

Sorting Cayley permutations with pattern-avoiding machines*

GIULIO CERBAI

*Dipartimento di Matematica e Informatica “U. Dini”
University of Firenze
Italy
giulio.cerbai@unifi.it*

Abstract

Pattern-avoiding machines were recently introduced by Claesson, Ferrari and the current author to gain a better understanding of the classical 2-stacksort problem. In this paper we generalize these devices by allowing permutations with repeated elements, also known as Cayley permutations. The main result is a description of those patterns such that the corresponding set of sortable permutations is a class. We also show a new involution on the set of Cayley permutations, obtained by regarding a pattern-avoiding stack as an operator. Finally, we analyze two generalizations of pop-stack sorting on Cayley permutations. In both cases we describe sortable permutations in terms of pattern avoidance.

1 Introduction

The problem of sorting a permutation using a stack, together with its many variants, has been widely studied in the literature. The original version was proposed by Knuth in [22]: given an input permutation π , either *push* the next element of π into the stack or *pop* the top element of the stack, placing it into the output. The goal is to describe and enumerate sortable permutations. To sort a permutation means to produce a sorted output, i.e. the identity permutation. An elegant answer can be given in terms of pattern avoidance: a permutation is sortable if and only if it does not contain a subsequence of three elements which is order isomorphic to 231. A set of permutations that can be characterized in terms of pattern avoidance is called a *class* and the minimal excluded permutations are its *basis*. The notion of pattern

* The author is a member of the INdAM Research group GNCS; he is partially supported by INdAM-GNCS 2020 project “Combinatoria delle permutazioni, delle parole e dei grafi: algoritmi e applicazioni”.

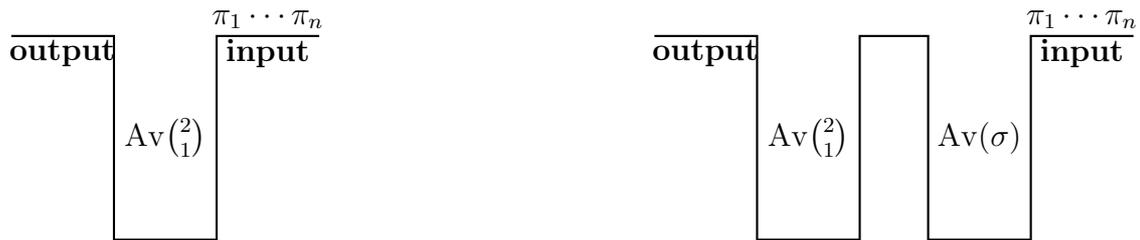


Figure 1: Sorting with one stack (on the left) and sorting with two stacks, where the first one is σ -restricted (on the right).

avoidance turns out to be a fundamental tool to approach a great variety of problems in combinatorics. We refer the reader to [8] for a more detailed survey on stack-sorting disciplines, and to [9] and [21] for an overview on patterns in permutations and words. It is easy to realize that the optimal algorithm for the classical stacksorting problem has two key properties. First, the elements in the stack are maintained in increasing order, reading from top to bottom. Moreover, the algorithm is *right-greedy*, meaning that it always performs a push operation, unless this violates the previous condition. Note that the expression “right-greedy” refers to the usual (and most natural) representation of this problem, depicted in Figure 1.

Although the classical problem is rather simple, as soon as one allows several stacks connected in series things become much harder. For example, it is known that the permutations that can be sorted using two stacks in series form a class, but in this case the basis is infinite [24], and still unknown. The enumeration of such permutations is still unknown too. In the attempt of gaining a better understanding of this device, some (simpler) variants have been considered. A *pop-stack* is a (restricted) stack where all the elements are extracted every time a pop operation is performed. Pop-stacks were introduced by Avis and Newborn [5], where the authors prove that permutations sortable through a pop-stack are the so called layered permutations. A permutation is *layered* if it avoids 231 and 312. More recently, two or more pop-stacks in series were considered in [4, 15, 25]. In his PhD thesis [28], West considered two passes through a classical stack, which is equivalent to performing a *right-greedy* algorithm on two stacks in series. In [27], Smith considered a decreasing stack followed by an increasing stack. This machine was then generalized in [12] to the case of many decreasing stacks, followed by an increasing one. Recently, the authors of [13] considered *pattern-avoiding machines*, an even more general device consisting of two stacks in series with a right-greedy procedure, where a restriction on the first stack is given in terms of pattern avoidance. More precisely, the first stack is not allowed to contain an occurrence of a forbidden pattern σ , for a fixed σ . West’s device is obtained by choosing $\sigma = 21$. The pattern $\sigma = 12$ corresponds to the device analyzed in [27], but with a right-greedy (and thus less powerful) algorithm. Pattern-avoiding machines have been discussed in [6, 7, 14, 20].

Other than imposing restrictions on devices and sorting algorithms, one can also allow a larger set of input sequences. Since the notion of pattern itself is inherently more general, it is natural to consider sorting procedures on bigger sets of strings [1,

3, 19]. Here we pursue this line of research by analyzing the behaviour of pattern-avoiding machines on permutations with repeated letters, which are known as Cayley permutations. A more formal definition of Cayley permutation will be given in Section 2, together with the necessary background and tools.

In Section 3 we generalize a result of [13] by determining for which patterns σ the words that can be sorted by the σ -machine form a class. In such cases, we also give an explicit description of the basis, which is either a singleton or consists of two patterns.

In Section 4, we regard a σ -avoiding stack as a function \mathcal{S}^σ that maps an input word into the resulting output, characterizing the patterns σ that give rise to a bijective operator. The proof of this result relies on the encoding of \mathcal{S}^σ as a labeled Dyck path. By composing \mathcal{S}^σ with the reverse operator, we obtain a new involution on the set of Cayley permutations. This involution has the nice property of preserving the multiset of entries of a Cayley permutation. It also leads to a constructive description of the set of sortable permutations.

In Section 5 we analyze two generalizations of pop-stack sorting on Cayley permutations. We call them hare pop-stack and tortoise pop-stack, in analogy with a paper by Defant and Kravitz [19]. In both cases, we characterize the set of sortable permutations in terms of pattern avoidance. A simple geometrical description allows us to enumerate the set of tortoise pop-stack sortable permutations, while the hare case is left for a future investigation.

2 Tools and Notations

Let \mathbb{N}^* be the set of strings over the alphabet $\mathbb{N} = \{1, 2, \dots\}$ of positive integers. Let $x = x_1 \cdots x_n$ and $p = p_1 \cdots p_k$ in \mathbb{N}^* , with $k \leq n$. The word x *contains* the pattern p if there are indices $i_1 < i_2 < \cdots < i_k$ such that $x_{i_1} x_{i_2} \cdots x_{i_k}$ is order isomorphic to p . Equivalently, for each pair of indices u, v , $x_{i_u} < x_{i_v}$ if and only if $p_u < p_v$ and $x_{i_u} = x_{i_v}$ if and only if $p_u = p_v$. In this case, we write $p \leq x$ and we say that $x_{i_1} x_{i_2} \cdots x_{i_k}$ is an *occurrence* of p . Otherwise, we say that x *avoids* the pattern p . This notion generalizes the usual notion of pattern involvement on permutations. For example, the string $x = 142215$ contains the pattern 2113, since the substring 4225 is order isomorphic to 2113. On the other hand, x avoids the pattern 1234. A *class* is a set of words that is closed downwards with respect to pattern involvement. A class is determined by the minimal set of words it avoids, which is called its *basis*.

Denote by \mathcal{C} the set of strings π on \mathbb{N} where each integer from 1 to $\max(\pi)$ appears at least once. Following [23], we call these strings *Cayley permutations* (they are called *normalized words* in [19], and sometimes also surjective words, Fubini words or packed words). Cayley permutations, with respect to their length, are enumerated by sequence A000670 in the OEIS [26]. For example, the only Cayley permutation of length one is the string 1, there are three Cayley permutations of length two, namely 11, 12 and 21, and thirteen Cayley permutations of length three, which are 111, 112, 121, 211, 122, 212, 221, 123, 132, 213, 231, 312, 321. Since only the relative order

of the elements is relevant for avoidance and containment, patterns live naturally in the set \mathcal{C} . More precisely, given $x \in \mathbb{N}^*$, an order-isomorphic string $\pi \in \mathcal{C}$ can be produced by suitably rescaling the elements of x , so to remove gaps. For this reason, and because we think that the most natural setting is the one where patterns and words belong to the same set, in the rest of the paper we will work on \mathcal{C} rather than on \mathbb{N}^* . We denote by $\mathcal{C}(\pi)$ the set of Cayley permutations that avoid the pattern π , for $\pi \in \mathcal{C}$; for a set of patterns $B = \{\pi_1, \dots, \pi_k\}$, the set of Cayley permutations that avoid all of the patterns π_1, \dots, π_k is denoted by $\mathcal{C}(B)$. The *reverse* of the Cayley permutation $\pi = \pi_1 \cdots \pi_k$ is $\pi^r = \pi_k \cdots \pi_1$. A *weak descent* is a pair of consecutive elements π_i, π_{i+1} such that $\pi_i \geq \pi_{i+1}$. If $\pi > \pi_{i+1}$, the pair is a (*strong*) *descent*. *Weak ascents* and (*strong*) *ascents* are defined analogously.

3 σ -machines on Cayley Permutations

The authors of [13] introduced pattern-avoiding machines on permutations. Here we generalize these devices by allowing Cayley permutations both as inputs and as forbidden patterns. Let σ be a Cayley permutation of length at least two. A σ -stack is a stack that is not allowed to contain an occurrence of the pattern σ when reading the elements from top to bottom. Before introducing σ -machines, we recall some useful results. Classical stacksort on \mathbb{N}^* has been discussed in [19]. Note that there are two possibilities when defining the analogue of the stacksort algorithm on \mathbb{N}^* . One can either allow a letter to sit on a copy of itself in the stack, or force a pop operation if the next element of the input is equal to the top element of the stack. Here we choose the former possibility, leaving the latter for future investigation. This is equivalent to regarding a classical stack as a 21-avoiding stack. The following theorem, proved in [19] for \mathbb{N}^* , applies, in particular, to Cayley permutations.

Theorem 3.1. *Let π be a Cayley permutation. Then π is sortable using a 21-stack if and only if π avoids 231.*

The term σ -machine refers to performing a right-greedy algorithm on two stacks in series: a σ -stack, followed by a 21-avoiding stack (see Figure 1). A Cayley permutation π is σ -sortable if the output of the σ -machine on input π is weakly increasing. The set of σ -sortable permutations is denoted by $\text{Sort}(\sigma)$. We use the notation $s_\sigma(\pi)$ to denote the output of the σ -stack on input π . Note that, since $s_\sigma(\pi)$ is the input of the 21-stack, Theorem 3.1 guarantees that $\pi \in \text{Sort}(\sigma)$ if and only if $s_\sigma(\pi)$ avoids 231. This fact will be used repeatedly for the rest of the paper. In [13], the authors provided a characterization of the (permutation) patterns σ such that the set of σ -sortable permutations is a class. The main goal of this section is to extend this result to Cayley permutations.

Remark 1. *Let $\sigma = \sigma_1 \cdots \sigma_k$ be a Cayley permutation. If an input Cayley permutation π avoids σ^r , then the restriction of the σ -stack is never triggered and $s_\sigma(\pi) = \pi^r$. Otherwise, the leftmost occurrence of σ results necessarily in an occurrence of $\hat{\sigma}$ in $s_\sigma(\pi)$, where $\hat{\sigma} = \sigma_2 \sigma_1 \sigma_3 \sigma_4 \cdots \sigma_k$.*

From now on, we denote by $\hat{\sigma}$ the Cayley permutation obtained from σ by interchanging σ_1 and σ_2 .

Theorem 3.2. *Let σ be a Cayley permutation. If $\hat{\sigma}$ contains 231, then $\text{Sort}(\sigma) = \mathcal{C}(132, \sigma^r)$.*

Proof. We start by proving that $\text{Sort}(\sigma) \subseteq \mathcal{C}(132, \sigma^r)$. Let $\pi \in \text{Sort}(\sigma)$. Equivalently, suppose that $s_\sigma(\pi)$ avoids 231. Suppose by contradiction that π contains σ^r . Then $s_\sigma(\pi)$ contains $\hat{\sigma}$ due to Remark 1 and $\hat{\sigma}$ contains 231 by hypothesis, which is impossible. Otherwise, if π avoids σ^r , but contains 132, then $s_\sigma(\pi) = \pi^r$ due to the same remark. Moreover π^r contains 231 by hypothesis, again a contradiction with $\pi \in \text{Sort}(\sigma)$. This proves that $\text{Sort}(\sigma) \subseteq \mathcal{C}(132, \sigma^r)$.

Conversely, suppose that π avoids both 132 and σ^r . Then $s_\sigma(\pi) = \pi^r$, which avoids $132^r = 231$ by hypothesis, therefore π is σ -sortable. This completes the proof. \square

We will show that the condition of Theorem 3.2 is also necessary for $\text{Sort}(\sigma)$ to be a class except when $\sigma = 12$.

Theorem 3.3. $\text{Sort}(12) = \mathcal{C}(213)$.

Proof. Let π be a Cayley permutation. Suppose that the element 1 appears k times in π and write $\pi = A_1 1 A_2 1 \cdots A_k 1 A_{k+1}$. It is easy to see that:

$$s_{12}(\pi) = s_{12}(A_1) s_{12}(A_2) \cdots s_{12}(A_k) s_{12}(A_{k+1}) 1 \cdots 1.$$

Indeed a copy of 1 can enter the 12-stack only if the 12-stack is either empty or it contains only other copies of 1. Finally, the element 1 cannot play the role of 2 in an occurrence of the (forbidden) pattern 12. Therefore the presence of some copies of 1 at the bottom of the 12-stack does not affect the sorting process of the block A_i , for each i .

Now, suppose that π contains an occurrence bac of 213. We prove that π is not 12-sortable by showing that $s_{12}(\pi)$ contains 231. We proceed by induction on the length of π . Write $\pi = A_1 1 A_2 1 \cdots A_k 1 A_{k+1}$ as above. Suppose that $b \in A_i$ and $c \in A_j$, for some $i \leq j$ (note that $b, c \neq 1$). If $i = j$, then A_i contains an occurrence bac of 213. Thus $s_{12}(A_i)$ contains 231 by induction, as wanted. Otherwise, let $i < j$. Then $b \in s_{12}(A_i)$ and $c \in s_{12}(A_j)$ and the elements b and c , together with any copy of 1, realize an occurrence of 231 in $s_{12}(\pi)$, as desired.

Conversely, suppose that $\pi = \pi_1 \cdots \pi_n$ is not sortable, i.e. $s_{12}(\pi)$ contains 231. We prove that π contains 213. Let bca be an occurrence of 231 in $s_{12}(\pi)$. Note that b has to precede c in π . This is due to the fact that a non-inversion in the output necessarily comes from a non-inversion in the input, since the stack is 12-avoiding. However, b is pushed out before c enters. Denote with x the next element of the input when b is extracted. Then we have $x < b$ and also $x \neq c$, since $c > b$. Finally, the triple bxc forms an occurrence of 213 in π , as desired. \square

σ	$\alpha \notin \text{Sort}(\sigma)$	$\beta \geq \alpha, \beta \in \text{Sort}(\sigma)$
11	132	3132
21	3241	35241
231	1324	361425

Figure 2: The case-by-case analysis of Theorem 3.4.

Theorem 3.4. *Let σ be a Cayley permutation and suppose $\sigma \neq 12$. If $\hat{\sigma}$ avoids 231, then $\text{Sort}(\sigma)$ is not a class.*

Proof. Let $\sigma = \sigma_1 \cdots \sigma_k$, with $k \geq 2$. We show that there are two Cayley permutations α, β such that α is contained in β , α is not σ -sortable and β is σ -sortable. This proves that $\text{Sort}(\sigma)$ is not closed downwards, as desired. Figure 2 shows an example of such α and β for patterns σ of length two and for $\sigma = 231$. Now, suppose that σ has length at least three and $\sigma \neq 231$. Then the Cayley permutation $\alpha = 132$ is not σ -sortable. Indeed, $s_\sigma(\alpha) = \alpha^r = 231$, since α avoids σ^r . Next we define the permutation β according to the following case-by-case analysis.

- Suppose that σ_1 is the strict minimum of σ , i.e. $\sigma_1 = 1$ and $\sigma_i \geq 2$ for each $i \geq 2$. Define:

$$\beta = \sigma'_k \cdots \sigma'_3 1 \sigma'_2 \sigma'_1,$$

where $\sigma'_i = \sigma_i + 1$ for each i . Note that β is a Cayley permutation and $1\sigma'_2\sigma'_1$ is an occurrence of 132 in β . We prove that β is σ -sortable by showing that $s_\sigma(\beta)$ avoids 231. The action of the σ -stack on input β is depicted in Figure 3. The first $k - 1$ elements of β are pushed into the σ -stack, since σ has length k . Then the σ -stack contains $1\sigma'_3 \cdots \sigma'_k$, reading from top to bottom, and the next element of the input is σ'_2 . Note that $\sigma'_2 > 1$, whereas $\sigma_1 < \sigma_2$, therefore $\sigma'_2 1 \sigma'_3 \cdots \sigma'_k$ is not an occurrence of σ and σ'_2 is pushed. The next element of the input is now σ'_1 . Here $\sigma'_1 \sigma'_2 \sigma'_3 \cdots \sigma'_k$ is an occurrence of σ , thus we have to pop σ'_2 before pushing σ'_1 . After the pop operation, the σ -stack contains $1\sigma'_3 \cdots \sigma'_k$. Again $\sigma'_1 > 1$, whereas $\sigma_1 < \sigma_2$, therefore σ'_1 is pushed. The resulting string is:

$$s_\sigma(\beta) = \sigma'_2 \sigma'_1 1 \sigma'_3 \sigma'_4 \cdots \sigma'_k.$$

We wish to show that $s_\sigma(\beta)$ avoids 231. Note that $\sigma'_2 \sigma'_1 \sigma'_3 \sigma'_4 \cdots \sigma'_k \simeq \hat{\sigma}$ avoids 231 by hypothesis. Moreover, the element 1 cannot be part of an occurrence of 231, because $\sigma'_2 > \sigma'_1$ and 1 is strictly less than the other elements of β . Therefore $s_\sigma(\beta)$ avoids 231, as desired.

- Otherwise, suppose that σ_1 is not the strict minimum of σ , i.e. either $\sigma_1 \neq 1$ or $\sigma_i = 1$ for some $i \geq 2$. Define:

$$\beta = \sigma''_k \cdots \sigma''_2 1 \sigma''_1 2,$$

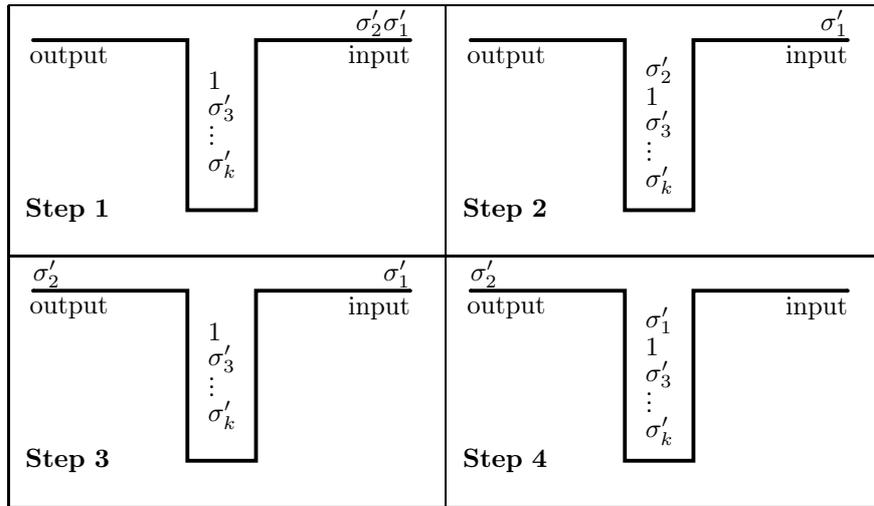


Figure 3: The action of the σ -stack on input β described in the proof of Theorem 3.4.

where $\sigma''_i = \sigma_i + 2$ for each i . Note that β is a Cayley permutation and $1\sigma''_2$ is an occurrence of 132 in β . Consider the action of the σ -stack on β . Again the first $k - 1$ elements of β are pushed into the σ -stack. Then the σ -stack contains $\sigma''_2 \cdots \sigma''_k$, reading from top to bottom, and the next element of the input is 1. Note that $1\sigma''_2 \cdots \sigma''_k$ is not an occurrence of σ . Indeed $1 < \sigma''_i$ for each i , while σ_1 is not the strict minimum of σ by hypothesis. Therefore 1 enters the σ -stack. The next element of the input is then σ''_1 , which realizes an occurrence of σ together with $\sigma''_2 \cdots \sigma''_k$. Thus 1 and σ''_2 are extracted before σ''_1 is pushed. Finally, the last element of the input is 2. Again 2 can be pushed into the σ -stack because 2 is strictly smaller than every element in the σ -stack, whereas σ_1 is not the strict minimum of σ by hypothesis. The resulting string is:

$$s_\sigma(\beta) = 1\sigma''_2\sigma''_1\sigma''_3 \cdots \sigma''_k.$$

Note that $\sigma''_2\sigma''_1\sigma''_3 \cdots \sigma''_k \simeq \hat{\sigma}$ avoids 231 by hypothesis. Finally, it is easy to realize that the elements 1 and 2 cannot be part of an occurrence of 231, similarly to the previous case. This completes the proof. □

Corollary 3.5. *Let σ be a Cayley permutation of length three or more. Then the set of σ -sortable permutations $\text{Sort}(\sigma)$ is not a class if and only if $\hat{\sigma}$ avoids 231. If $\hat{\sigma}$ contains 231, then $\text{Sort}(\sigma) = \mathcal{C}(132, \sigma^r)$.*

We end this section by analyzing the 21-machine. The 11-machine will be discussed in Section 4, thus completing the analysis of the σ -machines on Cayley permutations for patterns σ of length two. The classical permutation analogue of the 21-machine is exactly the (well known) case of the West 2-stack sortable permutations [29]. In this case, although sortable permutations do not form a class, it is possible to describe them efficiently in terms of avoidance of barred patterns.

Theorem 3.6. [29] *A permutation π is not 21-sortable if and only if π contains 2341 or π contains an occurrence of the barred pattern $3\bar{5}241$, i.e. an occurrence 3241 which is not part of an occurrence of 35241.*

The previous theorem can be reformulated in terms of a more general notion of pattern, which will be useful later when dealing with Cayley permutations. A *mesh pattern* [11] of length k is a pair (τ, A) , where τ is a permutation of length k and $A \subseteq [0, k] \times [0, k]$ is a set of pairs of integers. The elements of A identify the lower left corners of shaded squares in the plot of τ (see Figure 4). An occurrence of the mesh pattern (τ, A) in the permutation π is then an occurrence of the classical pattern τ in π such that no other elements of π are placed into a shaded square of A .

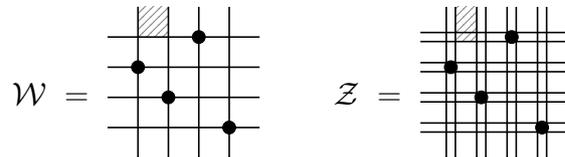


Figure 4: On the left, the barred pattern $3\bar{5}241$, which is equivalent to the mesh pattern $\mathcal{W} = (3241, \{(1, 4)\})$. The shaded box keeps into account the case of an occurrence of 3241 that is part of a 35241. On the right, the Cayley-mesh pattern \mathcal{Z} . The additional shaded region in \mathcal{Z} keeps into account the case of an occurrence of 3241 that is part of an occurrence of 34241.

Note that the barred pattern $3\bar{5}241$ is equivalent to the mesh pattern \mathcal{W} depicted in Figure 4. Now, in order to prove an analogous characterization for the 21-machine on Cayley permutations, we need to adapt the definition of mesh pattern to strings that may contain repeated elements. In other words, we allow the shading of regions that correspond to repeated elements. Instead of giving a formal definition, we refer to the example illustrated in Figure 4. We will use the term *Cayley-mesh pattern* to denote mesh patterns on Cayley permutations.

Lemma 3.7. *Let $\pi = \pi_1 \cdots \pi_n$ be a Cayley permutation. Suppose that $\pi_i < \pi_j$, for some $i < j$. Then π_i precedes π_j in $s_{21}(\pi)$.*

Proof. This follows from the definition of 21-stack. □

Theorem 3.8. *A Cayley permutation π is not 21-sortable if and only if π contains 2341 or π contains the Cayley-mesh pattern \mathcal{Z} depicted in Figure 4. In particular, $\text{Sort}(21)$ is not a class. For example, the 21-sortable Cayley permutation 34241 contains the non-sortable pattern 3241.*

Proof. We can basically repeat the argument used by West for classical permutations. The only difference is the additional shaded box, which corresponds to an occurrence of 3241 that is part of an occurrence of 34241. We sketch the proof anyway for completeness.

Let π be a Cayley permutation and suppose that π is 21-sortable. Suppose by contradiction that π contains an occurrence $bcda$ of 2341 and consider the action of the 21-stack on π . By Lemma 3.7, b is extracted from the 21-stack before c enters. Similarly, c is extracted before d enters. Thus $s_{21}(\pi)$ contains the occurrence bca of 231, contradicting the assumption that π is sortable. Otherwise, suppose that π contains an occurrence $cbda$ of 3241. We show that there is an element x between c and b in π such that $x \geq d$. If $x < c$ for each x in between c and b , then b is pushed into the 21-stack before c is popped. This results in the occurrence bca of 231 in $s_{21}(\pi)$, a contradiction with π 21-sortable. Otherwise, suppose there is at least one element x between c and b in π , with $x \geq c$. If $x = c$, we can repeat the same argument with $xbda$ instead of $cbda$. If $c < x < d$, then $cxda \simeq 2341$, which is impossible due to what was said in the above case. Therefore it has to be $x \geq d$, as desired.

Conversely, suppose that π is not 21-sortable. Equivalently, let bca be an occurrence of 231 in $s_{21}(\pi)$. We show that either π contains 2341 or π contains an occurrence $cbda$ of 3241 such that $x < d$ for each x between c and b in π . Observe that a follows c and b in π due to Lemma 3.7. Suppose that b comes before c in π . Note that c is extracted from the 21-stack before a enters. Let d be the next element of the input when c is extracted. Then $d > c$ and $bcda$ is an occurrence of 2341, as wanted. Otherwise, suppose that b follows c in π , and thus π contains cba . Since c is not extracted before b enters, it has to be $x \leq c$ for each x between c and b in π . Moreover, c is extracted before a enters. When c is extracted, the next element d of the input is such that $d > c$. This results in an occurrence $cbda$ of 3241 with the desired propriety. \square

Open Problem 1. *Enumerate the set of 21-sortable Cayley permutations. The initial terms of the sequence are 1, 3, 13, 73, 483, 3547, 27939, 231395 (not in [26]).*

Notice that, using the language of [19], a Cayley permutation π is 21-sortable if and only if $\text{hare}(\text{hare}(\pi))$ is weakly increasing (see also Section 5).

4 σ -stacks as Operators

In this section we regard σ -stacks as operators. Let σ be a Cayley permutation and define the map $\mathcal{S}^\sigma : \mathcal{C} \mapsto \mathcal{C}$ by $\mathcal{S}^\sigma(\pi) = s_\sigma(\pi)$, for each $\pi \in \mathcal{C}$. We are interested in the behavior of the map \mathcal{S}^σ . This line of inquiry for stacksort operators is not new in the literature. More generally, suppose one is to perform a deterministic sorting procedure. Then it is natural to consider the map \mathcal{S} that associates to an input string π the (uniquely determined) output of the sorting process. Some of the arising problems are the following.

- Determine the *fertility* of a string, which is the number of its pre-images under \mathcal{S} . Fertility under classical stacksort has been recently investigated by Defant [16, 17, 18].

- Determine the image of \mathcal{S} , i.e. the strings with positive fertility. These are often called *sorted permutations* [10].

We start by discussing the case $\sigma = 11$. Here we provide a useful decomposition that allows us to determine explicitly the image $\mathcal{S}^\sigma(\pi)$ of any input Cayley permutation π . From now on, we denote by \mathcal{R} the *reverse* operator, i.e. $\mathcal{R}(\pi) = \pi^r$, for each $\pi \in \mathcal{C}$.

Lemma 4.1. *Let $\sigma = 11$ and let $\pi = \pi_1 \cdots \pi_n$ be a Cayley permutation. Suppose that π contains $k + 1$ occurrences $\pi_1, \pi_1^{(1)}, \dots, \pi_1^{(k)}$ of π_1 , for some $k \geq 0$. Write $\pi = \pi_1 B_1 \pi_1^{(1)} B_2 \cdots \pi_1^{(k)} B_k$. Then*

$$\mathcal{S}^{11}(\pi) = \mathcal{S}^{11}(B_1)\pi_1 \mathcal{S}^{11}(B_2)\pi_1^{(1)} \cdots \mathcal{S}^{11}(B_k)\pi_1^{(k)}.$$

Proof. Consider the action of the 11-stack on input π . Since $x \neq \sigma_1$ for each $x \in B_1$, the sorting process of B_1 is not affected by the presence of σ_1 at the bottom of the 11-stack. Then, when the next element of the input is the second occurrence $\sigma_1^{(1)}$ of σ_1 , the 11-stack is emptied, since $\sigma_1 \sigma_1^{(1)}$ is an occurrence of the forbidden 11. The first elements of $\mathcal{S}^{11}(\pi)$ are thus $\mathcal{S}^{11}(B_1)\sigma_1$. Finally, $\sigma_1^{(1)}$ is pushed into the (empty) 11-stack and the same argument can be repeated. □

Theorem 4.2. *Let $\sigma = 11$. Then $(\mathcal{R} \circ \mathcal{S}^{11})$ is an involution on \mathcal{C} . Moreover, \mathcal{S}^{11} is a length-preserving bijection on \mathcal{C} . Therefore, the number of 11-sortable Cayley permutations of length n is equal to the number of 231-avoiding Cayley permutations of length n .*

Proof. We proceed by induction on the length of the input permutation. Let $\pi = \pi_1 \cdots \pi_n$ be a Cayley permutation of length n . The case $n = 1$ is trivial. If $n \geq 2$, write $\pi = \pi_1 B_1 \pi_1^{(1)} B_2 \cdots \pi_1^{(k)} B_k$ as in the previous lemma. Then, using the same lemma and the inductive hypothesis:

$$\begin{aligned} & [\mathcal{R} \circ \mathcal{S}^{11}]^2(\pi) \\ &= [\mathcal{R} \circ \mathcal{S}^{11}]^2\left(\pi_1 B_1 \pi_1^{(1)} B_2 \cdots \pi_1^{(k)} B_k\right) \\ &= [\mathcal{R} \circ \mathcal{S}^{11} \circ \mathcal{R}]\left(\mathcal{S}^{11}(B_1)\pi_1 \mathcal{S}^{11}(B_2)\pi_1^{(1)} \cdots \mathcal{S}^{11}(B_k)\pi_1^{(k)}\right) \\ &= [\mathcal{R} \circ \mathcal{S}^{11}]\left(\pi_1^{(k)} \mathcal{R}(\mathcal{S}^{11}(B_k)) \cdots \pi_1^{(1)} \mathcal{R}(\mathcal{S}^{11}(B_2)) \pi_1 \mathcal{R}(\mathcal{S}^{11}(B_1))\right) \\ &= \mathcal{R}\left(\mathcal{S}^{11}(\mathcal{R}(\mathcal{S}^{11}(B_k)))\pi_1^{(k)} \cdots \mathcal{S}^{11}(\mathcal{R}(\mathcal{S}^{11}(B_2)))\pi_1^{(1)} \mathcal{S}^{11}(\mathcal{R}(\mathcal{S}^{11}(B_1)))\pi_1\right) \\ &= \pi_1 [\mathcal{R} \circ \mathcal{S}^{11}]^2(B_1)\pi_1^{(1)} [\mathcal{R} \circ \mathcal{S}^{11}]^2(B_2) \cdots \pi_1^{(k)} [\mathcal{R} \circ \mathcal{S}^{11}]^2(B_k) \\ &= \pi_1 B_1 \pi_1^{(1)} B_2 \cdots \pi_1^{(k)} B_k \\ &= \pi. \end{aligned}$$

Therefore $(\mathcal{R} \circ \mathcal{S}^{11})^2(\pi) = \pi$, as desired. Finally, the reverse map \mathcal{R} is bijective, thus \mathcal{S}^{11} is a bijection on \mathcal{C} with inverse $\mathcal{R} \circ \mathcal{S}^{11} \circ \mathcal{R}$. □

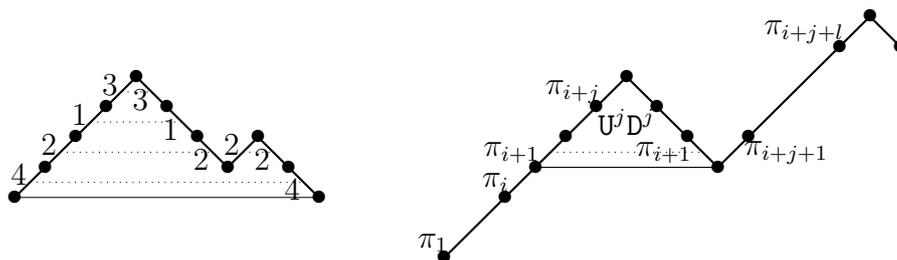


Figure 5: On the left, the Dyck path $UUUUDDDUDD$ which encodes $\mathcal{S}^{11}(42132)$. On the right, the (prefix of the) path $P_\sigma(\pi)$ mentioned in the proof of Corollary 4.5. Dotted lines connect matching steps, which have the same label.

Theorem 4.2 provides a constructive description of the set $\text{Sort}(11)$. Indeed, since $\text{Sort}(11) = \mathcal{R} \circ \mathcal{S}^{11} \circ \mathcal{R}(\mathcal{C}(231))$, every 11-sortable permutation π is obtained from a 231-avoiding Cayley permutation by applying $\mathcal{R} \circ \mathcal{S}^{11} \circ \mathcal{R}$. Next we generalize the above result by providing a characterization of all patterns σ such that \mathcal{S}^σ is bijective on \mathcal{C} . The main tool is an encoding of the action of \mathcal{S}^σ as a Dyck path.

A *Dyck path* is a path in the discrete plane $\mathbb{Z} \times \mathbb{Z}$ starting at the origin, ending on the x -axis, never falling below the x -axis and using two kinds of steps (of length 1), namely up steps $U = (+1, +1)$ and down steps $D = (+1, -1)$. The *height* of a step is its final ordinate. For each up step U , there is a unique *matching* step D defined as the first D step after U with height one less than U . The *length* of a Dyck path is the total number of its steps. A *valley* of a Dyck path is an occurrence of two consecutive steps DU . An example of Dyck path is illustrated in Figure 5. It is well known that Dyck paths, according to the semilength, are enumerated by Catalan numbers (sequence $A000108$ in [26]). A *labeled Dyck path* is a Dyck path where each step has a positive integer label. In this paper we consider labeled Dyck paths where the label of each up step is the same as the label of its matching down step. Therefore we can represent a labeled Dyck path \mathcal{P} as a pair $\mathcal{P} = (P, \pi)$, where P is the underlying Dyck path and π is the string obtained by reading the labels of the up steps from left to right. Given an unlabeled Dyck path P of length $2n$, the *reverse* path $\mathcal{R}(P)$ of P is obtained by taking the symmetric path with respect to the vertical line $x = n$.

Now let σ be a Cayley permutation and suppose we are applying \mathcal{S}^σ to the input Cayley permutation π , i.e. we are sorting π using a σ -stack. Then define a labeled Dyck path $\mathcal{P}_\sigma(\pi)$ as follows:

- Insert an up step U labeled a whenever the algorithm pushes an element a into the σ -stack.
- Insert a down step D labeled a whenever the algorithm pops an element a from the σ -stack.

Equivalently, define $P_\sigma(\pi)$ as the unlabeled Dyck path obtained by recording the push operations of the σ -stack with U and the pop operations with D . Then $\mathcal{P}_\sigma(\pi) =$

$(P_\sigma(\pi), \pi)$. Note that $P_\sigma(\pi)$ is a Dyck path. Indeed the number of push and pop operations performed when processing π is the same, therefore the number of U steps matches the number of D steps (and thus the path ends on the x -axis). Moreover, the path cannot go below the x -axis, since this would correspond to performing a pop operation when the σ -stack is empty, which is not possible. An example of this construction, when $\sigma = 11$, is depicted in Figure 5. Some basic properties of $\mathcal{P}_\sigma(\pi)$ are listed in the following lemma, whose straightforward proof is omitted.

Lemma 4.3. *Let σ be a Cayley permutation. Let $\pi = \pi_1 \cdots \pi_n$ be a Cayley permutation of length n and let $\mathcal{P}_\sigma(\pi) = (P_\sigma(\pi), \pi)$. Then:*

1. *The input π is obtained by reading the labels of the up steps of $P_\sigma(\pi)$ from left to right. The output $s_\sigma(\pi)$ is obtained by reading the labels of the down steps from left to right.*
2. *The height of each up (respectively down) step of $P_\sigma(\pi)$ is equal to the number of elements contained in the σ -stack after having performed the corresponding push (respectively pop) operation.*
3. *The σ -stack is emptied by a pop operation if and only if the corresponding D step of $P_\sigma(\pi)$ is a return on the x -axis. In other words, the decomposition of π considered in Lemma 4.1 corresponds to the decomposition of $P_\sigma(\pi)$ obtained by considering the returns on the x -axis.*
4. *The labels of the down steps are uniquely determined by the labels of the up steps. Conversely, the labels of the down steps uniquely determine the labels of the up steps. More precisely, matching steps have the same label. Indeed the element pushed into the σ -stack by an up step is then popped by the matching down step.*
5. *Let DU be a valley in $P_\sigma(\pi)$. Let a be the label of D and b the label of U. Then b plays the role of σ_1 in an occurrence of σ that triggers the restriction of the σ -stack, whereas a plays the role of σ_2 in such an occurrence. Moreover the number of valleys of $P_\sigma(\pi)$ is equal to the number of elements of π that trigger the restriction of the σ -stack.*
6. *If $\sigma_1 = \sigma_2$, then, for each valley DU, the labels of D and U are the same.*

Theorem 4.4. *Let $\sigma = \sigma_1 \cdots \sigma_k$ be a Cayley permutation. Let $\pi = \pi_1 \cdots \pi_n$ be a Cayley permutation and let $\gamma = \mathcal{R}(\mathcal{S}^\sigma(\pi))$. Consider the two labeled Dyck paths $\mathcal{P}_\sigma(\pi) = (P_\sigma(\pi), \pi)$ and $\mathcal{P}_\sigma(\gamma) = (P_\sigma(\gamma), \gamma)$.*

1. *If $\sigma_1 = \sigma_2$, then $P_\sigma(\pi) = \mathcal{R}(P_\sigma(\gamma))$.*
2. *If $P_\sigma(\pi) = \mathcal{R}(P_\sigma(\gamma))$, then $(\mathcal{R} \circ \mathcal{S}^\sigma)^2(\pi) = \pi$.*

Proof. 1. Suppose that $\sigma_1 = \sigma_2$. We proceed by induction on the number of valleys of $P_\sigma(\pi)$. If $P_\sigma(\pi)$ has zero valleys, then π avoids $\mathcal{R}(\sigma)$ by item 5 of Lemma 4.3.

Therefore $\mathcal{S}^\sigma(\pi) = \mathcal{R}(\pi)$ and $\gamma = \mathcal{R}^2(\pi) = \pi$. Since $P_\sigma(\pi) = \mathcal{U}^n \mathcal{D}^n$ is a pyramid, and each pyramid is equal to its reverse, the thesis follows immediately.

Now suppose that $P_\sigma(\pi)$ has at least one valley. Let $P_\sigma(\pi) = p_1 \cdots p_{2n}$ and write $P_\sigma(\pi) = \mathcal{U}^i \mathcal{U}^j \mathcal{D}^j \mathcal{U}^l \mathcal{D}^l Q$, where the steps p_{i+2j} and p_{i+2j+1} form the leftmost valley and $Q = p_{i+2j+l+2} \cdots p_n$ is the remaining suffix of $P_\sigma(\pi)$ (see Figure 5). Note that the label of both p_{i+2j} and p_{i+2j+1} is equal to π_{i+1} because of items 4, 5 and 6 of Lemma 4.3. Item 5 also implies that p_{i+2j+1} plays the role of σ_1 in an occurrence of σ that triggers the restriction of the σ -stack. More precisely, immediately after the push of π_{i+j} (i.e. after the up step p_{i+j} in $P_\sigma(\pi)$), π_{i+j+1} is the next element of the input. Since the next segment of the path is \mathcal{D}^j , j pop operations are performed before pushing π_{i+j+1} . This means that the element π_{i+1} , corresponding to the last down step, plays the role of σ_2 in an occurrence of σ , while π_{i+j+1} plays the role of σ_1 . Moreover there are $k - 2$ elements in the σ -stack that play the role of $\sigma_3, \dots, \sigma_k$. Since the elements in the σ -stack correspond to the labels of the initial prefix \mathcal{U}^i , $\pi_1 \cdots \pi_i$ contains an occurrence of $\sigma_k \cdots \sigma_3$ (claim I). Then, after j pop operations are performed, the σ -stack contains $\pi_i \cdots \pi_1$, reading from top to bottom, and the elements $\pi_{i+j+1}, \pi_{i+j+2}, \dots, \pi_{i+j+l}$ are pushed (claim II).

Now, write:

$$\pi = \underbrace{\pi_1 \cdots \pi_i}_A \underbrace{\pi_{i+1} \cdots \pi_{i+j}}_B \underbrace{\pi_{i+j+1} \cdots \pi_{i+j+l}}_C \underbrace{\pi_{i+j+l+1} \cdots \pi_n}_D,$$

where the elements of A correspond to the initial prefix \mathcal{U}^i of $P_\sigma(\pi)$, B corresponds to \mathcal{U}^j , C to \mathcal{U}^l and D to the remaining up steps. Consider the string $ACD = \pi_1 \cdots \pi_i \pi_{i+j+1} \cdots \pi_n$ obtained by removing the segment $B = \pi_{i+1} \cdots \pi_{i+j}$ from π . Let $\tilde{\pi}$ be the only Cayley permutation that is order isomorphic to ACD (i.e. obtained by suitably rescaling the elements of ACD , if necessary). Note that $\mathcal{P}_\sigma(\tilde{\pi})$ is obtained from $\mathcal{P}_\sigma(\pi)$ by cutting out the pyramid $\mathcal{U}^j \mathcal{D}^j$, which corresponds to the removed segment B . This is because the elements contained in the σ -stack after having pushed π_i are exactly the same as the elements contained in the σ -stack after having pushed π_{i+j+1} , thus we can safely cut out the pyramid $\mathcal{U}^j \mathcal{D}^j$ without affecting the sorting procedure. Therefore:

$$\mathcal{S}^\sigma(\pi) = \mathcal{R}(B) \mathcal{S}^\sigma(\tilde{\pi}) \quad \text{and} \quad \gamma = \mathcal{R}(\mathcal{S}^\sigma(\pi)) = \mathcal{R}(\mathcal{S}^\sigma(\tilde{\pi}))B.$$

Now, since $P_\sigma(\tilde{\pi})$ has one valley less than $P_\sigma(\pi)$, by inductive hypothesis $P_\sigma(\tilde{\pi}) = \mathcal{R}(P_\sigma(\tilde{\gamma}))$, where $\tilde{\gamma} = \mathcal{R}(\mathcal{S}^\sigma(\tilde{\pi}))$. The only difference between $P_\sigma(\pi)$ and $P_\sigma(\tilde{\pi})$ is the removed pyramid $\mathcal{U}^j \mathcal{D}^j$. We wish to show that $P_\sigma(\gamma)$ is obtained from $P_\sigma(\tilde{\gamma})$ by reinserting the same pyramid $\mathcal{U}^j \mathcal{D}^j$ in the same place, from which the thesis follows.

We have $\gamma = \mathcal{R}(\mathcal{S}^\sigma(\tilde{\pi}))B$ and $\tilde{\gamma} = \mathcal{R}(\mathcal{S}^\sigma(\tilde{\pi}))$. Consider the last push performed by the σ -stack when processing $\tilde{\gamma}$, which corresponds to the last up step of $\mathcal{P}_\sigma(\tilde{\gamma})$. Notice that, since $P_\sigma(\tilde{\pi}) = \mathcal{R}(P_\sigma(\tilde{\gamma}))$, this is also the first down step of $P_\sigma(\tilde{\pi})$, and thus the first pop performed when processing $\tilde{\pi}$. Therefore the elements contained in the σ -stack after the last push performed while processing $\tilde{\gamma}$ are $\pi_{i+j+l} \cdots \pi_{i+j+1} \pi_i \cdots \pi_1$, reading from top to bottom. If we sort γ instead of $\tilde{\gamma}$, we have to process the

additional segment B . Now, the first element of B is π_{i+1} . Since the same happened when sorting π (see claim I), π_{i+1} realizes an occurrence of σ together with π_{i+j+1} (which plays the role of σ_2) and $k-2$ other elements in $\pi_1 \cdots \pi_i$. The only difference is that, contrary to what happened when sorting π , the roles of π_{i+1} and π_{i+j+1} are interchanged: here the hypothesis $\sigma_1 = \sigma_2$ is relevant. As a result, before pushing the first element π_{i+1} of B , we have to pop each element of the σ -stack up to π_{i+j+1} , π_{i+j+1} included. After that, the σ -stack contains $\pi_i \cdots \pi_1$, reading from top to bottom. Therefore we can push $\pi_{i+1} = \pi_{i+j+1}$ and the remaining elements of B because of claim II. This means that $P_\sigma(\gamma)$ is obtained by inserting a pyramid $U^j D^j$ immediately before the last i down steps of $P_\sigma(\tilde{\gamma})$, as desired.

2. By hypothesis, $P_\sigma(\gamma) = \mathcal{R}(P_\sigma(\pi))$, therefore the word w obtained by reading the labels of the down steps of $P_\sigma(\gamma)$ (from left to right) is $w = \mathcal{R}(\pi)$. By definition of $\mathcal{P}_\sigma(\gamma)$, we also have $w = \mathcal{S}^\sigma(\gamma)$. Thus:

$$\mathcal{R}(\pi) = \mathcal{S}^\sigma(\gamma) = \mathcal{S}^\sigma(\mathcal{R}(\mathcal{S}^\sigma(\pi)))$$

and the thesis follows by applying the reverse operator to both sides of the equality. \square

Corollary 4.5. *Let $\sigma = \sigma_1 \cdots \sigma_k \in \mathcal{C}$. Then \mathcal{S}^σ is bijective if and only if $\sigma_1 = \sigma_2$. In this case, \mathcal{S}^σ is a bijection on \mathcal{C} that preserves the multiset of entries of a Cayley permutation and $\mathcal{R} \circ \mathcal{S}^\sigma$ is an involution on the set \mathcal{C} .*

Proof. Suppose that $\sigma_1 \neq \sigma_2$. Then $\hat{\sigma} \neq \sigma$, thus also $\mathcal{R}(\sigma) \neq \mathcal{R}(\hat{\sigma})$. Finally, $\mathcal{S}^\sigma(\mathcal{R}(\sigma)) = \hat{\sigma} = \mathcal{S}^\sigma(\mathcal{R}(\hat{\sigma}))$, therefore \mathcal{S}^σ is not injective.

Conversely, suppose that $\sigma_1 = \sigma_2$. By Theorem 4.4, we have that $(\mathcal{R} \circ \mathcal{S}^\sigma)^2$ is the identity on \mathcal{C} , therefore $\mathcal{R} \circ \mathcal{S}^\sigma$ is bijective. Finally, since the reverse map \mathcal{R} is bijective, \mathcal{S}^σ is a bijection too, as desired. \square

5 Pop-stack on Cayley permutations

This section is devoted to the study of pop-stack sorting on Cayley permutations. Recall from Section 1 that a *pop-stack* is a stack where all the elements are extracted every time a pop operation is performed. In analogy with [19], we introduce the hare and tortoise variants of a pop-stack, according to whether or not a letter is allowed to sit on a copy of itself.

A *hare pop-stack* is a 21-pop-stack, i.e. a 21-stack that is emptied every time a pop operation is performed. A Cayley permutation π is *hare pop-stack sortable* if π is sortable using a right-greedy algorithm on a hare pop-stack.

A *tortoise pop-stack* is a $\{21, 11\}$ -pop-stack, i.e. a $\{21, 11\}$ -stack that is emptied every time a pop operation is performed. A Cayley permutation π is *tortoise pop-stack sortable* if π is sortable using a right-greedy algorithm on a tortoise pop-stack.

Denote by $\mathcal{HPS}(\pi)$ and $\mathcal{TPS}(\pi)$ the output of a hare pop-stack and, respectively, a tortoise pop-stack, on input π . Recall that, since we are allowing repeated

elements, to sort a Cayley permutation means to produce a weakly increasing Cayley permutation (and not necessarily the identity permutation as in the classical case). Equivalently, the output of either a hare or tortoise pop-stack is not sorted if and only if it contains a strong descent. We shall provide a characterization of hare and tortoise pop-stack sortable permutations in terms of forbidden patterns, starting with hare pop-stack. The next lemma is a straightforward consequence of the definition of hare pop-stack.

Lemma 5.1. *Let π be a Cayley permutation. Write $\pi = B_1B_2\cdots B_k$, where each block B_t is maximally weakly decreasing (i.e. the last element of each block B_t forms a strong ascent together with the first element of the next block B_{t+1}). Then:*

$$\mathcal{HPS}(x) = \mathcal{R}(B_1) \cdots \mathcal{R}(B_k).$$

Theorem 5.2. *Let π be a Cayley permutation. Then π is hare pop-stack sortable if and only if π avoids 231, 312 and 2121.*

Proof. Suppose that π is hare pop-stack sortable. Observe that hare pop-stack sortable Cayley permutations are a subset of 21-sortable Cayley permutations. Thus π avoids 231 due to Theorem 3.1. Now suppose by contradiction that π contains an occurrence cab of 312. Then, when b enters the hare pop-stack, a has been already extracted. Otherwise $\mathcal{HPS}(\pi)$ would not be weakly increasing, contradicting the hypothesis. Therefore also c has been extracted and thus $b < c$, which again contradicts the fact that π is hare pop-stack sortable. Similarly, suppose that π contains an occurrence $bab'a'$ of 2121. Then a , and thus also b , must have been extracted before b' enters the stack, since $b' > a$. Therefore b is extracted before a' enters the hare pop-stack, which is impossible because $a' < b$.

Conversely suppose that π is not hare pop-stack sortable. We wish to show that π contains an occurrence of either 231, 312 or 2121. Write $\pi = B_1B_2\cdots B_k$ as in Lemma 5.1. Then $\mathcal{HPS}(\pi) = B_1^R \cdots B_k^R$ and $\mathcal{HPS}(\pi)$ contains at least one strict descent. Let $a > b$ the leftmost strict descent in $\mathcal{HPS}(\pi)$. Due to Lemma 5.1, it must be $a \in B_i$ and $b \in B_{i+1}$, for some i . The same result implies that a is the first element of B_i and b is the last element of B_{i+1} . Now, denote by u the last element of B_i and by v the first element of B_{i+1} , as illustrated in Figure 6. We have $a \geq u$, $v \geq b$ and $v > u$. Consider the following case-by-case analysis.

- If B_i is a singleton, then $u = a > b$ and $a = u < v$, therefore $b \neq v$ and avb is an occurrence of 231 in π .
- If B_{i+1} is a singleton, then $b = v > u$ and $v = b < a$, therefore $a \neq u$ and auv is an occurrence of 312 in π .
- Finally, suppose that both B_i and B_{i+1} are not singletons and consider the four elements $auvb$ in π . If $a > v$, then auv is an occurrence of 312. If $a < v$, then avb is an occurrence of 231. Otherwise, suppose that $a = v$ (and so $a = v > u$). Then $auvb$ is an occurrence of 2121, if $u = b$; aub is an occurrence of 231, if $u > b$; and aub is an occurrence of 312, if $u < b$.

$$\begin{aligned} \pi &= \underbrace{\cdots}_{B_1} | \cdots | \underbrace{a \cdots u}_{B_i} | \underbrace{v \cdots b}_{B_{i+1}} | \cdots | \underbrace{\cdots}_{B_k} \\ \mathcal{H}/\mathcal{TPS}(\pi) &= \underbrace{\cdots}_{B_1} | \cdots | \underbrace{u \cdots a}_{B_i} | \underbrace{b \cdots v}_{B_{i+1}} | \cdots | \underbrace{\cdots}_{B_k} \end{aligned}$$

Figure 6: The decomposition of π used in the proofs of Theorem 5.2 and Theorem 5.4.

□

Next we consider tortoise pop-stack.

Lemma 5.3. *Let π be a Cayley permutation. Write $\pi = B_1 B_2 \cdots B_k$, where each block B_t is maximally strictly decreasing (i.e. the last element of each block B_t forms a weak ascent together with the first element of the next block B_{t+1}). Then:*

$$\mathcal{TPS}(x) = \mathcal{R}(B_1) \cdots \mathcal{R}(B_k).$$

Theorem 5.4. *Let π be a Cayley permutation. Then π is tortoise pop-stack sortable if and only if π avoids 231, 312, 221 and 211.*

Proof. The proof is similar to that of Theorem 5.2. It is not difficult to show that if π contains an occurrence of either 231, 312, 221 or 211, then π is not tortoise pop-stack sortable. We leave the details to the reader.

Conversely, suppose that π is not tortoise pop-stack sortable. Then $\mathcal{TPS}(\pi) = \mathcal{R}(B_1) \cdots \mathcal{R}(B_k)$ and $\mathcal{TPS}(\pi)$ contains at least one strict descent $a > b$. Suppose a, b is the leftmost strict descent in $\mathcal{TPS}(\pi)$. Again it has to be $a \in B_i$ and $b \in B_{i+1}$ for some i , due to Lemma 5.3. Denote with u the last element of B_i and with v the first element of B_{i+1} . Notice that if a and u are distinct, then $a > u$; similarly, $v > b$ if v and b are distinct. Moreover, we have $u \leq v$ by Lemma 5.3 (see Figure 6). Suppose that B_i is a singleton and thus $a = u$. Then $v \neq b$, since $v \geq a$, whereas $b < a$. Now, if $a < v$, then avb is an occurrence of 231. Otherwise, if $a = v$, then avb is an occurrence of 221. Otherwise, suppose that B_i is not a singleton and thus $a \neq u$. If $a > v$, then auv is an occurrence of either 312, if $u < v$, or 211, if $u = v$. If $a < v$, then $v \neq b$, since $v > a$ and $b < a$, and avb is an occurrence of 231. Finally, suppose that $a = v$. Note that again $v \neq b$, since $v = a$ and $b < a$. Therefore, if $b = u$, then aub is an occurrence of 211. If $b < u$, then uvb is an occurrence of either 231, if $u < v$, or 221, if $u = v$. Instead, if $b > u$, then aub is an occurrence of 312. □

Let us now enumerate tortoise pop-stack sortable Cayley permutations. First a geometrical description. Write again $\pi = B_1 B_2 \cdots B_k$, where each block B_i is maximally strictly decreasing. Denote by m_i the first element of B_i and let $B_i = m_i A_i$, where A_i contains the remaining elements of B_i . Suppose that π is tortoise pop-stack sortable. Then:

1. $m_i \leq m_{i+1}$ for each i . Otherwise, suppose by contradiction that $m_i > m_{i+1}$. Let x the last element of B_i . Then $x \leq m_{i+1}$, therefore $x \neq m_i$ and $m_i x m_{i+1}$ is an occurrence of 312, contradicting Theorem 5.4.
2. $A_i < A_{i+1}$ for each i . In other words, $x < y$ for each $x \in A_i$ and $y \in A_{i+1}$. Otherwise, if $x = y$, then $m_i x y$ is an occurrence of 211, against Theorem 5.4. Instead, if $x > y$, then $x m_{i+1} y$ is an occurrence of 231, which is impossible due to the same result.
3. $y \geq m_i$ for each $y \in A_{i+1}$. Otherwise $m_i m_{i+1} y$ is an occurrence of either 231, if $m_i < m_{i+1}$, or 221, if $m_i = m_{i+1}$. In both cases this is impossible due to Theorem 5.4.

Denote by $f_{n,k}$ the number of tortoise pop-stack sortable Cayley permutations of length n and with k maximally strictly decreasing blocks. As a consequence of what was said above, each of these Cayley permutations is determined uniquely by choosing:

- The length of each block, which can be done in $\binom{n-1}{k-1}$ distinct ways.
- Whether the last element of a block is equal to or greater than the first element of the previous block. Equivalently, if ℓ_i is the last element of the block B_i , whether $\ell_{i+1} = m_i$ or $\ell_{i+1} = m_i + 1$, for each $i \geq 2$.

Therefore $f_{n,k} = \binom{n-1}{k-1} 2^{k-1}$.

Corollary 5.5. *For each $n \geq 1$, there are 3^{n-1} tortoise pop-stack sortable permutations of length n .*

Proof. Let f_n be the number of tortoise pop-stack sortable permutations of length n . We have:

$$3^{n-1} = (2+1)^{n-1} = \sum_{j=0}^{n-1} \binom{n-1}{j} 2^j = \sum_{k=1}^n \binom{n-1}{k-1} 2^{k-1} = \sum_{k=1}^n f_{n,k} = f_n.$$

□

The enumeration of hare pop-stack sortable Cayley permutations, or equivalently the set of Cayley permutations avoiding 231, 312 and 2121, is rather more complicated, thus we leave it for a future work.

Open Problem 2. *Enumerate the hare pop-stack sortable Cayley permutations. The sequence starts 1, 3, 11, 41, 151, 553 and it does not match any sequence in the OEIS [26].*

The authors of [2] studied *pop-stacked permutations*, that is the image of the deterministic pop-stack operator on classical permutations. They show a structural characterization of such permutations, as well as a proof that the generating function is rational when the number of runs is fixed. Some algorithmic and asymptotic aspects of pop-stacked permutations are also considered. It would thus be interesting to study the analogues of their results for hare pop-stack and tortoise pop-stack on Cayley permutations.

Acknowledgements

The author would like to sincerely thank Anders Claesson for the fruitful discussions and the precious data he provided. The author is grateful to the anonymous referees for carefully reading this manuscript and providing a lot of useful suggestions and ideas to improve its readability.

References

- [1] M. H. Albert, R. E. L. Aldred, M. D. Atkinson, C. Handley and D. Holton, Permutations of a multiset avoiding permutations of length 3, *Europ. J. Combin.* **22** (2001), 1021–1031.
- [2] A. Asinowski, C. Banderier and B. Hack, Flip-sort and combinatorial aspects of pop-stack sorting, *Discrete Math. Theor. Comput. Sci.* **22**(2) (2021).
- [3] M. D. Atkinson, S. A. Linton and L. A. Walker, Priority queues and multisets, *Electron. J. Combin.* **2** (1995), #R24.
- [4] M. D. Atkinson and T. Stitt, Restricted permutations and the wreath product, *Discrete Math.* **259** (1-3) (2002), 19–36.
- [5] D. Avis and M. Newborn, On pop-stacks in series, *Util. Math.* **19** (1981), 129–140.
- [6] J.-L. Baril, G. Cerbai, C. Khalil and V. Vajnovszki, Catalan and Schröder permutations sortable by two restricted stacks, *Inform. Process. Lett.* **171** (2021), 106138.
- [7] K. Berlow, Restricted stacks as functions, arXiv:2008.01164.
- [8] M. Bona, A survey of stack sorting disciplines, *Electron. J. Combin.* **9** (2) (2003), #A1.
- [9] M. Bona, *Combinatorics of Permutations*, *Discrete Math. Appl.*, CRC Press (2004).

- [10] M. Bousquet-Melou, Sorted and/or sortable permutations, *Discrete Math.* **225** (1-3) (2000), 25–50.
- [11] P. Brändén and A. Claesson, Mesh Patterns and the expansion of permutation statistics as sums of permutation patterns, *Electron. J. Combin.* **18** (2011), #P5.
- [12] G. Cerbai, L. Cioni and L. Ferrari, Stack sorting with increasing and decreasing stacks, *Electron. J. Combin.* **27** (1) (2020), #P1.1.
- [13] G. Cerbai, A. Claesson and L. Ferrari, Stack sorting with restricted stacks, *J. Combin. Theory Ser. A* **173** (2020), 105230.
- [14] G. Cerbai, A. Claesson, L. Ferrari and E. Steingrímsson, Sorting with pattern-avoiding stacks: the 132-machine, *Electron. J. Combin.* **27** (3) (2020), #P3.32.
- [15] A. Claesson and B. A. Guðmundsson, Enumerating permutations sortable by k passes through a pop-stack, *Adv. Appl. Math.* **108** (2019), 79–96.
- [16] C. Defant, Postorder preimages, *Discrete Math. Theor. Comput. Sci.* **19** (1) (2017).
- [17] C. Defant, Counting 3-stack-sortable permutations, *J. Combin. Theory Ser. A* **172** (2020), 105209.
- [18] C. Defant, Troupes, Cumulants, and Stack-Sorting, arXiv:2004.11367.
- [19] C. Defant and N. Kravitz, Stack-sorting for words, *Australas. J. Combin.* **77** (1) (2020), 51–68.
- [20] C. Defant and K. Zheng, Stack-Sorting with Consecutive-Pattern-Avoiding Stacks, *Adv. Appl. Math.* **128** (2021), 102192.
- [21] S. Kitaev, *Patterns in permutations and words*, Monographs in Theoretical Computer Science; an EATCS Series, Springer, Heidelberg (2011).
- [22] D. E. Knuth, *The art of computer programming, vol. 1, Fundamental Algorithms*, Addison-Wesley, Reading, Massachusetts (1973).
- [23] M. Mor and A. S. Fraenkel, Cayley permutations, *Discrete Math.* **48** (1) (1984), 101–112.
- [24] M. M. Murphy, Restricted permutations, antichains, atomic classes and stack sorting, PhD thesis, University of St. Andrews (2002).
- [25] L. Pudwell and R. Smith, Two-stack-sorting with pop stacks, *Australas. J. Combin.* **74** (1) (2019), 179–195.
- [26] N. J. A. Sloane, The On-Line Encyclopedia of Integer Sequences, at oeis.org.
- [27] R. Smith, Two stacks in series: a decreasing stack followed by an increasing stack, *Ann. Comb.* **18** (2014), 359–363.

- [28] J. West, Permutations with forbidden subsequences and stack-sortable permutations, PhD thesis, Massachusetts Institute of Technology, 1990.
- [29] J. West, Sorting twice through a stack, *Theoret. Comput. Sci.* **117** (1-2) (1993), 303–313.

(Received 22 Apr 2020)