta_1'\beta_2'\cdots\beta_i'$ is the substring $A_1A_2\cdots A_k(\beta_1\beta_2\cdots\beta_i)$ rotated one (A_1 is withouth $n-k$ in this rotation) to the right and $r_2(\pi) = \pi'$.

Now the mapping is bijective for all cases of k and hence the recursion holds for all $n > 1$. □

For every permutation in $\mathcal{S}_{n,k}(123)$ there is a unique path $p_{r_2} : \mathcal{S}_{n,k}(123) \to (\mathbb{N}\times\mathbb{N})^{\mathbb{N}}$, from $(1,1)$ to (n,k), via the mapping r_2. The same goes for permutations in $\mathcal{S}_{n,k}(231)$, via r_1, and we can construct a bijection $\varphi_3 : \mathcal{S}_{n,k}(123) \to \mathcal{S}_{n,k}(231)$ such that $\varphi_3(\pi) = \pi'$ if and only if $p_{r_2}(\pi) = p_{r_1}(\pi')$.

Example 3.12. For the permutation $43251 \in \mathcal{S}_{5,1}(123)$ we have $\varphi_3(43251) = 32145 \in \mathcal{S}_{5,3}(231)$ since

$$p_{r_2}(43251) = ((1,1),(2,2),(2,1),(3,2),(4,3),(5,3),(5,2),(5,1),(5,4)) = p_{r_1}(32145).$$

In terms of permutations, the paths are

$$\begin{aligned}&1 \to 21 \to 12 \to 312 \to 4312 \to 54312 \to 25431 \to 32541 \to 43251,\\ &1 \to 21 \to 12 \to 312 \to 4312 \to 54312 \to 15423 \to 21534 \to 32145\end{aligned}$$

for 123- and 231-avoidance respectively.

4 Dyck paths with the first descent of length k

Every permutation in $\mathcal{S}_{n,k}(132)$ can be written as $\pi = A_1 n A_2 (n-1) \cdots A_k (n-k+1) A_{k+1}$ where A_i is a sub-permutation of a permutation on $[n-k]$. If any of the sub-permutations $A_1, A_2, \ldots, A_{k-1}$ are nonempty, then they are an occurrence of 132 with $n-k+2$ and $n-k+1$. Every element of A_k must be larger then every element of A_{k+1} or they would be an occurrence of 132 with $n-k+1$. If $A_k = \emptyset$, then $A_{k+1} = \emptyset$ or $n-k$ would be in A_{k+1} and $\Lambda(\pi) > k$. For convenience, we form the permutation again with A_1 and A_2 instead of A_k and A_{k+1} and $\pi = n(n-1)\cdots(n-k+2)A_1(n-k+1)A_2$.

Example 4.1. The permutation $(10)987456231 \in \mathcal{S}_{10,5}(132)$ has $A_1 = 45$ and $A_2 = 231$.

Similarly every permutation in $\mathcal{S}_{n,k}(312)$ can be written as $\pi = A_1 n A_2 (n-1) \cdots A_k (n-k+1) A_{k+1}$ where A_i is a sub-permutation of a permutation on $[n-k]$. If $A_i \neq \emptyset$ for $1 < i < k+1$, then $nA_i(n-k+1)$ has an occurrence of 312 and therefore they must be empty. For convenience, we form the permutation again with A_2 instead of A_{k+1}, $\pi = A_1 n(n-1)\cdots(n-k+1)A_2$. If A_1 is empty then A_2 must also be empty since it can't contain $n-k$. The sub-permutation A_2 must avoid the pattern 12 or $n-k+1$ and A_2 would have an occurrence of 312, that is A_2 is in decreasing order.

Example 4.2. The permutation $3476(10)98521 \in \mathcal{S}_{10,3}(312)$ has $A_1 = 3476$, $A_2 = 521$ and $t = 5$.

$n \backslash k$	1	2	3	4	5	6	7	8
1	1							
2	1	1						
3	3	1	1					
4	9	3	1	1				
5	28	9	3	1	1			
6	90	28	9	3	1	1		
7	297	90	28	9	3	1	1	
8	1001	297	90	28	9	3	1	1

Table 4: Computer observation for $M_{132}(n,k)$, $M_{312}(n,k)$ and $\left|\mathcal{D}^{\text{des}}_{n,k}\right|$

4.1 Cardinality and recursion

Proposition 4.1. For $k < n$ we have $M_{132}(n,k) = C_{n-k+1} - C_{n-k}$ and $M_{132}(n,n) = 1$.

Proof. Lets assume $k < n$. Every permutation must have $n-k+2, n-k+3, \ldots, n$ in decreasing order in the first $k-1$ positions since every element smaller then any of them would be an occurrence of 132 along with one of them and $n-k+1$. Now there are only $n-k+1$ positions left and they only have to avoid 132 amongst each other (since they avoid 132 with any of the other elements) and $n-k$ must be to the left of $n-k+1$. After having placed $n-k+2, n-k+3, \ldots, n$ in the first $k-1$ positions, there are C_{n-k+1} possible ways to place the rest of the elements such that they avoid 132. Now we subtract the number of ways $n-k+1$ can be to the left of $n-k$. If $n-k+1$ is to the left of $n-k$ then it can only be in the first position (out of the remaining $n-k+1$ positions) or the element in that position would be an occurrence of 132 with $n-k+1$ and $n-k$. Hence there are C_{n-k} such permutations and $M_{132}(n,k) = C_{n-k+1} - C_{n-k}$. If $k = n$ there is only one permutation and it avoids 132. □

We can derive a recursion from Proposition 4.1 but we will provide a combinatorial proof.

Proposition 4.2. For $n \geq 1$ we have $M_{132}(n,k) = M_{132}(n-1,k-1)$.

Proof. Define $r : \mathcal{S}_{n-1,k-1}(132) \to \mathcal{S}_{n,k}(132)$ such that for $\pi = \pi_1\pi_2\cdots\pi_{n-1} \in \mathcal{S}_{n-1,k-1}(132)$ we have $r(\pi) = n\pi_1\pi_2\cdots\pi_{n-1}$. This maps the permutations in $\mathcal{S}_{n-1,k-1}(132)$ to $\mathcal{S}_{n,k}(132)$ given that $n \geq 1$. Let $\pi_1, \pi_2 \in \mathcal{S}_{n-1,k-1}(132)$ such that $\pi_1 \neq \pi_2$, then $r(\pi_1)$ and $r(\pi_2)$ differ in at least one element other then the first one. Let $\pi' \in \mathcal{S}_{n,k}(132)$. We know that every permutation in $\mathcal{S}_{n,k}(132)$ must have the elements $n-k+2, n-k+3, \ldots, n$ in decreasing order in the first $k-1$ positions and therefore the permutation π, constructed by removing n from π' is in $\mathcal{S}_{n-1,k-1}(132)$ and $r(\pi) = \pi'$. Therefore the mapping is bijective. □

4.2 Bijections

4.2.1 A bijection from $\mathcal{S}_{n,k}(132)$ to $\mathcal{D}_{n,k}^{\text{des}}$

Let ω_4 be any bijection from $\mathcal{S}_{n,k}(132)$ to $\mathcal{D}_n$ and let $\Omega(D)$ be the string of first consecutive up steps in a Dyck path D and $\overline{\Omega}(D)$ the remaining string. Let u^i (d^i) be i consecutive up (down) steps and define $\varphi_4 : \mathcal{S}_{n,k}(132) \to \mathcal{D}_{n,k}^{\text{des}}$ such that, for $\pi = n(n-1)\cdots(n-k+2)A_1(n-k+1)A_2 \in \mathcal{S}_{n,k}(132)$, we have

$$\varphi_4(\pi) = \Omega(\omega_4(A_2))u^k d^k \omega_4(A_1)\overline{\Omega}(\omega_4(A_2)).$$

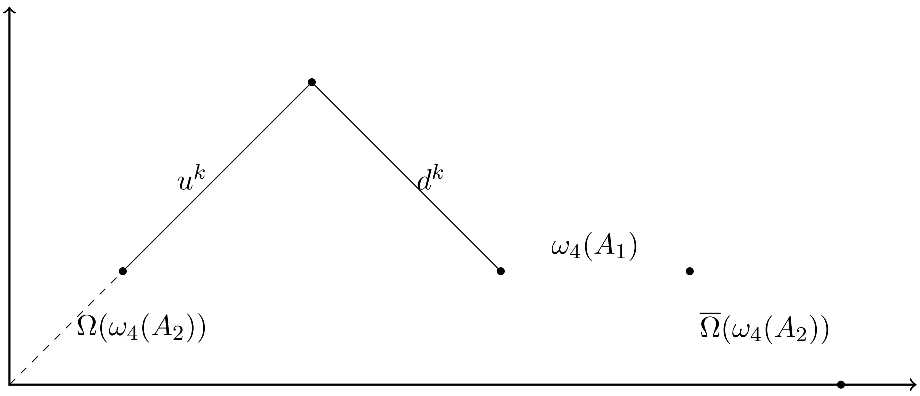

Figure 4: The mapping $\varphi_4(\pi)$

Example 4.3. For $7634251 \in \mathcal{S}_{7,3}(132)$ we have

$$\varphi_4(7634251) = \Omega(\omega_4(1))u^3d^3\omega_4(342)\overline{\Omega}(\omega_4(1)) = \Omega(ud)u^3d^3\omega_4(342)\overline{\Omega}(ud) = uu^3d^3\omega_4(342)d.$$

If ω_4 is Krattenthaler's bijection then

$$\varphi_4(7634251) = uu^3d^3\omega_4(342)d = uu^3d^3uuddudd$$

Since every nonempty Dyck path starts with an up step, then the first descent of $\varphi_4(\pi)$ is of length k (if $\omega_4(A_1) = \epsilon$, then $\omega_4(A_2) = \epsilon$ and that still holds) and the mapping maps permutations from $\mathcal{S}_{n,k}(132)$ to Dyck paths of length $2n$ with the first descent of length k.

Proposition 4.3. The mapping φ_4 is bijective.

Proof. We will show that the inverse of φ_4 is the mapping $\varphi_4^{-1} : \mathcal{D}_{n,k}^{\text{des}} \to \mathcal{S}_{n,k}(132)$ such that for $D = \Omega(D_2)u^kd^kD_1\overline{\Omega}(D_2) \in \mathcal{D}_{n,k}^{\text{des}}$ we have

$$\varphi_4^{-1}(D) = n(n-1)\cdots(n-k+2)\omega_4^{-1}(D_1)(n-k+1)\omega_4^{-1}(D_2).$$

Let $\pi = n(n-1)\cdots(n-k+2)A_1(n-k+1)A_2 \in \mathcal{S}_{n,k}(132)$, then

$$\begin{aligned}\varphi_4(\varphi_4^{-1}(D)) &= \varphi_4(n(n-1)\cdots(n-k+2)\omega_4^{-1}(D_1)(n-k+1)\omega_4^{-1}(D_2))\\ &= \Omega(\omega_4(\omega_4^{-1}(D_2)))u^kd^k\omega_4(\omega_4^{-1}(D_1))\overline{\Omega}(\omega_4(\omega_4^{-1}(D_2)))\\ &= \Omega(D_2)u^kd^kD_1\overline{\Omega}(D_2) = D\end{aligned}$$

and

$$\begin{aligned}\varphi_4^{-1}(\varphi_4(\pi)) &= \varphi_4^{-1}(\Omega(\omega_4(A_2))u^kd^k\omega_4(A_1)\overline{\Omega}(\omega_4(A_2)))\\ &= n(n-1)\cdots(n-k+2)\omega_4^{-1}(\omega_4(A_1))(n-k+1)\omega_4^{-1}(\omega_4(A_2))\\ &= n(n-1)\cdots(n-k+2)A_1(n-k+1)A_2 = \pi.\end{aligned}$$

Therefore φ_4 is invertible and hence bijective. $\square$

4.2.2 A bijection from $\mathcal{S}_{n,k}(312)$ to $\mathcal{S}_{n,k}(132)$

Let $\pi = A_1 n(n-1)\cdots(n-k+1)A_2 \in \mathcal{S}_{n,k}(312)$, $A_2 = c_1c_2\cdots c_s$ and $t = \max\{c_1, c_2, \ldots, c_s\}$, then every element x of A_1 such that $x < t$ must be to the left of every element y of A_1 such that $y > t$ or they would be an occurrence of 312 with t. Let $A_1 = a_1a_2\cdots a_u b_1b_2\cdots b_v$ where $a_i < t$ and $b_j > t$ for $i = 1, 2, \ldots, u$ and $j = 1, 2, \ldots, v$. Let

$$h : \mathcal{S}_{n,k}(312) \to \bigcup_{\substack{j_1+j_2=n-k \\ j_1>0, j_2\geq 0}} \mathcal{S}_{j_1}(312) \times \mathcal{S}_{j_2}(312)$$

for $k < n$ such that, for

$$\pi = A_1 n(n-1)\cdots(n-k+1)A_2 = a_1a_2\cdots a_u b_1b_2\cdots b_v n\cdots(n-k+1)c_1c_2\cdots c_s \in \mathcal{S}_{n,k}(312),$$

we have

$$h(\pi) = (h_1(\pi), h_2(\pi)) = (b_1b_1\cdots b_v, a_1a_2\cdots a_u A_2).$$

For $n = k$ we let $h : \mathcal{S}_{n,n}(312) \to \{(\epsilon, \epsilon)\}$. Note that the $\mathcal{S}_{j_1}(312)$ are permutations on $\{j_2+1, j_2+2, \ldots, n-k\}$.

Example 4.4. For $25487631 \in \mathcal{S}_{8,3}(312)$ we have $h(25487631) = (54, 231)$.

Since A_2 is in decreasing order, it will avoid 312 by moving $a_1a_2\cdots a_u$ to the left of it. Hence h maps $\pi \in \mathcal{S}_{n,k}(312)$ to $\bigcup \mathcal{S}_{j_1}(312) \times \mathcal{S}_{j_2}(312)$.

Lemma 4.1. The mapping h is bijective.

Proof. We will show that the inverse of h is the mapping $h^{-1} : \bigcup \mathcal{S}_{j_1}(312) \times \mathcal{S}_{j_2}(312) \to \mathcal{S}_{n,k}(312)$ such that for

$$P = (b'_1b'_2\cdots b'_m, a'_1a'_2\cdots a'_l c'_1c'_2\cdots c'_w)$$

where $c'_1 = \max\{a'_1, a'_2, \ldots, a'_l, c'_1, c'_2, \ldots, c'_w\}$ we have

$$h^{-1}(P) = a'_1a'_2\cdots a'_l b'_1b'_2\cdots b'_m n(n-1)\cdots(n-k+1)c'_1c'_2\cdots c'_w.$$

Let $\pi = a_1a_2\cdots a_u b_1b_2\cdots b_v n(n-1)\cdots(n-k+1)c_1c_2\cdots c_s$ such that $\max\{a_1, a_2, \ldots, a_u\} < t$ and $\max\{b_1, b_2, \ldots, b_v\} > t$ for $t = \max\{c_1, c_2, \ldots, c_s\}$, then

$$\begin{aligned} h(h^{-1}(P)) &= h(a'_1a'_2\cdots a'_l b'_1b'_2\cdots b'_m n(n-1)\cdots(n-k+1)c'_1c'_2\cdots c'_w) \\ &= (b'_1b'_2\cdots b'_m, a'_1a'_2\cdots a'_l c'_1c'_2\cdots c'_w) = P \end{aligned}$$

and

$$\begin{aligned} h^{-1}(h(\pi)) &= h^{-1}((b_1b_1\cdots b_v, a_1a_2\cdots a_u c_1c_2\cdots c_s)) \\ &= a_1a_2\cdots a_u b_1b_2\cdots b_v n(n-1)\cdots(n-k+1)c_1c_2\cdots c_s = \pi. \end{aligned}$$

Therefore h is invertivle and hence bijective. □

Let ω_5 be any bijection from $\mathcal{S}_n(312)$ to $\mathcal{S}_n(132)$ and define $\varphi_5 : \mathcal{S}_{n,k}(312) \to \mathcal{S}_{n,k}(132)$ such that, for $\pi = A_1 n(n-1)\cdots(n-k+1)A_2 \in \mathcal{S}_{n,k}(312)$, we have

$$\varphi_5(\pi) = n(n-1)\cdots(n-k+2)\omega_5(h_1(\pi))(n-k+1)\omega_5(h_2(\pi))$$

Example 4.5. For the same permutation as in example 4.4, we have

$$\varphi_5(\pi) = 87\omega_5(54)6\omega_5(231).$$

If $\omega_5 = \text{com}$ then

$$\varphi_5(\pi) = 87\omega_5(54)6\omega_5(231) = 87456213.$$

Since the elements of $\omega_5(h_1(\pi))$ are all larger then the elements of $\omega_5(h_2(\pi))$, they avoid 132 and the mapping maps a 312-avoiding permutation to a 132-avoiding one, preserving the maximal decreasing run.

Proposition 4.4. The mapping φ_5 is bijective.

Proof. We will show that the inverse of φ_5 is the mapping $\varphi_5^{-1} : \mathcal{S}_{n,k}(132) \to \mathcal{S}_{n,k}(312)$ such that for $\pi' = n(n-1)\cdots(n-k+2)A_1'(n-k+1)A_2' \in \mathcal{S}_{n,k}(132)$ we have

$$\varphi_5^{-1}(\pi') = h^{-1}(\omega_5^{-1}(A_1'), \omega_5^{-1}(A_2')).$$

Let $\pi = A_1 n(n-1)\cdots(n-k+1)A_2 \in \mathcal{S}_{n,k}(312)$, then

$$\begin{aligned}
\varphi_5(\varphi_5^{-1}(\pi')) &= \varphi_5(h^{-1}(\omega_5^{-1}(A_1'), \omega_5^{-1}(A_2'))) \\
&= \varphi_5(h^{-1}(\omega_5^{-1}(A_1'), \alpha\beta)) \\
&= \varphi_5(\alpha\omega_5^{-1}(A_1')n(n-1)\cdots(n-k+1)\beta) \\
&= n(n-1)\cdots(n-k+2)\omega_5(\omega_5^{-1}(A_1'))(n-k+1)\omega_5(\alpha\beta) \\
&= n(n-1)\cdots(n-k+2)A_1'(n-k+1)A_2' = \pi'
\end{aligned}$$

and

$$\begin{aligned}
\varphi_5^{-1}(\varphi_5(\pi)) &= \varphi_5^{-1}(n(n-1)\cdots(n-k+2)\omega_5(h_1(\pi))(n-k+1)\omega_5(h_2(\pi))) \\
&= h^{-1}(\omega_5^{-1}(\omega_5(h_1(\pi))), \omega_5^{-1}(\omega_5(h_2(\pi)))) \\
&= h^{-1}(h_1(\pi), h_2(\pi)) = h^{-1}(h(\pi)) = \pi
\end{aligned}$$

where β is the string from (and including) the largest element to the last and α the rest of the elements in $\omega_5^{-1}(A_2')$. Therefore φ_5 is invertible and hence bijective. $\square$

Appendix - Listings

Listing 1 - Counters and generators

```
# input: permutation p
# output: Maximal decreasing run
def MDR(p):
    nextMDR = len(p)
    for i in p:
        if (i == nextMDR):
            nextMDR -= 1
    return len(p) - nextMDR

# input: length of permutations n
# output: [M(n,1),M(n,2),...,M(n,n)]
def distribution_MDR(n):
    M = [0 for i in [1..n]]
    for p in Permutations(n):
        M[MDR(p)-1] += 1
    return M

# input: length of permutations n, MDR k, type of data structure
# output: a list/dictionary with all permutations in S_(n,k)
def generate_MDR(n,data_structure="list"):
    if (data_structure == "list"):
        A = [[] for i in [1..n]]
        for p in Permutations(n):
            A[MDR(p)-1] += [p]
    if (data_structure == "dictionary"):
        A = [{} for i in [1..n]]
        for p in Permutations(n):
            A[MDR(p)-1][p] = 0
    return A

# input: length of permutations n, pattern
# output: [M_{pattern}(n,1),M_{pattern}(n,2),...,M_{pattern}(n,n)]
def distribution_avoiding_MDR(n,pattern):
    M = [0 for i in [1..n]]
    for p in Permutations(n,avoiding=pattern):
        M[MDR(p)-1] += 1
    return M

# input: length of permutation n, MDR k, pattern, type of data structure
# output: a list/dictionary with all permutations in S_{n,k}(pattern)
def generate_avoiding_MDR(n,pattern,data_structure="list"):
    if (data_structure == "list"):
        A = [[] for i in [1..n]]
        for p in Permutations(n,avoiding=pattern):
            A[MDR(p)-1] += [p]
    if (data_structure == "dictionary"):
        A = [{} for i in [1..n]]
        for p in Permutations(n,avoiding=pattern):
            A[MDR(p)-1][p] = 0
    return A

# input: Dyck path d
# output: number of returns to the x-axis
def returns(d):
    c = 0
    height = 0
    for i in d: # d is a list of binary numbers
        if (i):
            height += 1
        else:
            height -= 1
            if (height == 0):
                c += 1
    return c

# input: length of Dyck path n
# output: [|D_{n,1}^{ret}|,|D_{n,2}^{ret}|,...|D_{n,n}^{ret}|]
def distribution_return_Dyck(n):
    M = [0 for i in [1..n]]
    for d in DyckWords(n):
        M[returns(d)-1] += 1
```

```
    return M

# input: length of Dyck path n, number of returns k, type of data structure
# output: a list/dictionary of all Dyck paths in D_{n,k}^{ret}
def generate_return_Dyck(n,data_structure="list"):
    if (data_structure == "list"):
        A = [[] for i in [1..n]]
        for p in DyckWords(n):
            A[returns(p)-1] += [p]
    if (data_structure == "dictionary"):
        A = [{} for i in [1..n]]
        for p in DyckWords(n):
            A[returns(p)-1][p] = 0
    return A

# input: Dyck path d
# output: legnth of first descent
def first_descent(d):
    c = 0
    index = 0
    while (d[index] == 1):
        index += 1
        if (index * 2 == len(d)): # first half of length are up steps
            return index
    while (d[index] == 0):
        c += 1
        index += 1
    return c

# input: length of Dyck path n
# output: [|D_{n,1}^{des}|,|D_{n,2}^{des}|,...|D_{n,n}^{des}|]
def distribution_descent_Dyck(n):
    M = [0 for i in [1..n]]
    for d in DyckWords(n):
        M[first_descent(d)-1] += 1
    return M

# input: length of Dyck path n, number of returns k, type of data structure
# output: a list of all Dyck paths in D_{n,k}^{des}
def generate_descent_Dyck(n,data_structure="list"):
    if (data_structure == "list"):
        A = [[] for i in [1..n]]
        for p in DyckWords(n):
            A[first_descent(p)-1] += [p]
    if (data_structure == "dictionary"):
        A = [{} for i in [1..n]]
        for p in DyckWords(n):
            A[first_descent(p)-1][p] = 0
    return A
```

Listing 1: Counters and generators

Listing 2 - The mapping γ

```
# input: a permutation p with length > MDR - 1
# output: a list of permutations with length incremented by 1 by adding 1s
def gamma(p):
    p = [i + 1 for i in p]
    P = [[1] + [i for i in p]] # left end special case
    for i in [0..len(p)-1]: # add 1 to all possible position
        P += [[p[j] for j in [0..i]] + [1] + [p[j] for j in [i+1..len(p)-1]]]
    return P
```

Listing 2: The mapping γ

Listing 3 - The mapping r_1^{-1}

```
# input: a permutation from S_{n,k}(231)
# output: a permutation p from S_{n-1,k-1}(231) or S_{n,k+1}
def r1_inv(p,k):
    n = len(p)
    if (p[0] == n):
        return Permutation([p[i] for i in [1..n-1]])
    else:
        return Permutation([p[i] for i in [1..p[0]-1]]+[n]+[p[i]-1 for i in [p[0]..n-1]])
```

Listing 3: The mapping r_1^{-1}

Listing 4 - The mapping r_2^{-1}

```
# input: list of sub permutations L [[...],[...],...,[...]]
# output: same list with elements rotated amongst non empty inner lists
def rotator(L):
    elements = [] # list with elements, no inner lists
    for i in [0..len(L)-1]:
        if (len(L[i]) > 0):
            for j in L[i]:
                elements += [j]
    rotation = {}
    for i in [0..len(elements)-1]: # map each element to it's left neighbour (circular)
        rotation[elements[i]] = elements[(i-1+len(elements)) % len(elements)]
    for i in L: # replace domain elements with corresponding image elements
        if (len(i) > 0):
            for j in [0..len(i)-1]:
                i[j] = rotation.get(i[j])
    return L

# input: 123 avoiding permutation p, MDR k, boolean return_path (for phi_3)
# output: see cases in comments, but always a 123 avoiding permutation
def r2_inv(p,k,return_path=false): # inverse chosen for phi_3, direction dosent matter
    n = len(p)
    if (p[0] == n): # returns a permutation with MDR & length decremented by 1
        if (return_path):
            return (0,Permutation([p[i] for i in [1..n-1]]))
        else:
            return Permutation([p[i] for i in [1..n-1]])
    else: # returns a permutation of same length with MDR incremented by 1
        cut_point = -1 # n-k-1
        last_run_element = -1 # n-k+1
        for i in [0..n-1]: # find n-k-1 and n-k+1
            if p[n-1-i] == n-k+1:
                last_run_element = n-1-i
            if p[n-1-i] == n-k-1:
                cut_point = n-1-i
            if (cut_point >= 0 and last_run_element >= 0):
                break
        if (cut_point < last_run_element): # more trivial subcase, no rotation
            return_value = [p[i] for i in [1..last_run_element]]
            return_value += [n-k] + [p[i] for i in [last_run_element+1..n-1]]
            if (return_path):
                return (1, Permutation(return_value))
            else:
                return Permutation(return_value)
        else:
            # A_{k+1} split in 2 parts (rotated, not rotated)
            B1 = [p[i] for i in [last_run_element+1..cut_point]]
            B2 = [p[i] for i in [cut_point+1..n-1]]
            rotated = []
            temp = []
            next_parse_sign = n
            # gather sub-permutations A_1,A_2,...,A_k to a list
            for i in [1..last_run_element]:
                if (p[i] == next_parse_sign):
                    next_parse_sign -= 1
                    rotated += [temp]
                    temp = []
                else:
                    temp += [p[i]]
            rotated += [B1] # [[A_1],[A_2],...,[A_k], [part of A_{k+1}]]
            rotated = rotator(rotated)
            return_value = []
            run_value = n
             # alternate rotated sub-permutations and run elements
            for i in [0..len(rotated)-1]:
                return_value += rotated[i] + [run_value-i]
            return_value += B2
            if (return_path):
                return (1, Permutation(return_value))
            else:
                return Permutation(return_value)
```

Listing 4: The mapping r_2^{-1}

Listing 5 - The mapping r

```
# input: a 132 avoiding permutation p
# output: a 132 avoiding permutation with length & MDR incremented by 1
def r(p,k):
    return Permutation([len(p)+1] + [i for i in p]) # n added to left end
```

Listing 5: The mapping r

Listing 6 - The mapping φ_1

```
# input: 132 avoiding permutation p, index i
# output: number of larger elements to the right of i
def larger_to_right(p,i):
    c = 0
    for j in [i+1..len(p)-1]: # iterate right of index
        if (p[j] > p[i]):
            c += 1
        if (c == len(p) - p[i]): # max possible value of c
            return c
    return c

# input: 132 avoiding permutation p
# output: Dyck path
def Krattenthaler(p):
    """
    Krattenthaler's algorithm:
        - input: permutation p, outout: Dyck path d
        - Read through p from left to right. For element i, count number of
          larger elements to it's right. Suppose there are x elements larger
          than i to it's right. If x is larger or equal than current height
          of d (as it stands now) then add (to right end of d) a as many up
          steps as needed to d so that we can take a last step (for i) from
          height x+1 to x. If x is less than current height than as many
          down steps are added as needed (can be none) to achieve the same.
    """
    d = []
    curr_height = 0
    for i in [0..len(p)-1]:
        r = larger_to_right(p,i)
        while (curr_height <= r):
            curr_height += 1
            d += [1]
        while (curr_height > r):
            curr_height -= 1
            d += [0]
    return d

# input: 231 avoiding permutation p
# output: Dyck path with p's MDR returns
def phi_1(p,k):
    d = [] # stores return value (Dyck path)
    sub = [] # stores sub permutation for helper maps
    nextMDR = len(p) # next run element
    for i in [0..len(p)-1]:
        if (p[i] != nextMDR): # add non-run elements to sub permutation
            sub += [p[i]]
        else: # (p[i] == nextMDR)
            nextMDR -= 1 # update next run element
            if (len(sub) == 0): # empty sub permutation case
                d += [1,0]
            else:
                # reverse: 231->132, also handles lovering elements to 1,2,...,x
                temp = Permutation([j - (min(sub)-1) for j in sub]).reverse()
                d += ([1] + Krattenthaler(temp) + [0])
            sub = []
    return DyckWord(d)
```

Listing 6: The mapping φ_1

Listing 7 - The mapping φ_2

```
# input: 213 avoiding permutation p, MDR k
# output: [A_1,A_2,...,A_(k+1)] (see structure of 213's)
def parser(p,k):
    n = len(p)
    parsed = [] # hols all sub permutations, return value
```

```
    temp = [] # holds each sub permutation
    nextMDR = n # next run value
    for i in [0..n-1]:
        if (p[i] == nextMDR): # if in {n,n-1,...,n-k+1}
            parsed += [temp]
            temp = []
            if (nextMDR == n-k+1): # end case
                parsed += [[p[j] for j in [i+1..n-1]]]
                break
            nextMDR -= 1
        else:
            temp += [p[i]]
    return parsed

# input [A_1,A_2,...,A_(k+1)] (parsed)
# output: [alpha_0,alpha_1,...,alphak] (max values of intervals, see text)
def alphas(parsed):
    alph = [0] # alpha_0 = 0 for all parsings
    for i in [1..len(parsed)-1]:
        if (len(parsed[i-1]) == 0): # if empty sub permutation
            alph += [alph[i-1]] # alpha_i copies alpha_{i-1}
        else:
            alph += [max(parsed[i-1])] # largests element of sub permutation
    return alph

# input: sub permutation A_(k+1), alpha-intervals
# output: [A^1, A^2,...,A^k] (reverse order as in A_{k+1})
def parser_k_1(sub, alph):
    parts = [[] for i in [1..len(alph)-1]] # parsin of A_{k+1} into A^k...A^1
    temp = [] # stores sub permutation of a sub permutation
    interval = 0 # which interval
    while (interval < len(alph) - 1):
        if (alph[interval + 1] != alph[interval]): # if interval is not empty
            for i in sub: # iterate through A_{k+1}, store all values on interval
                if (alph[interval] < i and alph[interval +  1] > i):
                    temp += [i]
            parts[interval] += temp # add to list of lists
        interval += 1
        temp = []
    return parts

# input: 213 avoiding permutations p, MDR k
# output: g of the permutation (see text)
def g(p,k):
    parsedLeft = parser(p, k) # A_1, A_2, ..., A_k (and A_{k+1})
    # A^1, A^2, ..., A_k
    parsedRight = parser_k_1(parsedLeft[len(parsedLeft)-1], alphas(parsedLeft))
    # [[A_1A^1],[A_2A^2],...,[A_kA^k]]
    permutations = [parsedLeft[i] + parsedRight[i] for i in [0..k-1]]
    return permutations

# input: 213 avoiding permutation p
# output: 231 avoiding permutation
def omega2(p):
    if (len(p) == 0):
        return []
    # decrement to a permutation on {1,2,...,|p|}
    m = min(p)
    p = Permutation([i - (m-1) for i in p])
    p = p.reverse().complement().reverse() # composition of symmetry maps
    p = [i + (m-1) for i in p] # increment back to original values
    return p

# input: 213 avoiding permutation p, MDR k
# output: 231 avoiding permutation with k MDR
def phi_2(p,k):
    n = len(p)
    parts = g(p,k)
    return_permutation = []
    for i in [0..len(parts)-1]:
        return_permutation += omega2(parts[i]) + [n-i]
    return Permutation(return_permutation)
```

Listing 7: The mapping φ_2

Listing 8 - The mapping φ_3

```
# input: a permutation p from S_{n-1,k-1}(231)
# output: a permutation from S_{n,k}(231)
def first_case_r1(p):
    return Permutation([len(p) + 1] + [i for i in p]) # add n to left end

# input: a permutation p from S_{n,k+1}(231)
# output: a permutation from S_{n,k}(231)
def second_case_r1(p):
    temp = []
    index = 1
    while (1):
        if (p[index-1] == len(p)): # find index of n
            break
        else:
            temp += [p[index-1]] # elements up to n
            index += 1
    return [index] + temp + [p[i-1] + 1 for i in [index+1..len(p)]]

# input: a 123 avoiding permutation
# output: a 231 avoiding permutation
def phi_3(p,k,return_path=false):
    permutation_path = [p]
    path = [] # stores unique path from (n,k) to (1,1)
    while (p != [1]): # map p (and store in self) until we reach (1,1)
        A = r2_inv(p,MDR(p),true) # A = [path value, image permutation]
        path = [A[0]] + path # add to front to get path from (1,1) to (n,k)
        p = A[1]
        permutation_path += [p]
    for i in path: # follow the same path for r1
        if (i):
            p = second_case_r1(p)
        else:
            p = first_case_r1(p)
        permutation_path += [p]
    if (return_path):
        return [Permutation(p),permutation_path]
    else:
        return Permutation(p)
```

Listing 8: The mapping φ_3

Listing 9 - The mapping φ_4

```
# input: a 132 avoiding permutation, MDR k
# output: a Dyck path with first descent of length MDR
def phi_4(p,k):
    n = len(p)
    if (n == k): # sepcial case, [u,u,...,u,d,d,...,d]
        return DyckWord([1 for i in [1..k]] + [0 for i in [1..k]])
    A = [[],[]] # [A_1,A_2]
    index = 0
    for i in [k-1..n-1]: # iterate through permutation (can skip a few)
        if (p[i] == n-k+1): # n-k+1
            index += 1 # start adding to A[1]
        else:
            A[index] += [p[i]] # non run elements
    if (len(A[0]) > 0): # if A_1 is not empty
        m = min(A[0])
        if (m != 1): # decrement to permutation on {1,2,...,|A_1|}
            A[0] = [i-(m-1) for i in A[0]]
    D = [Krattenthaler(A[1]),Krattenthaler(A[0])] # notice order of A's
    d = []
    index = 0
    if (len(D[0]) > 0):
        while (D[0][index]): # add consecutive up steps
            d += [1]
            index += 1
        d += [1 for i in [0..k-1]] # k up steps
        d += [0 for i in [0..k-1]] # k down steps
        d += [i for i in D[1]]
        d += [D[0][i] for i in [index..len(D[0])-1]] # add remaining steps
        return DyckWord(d)
    else:
```

```
        return DyckWord([1 for i in [1..k]] + [0 for i in [1..k]] + [i for i in D[1]])
```

Listing 9: The mapping φ_4

Listing 10 - The mapping φ_5

```
# input: 312 avoiding permutation p, MDR k
# output: h of that permutation (see text)
def h(p,k):
    n = len(p)
    A2 = []
    a2_max = 0
    passed = false
    for i in [k..n-1]: # iterate through p to find A_2 (can skip some)
        if (p[i] == n - k + 1): # everything after n-k+1 belongs to A_2
            passed = true
            continue
        if (passed):
            A2 += [p[i]]
            if (a2_max < p[i]): # finds max of A_2
                a2_max = p[i]
    A1 = []
    for i in [0..n-1]: # iterate through p to find A_1
        if (p[i] == n): # won't go further than n
            break
        A1 += [p[i]]
    temp = [[],[]] # stores A_1 when split by size
    for i in [0..len(A1)-1]: # iteration A1
        if (A1[i] > a2_max): # larger than max(A2) elements
            temp[1] += [A1[j] for j in [i..len(A1)-1]]
            break
        temp[0] += [A1[i]] # smaller than max(A2) elements
    return [temp[1],temp[0]+A2]

# input: 312 avoiding permutation on {a,...,b}
# output: complement of that permutation (132 avoiding)
def com(p):
    if (len(p) == 0): # special case
        return []
    m = min(p)
    p = [i-(m-1) for i in p] # decrement to permutation on {1,2,...,len(p)}
    p = Permutation(p)
    p = p.complement() # 312 -> 132
    p = [i+(m-1) for i in p] #increment to original
    return p

# input: 312 avoiding permutation
# output: 132 avoiding permutation
def phi_5(p,k):
    temp = h(p,k)
    n = len(p)
    perm = []
    for i in [0..k-2]: # add first k-1 run elements
        perm += [n-i]
    perm += com(temp[0])
    perm += [n-k+1] # add last run element
    perm += com(temp[1])
    return Permutation(perm)
```

Listing 10: The mapping φ_5

Listing 11 - Validations

```
def identity_checker(n):
    for i in [1..n]:
        P = generate_MDR(i)
        P1 = generate_avoiding_MDR(i,[2,3,1])
        P2 = generate_avoiding_MDR(i,[2,1,3])
        P3 = generate_avoiding_MDR(i,[1,2,3])
        P4 = generate_avoiding_MDR(i,[3,1,2])
        P5 = generate_avoiding_MDR(i,[1,3,2])
        D1 = generate_return_Dyck(i)
        D2 = generate_descent_Dyck(i)
        for j in [1..i]:
            A = [len(P[j-1])] # lengths of S_{i,j}
            if (j < i):
                A += [sum([x*(binomial(x-1,j-1))*factorial(i-j-1) for x in [0..i-1]])]
```

```
                A += [j * factorial(i) / (factorial(j+1))]
            else:
                A += [1]
            for x in [0..len(A)-2]: # are all values in list equal?
                if (A[x] != A[x+1]):
                    print "identiy does not hold"
            B = [len(P1[j-1]),len(P2[j-1]),len(P3[j-1]),len(D1[j-1])]
            B += [binomial(2*i - j - 1, i - j) * j / i]
            for x in [0..len(B)-2]: # are all values in list equal?
                if (B[x] != B[x+1]):
                    print "identity does not hold"
            C = [len(P4[j-1]),len(P5[j-1]),len(D2[j-1])]
            if (i > j):
                C += [catalan_number(i-j+1)-catalan_number(i-j)]
            else:
                C += [1]
            for x in [0..len(C)-2]: # are all values in list equal?
                if (C[x] != C[x+1]):
                    print "identity does not hold"
            print i, j, "DONE"

# A general purpose map checker!
# Checks if the map func is a bijection from domain to image
# Dommain must be a list of permutations, image must be a hashmap
# MDR k is needed for some function, might be unused for others
def map_checker(domain, image, func, k):
    A = {}
    for p in domain:
        p = func(p,k)
        if (A.has_key(p)): # have we mapped to this element before?
            print "Not injective!"
        A[p] = 0
        if (not image.has_key(p)): # is this element in the image?
            print "Invalid mapping!"
    if (len(A) != len(image)): # is there anything we missed?
        print "Not surjective!"
    return "DONE!"

def check_gamma(n):
    for i in [1..n]:
        X = generate_MDR(i)
        Y = generate_MDR(i+1,"dictionary")
        for j in [1..i-1]:
            A = {}
            dom = []
            for p in X[j-1]:
                dom += gamma(p)
            print i, j, map_checker(dom, Y[j-1],Permutation,j)

def check_r1_inv(n):
    Y = [generate_avoiding_MDR(i,[2,3,1],"dictionary") for i in [1..n]]
    for i in [2..n]:
        X = generate_avoiding_MDR(i,[2,3,1])
        for j in [1..i]:
            A = {}
            IM = {}
            if (j < i):
                IM.update(Y[i-1][j])
            if (j > 1):
                IM.update(Y[i-2][j-2])
            print i,j,map_checker(X[j-1],IM,r1_inv,j)

def check_r2_inv(n):
    Y = [generate_avoiding_MDR(i,[1,2,3],"dictionary") for i in [1..n]]
    for i in [2..n]:
        X = generate_avoiding_MDR(i,[1,2,3])
        for j in [1..i]:
            A = {}
            IM = {}
            if (j < i):
                IM.update(Y[i-1][j])
            if (j > 1):
                IM.update(Y[i-2][j-2])
            print i,j,map_checker(X[j-1],IM,r2_inv,j)
```

```
def check_r(n):
    X = [generate_avoiding_MDR(i,[1,3,2]) for i in [1..n]]
    Y = [generate_avoiding_MDR(i,[1,3,2],"dictionary") for i in [1..n+1]]
    for i in [1..n]:
        for j in [1..i]:
            print i,j,map_checker(X[i-1][j-1],Y[i][j],r,j)

def check_phi_1(n):
    for i in [1..n]:
        X = generate_avoiding_MDR(i,[2,3,1])
        Y = generate_return_Dyck(i,"dictionary")
        for j in [1..i]:
            print i,j,map_checker(X[j-1],Y[j-1],phi_1,j)

def check_phi_2(n):
    for i in [1..n]:
        X = generate_avoiding_MDR(i,[2,1,3])
        Y = generate_avoiding_MDR(i,[2,3,1],"dictionary")
        for j in [1..i]:
            print i,j,map_checker(X[j-1],Y[j-1],phi_2,j)

def check_phi_3(n):
    for i in [1..n]:
        X = generate_avoiding_MDR(i,[1,2,3])
        Y = generate_avoiding_MDR(i,[2,3,1],"dictionary")
        for j in [1..i]:
            print i,j,map_checker(X[j-1],Y[j-1],phi_3,j)

def check_phi_4(n):
    for i in [1..n]:
        X = generate_avoiding_MDR(i,[1,3,2])
        Y = generate_descent_Dyck(i,"dictionary")
        for j in [1..i]:
            print i,j,map_checker(X[j-1],Y[j-1],phi_4,j)

def check_phi_5(n):
    for i in [1..n]:
        X = generate_avoiding_MDR(i,[3,1,2])
        Y = generate_avoiding_MDR(i,[1,3,2],"dictionary")
        for j in [1..i]:
            print i,j,map_checker(X[j-1],Y[j-1],phi_5,j)
```

Listing 11: Validation

Listing 12 - Demonstration

```
def show_gamma(n,k):
    for p in generate_MDR(n)[k-1]:
        print p, " -> ", gamma(p)

def show_r1_inv(n,k):
    for p in generate_avoiding_MDR(n,[2,3,1])[k-1]:
        print p, " -> ", r1_inv(p,k)

def show_r2_inv(n,k):
    for p in generate_avoiding_MDR(n,[1,2,3])[k-1]:
        print p, " -> ", r2_inv(p,k)

def show_r(n,k):
    for p in generate_avoiding_MDR(n,[1,3,2])[k-1]:
        print p, " -> ", r(p,k)

# input: dyck path d
# output: list of coordinates it passes through in correct order
def Dyck_to_cords(d):
    current_cord = [0,0]
    cords = [(current_cord[0],current_cord[1])]
    for i in d:
        current_cord[0] += 1
        if (i):
            current_cord[1] += 1
        else:
            current_cord[1] -= 1
        cords += [(current_cord[0],current_cord[1])]
    return cords
```

```
# print/plot elements of domain with corresponding elements of image
def show_phi_1(n,k,show_plot=false):
    if (show_plot):
        for p in generate_avoiding_MDR(n,[2,3,1])[k-1]:
            print p
            cords = Dyck_to_cords(phi_1(p,k))
            lines = line2d(cords)
            points = point2d(cords,size=50,color='red')
            plot(lines+points).show(figsize=3)
    else:
        for p in generate_avoiding_MDR(n,[2,3,1])[k-1]:
            print p, " -> ", phi_1(p,k)

def show_phi_2(n,k):
    for p in generate_avoiding_MDR(n,[2,1,3],"list")[k-1]:
        print p, " -> ", phi_2(p,k)

# print elements of domain with corresponding elements of image
def show_phi_3(n,k,show_path=false):
    if (show_path):
        for p in generate_avoiding_MDR(n,[1,2,3])[k-1]:
            print [i for i in phi_3(p,k,true)[1]]
    else:
        for p in generate_avoiding_MDR(n,[1,2,3])[k-1]:
            print p, " -> ", phi_3(p,k)

def show_phi_4(n,k,show_plot=false):
    if (show_plot):
        for p in generate_avoiding_MDR(n,[1,3,2])[k-1]:
            print p
            cords = Dyck_to_cords(phi_4(p,k))
            lines = line2d(cords)
            points = point2d(cords,size=50,color='red')
            plot(lines+points).show(figsize=3)
    else:
        for p in generate_avoiding_MDR(n,[1,3,2])[k-1]:
            print p, " -> ", phi_4(p,k)

def show_phi_5(n,k):
    for p in generate_avoiding_MDR(n,[3,1,2])[k-1]:
        print p, " -> ", phi_5(p,k)
```

Listing 12: Demonstration

References

[1] J. West. Permutations with restricted subsequences and stack-sortable permutations. Doctoral Thesis, M.I.T., (1990).

[2] 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 Mathematics (1992) 102, 85-93.

[3] MacMahon, Percy A. (1915), *Combinatory Analysis*, London: Cambridge University Press, Volume I, Section III, Chapter V.

[4] D. E. Knuth, *The Art of Computer Programming*, Vol's. 1 and 3, Addison-Wesley, Reading, Mass. 1968, 1973.

[5] Rodica Simion and Frank W. Schmidt, *Restricted permutations*, Europ.l. Combinatorics (1985) 6, 383-406.

[6] C. Krattenthaler, *Permutations with restricted patterns and Dyck paths*, Adv. Appl. Math. 27 (2001), 510-530.