

The Insertion Encoding of Cayley Permutations

Christian Bean^a Paul C. Bell^b Abigail Ollson^a

Submitted: May 23, 2025; Accepted: Jun 16, 2026; Published: Aug 7, 2026

© The authors. Released under the CC BY license (International 4.0).

Abstract

We introduce the vertical and horizontal insertion encodings for Cayley permutations which naturally generalise the insertion encoding for permutations. In both cases, we fully classify the Cayley permutation classes for which these languages are regular, and provide an algorithm for computing the rational generating functions. We use our algorithm to solve an open problem of Cerbai by enumerating the hare pop-stack sortable Cayley permutations.

Mathematics Subject Classifications: 05A05, 05A15, 05A18

1 Introduction

A *Cayley permutation* is a word $\pi \in \mathbb{N}_1^*$ such that every number between 1 and the maximum value of π appears at least once. The plot of a Cayley permutation $\pi_1\pi_2\cdots\pi_n$ is the set of points (i, π_i) . For example, the plot of the Cayley permutation 21321 is shown in Figure 1.

The set of Cayley permutations is in bijection with the set of ordered set partitions; the value i in the j^{th} block of the ordered set partition implies the i^{th} index of the Cayley permutation has value j . For example, the Cayley permutation 21321 corresponds to the ordered set partition $\{2, 5\}, \{1, 4\}, \{3\}$. Therefore, the number of Cayley permutations of size n is the n^{th} Fubini number, A000670 in the Online Encyclopedia of Integer Sequences (OEIS) [25].

There are many objects which are equivalent to Cayley permutations, for example surjective words [15], packed words [13, 18], initial words [22], and preferential arrangements [1, 6, 21, 23]. Cayley permutations were first defined in 1983 by Mor and Fraenkel [20]. The name was inspired by a 1857 paper of Cayley [7] where he counted a subclass of trees which are enumerated by the ordered Bell numbers, also known as the Fubini numbers. Mor and Fraenkel called these trees *Cayley trees* and found that they

^aSchool of Computer Science and Mathematics, Keele University, Keele, United Kingdom
(c.n.bean@keele.ac.uk, a.n.ollson@keele.ac.uk).

^bSchool of Computer Science and Mathematics, Liverpool John Moores University, Liverpool, United Kingdom (p.c.bell@ljmu.ac.uk).

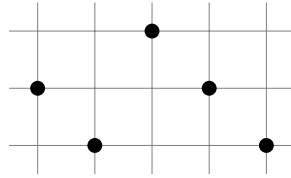


Figure 1: The plot of the Cayley permutation 21321.

are in bijection with Cayley permutations. They also gave a new formula for enumerating Cayley permutations by counting the number of nondecreasing Cayley permutations and then counting the number of ways in which each of these can be permuted.

The *standardisation* of a word in $\mathbb{N}_1^*$ is the Cayley permutation obtained by replacing every occurrence of the smallest integer by 1, the next smallest by 2, and so on. For example, the standardisation of 4674 is 1231. We say that the word $w = w_1w_2 \cdots w_n$ *contains* the Cayley permutation $p = p_1p_2 \cdots p_k$ if there exist indices $i_1 < i_2 < \cdots < i_k$ such that the standardisation of $w_{i_1}w_{i_2} \cdots w_{i_k}$ is p . We say w *avoids* p if w does not contain p . In this context, we call p a pattern. For example, the Cayley permutation 21321 contains multiple occurrences of the pattern 121, such as the standardisation of 232 or an occurrence of 121 directly, but 21321 avoids the pattern 123 as there is no strictly increasing subsequence of size 3 in 21321.

This containment order forms a partial order on the set of Cayley permutations. A set of Cayley permutations that is closed downwards is called a *Cayley permutation class*, i.e., if $\pi \in \mathcal{C}$ and π contains σ , then $\sigma \in \mathcal{C}$. A Cayley permutation class can be uniquely defined by the set of minimal Cayley permutations not in $\mathcal{C}$, called the *basis*. For a set of Cayley permutations B we say that a Cayley permutation π *contains* B if π contains any element of B , and *avoids* B if π avoids every element of B . The set of Cayley permutations that avoid B is denoted $\text{Av}(B)$.

Cayley permutations can be seen as a generalisation of permutations where repeated values are allowed, hence the Cayley permutation class $\text{Av}(11)$ describes all permutations. The theory of permutation classes has been very well developed. One of the first notable results was by MacMahon in 1915 [19] where he found that the Catalan numbers enumerate the class of permutations avoiding 123. In 1968, Knuth [17] asked as an exercise to prove that a permutation is stack sortable if and only if it avoids 231. The permutation class $\text{Av}(231)$ is also enumerated by the Catalan numbers. These classes can be thought of as the Cayley permutation classes $\text{Av}(11, 123)$ and $\text{Av}(11, 231)$ respectively. Knuth's work brought much more focused attention to the area of pattern-avoiding permutations eventually leading to the first systematic study of pattern-avoiding permutations in 1985 by Simion and Schmidt [24]. For a more detailed overview see for example the survey by Vatter [27].

The goal of this work is to extend some of the methods that have been developed for permutation classes to study Cayley permutation classes. In particular, we extend the insertion encoding of permutations introduced by Albert, Linton and Ruškuc [2] to Cayley permutations in two different ways; first, by inserting new maxima, which we will

call the *vertical insertion encoding* and then by inserting new rightmost values, which we will call the *horizontal insertion encoding*. As permutations are closed with respect to the inverse symmetry, inserting a new maximum is equivalent to inserting a new rightmost value so these two methods are equivalent. For Cayley permutations this is no longer the case. Section 2 and Section 3 explain the method for the vertical insertion encoding and Section 5 considers the horizontal insertion encoding.

Our main results are in Section 3 and Section 5 which fully classify the Cayley permutation classes for which the vertical and horizontal insertion encodings form a regular language and follow a similar theory to the theory for permutations. We outline a method for computing these languages for the vertical insertion encoding in Section 4, which we also adapted to the horizontal case. Our implementation of these methods has been made available on GitHub [5].

Table 1 shows the number of Cayley permutation classes defined by avoiding size 3 patterns which have either a set of vertical or horizontal insertion encodings which form a regular language. Every class defined by avoiding nine or more size three patterns has either a set of vertical or horizontal insertion encodings which is regular. Of the 8191 bases containing Cayley permutations of size 3 only, 7498 of them have a set of vertical or horizontal insertion encodings which is regular.

Size of basis	Number of classes	Set of vertical ins. encodings are regular	Set of horizontal ins. encodings are regular	Either vertical or horizontal ins. encodings are regular
1	13	0	0	0
2	78	0	13	13
3	286	87	111	145
4	715	435	428	528
5	1287	1028	986	1124
6	1716	1550	1513	1625
7	1716	1645	1631	1687
8	1287	1269	1267	1283
9	715	713	713	715

Table 1: The number of Cayley permutation classes defined by avoiding a set of size 3 patterns. In the third and fourth columns, the number of classes with a set of vertical insertion encodings and a set of horizontal insertion encodings which are regular are shown. The last column shows the number of classes with either a set of vertical or horizontal insertion encodings which are regular.

Recently, Claesson, Cerbai, Ernst and Golab [10] enumerated $\text{Av}(111)$, $\text{Av}(112)$, $\text{Av}(121)$, $\text{Av}(123)$ and $\text{Av}(132)$ using combinatorial species. Golab's thesis [14] went on to enumerate $\text{Av}(112, 121)$ and $\text{Av}(111, 112)$. None of these classes have either a set of vertical or horizontal insertion encodings which are regular.

Defant and Kravitz [11] have adapted the theory behind stack sortable permutations to Cayley permutations. They defined two different types of stack sorting, a hare-stack sort

and a tortoise-stack sort. In a hare-stack sort, consecutive repeated values are allowed in the stack and in a tortoise-stack sort they are not. They identified the classes of Cayley permutations that are hare-stack sortable and tortoise-stack sortable as $\text{Av}(231)$ and $\text{Av}(221, 231)$ respectively, and found a recurrence relation for their counting sequences. Cerbai [8] then found the bases for tortoise pop-stack sortable Cayley permutations to be $\text{Av}(211, 221, 231, 312)$ and hare pop-stack sortable to be $\text{Av}(231, 312, 2121)$, enumerating $\text{Av}(211, 221, 231, 312)$. Using the insertion encoding, implemented using Tilings described in Section 4, we present the generating function for $\text{Av}(231, 312, 2121)$ in Section 4.2.

2 The insertion encoding

Albert, Linton and Ruškuc [2] introduced the *insertion encoding* for permutations and studied the languages which these form, which generalises the finitely labelled generating trees of Chung, Graham, Hoggatt, and Kleiman [9]. The insertion encoding encodes how a permutation is built up by iteratively adding new maxima. The set of all insertion encodings of permutations in a class forms a language. Albert, Linton and Ruškuc [2] studied these languages for permutation classes, in particular, classifying for which permutation classes the set of insertion encodings forms a regular language. They also gave a linear time algorithm in the size of the basis elements for checking whether a finitely based permutation class has a set of insertion encodings which form a regular language. Vatter [26] gave an algorithm for computing a Deterministic Finite Automaton (DFA) for these regular languages which could then be used to compute the rational generating function of the permutation classes. An alternative algorithm using the telescope algorithm [3] was later given, which experimentally appears to be significantly faster.

Unlike permutations, Cayley permutations can have repeated numbers, so in order to generalise this process to build Cayley permutations we will need to allow the insertion of a repeated maximum number. To ensure that it is unique we will build the Cayley permutations by inserting repeated values from left to right, which we call the *vertical insertion encoding* in Section 1. Choosing leftmost values was an arbitrary decision and would work just as well inserting repeated values from right to left.

Tracing how to build a Cayley permutation in this way starting from the empty Cayley permutation is called the *evolution* of a Cayley permutation. In the evolution, we keep track of where future points will be inserted using a symbol ‘ $\diamond$ ’ called a *slot*. For example, the evolution of the Cayley permutation 31232 is shown in Figure 2.

$$\begin{array}{c} \diamond \\ \diamond 1 \diamond \\ \diamond 1 2 \diamond \\ \diamond 1 2 \diamond 2 \\ 3 1 2 \diamond 2 \\ 3 1 2 3 2 \end{array}$$

Figure 2: The evolution of the Cayley permutation 31232.

$$\begin{array}{llll}
\ell: & \diamond & \rightarrow & n \diamond \\
m: & \diamond & \rightarrow & \diamond n \diamond \\
r: & \diamond & \rightarrow & \diamond n \\
f: & \diamond & \rightarrow & n
\end{array}$$

Figure 3: The four ways to insert a number n into a slot.

Each line of the evolution, such as the string $\diamond 12 \diamond$, is called a *configuration*.

For a configuration, we index the slots from left to right, starting with 1. An insertion is determined by three things: the index of the slot inserted into, the way the slot is affected by the insertion, and whether or not the insertion is a repeated current maximum value.

There are four possibilities for how n can be inserted into a slot, which we will represent with the letters ℓ, m, r, f denoting left, middle, right and fill respectively. Figure 3 shows the four ways to replace a slot with a combination of the number n and some new slots.

To encode Cayley permutations, for each insertion we will use letters of the form $a_{i,j}$ with $a \in \{\ell, r, m, f\}$, $i \in \mathbb{N}_1$ is the index of the slot being inserted into, and $j \in \{0, 1\}$ tell us if we are inserting a repeated maximum value (0) or a new maximum value (1). Returning to our example 31232, the vertical insertion encoding for this Cayley permutation will be the word $m_{1,1}\ell_{2,1}r_{2,0}f_{1,1}f_{1,0}$, shown in Figure 4.

$$\begin{array}{ll}
& \diamond \\
& \diamond 1 \diamond & m_{1,1} \\
& \diamond 1 2 \diamond & \ell_{2,1} \\
& \diamond 1 2 \diamond 2 & r_{2,0} \\
& 3 1 2 \diamond 2 & f_{1,1} \\
& 3 1 2 3 2 & f_{1,0}
\end{array}$$

Figure 4: The evolution of a Cayley permutation 31232 with the encoding shown with each insertion.

For a given configuration, not every slot can be inserted into with a repeated value to ensure a unique evolution for every Cayley permutation. For example, for the configuration $\diamond 1$ only new maxima can be inserted, so the only valid letters are $\ell_{1,1}, m_{1,1}, r_{1,1}$ and $f_{1,1}$ giving the configurations $2 \diamond 1$, $\diamond 2 \diamond 1$, $\diamond 21$ and 21 respectively. But for the configuration $1 \diamond$, either a new maximum or a repeated value can be inserted into the slot so any of the letters $\ell_{1,0}, m_{1,0}, r_{1,0}, f_{1,0}, \ell_{1,1}, m_{1,1}, r_{1,1}$ and $f_{1,1}$ can be applied, resulting in the configurations $11 \diamond$, $1 \diamond 1 \diamond$, $1 \diamond 1$, 11 , $12 \diamond$, $1 \diamond 2 \diamond$, $1 \diamond 2$ and 12 respectively.

For a configuration c we call the *derivations* of c the set of all Cayley permutations which have c in its evolution.

2.1 An example - the class $\text{Av}(211, 312)$

In this subsection we will show how to enumerate a class of Cayley permutations using the vertical insertion encoding. In their paper on the insertion encoding of permutations,

Albert, Linton and Ruškuc [2] prove that the permutation class $\text{Av}(11, 312)$ is counted by the Catalan numbers by finding a context-free grammar for it. They also prove that every insertion is in the first slot of a configuration in the evolution of the permutations in $\text{Av}(11, 312)$. We will prove in Proposition 1 that this is also true of the Cayley permutation class $\text{Av}(211, 312)$ for the vertical insertion encoding and show that is context-free by giving a non-ambiguous grammar for it.

Proposition 1. *A Cayley permutation avoids 211 and 312 if and only if its vertical insertion encoding only uses letters of the form $a_{1,j}$ for $a \in \{\ell, r, m, f\}$ and $j \in \{0, 1\}$.*

Proof. Only using letters of this form is equivalent to every insertion being into the first slot of the configuration. We will prove the statement by proving the contrapositive of each direction, first that if the evolution of a Cayley permutation inserts into a slot that is not the first slot then it contains 211 or 312. Let π be a Cayley permutation whose evolution contains an insertion into a slot that is not the first slot. Suppose this slot has index $i > 1$. As there exists at least one value between any two slots in a configuration, there is at least one value between the first slot and the i^{th} slot. This value, together with the value inserted in the i^{th} slot, will form an occurrence of either 12 or 11.

At some point in the evolution, the first slot will be inserted into. The value inserted into the first slot will be strictly larger than all values to the right. As there is an occurrence of either 12 or 11 to the right of the first slot, inserting this new value will create an occurrence of either 211 or 312. As π contains either 211 or 312, $\pi \notin \text{Av}(211, 312)$.

We now prove that if a Cayley permutation contains 211 or 312 then its evolution inserts into a slot that is not the first slot. Suppose π is a Cayley permutation which contains 211 or 312. In both 211 and 312, the middle 1 is the leftmost smallest value so if π contains either of these patterns then this middle 1 of any occurrence was inserted first in the evolution of π .

Suppose in the evolution of π some value n is inserted to create a configuration c and this occurrence of n is at index j in c . For this to later play the role of the middle 1 in either 211 or 312, there must be a slot at an index smaller than j in c . By the rule of inserting smallest leftmost values first, the next value to be inserted for either 211 or 312 will be the 2 in 312 or the other 1 in 211. Both of these must be inserted into a slot at index larger than j in c which is not the first slot, so the evolution of π must insert into a slot that is not the first slot. $\square$

We have the following conditions for the set of vertical insertion encodings of Cayley permutations in $\text{Av}(211, 312)$:

- As usual, when inserting the first value we must insert a new maximum value.
- After an ℓ insertion, the next slot will be on the right of the last value that was inserted, so may be a new maximum or a repeated value.
- After an r insertion, the next slot will be on the left of the last value that was inserted, so must be a new maximum.

- After an m insertion, of the two slots we have inserted the first must begin with a new maximum and the second may be a new maximum or a repeated value.

This gives us the following non-ambiguous context free grammar for the set of vertical insertion encodings of $\text{Av}(211, 312)$. The start symbol is S which represents inserting a new maximum and C represents inserting either a new maximum or a repeated value.

$$\begin{aligned} S &\rightarrow f_{1,1} \mid \ell_{1,1}C \mid r_{1,1}S \mid m_{1,1}SC \\ C &\rightarrow S \mid f_{1,0} \mid \ell_{1,0}C \mid r_{1,0}S \mid m_{1,0}SC \end{aligned}$$

Let $S(x)$ be the generating function for S and $C(x)$ be the generating function for C . Then we have

$$\begin{aligned} S(x) &= x + xC(x) + xS(x) + xS(x)C(x), \\ C(x) &= S(x) + x + xC(x) + xS(x) + xS(x)C(x), \end{aligned}$$

which solves to give the generating function

$$S(x) = \frac{1 - 3x - \sqrt{1 - 6x + x^2}}{4x}$$

for the sequence beginning 1, 3, 11, 45, 197, 903, 4279, 20793, 103049, 518859 which is the little Schroeder numbers, A001003 in the OEIS [25].

3 Classifying regular languages

We will now focus on regular languages. The set of insertion encodings of a Cayley permutation class forms a regular language if it can be represented by a deterministic finite automaton for which there exists some $k \in \mathbb{N}_1$ such that from any state there exists a path of size $\leq k$ to an accept state. Representing states as configurations and accept states as Cayley permutations, from any configuration with s slots the minimum number of insertions needed to obtain a Cayley permutation is s . Hence, for any class with a set of vertical insertion encodings which form a regular language, there is a bound on the number of slots in any configuration in the evolution of any Cayley permutation in the class. We will call such a class *slot-bounded*.

In Section 4, we will show for finitely-based Cayley permutation classes that being slot-bounded is also a sufficient condition for its vertical insertion encodings to form a regular language. We will do this by presenting an algorithm for finding a DFA for the language. In this section, we focus on giving a characterisation of slot-bounded Cayley permutation classes.

Let $\mathcal{SB}(k)$ be the set of Cayley permutations with evolutions that contain configurations with at most k slots. For each k , this is a finitely based Cayley permutation class. The following proposition allows us to compute the minimal Cayley permutations not in the class.

Proposition 2. For each positive integer k , the set $\mathcal{SB}(k)$ is the set of Cayley permutations that avoid all size $2k + 1$ derivations of configurations of the form

$$\diamond a_1 \diamond a_2 \diamond \cdots \diamond a_k \diamond ,$$

where $a_1 a_2 \cdots a_k$ is any size k Cayley permutation.

Proof. Clearly, any Cayley permutation that is a derivation from such a configuration is not in $\mathcal{SB}(k)$.

Take a Cayley permutation π not in $\mathcal{SB}(k)$. The evolution of π must have a configuration with exactly $k + 1$ slots in its evolution as each insertion increases the number of slots by at most 1. First, pick a number between each of the $k + 1$ slots, say $a_1, \dots, a_k$. Second, pick a value of π to fill each slot so that we have $\sigma_1 a_1 \sigma_2 a_2 \cdots a_k \sigma_{K+1}$. These $2k + 1$ numbers will create a Cayley permutation of size $2k + 1$ that is a derivation of the configuration $\diamond a_1 \diamond a_2 \diamond \cdots \diamond a_k \diamond$. $\square$

For $k = 1$, the class $\mathcal{SB}(1)$ contains Cayley permutations whose evolutions have at most 1 slot. By Proposition 2, the basis of $\mathcal{SB}(1)$ is all size 3 derivations of the configuration $\diamond 1 \diamond$, so $\mathcal{SB}(1) = \text{Av}(112, 212, 213, 312)$. This class can also be described as the set of all Cayley permutations whose vertical insertion encodings do not use letters of the form $m_{i,j}$ for any i and j . If the slot in a configuration can have a repeated value inserted into it then we will describe that state as A , otherwise we will describe the state as S . Using this notation and beginning at S , we give the following regular grammar for the set of vertical insertion encodings of $\text{Av}(112, 212, 213, 312)$.

$$\begin{aligned} S &\rightarrow f_{1,1} \mid \ell_{1,1} A \mid r_{1,1} S \\ A &\rightarrow S \mid f_{1,0} \mid \ell_{1,0} A \mid r_{1,0} S \end{aligned}$$

Let $S(x)$ be the generating function for S and $A(x)$ be the generating function for A . Then

$$\begin{aligned} S(x) &= x + xA(x) + xS(x), \\ A(x) &= S(x) + x + xA(x) + xS(x), \\ &= 2S(x), \end{aligned}$$

which solves to give the generating function

$$S(x) = \frac{x}{1 - 3x}$$

for the sequence beginning 1, 3, 9, 27, 81, 243, 729, 2187, 6561, 19683 which is A000244 in the OEIS [25].

For a Cayley permutation class to be slot-bounded it must be a subclass of $\mathcal{SB}(k)$ for some k . In order to determine when this is the case, we need a few definitions. A Cayley permutation π with maximum value $n + k$ is a *vertical juxtaposition* of a Cayley

permutation σ with maximum value n and a Cayley permutation τ with maximum value k if all entries in π with values n or smaller standardise to σ and all entries in π with values larger than n standardise to τ . For example, a vertical juxtaposition of 12432 and 221 is 16624352 as the entries with value four and smaller are 12432 and the entries with value larger than four are 665.

There are nine classes of vertical juxtapositions which are central to our theory. They arise from vertically juxtaposing strictly increasing, strictly decreasing or constant sequences. We will use $\mathcal{V}_{a,b}$ with $a, b \in \{D, I, C\}$ to denote the Cayley permutation class which is the vertical juxtaposition of a strictly increasing sequence if $a = I$, strictly decreasing sequence if $a = D$ and constant sequence if $a = C$ with a strictly increasing sequence if $b = I$, strictly decreasing sequence if $b = D$ and constant sequence if $b = C$. For example, the Cayley permutation 23141 belongs to the class $\mathcal{V}_{C,I}$ and 21122 to $\mathcal{V}_{C,C}$, as shown in Figure 5. Each of I , D and C can be empty so $111 \in \mathcal{V}_{C,C}$ for example.

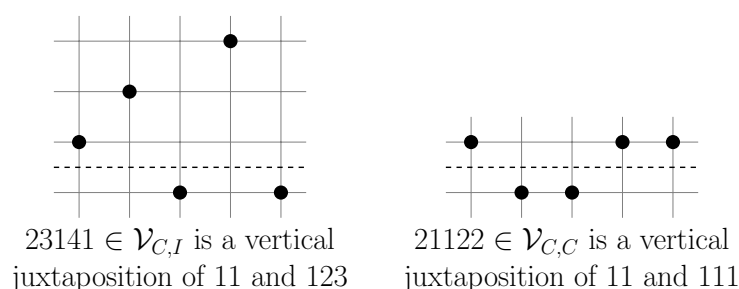


Figure 5: Examples of vertical juxtapositions.

A Cayley permutation can be in more than one of these nine classes, such as 23145. The sequence 2345 is strictly increasing, and the sequence 1 is strictly increasing, strictly decreasing, and constant, hence 23145 is in $\mathcal{V}_{I,I}$, $\mathcal{V}_{D,I}$ and $\mathcal{V}_{C,I}$. This is always the case when the values can be split so that there is a single 1 on its own, hence the Cayley permutations 12 and 21 are in all nine of these classes of vertical juxtapositions. Atkinson [4, Theorem 2.2] showed how to compute the basis for the juxtaposition of two permutation classes, a method which can be adapted to Cayley permutations. Table 2 shows these bases for Cayley permutations.

$A \setminus B$	I	D	C
I	$\text{Av}(11, 321, 2143, 2413)$	$\text{Av}(11, 132, 312)$	$\text{Av}(122, 132, 212, 221, 312, 321)$
D	$\text{Av}(11, 213, 231)$	$\text{Av}(11, 123, 3142, 3412)$	$\text{Av}(122, 123, 212, 213, 221, 231)$
C	$\text{Av}(112, 121, 211, 213, 231, 321)$	$\text{Av}(112, 121, 123, 132, 211, 312)$	$\text{Av}(123, 132, 213, 231, 312, 321)$

Table 2: The bases for the nine classes of vertical juxtapositions $\mathcal{V}_{A,B}$.

A *vertical alternation* is a Cayley permutation $a_1b_1a_2b_2\cdots a_nb_n$ with $b_i > a_j$ for all $1 \leq i \leq n$ and $1 \leq j \leq n$. There is a unique vertical alternation of size $2n$ for each of the nine classes of vertical juxtapositions that we introduced. These have the following useful property.

Lemma 3. *For each $\mathcal{V} \in \{\mathcal{V}_{I,I}, \mathcal{V}_{I,D}, \mathcal{V}_{I,C}, \mathcal{V}_{D,I}, \mathcal{V}_{D,D}, \mathcal{V}_{D,C}, \mathcal{V}_{C,I}, \mathcal{V}_{C,D}, \mathcal{V}_{C,C}\}$, the vertical alternation of size $2n$ in $\mathcal{V}$ contains every vertical juxtaposition in $\mathcal{V}$ of size up to n .*

Proof. Let $a_1b_1a_2b_2\cdots a_nb_n$ be the vertical alternation of size $2n$ in $\mathcal{V}$. Take an arbitrary π of size n in $\mathcal{V}$ that is a vertical juxtaposition of σ and τ . If the i^{th} value of π is from σ select a_i or if it is from τ select b_i . This selection of n numbers will standardise to π as the b_i 's are all greater than the a_i 's. Therefore, the vertical alternation of size $2n$ in $\mathcal{V}$ contains every vertical juxtaposition in $\mathcal{V}$ of size up to n . $\square$

Using Lemma 3 we will show in Theorem 6 that to find when a Cayley permutation class is slot-bounded we need only to find out if its basis contains a vertical juxtaposition from each of our nine classes. In order to prove this, Lemma 4 and Lemma 5 are needed first.

Lemma 4. *A nondecreasing sequence of size m contains a subsequence of size at least $\lceil \sqrt{m} \rceil$ that is either strictly increasing or constant.*

Proof. First, consider the case when $m = n^2$ for some $n \in \mathbb{N}_1$. If there are at least n distinct values then there would be a strictly increasing sequence of size at least n . If there were fewer than n distinct values then there would be at least one value with at least n occurrences in the sequence, forming a constant subsequence of size at least n .

If $n^2 < m < (n+1)^2$ then either there are more than n distinct values, in which case there is a strictly increasing subsequence of size at least $n+1$, or there is at least one value with at least $n+1$ occurrences, in which case there is a constant subsequence of size at least $n+1$. Hence, a strictly increasing sequence of size m , with $n^2 < m < (n+1)^2$ contains either a strictly increasing sequence or a constant sequence of size at least $\lceil \sqrt{m} \rceil$. $\square$

Lemma 4 is a generalisation of the Erdős-Szekeres theorem [12] which states that for $r, s \in \mathbb{N}_1$ any sequence of length $(r-1)(s-1)+1$ contains a subsequence of length r that is nondecreasing or a subsequence of length s that is nonincreasing. Using this theorem we can prove Lemma 5.

Lemma 5. *Any sequence of size n^4 contains a subsequence of size n that is either strictly increasing, strictly decreasing or constant.*

Proof. By the Erdős-Szekeres theorem [12], any sequence of size n^4 contains a subsequence of size n^2 that is nondecreasing or nonincreasing, and by Lemma 4 any nondecreasing (or nonincreasing) sequence of size n^2 contains a subsequence of size n that is either strictly increasing (or strictly decreasing) or constant. $\square$

Using these lemmas, we generalise a proof on permutations by Albert, Linton and Ruškuc [2, Proposition 13] to prove a condition in Theorem 6 for a Cayley permutation class to be a subclass of $\mathcal{SB}(k)$. As the results in Section 4 show that a finitely based Cayley permutation class is regular if and only if it is slot-bounded, Theorem 6 gives a linear time algorithm in the length of the basis elements for checking if a finitely based Cayley permutation class is regular.

Theorem 6. *A Cayley permutation class $\text{Av}(B)$ is a subclass of $\mathcal{SB}(k)$ for some k if and only if B contains a Cayley permutation from each of the nine classes $\mathcal{V}_{I,I}$, $\mathcal{V}_{I,D}$, $\mathcal{V}_{I,C}$, $\mathcal{V}_{D,I}$, $\mathcal{V}_{D,D}$, $\mathcal{V}_{D,C}$, $\mathcal{V}_{C,I}$, $\mathcal{V}_{C,D}$, and $\mathcal{V}_{C,C}$ of vertical juxtapositions.*

Proof. Suppose that $\text{Av}(B) \subseteq \mathcal{SB}(k)$. The nine Cayley permutations of the form

$$b_0 a_1 b_1 \cdots a_k b_k$$

where $a_1 \cdots a_k$ and $b_0 b_1 \cdots b_k$ are each strictly increasing, strictly decreasing or constant are examples from each of the nine types of vertical juxtapositions. They are also not in $\text{Av}(B)$ as they are all derivations of configurations of the form $\diamond a_1 \diamond a_2 \diamond \cdots \diamond a_k \diamond$. As each of these nine forms are downwards closed, B must contain a Cayley permutation from each of the nine classes of vertical juxtapositions.

Suppose $\text{Av}(B)$ is a Cayley permutation class with B containing Cayley permutations from each of the nine classes of vertical juxtapositions. Suppose the longest vertical juxtaposition in B has size n . We will show that there exists some integer k such that every Cayley permutation in the basis of $\mathcal{SB}(k)$ contains a vertical juxtaposition that is contained in B , so $\text{Av}(B) \subseteq \mathcal{SB}(k)$ for this k .

Let $k = n^{16}$ and $\sigma = b_1 a_1 b_2 a_2 b_3 \cdots b_k a_k b_{k+1}$ be a basis element of $\mathcal{SB}(k)$. Although a smaller k may also work, we will show that $k = n^{16}$ is sufficient. The sequence of a 's has size $k = n^{16}$, so by Lemma 5 there exists a subsequence of size n^4 which is either strictly increasing, strictly decreasing or constant. We will denote this as $a_{i_1} a_{i_2} \cdots a_{i_{n^4}}$.

Next, we choose the n^4 values from the b 's that are adjacent to the a_i 's so that they form the pattern $a_{i_1} b_{i_1+1} a_{i_2} b_{i_2+1} \cdots a_{i_{n^4}} b_{i_{n^4}+1}$. As before, among these b_i 's there exists a subsequence of size n that is strictly increasing, strictly decreasing, or constant by Lemma 5. We will denote these $b_{i_{j_1}} b_{i_{j_2}} \cdots b_{i_{j_n}}$. Taking a subset of the a_i 's that remain adjacent to the b_{i_j} 's we create the pattern $a_{i_{j_1}-1} b_{i_{j_1}} a_{i_{j_2}-1} b_{i_{j_2}} \cdots a_{i_{j_n}-1} b_{i_{j_n}}$.

This sequence is a vertical alternation in one of the nine classes of vertical juxtapositions of length $2n$, so by Lemma 3 the sequence contains every vertical juxtaposition from the same class of size up to n , specifically a vertical juxtaposition that is in B . Therefore, this basis element σ of $\mathcal{SB}(k)$ with $k = n^{16}$ contains a basis element from B . As σ was arbitrary, every basis element of $\mathcal{SB}(k)$ contains an element from B , so none of them are in $\text{Av}(B)$, hence $\text{Av}(B) \subseteq \mathcal{SB}(k)$. $\square$

In Section 4 we will give an algorithm for computing a DFA that accepts the regular language for the set of vertical insertion encodings of any finitely based slot-bounded Cayley permutation class. Hence, if a finitely based Cayley permutation class avoids arbitrarily long vertical juxtapositions then we can compute a DFA for it.

4 Computing vertical insertion encodings

Vatter [26] gave a method for constructing a DFA for the insertion encodings of slot-bounded permutation classes. The method from that paper can be extended to slot-bounded Cayley permutation classes. However, we present an alternative approach that is more efficient for permutations and can be applied to Cayley permutations. Our definitions and methods follow those introduced by Albert, Bean, Claesson, Nadeau, Pantone, and Ulfarsson [3] for permutations, but here we will adapt them to Cayley permutations.

The heart of our approach is essentially the same as Vatter's method: consider an infinite accepting automata whose states are configurations with the start state $\diamond$, accept states being Cayley permutations, and transitions being determined by the possible insertions. Vatter outlines a method for determining when a letter can be removed from a configuration without changing the transitions, and calls such a letter *insertion-encoding-reducible* (or *IE-reducible*). From their results, in a slot-bounded permutation class any sufficiently long configuration will have an IE-reducible letter, which in turns tells us that this method will result in a DFA.

To implement this, one needs to verify that any permutation that is a derivation of the configuration that is not in the permutation class is still not in the permutation class after removing the letter. This is often a computationally expensive check. Vatter showed that it is possible to determine if a letter in a configuration is IE-reducible by checking finitely many derivations of the configuration.

We will instead represent configurations as *tilings*. Tilings give a convenient and computer implementable way of representing sets of Cayley permutations and the strategies needed to construct the DFA. To check if a letter is IE-reducible, Vatter's method requires the generation of permutations up to a fixed size with some checks on whether the permutations contain or avoid the basis of the class. Tilings keep track of the occurrences of basis elements as the configurations are expanded, so checks required to determine if a letter is IE-reducible are already done. In fact, by moving to tilings every letter will be IE-reducible, and therefore more states in our DFA are shown to be equivalent. For these reasons, our approach appears to produce significantly fewer states and require less computation time than Vatter's method, which we observed in the computational experiments that we ran.

In Table 3, we compare the number of states in the DFAs found by our implementation of Vatter's method and the tilings approach for the classes $\text{Av}(B)$ for subsets B of the Cayley permutations of size 3. It appears to show that the tilings approach results in a DFA with significantly fewer states.

We will now outline some details of this implementation in the remainder of this section. Much of this is a direct adaptation of the work of Albert *et al.* [3] to the context of Cayley permutations. The material in Section 5.1 is technical with details that are independent of the rest of the paper, so the reader may skip to Section 5.2 on their first pass through this paper if they wish.

$ B $	Vatter's method			The tilings approach		
	Min	Average	Max	Min	Average	Max
3	33	180.7	843	5	17.9	38
4	12	124.1	616	3	15.6	35
5	9	92.2	434	4	13.7	28
6	13	70.7	330	4	12.1	23

Table 3: Comparison of the number of states of the DFA found by Vatter's method and the tilings approach. This is for the classes $\text{Av}(B)$ for subsets B of the Cayley permutations of size 3.

4.1 Tilings

A Cayley permutation π can be represented on a grid such that each index of π is given a position in a cell on the grid. For example, the Cayley permutation 35125224 is shown in Figure 6 with positions $((0, 1), (0, 1), (0, 0), (0, 0), (1, 1), (1, 0), (1, 0), (1, 1))$.

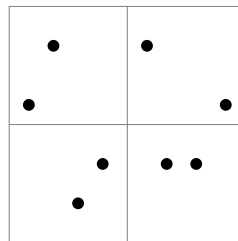


Figure 6: The Cayley permutation 35125224 on a grid with positions $((0, 1), (0, 1), (0, 0), (0, 0), (1, 1), (1, 0), (1, 0), (1, 1))$.

More formally, a Cayley permutation π of size n is given an n -tuple of cells $(c_1, c_2, \dots, c_n)$ with $c_i \in \mathbb{N}_0 \times \mathbb{N}_0$, called the *positions*. This represents a gridding of the points of the Cayley permutation on the positive quadrant; the point $(i, \pi(i))$ with $c_i = (x, y)$ is in the square $[x, x + 1) \times [y, y + 1)$.

We only consider tuples of positions for Cayley permutations which are consistent with the Cayley permutation, i.e., a Cayley permutation $\pi(1)\pi(2)\cdots\pi(n)$ and positions $((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n))$ are consistent if for every pair of indices i and j with $i < j$, we have

- $x_i \leq x_j$,
- if $\pi(i) < \pi(j)$ then $y_i \leq y_j$,
- if $\pi(i) > \pi(j)$ then $y_i \geq y_j$, and
- if $\pi(i) = \pi(j)$ then $y_i = y_j$.

The example shown in Figure 6 is consistent but the Cayley permutation 212 with positions $((0, 1), (1, 0), (1, 0))$ is not consistent, for example.

We define a *gridded Cayley permutation* to be a pair (π, P) of a Cayley permutation π , called the *underlying pattern*, with a tuple of positions P such that π and P are consistent. For example,

$$(35125224, ((0, 1), (0, 1), (0, 0), (0, 0), (1, 1), (1, 0), (1, 0), (1, 1)))$$

is a gridded Cayley permutation shown in Figure 6, but $(212, ((0, 1), (1, 0), (1, 0)))$ is not consistent, so is not a gridded Cayley permutation. For clarity when there are multiple gridded Cayley permutations on the same grid, we will represent each gridded Cayley permutation with lines joining its points. We define the *size* of a gridded Cayley permutation (π, P) to be the size of its underlying pattern π . As a shorthand, when all of the cells in P are the same, say c , then we will write (π, c) .

Let $\mathcal{GC}$ be the set of all gridded Cayley permutations, and $\mathcal{GC}^{(t,u)}$ be the set of gridded Cayley permutations with points in the region $[0, t] \times [0, u]$. A gridded Cayley permutation $(\pi, (c_1, \dots, c_n))$ *contains* a gridded Cayley permutation $(\sigma, (d_1, \dots, d_k))$ if there is a subsequence $\pi(i_1)\pi(i_2)\cdots\pi(i_k)$ of π that standardises to σ , and the position $c_{i_j} = d_j$ for all $1 \leq j \leq k$. As for sets of Cayley permutations, we say that a gridded Cayley permutation avoids a set of gridded Cayley permutations if it avoids every gridded Cayley permutation in the set. Otherwise, we say the gridded Cayley permutation contains the set of gridded Cayley permutations. In Figure 7, the points circled in red show an occurrence of the gridded Cayley permutation $(2311, ((0, 1), (0, 1), (1, 0), (1, 0)))$ in $(35125224, ((0, 1), (0, 1), (0, 0), (0, 0), (1, 1), (1, 0), (1, 0), (1, 1)))$.

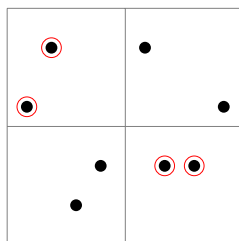


Figure 7: An occurrence of the gridded Cayley permutation $(2311, ((0, 1), (0, 1), (1, 0), (1, 0)))$.

A *(Cayley) tiling* is a triple $\mathcal{T} = ((t, u), \mathcal{O}, \mathcal{R})$ where $(t, u) \in \mathbb{N}_0 \times \mathbb{N}_0$, $\mathcal{O} \subseteq \mathcal{GC}^{(t,u)}$ is a set of gridded Cayley permutations called *obstructions*, and $\mathcal{R} = \{\mathcal{R}_1, \dots, \mathcal{R}_k\}$ is a set of sets of gridded Cayley permutations called *requirements*. Each $\mathcal{R}_i$ is called a *requirement list*. The tiling $\mathcal{T}$ represents the set $\text{Grid}(\mathcal{T})$ of gridded Cayley permutations in $\mathcal{GC}^{(t,u)}$ that avoid $\mathcal{O}$ and contain each $\mathcal{R}_i$.

For a Cayley permutation class $\text{Av}(B)$, the set of derivations from the configuration $\diamond$ is the same as the set of non-empty Cayley permutations in $\text{Av}(B)$. There exists a size-preserving bijection between the set $\text{Av}(B)$ and the gridded Cayley permutations represented by the 1×1 tiling that avoids each pattern in B in the cell $(0, 0)$ while containing a point in cell $(0, 0)$, as stated in Proposition 7.

Proposition 7. For a Cayley permutation class $\text{Av}(B)$, the set of derivations $D_{\text{Av}(B)}(\diamond)$ is in bijection with the gridded Cayley permutations represented by the tiling

$$((1, 1), \{(\pi, (0, 0)) \mid \pi \in B\}, \{(1, (0, 0))\}).$$

We usually represent obstructions in red with circles and requirements in blue with squares, as in Figure 8 which shows the tiling representing the set of derivations from the configuration $\diamond$ in the class $\text{Av}(11, 231, 321)$. Every basis element is represented as an obstruction in the cell $(0, 0)$ and the requirement represents a slot.

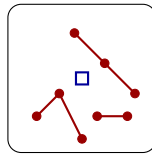


Figure 8: The tiling representing the set of derivations from the configuration $\diamond$ in the class $\text{Av}(11, 231, 321)$.

Every configuration can also be represented as a tiling. We effectively “draw” a configuration on a grid where each placed point in the configuration is represented by a cell in the tiling, point requirements are used to represent slots, and all other cells contain point obstructions to ensure that the order of points being placed in the tiling is consistent with the vertical insertion encoding. This creates an *intermediate tiling* representing the configuration. On top of this we add obstructions representing the basis of the class to ensure that the gridded Cayley permutations represented by the tiling are exactly the derivations from the configuration that are in the class.

For a configuration with n points already placed with m different values and k slots, we create a size $(n + k) \times (m + 1)$ tiling. Each placed point with value v at index i in the configuration is represented by a *point cell* in the cell (i, v) . This consists of the following.

- A requirement list containing the point requirement $\{(1, (i, v))\}$.
- Point obstructions in every other cell in column i to ensure that there are no other values in that column. That is, for every $u \in [m + 1]$ with $u \neq v$, add the obstruction $(1, (i, u))$.
- The obstructions with underlying Cayley permutations 12 and 21 and positions in cells (x, v) for all $x \in [n + k]$ to ensure that there can be no other values in that row.
- The obstruction $(11, (i, v))$ to ensure that there is exactly one point in the point cell.
- For each cell (x, y) such that $x < i$ and $y = v$, if (x, y) is not also a point cell then add the obstruction $(1, (x, y))$. This ensures that no occurrences of the same value can appear to the left of the placed point, following the rules of vertical insertion encoding.

Each slot at index i in the configuration is represented by a list requirement $\mathcal{R}$ containing the requirements $(1, (i, y))$ for each $y \in [m + 1]$ such that (i, y) does not contain a point obstruction. If i is to the left of the largest, rightmost point which has been placed in the configuration, then this will be a size 1 list as the only option for this slot is to insert a new maximum value. Otherwise, it will be a size 2 list, representing the options of inserting a new maximum value or a repeated value. The intermediate tiling for the configuration $2 \diamond 2 \ 1 \diamond$ is shown in Figure 9.

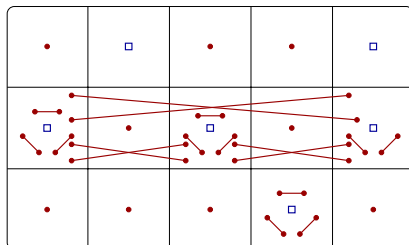


Figure 9: The intermediate tiling representing the configuration $2 \diamond 2 \ 1 \diamond$.

Formally, this tiling is $\mathcal{T} = ((2, 5), \mathcal{O}, \{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \mathcal{R}_4, \mathcal{R}_5\})$ where

- $\mathcal{O} = \{(1, (0, 0)), (1, (1, 0)), (1, (2, 0)), (1, (4, 0)), (1, (1, 1)), (1, (3, 1)), (1, (0, 2)), (1, (2, 2)), (1, (3, 2)), (11, (0, 1)), (12, (0, 1)), (21, (0, 1)), (11, (2, 1)), (12, (2, 1)), (21, (2, 1)), (11, (3, 0)), (12, (3, 0)), (21, (3, 0)), (12, ((0, 1), (2, 1))), (21, ((0, 1), (2, 1))), (12, ((0, 1), (4, 1))), (21, ((0, 1), (4, 1))), (12, ((2, 1), (4, 1))), (21, ((2, 1), (4, 1))), (12, (4, 1)), (21, (4, 1))\}$
- $\mathcal{R}_1 = \{(1, (0, 1))\}$,
- $\mathcal{R}_2 = \{(1, (2, 1))\}$,
- $\mathcal{R}_3 = \{(1, (3, 0))\}$,
- $\mathcal{R}_4 = \{(1, (1, 2))\}$, and
- $\mathcal{R}_5 = \{(1, (4, 1)), (1, (4, 2))\}$.

To reduce the notation in tilings, point cells will be represented by a black dot replacing the obstructions and requirement list in that cell, as shown in cells $(0, 1)$, $(2, 1)$ and $(3, 0)$ in Figure 10. Additionally, the size 2 obstructions in rows containing a point cell will be omitted for clarity and represented by a $*$ beside that row in the tiling, also shown in Figure 10. These rows are called *point rows*. Note that apart from the top row of a tiling every other row will be a point row.

•	◻	•	•	◻	
●	•	●	•	◻	*
•	•	•	●	•	*

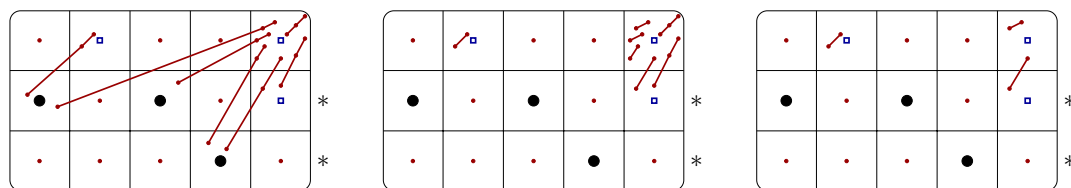
Figure 10: Intermediate tiling representing the configuration $2 \diamond 2 1 \diamond$ with point cells in black and point rows denoted $*$.

From this intermediate tiling, for a class $\text{Av}(B)$ we add all of the gridded Cayley permutations in $\mathcal{GC}^{(5,3)}$ with an underlying Cayley permutation in B as obstructions. Albert *et al.* [3] outlined how effective keeping track of occurrences of basis patterns can be in the context of permutation classes. We also simplify tilings where possible to create equivalent tilings. We say that two tilings $\mathcal{T}_1$ and $\mathcal{T}_2$ are *equivalent*, denoted $\mathcal{T}_1 \cong \mathcal{T}_2$ if there exists a size preserving bijection between the gridded Cayley permutations in $\text{Grid}(\mathcal{T}_1)$ and in $\text{Grid}(\mathcal{T}_2)$. A tiling $\mathcal{T}_1$ can often be simplified to an equivalent tiling $\mathcal{T}_2$ with fewer obstructions and requirements by the following.

1. If an obstruction contains another obstruction, remove the larger obstruction. Avoiding the smaller one is sufficient to avoid the larger one.
2. If an obstruction has a position in a point cell, then that position and index in the underlying pattern can be removed from the obstruction.
3. If a row or column in a tiling contains point obstructions in every cell, then that row or column can be removed from the tiling.
4. If there exists a point obstruction $g = (1, (x, y))$ and there exists a requirement list $\mathcal{R}$ containing g , then g is removed from $\mathcal{R}$. The obstruction implies that a point in that cell would create an occurrence of a basis pattern and a different requirement in the list must be satisfied. If there are no other requirements in $\mathcal{R}$, then the tiling represents a configuration with no derivations in the class.

By the first simplification, for the class $\text{Av}(123)$ we need only add obstructions with pattern 123 and positions in cells which do not contain point obstructions. Further, obstructions can have at most one index in point cells as they contain all size 2 obstructions and at most one value in point rows. Figure 11a shows the tiling after adding the basis pattern 123 in every possible way after this first simplification.

By the second simplification, any obstructions with an index in a point cell can be simplified to a smaller obstruction. This is done in Figure 11b. And finally, the obstructions can be simplified again by containment, as in Figure 11c. More details of simplifying tilings are given in Section 6.3.2 of Albert *et al.* [3], with some additional care needed when considering point rows. However, the above simplifications are sufficient for our purposes.



(a) Add the basis pattern 123 as an obstruction in every way. (b) Remove indices of basis patterns which are in point contained by other obstructions. (c) Simplify obstructions.

Figure 11: Adding basis elements from the class $\text{Av}(123)$ to an intermediate tiling as obstructions and simplifying.

We will now explain how a Cayley permutation class can be enumerated using tilings, using the class $\text{Av}(123)$ as an example. Begin with the tiling representing the configuration $\diamond$ for the class, as shown in Figure 12 for our example.

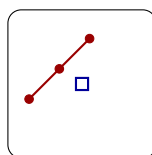


Figure 12: The tiling representing the the configuration $\diamond$ in the class $\text{Av}(123)$.

For each tiling, an insertion can be made into any cell containing a point requirement as these represent the slots in the configuration. From $\diamond$, there are four possible insertions that can be made; $f_{1,1}$, $\ell_{1,1}$, $m_{1,1}$, and $r_{1,1}$. Their intermediate tilings are shown in Figure 13. Each intermediate tiling is size 3×2 with a point cell in position $(1,0)$ representing the newly added point and all the obstructions and requirements necessary for a point cell. For each type of insertion ℓ , m , r , and f , we add the gridded Cayley permutations $g_1 = (1, (0, 1))$, $g_2 = (1, (2, 0))$, and $g_3 = (1, (2, 1))$ as either obstructions or requirements. For an f insertion the slot has been filled so both the first and third column are empty, so all of g_1 , g_2 , and g_3 are added as obstructions. For an ℓ insertion, g_1 is added as an obstruction and $R_2 = \{g_2, g_3\}$ is added as a requirement list, as the first column must be empty but the third column must contain a point. Similarly, for an r insertion, $R_1 = \{g_1\}$ is added as a requirement list and g_2 and g_3 are added as obstructions, and for an m insertion both R_1 and R_2 are added as requirement lists.

For each of these intermediate tilings, we add all of the gridded Cayley permutations in $\mathcal{GC}^{(3,2)}$ with an underlying Cayley permutation in $B = \{123\}$ as obstructions and simplify as before. These four tilings are shown in Figure 14 and represent all of the possible configurations after one insertion into the configuration $\diamond$ in the class $\text{Av}(123)$. These tilings are called the *children* of the original tiling, which is called the *parent* tiling. We only consider a tiling to be a child of it's parent if it has derivations in the class (see simplification 4). Together they form a *rule*, mapping the parent to its children which

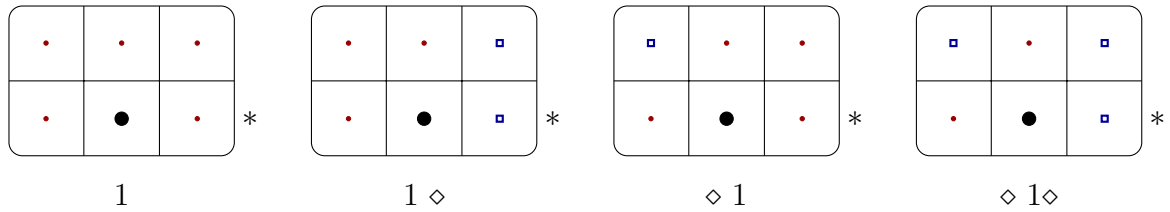


Figure 13: The four intermediate tilings representing configurations after one insertion into $\diamond$ with the configuration they represent underneath.

represent all possible configurations after one insertion. This is shown in Figure 14 with a label for each tiling.

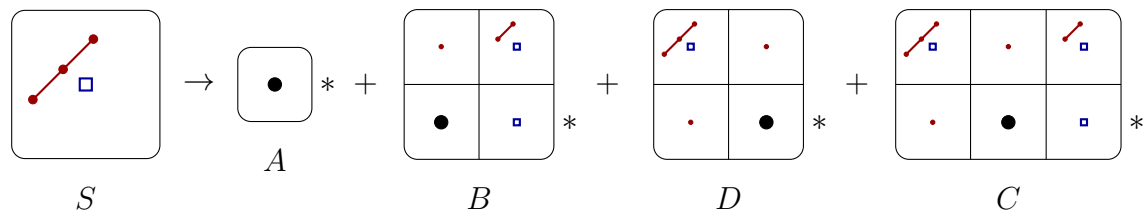


Figure 14: A rule showing the parent tiling representing $\diamond$ and its children representing all possible configurations after one insertion in the class $\text{Av}(123)$.

We can keep going in this manner by inserting the leftmost minimum point into each new tiling which contains requirements. If our letter is of the form $x_{i,0}$ then we place the point in a point row and if our letter is of the form $x_{i,1}$ then we place the point in a row which is not a point row. For example, the children of the tiling labelled B in Figure 14 after simplification are shown in Figure 15. The configuration B represents, $1 \diamond$, has eight possible insertions; $f_{1,0}$, $\ell_{1,0}$, $r_{1,0}$, $m_{1,0}$, $f_{1,1}$, $\ell_{1,1}$, $r_{1,1}$, and $m_{1,1}$. The children of B are shown in the same respective order.

As we keep track of occurrences of basis patterns through obstructions, the point cells do not add any information so we can remove them from our tilings. This is called *factoring* in Albert *et al.* [3]. We can apply the same factoring conditions that they use to our tilings, except that we can additionally factor cells in point rows. Every point cell and the column it is in can be factored out from the tiling. By the third simplification, any row which contains point obstructions in every cell can be removed from a tiling. Figure 16 shows the factorisation of some tilings found so far for the class $\text{Av}(123)$.

Factoring always creates a collection of 1×1 tilings containing only point cells and at most one tiling with no point cells and a requirement list in every column, representing slots in a configuration which we call the *factored tiling*. Additionally, as seen in Figure 16, the factored tilings are often the same even though the tiling represents different configurations. This allows us to reduce the number of tilings we need to consider. As tilings are also the states in the DFA created, this also reduces the number of states in the DFA we construct. In practice, the algorithm works directly on the factored tilings

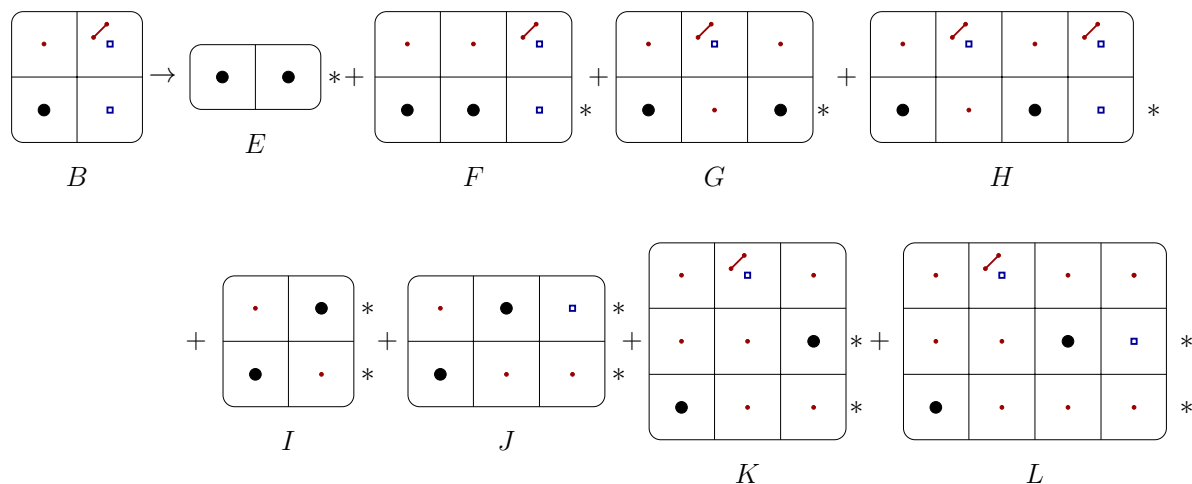


Figure 15: Another rule for the class $\text{Av}(123)$.

rather than considering the configurations they represent so the process is more efficient, but we will not go into the details here. For more information on how this works, please see Albert *et al.* [3].

By design factored tilings will never have more than two rows and for k -slot-bounded classes will never have more than k columns. Moreover, as the basis patterns are a fixed size and there is a finite number of ways in which they can appear on a tiling, there are finitely many tilings of a given size. This proves that continuing this process will eventually terminate and result in a DFA which recognises the vertical insertion encodings of the slot-bounded Cayley permutation class of interest. In the next section we give a full example of a DFA constructed using this method for the class $\text{Av}(231, 312, 2121)$.

4.2 Implementation of the algorithm

We have implemented the algorithm described in Section 4.1 in Python. For the horizontal insertion encoding, which will be described in Section 5, we have also implemented a similar algorithm. Both implementations can be found on GitHub [5] and can be used to enumerate any class of Cayley permutations with a set of vertical or horizontal insertion encodings that form a regular language. For example, as discussed in Section 1, Cerbai [8] found the class of hare pop-stack sortable Cayley permutations to be $\text{Av}(231, 312, 2121)$. Our implementation of the algorithm can be used to find a set of 9 rules for this class shown in Figures 17, 18, 19, 20, 21, 22, 23, 24, and 25. Each figure shows the rule with unfactored tilings to make it clear which insertion corresponds to which tiling and with factored tilings to make it clear which fully factored tilings are of interest for the class. The corresponding rule in the grammar is also shown.

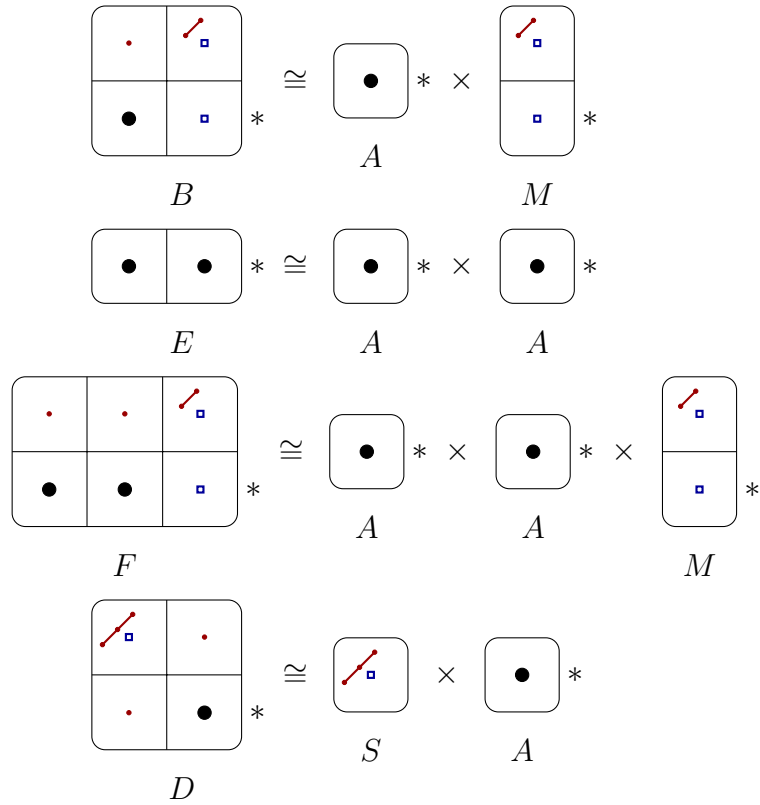


Figure 16: Examples of factorising tilings.

These 9 rules can be turned into the following grammar for the class.

$$\begin{aligned}
S &\rightarrow f_{1,1} \mid \ell_{1,1}A \mid r_{1,1}B \mid m_{1,1}C \\
A &\rightarrow f_{1,1} \mid \ell_{1,1}A \mid r_{1,1}B \mid m_{1,1}C \mid f_{1,0} \mid \ell_{1,0}A \mid r_{1,0}B \mid m_{1,0}C \\
B &\rightarrow f_{1,1} \mid \ell_{1,1}D \mid r_{1,1}B \mid m_{1,1}E \\
C &\rightarrow f_{2,0}B \mid \ell_{2,0}C \mid f_{1,1}A \mid \ell_{1,1}F \mid r_{1,1}G \mid m_{1,1}H \\
D &\rightarrow f_{1,0} \mid \ell_{1,0}D \\
E &\rightarrow f_{2,0}B \mid \ell_{2,0}E \\
F &\rightarrow f_{1,0}A \mid \ell_{1,0}F \\
G &\rightarrow f_{1,1}A \mid \ell_{1,1}F \mid r_{1,1}G \mid m_{1,1}H \\
H &\rightarrow f_{2,0}F \mid \ell_{1,0}F
\end{aligned}$$

Let $S(x)$ be the generating function for S , $A(x)$ be the generating function for A , and so on. Then we have the following system of equations.

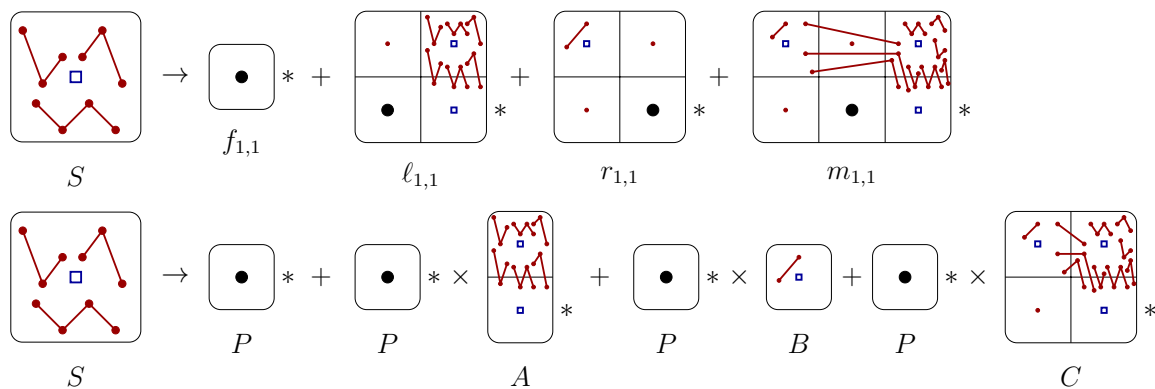


Figure 17: $S \rightarrow f_{1,1} \mid \ell_{1,1} A \mid r_{1,1} B \mid m_{1,1} C$

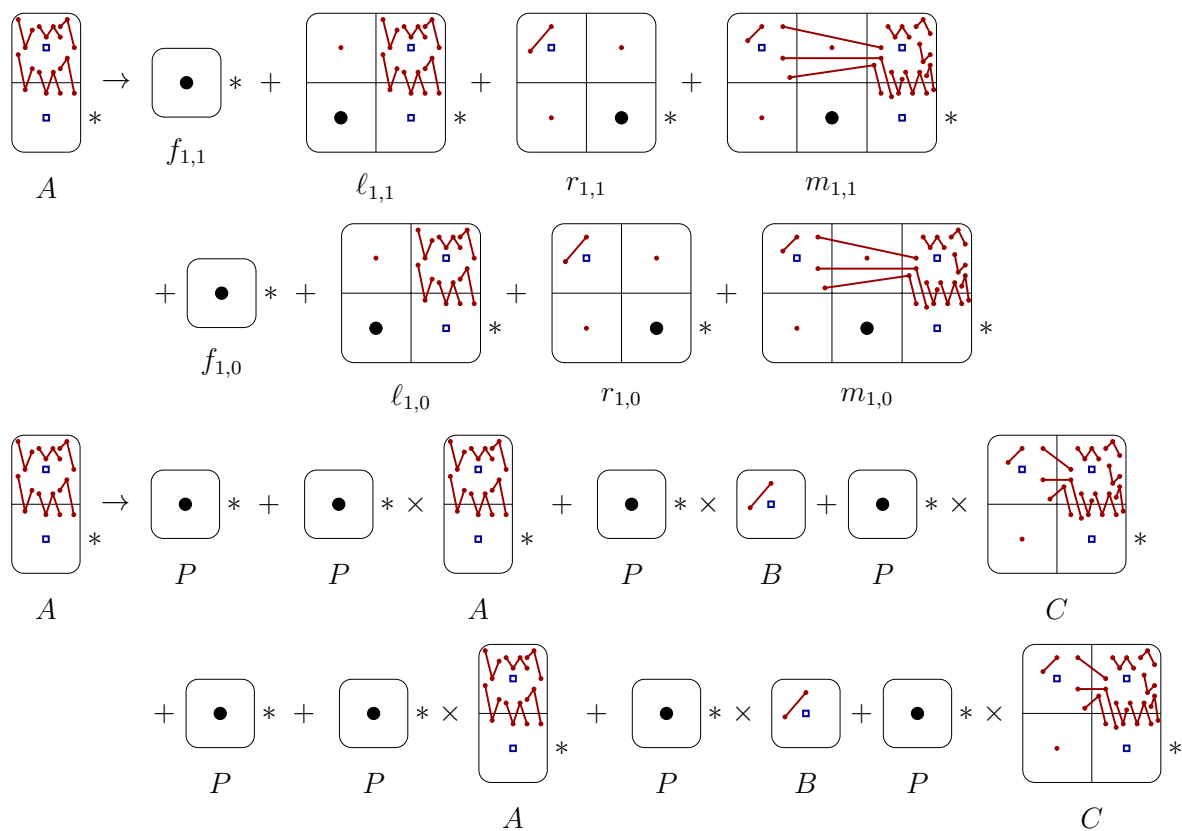


Figure 18: $A \rightarrow f_{1,1} \mid \ell_{1,1} A \mid r_{1,1} B \mid m_{1,1} C \mid f_{1,0} \mid \ell_{1,0} A \mid r_{1,0} B \mid m_{1,0} C$

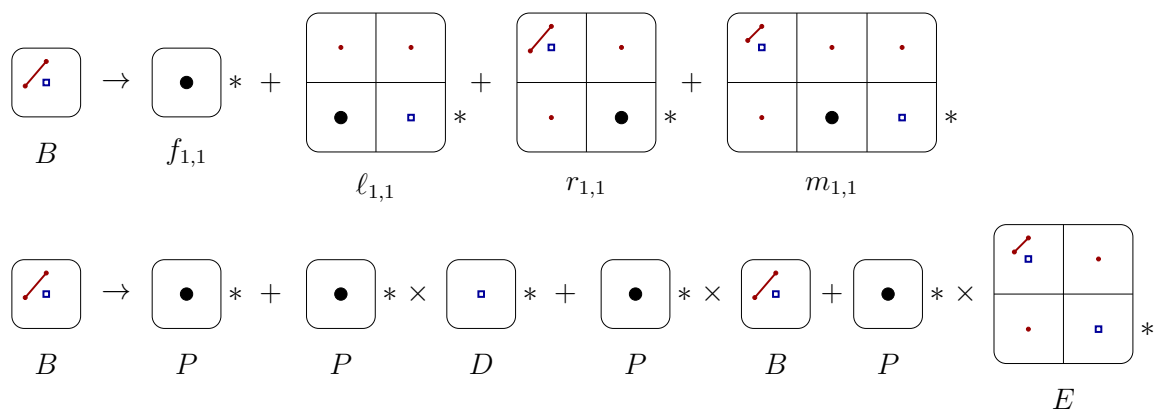


Figure 19: $B \rightarrow f_{1,1} \mid \ell_{1,1}D \mid r_{1,1}B \mid m_{1,1}E$

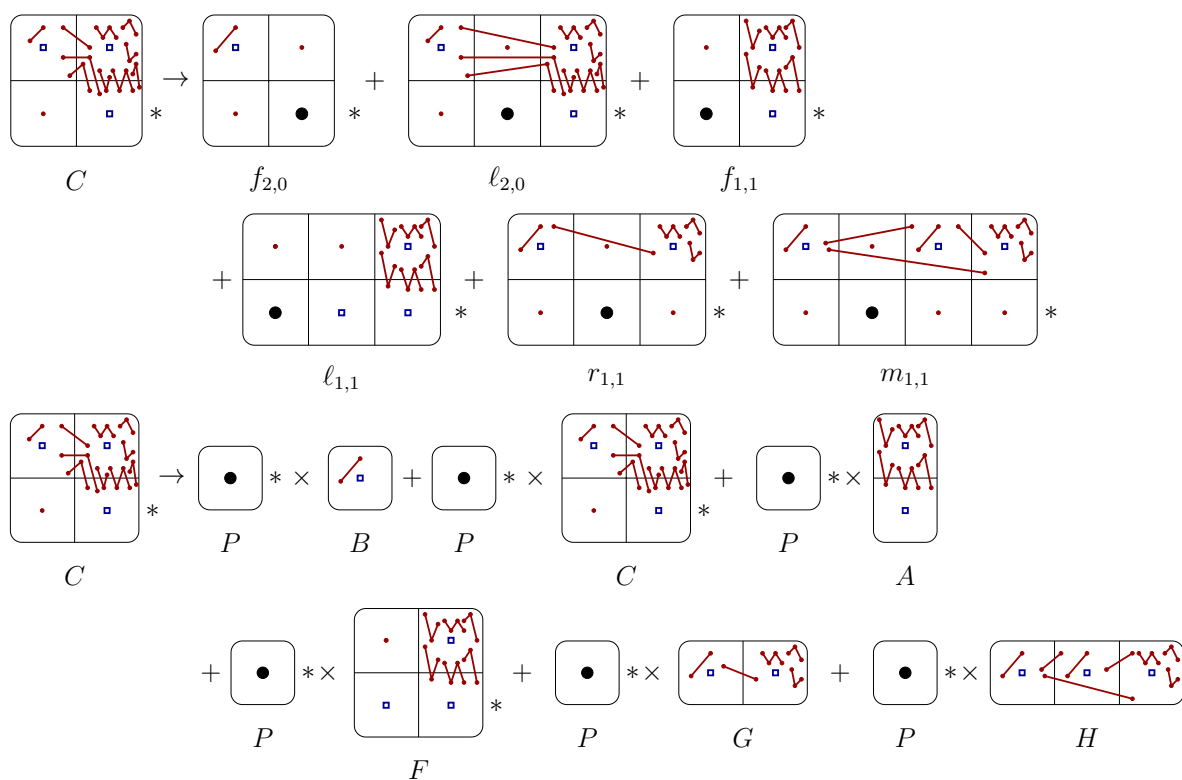


Figure 20: $C \rightarrow f_{2,0}B \mid \ell_{2,0}C \mid f_{1,1}A \mid \ell_{1,1}F \mid r_{1,1}G \mid m_{1,1}H$

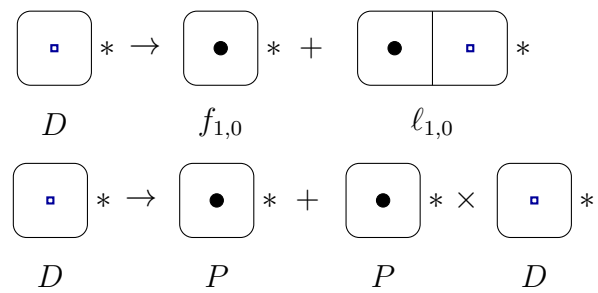


Figure 21: $D \rightarrow f_{1,0} \mid \ell_{1,0}D$

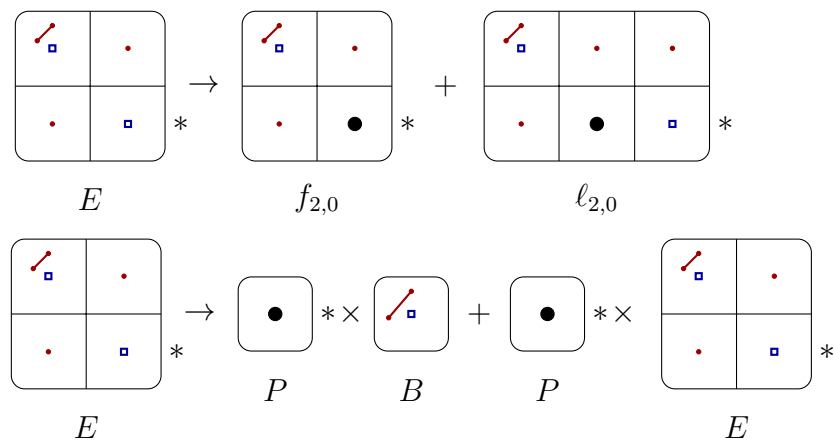


Figure 22: $E \rightarrow f_{2,0}B \mid \ell_{2,0}E$

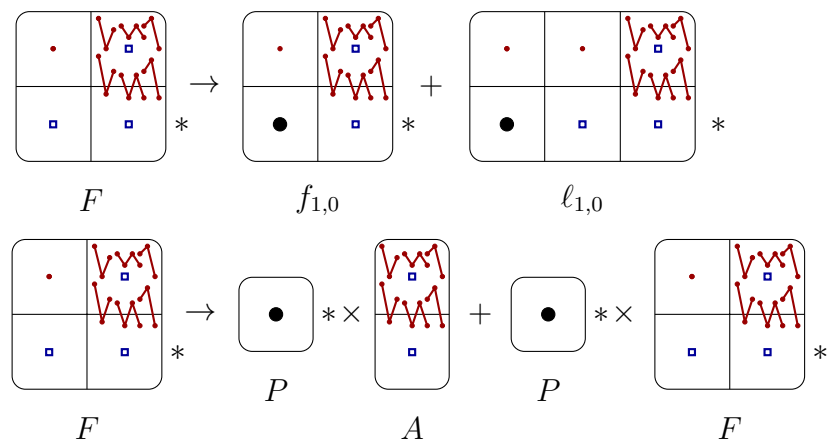


Figure 23: $H \rightarrow f_{1,0}A \mid \ell_{1,0}H$

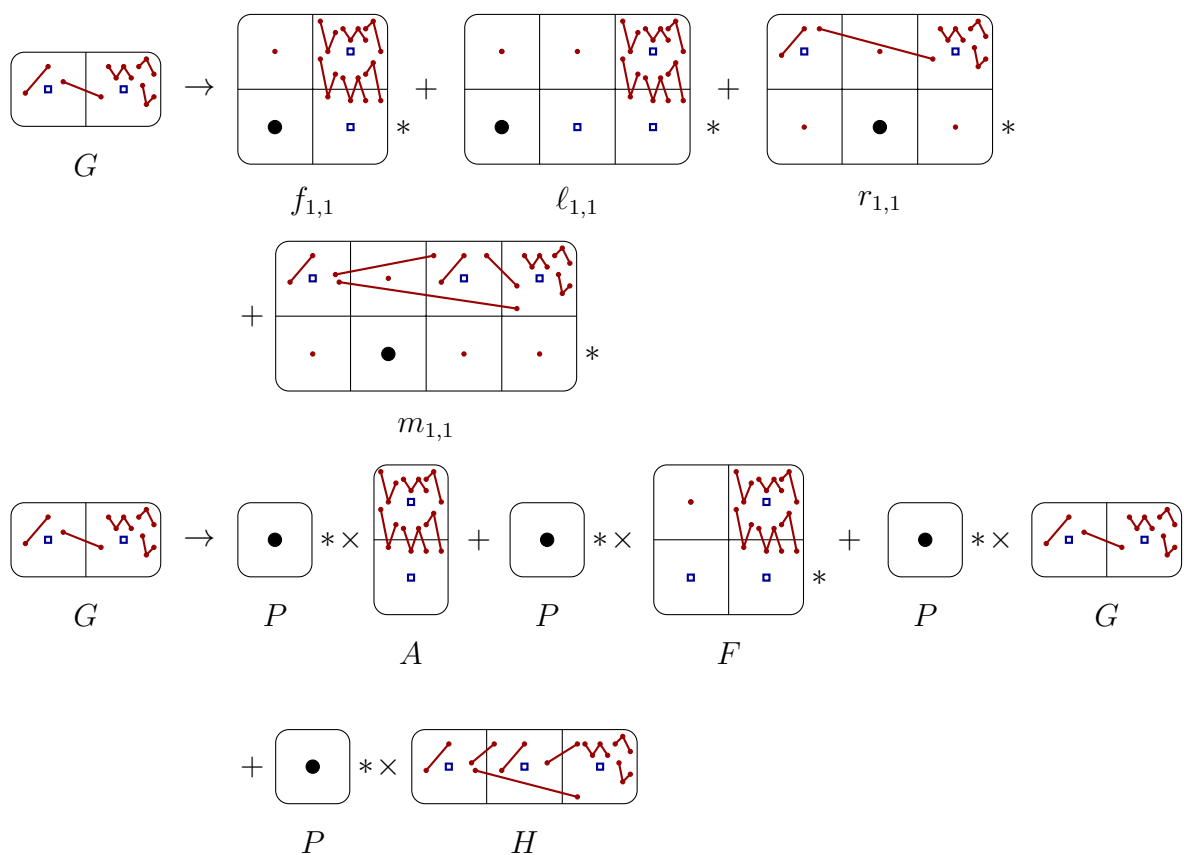


Figure 24: $G \rightarrow f_{1,1}A \mid l_{1,1}F \mid r_{1,1}G \mid m_{1,1}H$

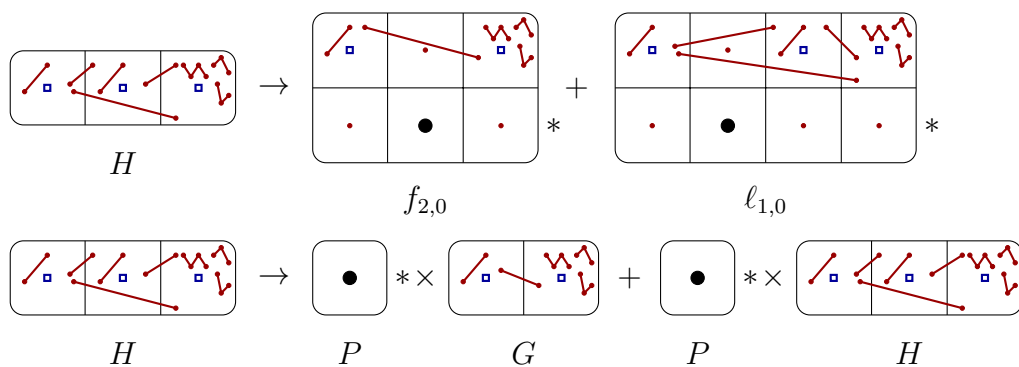


Figure 25: $H \rightarrow f_{2,0}F \mid l_{1,0}F$

$$\begin{aligned}
S(x) &= x + xA(x) + xB(x) + xC(x) \\
A(x) &= x + xA(x) + xB(x) + xC(x) + x + xA(x) + xB(x) + xC(x) \\
B(x) &= x + xD(x) + xB(x) + xE(x) \\
C(x) &= xB(x) + xC(x) + xA(x) + xF(x) + xG(x) + xH(x) \\
D(x) &= x + xD(x) \\
E(x) &= xB(x) + xE(x) \\
F(x) &= xA(x) + xF(x) \\
G(x) &= xA(x) + xF(x) + xG(x) + xH(x) \\
H(x) &= xA(x) + xF(x)
\end{aligned}$$

This solves to give the generating function $S(x)$ for $\text{Av}(231, 312, 2121)$

$$S(x) = \frac{2x^2 - 2x^3 - x}{4x^3 - 6x^2 + 5x - 1}.$$

This is the generating function for the sequence beginning 1, 3, 11, 41, 151, 553, 2023, 7401, 27079, 99081 which is sequence A335793 in the OEIS [25].

5 Horizontal insertion encoding

Unlike for permutations, Cayley permutations are not symmetric under inverses so the rules that govern inserting new rightmost values are different from those of inserting new maximum values. In this section, we will define the *horizontal insertion encoding* that instead traces how Cayley permutations are built from left to right. This is strictly different from the vertical insertion encoding for Cayley permutations although the vertical and horizontal versions are equivalent under the inverse symmetry for permutations.

In this section, we will also classify when the set of horizontal insertion encodings of a class is regular. A similar algorithm as in Section 4 can be used to enumerate Cayley permutation classes with horizontal insertion encodings which form a regular language. Although we will not describe it in this paper, we have an implementation for this on GitHub [5] as well.

The set of Cayley permutations can be built by iteratively adding new rightmost values. As Cayley permutations can have repeated values, the insertion of the new rightmost number can be either a new value or a repeated value already in the Cayley permutation. The *(horizontal) evolution* traces how a Cayley permutation is built in this way, starting from the empty Cayley permutation. Again, we use $\diamond$ to denote where future points will be inserted. To denote that the value to come is a repeated value, we use $\bar{\diamond}$. We call $\diamond$ a *new slot* and $\bar{\diamond}$ a *repeating slot*. For example, the evolution of the Cayley permutation 243441 is shown in Figure 26.

Again, we call each step of the evolution a *configuration* and call the points in a configuration c which have already been placed the *prefix* of c . The prefix of a configuration

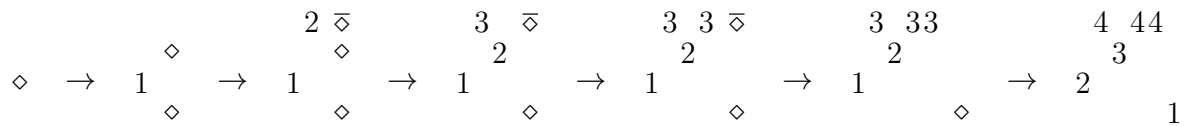


Figure 26: The horizontal evolution of the Cayley permutation 243441.

always forms a Cayley permutation. An insertion in a conconfiguration is determined by the index of the slot inserted into, the way the slot is effected by the insertion, and whether or not there are more of the same value to be inserted.

We use the letters u , d , m , and f denoting up, down, middle, and fill respectively. For each insertion, we use the letter $a_{i,j}$ with $a \in \{u, d, m, f\}$, $i \in \mathbb{N}_1$ as the index of the slot being inserted into counting from bottom to top, and $j \in \{0, 1\}$ to tell us if there are more of the same value still to be inserted. Only f can be inserted into a repeating slot. Table 4 shows all possible insertions of a value n into a slot at index i .

$u_{i,0}:$	$\diamond \rightarrow n$	$u_{i,1}:$	$\diamond \rightarrow n \overline{\diamond}$
	$\diamond$		$\diamond$
$m_{i,0}:$	$\diamond \rightarrow n$	$m_{i,1}:$	$\diamond \rightarrow n \overline{\diamond}$
	$\diamond$		$\diamond$
$d_{i,0}:$	$\diamond \rightarrow n$	$d_{i,1}:$	$\diamond \rightarrow n \overline{\diamond}$
$f_{i,0}:$	$\diamond \rightarrow n$	$f_{i,1}:$	$\diamond \rightarrow n \overline{\diamond}$
$f_{i,0}:$	$\overline{\diamond} \rightarrow n$	$f_{i,1}:$	$\overline{\diamond} \rightarrow n \overline{\diamond}$

Table 4: Types of insertions of the value n into a slot at index i . Only $f_{i,0}$ and $f_{i,1}$ can insert into $\overline{\diamond}$.

For example, the horizontal insertion encoding for the Cayley permutation 243441 is the word $m_{1,0}u_{2,1}f_{2,0}f_{2,1}f_{2,0}f_{1,0}$ and the evolution is shown in Figure 27.

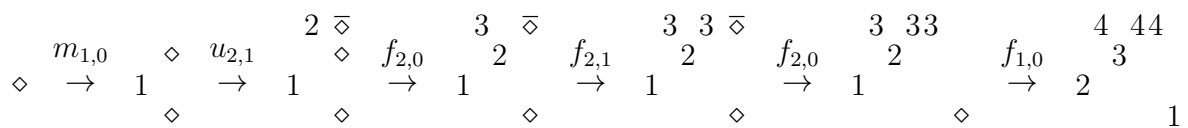


Figure 27: The horizontal evolution of the Cayley permutation 243441 with the encoding shown with each insertion.

As for vertical evolutions, for a configuration c , we call the Cayley permutations which have c as a configuration in its horizontal evolution the derivations of c , denoted $D(c)$.

Once again, for any configuration with s slots, the minimum number of insertions needed to create a Cayley permutation is s . Therefore, for any Cayley permutation class

with a set of horizontal insertion encodings which is regular there exists a bound k on the number of slots in any configuration leading to a Cayley permutation in the class. We call such classes *(horizontal) slot-bounded*. Being slot-bounded is also a sufficient condition for a Cayley permutation class to have a set of horizontal insertion encodings which is regular. This can be shown by adapting the algorithm in Section 4 to the horizontal case, which we will not do here for brevity. In this section, we will give a characterisation of the slot-bounded Cayley permutation classes.

Let $\mathcal{HSB}(k)$ be the set of Cayley permutations with evolutions that contain configurations with at most k slots. We will show that, as with the vertical case, for each k this is a finitely-based Cayley permutation class by proving certain properties of configurations with k slots, leading to Proposition 9 which describes the basis of $\mathcal{HSB}(k)$.

First, note that if a Cayley permutation has more than k slots in its evolution then it must contain a configuration with exactly $k + 1$ slots since it is only possible to reduce the number of slots by one with an $f_{i,0}$ insertion. For a configuration with $k + 1$ slots, the slots could either be a $\diamond$ or a $\bar{\diamond}$ slot. For every repeating slot, there is at least one occurrence of the same value in the prefix of the configuration and for every pair of new slots there must be at least one occurrence of a value between them in the prefix, as shown in Figure 28.

$$\begin{array}{c} \vdots \\ \diamond \\ \diamond \\ \vdots \end{array} \implies n \begin{array}{c} \vdots \\ \diamond \\ \diamond \\ \vdots \end{array} \quad \text{and} \quad \begin{array}{c} \vdots \\ \bar{\diamond} \\ \vdots \end{array} \implies n \begin{array}{c} \vdots \\ \bar{\diamond} \\ \vdots \end{array}$$

Figure 28: Slots imply the existence of values in the prefix of a configuration.

The set $\mathcal{HSB}(k)$ can be defined by avoiding all derivations of configurations with $k + 1$ slots. To give a characterisation of these configurations, we will first define *subconfigurations*. Let c be a configuration and c' be a configuration that can be obtained from c by deleting elements of the prefix of c (and standardising if necessary). In this process of deleting points, repeating slots become new slots if all the corresponding values in the prefix are removed. If enough points are removed from the prefix then slots may also be standardised to have different heights, but the number of slots and the relative heights of the slots will not change. We say that c' is a *subconfiguration* of c , and we have the following useful result. Note, in our definition, we will have the same number of slots at the same relative values as the configuration, as we are only allowing removal of values in the prefix. Removing some values in the prefix could result in something which is not a configuration, in which case, we do not consider this a subconfiguration.

Lemma 8. *Let c be a configuration and c' be a subconfiguration of c . Then for every derivation $\sigma \in D(c)$ there exists a derivation $\sigma' \in D(c')$ such that σ contains σ' .*

Proof. Let c be a configuration with insertion encoding w and c' be the subconfiguration of c with insertion encoding w' . Then, the insertion encoding of a Cayley permutation

$\sigma \in D(c)$ can be written as wv . Let $\sigma' \in D(c')$ be the Cayley permutation with insertion encoding $w'v$. As the prefix of c' is contained in the prefix of c at the same values, and the height of the slots are the same, it follows that σ contains σ' . $\square$

Given a configuration c and subconfiguration c' whose derivations must be avoided, this result tells us that avoiding the derivations $D(c)$ is implied by avoiding the derivations of $D(c')$. We're now ready to state and prove the result.

Proposition 9. *For each integer k , let C_k be the set of configurations with $k + 1$ slots such that the following conditions are fulfilled.*

- *The prefix of c is a permutation.*
- *There are no values in the prefix of c larger than the top slot or smaller than the bottom slot.*
- *There are no values in the prefix which are strictly between a repeating slot and another slot.*
- *There are no three consecutive repeating slots in c .*
- *There are no two consecutive repeating slots at the top or bottom.*

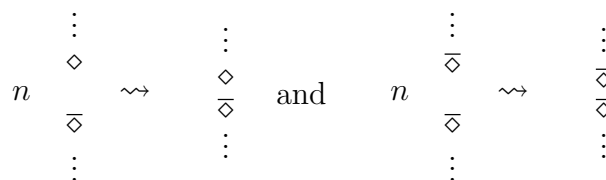
Then $\mathcal{HSB}(k)$ is the Cayley permutation class whose basis is the set of minimal Cayley permutations in the derivations of each configuration in $c \in C_k$ of size $|c|$.

Proof. The basis elements of $\mathcal{HSB}(k)$ are the minimal Cayley permutations which have an evolution containing a configuration with $k + 1$ slots, which is a superset of C_k . We will prove that for any configuration which has $k + 1$ slots that does not satisfy the conditions there exists a smaller subconfiguration. By Lemma 8, avoiding the derivations of a configuration is implied by avoiding the derivations of a subconfiguration, so we can ignore the configuration and work with the subconfiguration instead.

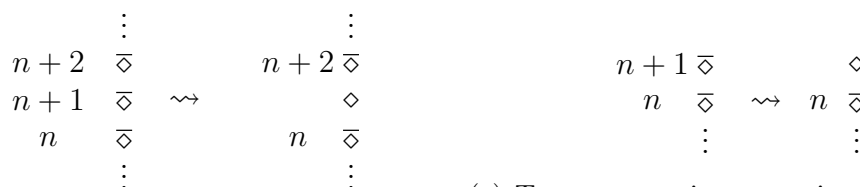
Let c be a configuration with $k + 1$ slots.

- Assume that the prefix of c is not a permutation. Let c' be the subconfiguration obtained by deleting every index that is not the first occurrence of each value. Then c' is a subconfiguration of c with a prefix that is a permutation.
- Assume there are values in the prefix of c larger than the topslot or smaller than the bottom slot. Then let c' be the subconfiguration obtained by deleting these values.
- Assume that there is value n in the prefix strictly between a repeating slot and another slot, as shown in Figure 29a. Let c' be the subconfiguration obtained by deleting all occurrences of the value n in the prefix.
- Assume there are three consecutive repeating slots with values n , $n + 1$, and $n + 2$ respectively, as shown in Figure 29b. Let c' be the configuration obtained by deleting all occurrences of $n + 1$. Note, the repeating slot at this height will become a new slot. Clearly, c' is a smaller subconfiguration of c .

- Assume there are two consecutive repeating slots at the top of the configuration with values n and $n+1$, as shown in Figure 29c. Then let c' be the subconfiguration obtained by deleting all occurrence of $n+1$, and replacing the repeating slot with a new slot. Similarly, if there are two consecutive repeating slots at the bottom with values 1 and 2, then we take c' to be the subconfiguration obtained by deleting all occurrences of 1 and replacing this repeating slot with a new slot.



(a) Values between a repeating slot and another slot.



(b) Three consecutive repeating slots.

(c) Two consecutive repeating slots at the top of a configuration.

Figure 29: Configurations which do not satisfy the conditions of Proposition 9 have smaller subconfigurations.

In each case, we reduce the size of the configuration, so the process of repeatedly replacing c with c' will terminate with a configuration in C_k . Hence, $\mathcal{HSB}(k)$ is defined by avoiding all the derivations of the configurations in C_k . $\square$

To compute the basis of $\mathcal{HSB}(k)$ we consider all configurations with $k+1$ slots in which the prefix forms a permutation which we call π_L . Any derivation of such a configuration will contain a derivation σ of size $|\pi_L| + k + 1$ from the same configuration. The values to the right of the prefix π_L will form a permutation of size $k + 1$ which we denote π_R . As there can not be three consecutive repeating slots, there are no three consecutive values in π_L or π_R . This gives a bound on the size of the Cayley permutations in the basis of $\mathcal{HSB}(k)$, stated in Remark 10.

Remark 10. A basis element $\pi_L\pi_R$ in $\mathcal{HSB}(k)$ will have $|\pi_R| = k + 1$, $\lfloor \frac{k+1}{2} \rfloor \leq |\pi_L|$ due to the restriction on repeating slots and $|\pi_L| \leq k$ which is achieved if all slots are new slots.

A slot-bounded Cayley permutation class is a subclass of $\mathcal{HSB}(k)$ for some k . In order to characterise the slot-bounded Cayley permutation classes we need some definitions.

A size $n + k$ permutation π is a *horizontal juxtaposition* of the size n permutation σ and the size k permutation τ if the size n prefix of π standardises to σ and the size k suffix of π standardises to τ . For example, a horizontal juxtaposition of 1423 and 312 is 1624735.

There are four classes of horizontal juxtapositions which we will need for our theory. They come from horizontally juxtaposing strictly increasing and strictly decreasing sequences. We denote these classes $\mathcal{H}_{I,I}$, $\mathcal{H}_{I,D}$, $\mathcal{H}_{D,I}$, and $\mathcal{H}_{D,D}$. For example, 13245 is a juxtaposition of the strictly increasing sequence 12 and the strictly increasing sequence 123, so is in $\mathcal{H}_{I,I}$, and 521346 is a juxtaposition of the strictly decreasing sequence 21 and the strictly increasing sequence 1234, so is in $\mathcal{H}_{D,I}$, as shown in Figure 30.

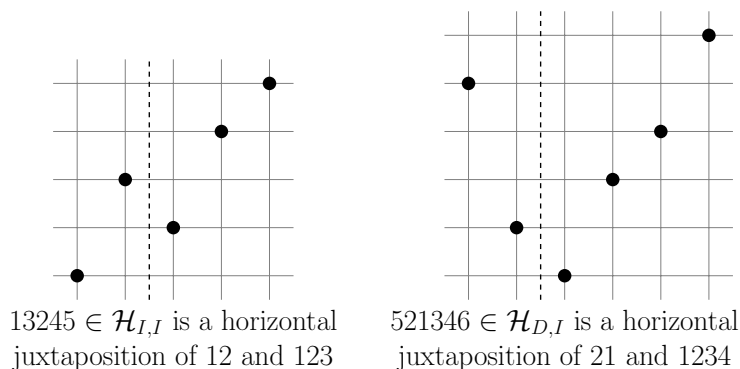


Figure 30: The horizontal juxtapositions $13124 \in \mathcal{H}_{I,I}$ and $421234 \in \mathcal{H}_{D,I}$.

A *horizontal alternation* is a permutation of size $2n$ for which the prefix of size n is a permutation of the odd values and the suffix of size n is a permutation of the even values. For example, 135264 is a horizontal alternation. By symmetry Lemma 3 can be immediately adapted to Lemma 11 for horizontal alternations.

Lemma 11. *For each $\mathcal{H} \in \{\mathcal{H}_{I,I}, \mathcal{H}_{I,D}, \mathcal{H}_{D,I}, \mathcal{H}_{D,D}\}$, the size $2n$ horizontal alternation in $\mathcal{H}$ contains every horizontal juxtaposition in $\mathcal{H}$ of size up to n .*

We are ready to prove the main theorem of this section, Theorem 12.

Theorem 12. *A Cayley permutation class $\text{Av}(B)$ is a subclass of $\mathcal{HSB}(k)$ for some k if and only if B contains a permutation from each of the classes $\mathcal{H}_{I,I}$, $\mathcal{H}_{I,D}$, $\mathcal{H}_{D,I}$, and $\mathcal{H}_{D,D}$.*

Proof. Suppose that $\text{Av}(B) \subseteq \mathcal{HSB}(k)$ for some k . Then $\text{Av}(B)$ avoids derivations from the configurations in C_k , as stated in Proposition 9. In particular, consider a configuration c in C_k with no repeating slots. There is a value between every pair of slots, so the $|c| = 2k + 1$. One of these configurations will have the values in strictly increasing order and another will have the values in strictly decreasing order. A derivation of size $2k + 1$ of these configurations will be formed by inserting values in strictly increasing or strictly decreasing order, creating a permutation from each of the classes $\mathcal{H}_{I,I}$, $\mathcal{H}_{I,D}$, $\mathcal{H}_{D,I}$, and $\mathcal{H}_{D,D}$. As they are not in $\text{Av}(B)$, B must contain either the permutation or a subpermutation from each of these four classes. But, as each of these classes are downwards closed, any subpermutation of a permutation in the class is also in the class. Therefore, B must contain a permutation from each of $\mathcal{H}_{I,I}$, $\mathcal{H}_{I,D}$, $\mathcal{H}_{D,I}$, and $\mathcal{H}_{D,D}$.

Suppose $\text{Av}(B)$ is a Cayley permutation class with B containing permutations from each of the four classes of horizontal juxtapositions. Suppose the length of the largest horizontal juxtaposition in B is n . We will show that there exists a nonnegative integer k such that every basis element of $\mathcal{HSB}(k)$ contains a horizontal juxtaposition that is in B , so $\text{Av}(B) \subseteq \mathcal{HSB}(k)$ for this k .

Let $k = 3n^4$ and σ be any basis element in $\mathcal{HSB}(k)$. Although a smaller k may work, we will show that $k = 3n^4$ is sufficient. We can write $\sigma = \pi_L \pi_R$ with $|\pi_R| = k + 1$ and $\pi_L = |\sigma| - k - 1$. From our discussion about the basis of $\mathcal{HSB}(k)$ it follows that π_L and π_R are permutations, and that σ is a derivation of some configuration $c \in C_k$. As c contains no three consecutive repeating slots, there must be at least $\lfloor \frac{k+1}{3} \rfloor$ new slots in the configuration, which correspond to values in π_R that are not in π_L . As $k = 3n^4$, this is at least n^4 values. By the Erdős-Szekeres theorem [12], there is a strictly increasing or strictly decreasing subsequence of size n^2 in π_R , which we call τ .

By Remark 10, there is at least $\lfloor \frac{k+1}{2} \rfloor = \lfloor \frac{3n^4+1}{2} \rfloor$ values in π_L . All the values in π_L are not in τ and since $n^2 < \lfloor \frac{3n^4+1}{2} \rfloor$, we can take n^2 values from π_L which perfectly interleave with τ . Of these n^2 values, by the Erdős-Szekeres theorem [12], there is a strictly increasing or strictly decreasing subsequence of size n , say α . We can take a subset of the values in τ that perfectly interleaves this subsequence, say β . The concatenation of α and β form a horizontal alternation of size $2n$ which is contained in σ in one of $\mathcal{H}_{I,I}$, $\mathcal{H}_{I,D}$, $\mathcal{H}_{D,I}$, or $\mathcal{H}_{D,D}$. By Lemma 11, this horizontal alternation contains every horizontal juxtaposition from the same class of size up to n , specifically a horizontal juxtaposition in B . Therefore, the basis element σ of $\mathcal{HSB}(k)$ contains a basis element in B . As σ was arbitrary, every basis element of $\mathcal{HSB}(k)$ contains a basis element in B , so none of them are in $\text{Av}(B)$, hence $\text{Av}(B) \subseteq \mathcal{HSB}(k)$. $\square$

An algorithm similar to the one given in Section 4 can be used for computing a DFA that accepts the horizontal insertion encodings of any finitely-based slot-bounded Cayley permutation class. Hence, if a finitely-based Cayley permutation class avoids arbitrarily long horizontal juxtapositions then its horizontal insertion encodings form a regular language and a DFA can be computed for it.

There are 13 Cayley permutation classes which avoid two size 3 patterns and have a set of horizontal insertion encodings which form a regular language, as shown in Table 1. For bases B and B' , obtained by taking the complement or reverse of all Cayley permutations in B , the size n Cayley permutations in $\text{Av}(B)$ are in bijection with all size n Cayley permutations in $\text{Av}(B')$, so $\text{Av}(B)$ and $\text{Av}(B')$ have the same generating function. These 13 Cayley permutation classes correspond to 5 distinct symmetry classes. Their generating functions and first 8 terms have been computed by an implementation of our algorithm [5] and are given in Table 5 with the corresponding references to the OEIS [25].

6 Concluding remarks

We have presented two generalisations of the insertion encoding of permutations introduced by Albert, Linton and Ruškuc [2] to Cayley permutations, namely the vertical and

Bases	Generating function	First 8 terms	OEIS
Av(123, 321)	$\frac{4x^4-2x^3+7x^2-4x+1}{(x-1)^3(2x-1)}$	1, 1, 3, 11, 37, 105, 263, 607	A393341
Av(123, 132) Av(132, 312)	$\frac{2x^2-4x+1}{(x-1)(4x-1)}$	1, 1, 3, 11, 43, 171, 683, 2731	A007583
Av(132, 213)	$\frac{x^4-6x^3+8x^2-5x+1}{(x^2-4x+1)(2x^2-2x+1)}$	1, 1, 3, 11, 42, 159, 596, 2225	A393343
Av(123, 231)	$\frac{12x^6-56x^5+96x^4-86x^3+41x^2-10x+1}{(x-1)^2(2x-1)^3(3x-1)}$	1, 1, 3, 11, 41, 145, 483, 1531	A393344

Table 5: Generating functions, first 8 terms and OEIS entry for all symmetry classes of Cayley permutations avoiding two size 3 patterns which have a set of horizontal insertion encodings that form a regular language.

horizontal insertion encodings. In both cases, we classified which Cayley permutation classes have a set of insertion encodings which form a regular language and provided an algorithm to find their rational generating functions, an implementation for which can be found on GitHub [5].

Albert, Linton and Ruškuc [2] also gave a classification for permutation classes with context free grammars, but we have not given such a condition for Cayley permutations. Our motivation is to enumerate Cayley permutations classes and their classification did not provide a method for automatically computing the algebraic generating functions, but this could be a direction for further research.

Our algorithm for enumerating the slot-bounded Cayley permutation classes extended the definition of tilings for permutations as introduced in Albert *et al.* [2] to Cayley permutations. By following similar methods to those outlined by Albert *et al.* [3] we can extend our implementation of tilings to create strategies that can automatically enumerate many more classes of Cayley permutations, including those which do not have insertion encodings which form a regular language.

Cayley permutations are in bijection with ordered set partitions. The unordered set partitions are in bijection with a subset of Cayley permutation, referred in the literature as *restricted growth functions* [16]. Restricted growth functions are the Cayley permutations with the additional condition that for all values k and ℓ in the Cayley permutation, if $k < \ell$ then the first occurrence of k appears before the first occurrence of ℓ . Our algorithm can be adapted to compute the rational generating functions for pattern-avoiding restricted growth functions with either a horizontal or vertical set of insertion encodings which are regular by restricting the letters that are used. It should be possible to classify these regular cases in a similar way as we have done for Cayley permutations, but this is left for future work.

We thank the anonymous referee for their thoughtful comments which have improved the exposition of our results.

References

- [1] Ahlback, C., Usatine, J., and Pippenger, N. (2012). Barred preferential arrangements. *Electronic Journal of Combinatorics*, (20):#P55.
- [2] Albert, M., Linton, S., and Ruškuc, N. (2005). The insertion encoding of permutations. *Electronic Journal of Combinatorics*, 12(1):#R47.
- [3] Albert, M. H., Bean, C., Claesson, A., Nadeau, E., Pantone, J., and Ulfarsson, H. (2025). Combinatorial exploration: An algorithmic framework for enumeration. *Memoirs of the American Mathematical Society*, to appear.
- [4] Atkinson, M. (1999). Restricted permutations. *Discrete Mathematics*, 195:27–38.
- [5] Bean, C. and Ollson, A. (2025). cperms_ins_enc. https://github.com/Ollson2921/cperms_ins_enc.
- [6] Biers-Ariel, Y., Zhang, Y., and Godbole, A. (2016). Some results on superpatterns for preferential arrangements. *Advances in Applied Mathematics*, 81.
- [7] Cayley, A. (1857). On the theory of analytic forms called trees. *Philos. Mag.* 13, 19-30, 1857. Reprinted in *Mathematical Papers*, 3:242–246.
- [8] Cerbai, G. (2021). Sorting Cayley permutations with pattern-avoiding machines. *Australasian Journal of Combinatorics*, 80(3):322–341.
- [9] Chung, F., Graham, R., Hoggatt, V., and Kleiman, M. (1978). The number of baxter permutations. *Journal of Combinatorial Theory, Series A*, 24(3):382–394.
- [10] Claesson, A., Cerbai, G., Ernst, D., and Golab, H. (2024). Pattern-avoiding Cayley permutations via combinatorial species. [arXiv:2407.19583](https://arxiv.org/abs/2407.19583).
- [11] Defant, C. and Kravitz, N. (2020). Stack-sorting for words. *Australasian Journal of Combinatorics*, 77(1):51–68.
- [12] Erdős, P. and Szekeres, G. (1935). A combinatorial problem in geometry. *Compositio Mathematica*, 2:463–470.
- [13] Foissy, L., Novelli, J., and Thibon, J.-Y. (2010). Polynomial realizations of some combinatorial Hopf algebras. *Journal of Noncommutative Geometry*, 8.
- [14] Golab, H. (2024). Pattern avoidance in Cayley permutations. Master’s thesis, Northern Arizona University.
- [15] Hazewinkel, M. (2004). Hopf algebras of endomorphisms of Hopf algebras. *Arabian Journal for Science and Engineering*, 33.
- [16] Jelínek, V. and Mansour, T. (2008). On pattern-avoiding partitions. *Electronic Journal of Combinatorics*, 15:#R39.

- [17] Knuth, D. E. (1968). *The Art of Computer Programming*, volume 1. Addison-Wesley.
- [18] Kroes, D. and Rhoades, B. (2022). Packed words and quotient rings. *Discrete Mathematics*, 345:112945.
- [19] MacMahon, P. A. (1915). *Combinatory analysis*, volume 1. Cambridge University Press.
- [20] Mor, M. and Fraenkel, A. (1984). Cayley permutations. *Discrete Mathematics*, 48:101–112.
- [21] Nkonkobe, S., Bényi, B., Corcino, R., and Corcino, C. (2020). A combinatorial analysis of higher order generalised geometric polynomials: A generalisation of barred preferential arrangements. *Discrete Mathematics*, 343:111729.
- [22] Patrias, R. and Pylyavskyy, P. (2016). Combinatorics of K -theory via a K -theoretic Poirier–Reutenauer bialgebra. *Discrete Mathematics*, 339:1095–1115.
- [23] Pippenger, N. (2010). The hypercube of resistors, asymptotic expansions, and preferential arrangements. *Mathematics Magazine*, 83.
- [24] Simion, R. and Schmidt, F. (1985). Restricted permutations. *European Journal of Combinatorics*, 6.
- [25] Sloane, N. J. A. (2025). The Online Encyclopedia of Integer Sequences. <https://oeis.org>.
- [26] Vatter, V. (2009). Finding regular insertion encodings for permutation classes. *Journal of Symbolic Computation*, 47.
- [27] Vatter, V. (2015). Permutation classes. In *Handbook of Enumerative Combinatorics*, pages 777–858. Chapman and Hall/CRC.