Rationality For Subclasses of 321-Avoiding Permutations

Michael H. Albert*
Department of Computer Science
University of Otago
Dunedin, New Zealand

Robert Brignall
Department of Mathematics and Statistics
The Open University
Milton Keynes, England UK

Nik Ruškuc*
School of Mathematics and Statistics
University of St Andrews
St Andrews, Scotland UK

Vincent Vatter*†
Department of Mathematics
University of Florida
Gainesville, Florida USA

We prove that every proper subclass of the 321-avoiding permutations that is defined either by only finitely many additional restrictions or is well-quasi-ordered has a rational generating function. To do so we show that any such class is in bijective correspondence with a regular language. The proof makes significant use of formal languages and of a host of encodings, including a new mapping called the panel encoding that maps languages over the infinite alphabet of positive integers avoiding certain subwords to languages over finite alphabets.

1. Introduction

It has been known since 1968, when the first volume of Knuth's *The Art of Computer Programming* [20] was published, that the 312-avoiding permutations and the 321-avoiding permutations are both enumerated by the Catalan numbers, and thus have algebraic generating functions. At least nine essentially different bijections between these two permutation classes have been devised in the intervening years, as surveyed by Claesson and Kitaev [15]. In one such bijection (shown in Figure 1 and first given in this non-recursive form by Krattenthaler [21]) we obtain Dyck paths from permutations of both types by drawing a path above their left-to-right maxima (an entry is a *left-to-right maximum* if it is greater than every entry to its left).

*Albert, Ruškuc, and Vatter were partially supported by EPSRC via the grant EP/J006440/1.
†Vatter's research was partially supported by the National Science Foundation under Grant Number DMS-1301692.

Date: December 4, 2017
AMS 2000 Subject Classification. 05A05, 05A15

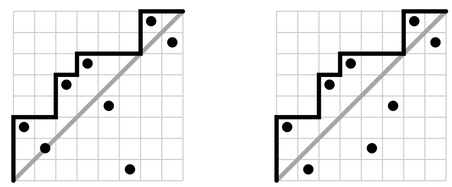


Figure 1: The bijections to Dyck paths from 312-avoiding permutations (left) and 321-avoiding permutations (right). Knowing the positions and values of the left to right maxima, the remaining elements can be added in a unique fashion to avoid 312, respectively 321.

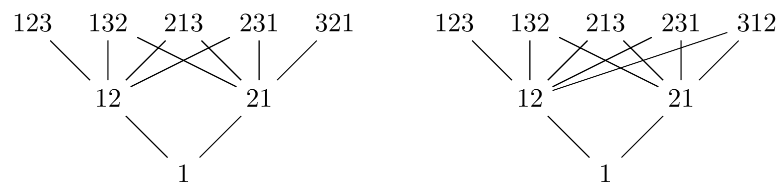


Figure 2: The Hasse diagrams of 312-avoiding (left) and 321-avoiding (right) permutations.

Despite their equinumerosity, there are fundamental differences between these two classes. Indeed, Miner and Pak [27] make a compelling argument that there are so many different bijections between these two classes precisely *because* they are so different, and thus there can be no “ultimate” bijection. In particular, both sets carry a natural ordering with respect to the containment of permutations (defined below) but they *are not* isomorphic as partially ordered sets. Indeed, this can be seen by examining the first three levels of their Hasse diagrams, drawn in Figure 2.

A more striking difference between the two classes is that the 321-avoiding permutations contain infinite antichains (see Section 9), while the 312-avoiding permutations do not. Following the standard terminology, we say that a permutation class without infinite antichains is *well-quasi-ordered*.

From a structural perspective, the avoidance of 312 imposes severe restrictions on permutations: the entries to the left of the minimum must lie below the entries to the right of this minimum. This restricted structure is known to imply that proper subclasses of the 312-avoiding permutations are very well-behaved: there are only countably many such subclasses, and as Albert and Atkinson [1] proved in their work on the substitution decomposition, each has a rational generating function. (Mansour and Vainshtein [25] had proved this rationality result for proper subclasses classes defined by a single additional restriction earlier.)

The 321-avoiding permutations also have a good deal of structure: their entries can be partitioned into two increasing subsequences. However, this property has proved much more difficult to work with. In particular, as noted above, there are infinite antichains of 321-avoiding permutations, so there are uncountably many proper subclasses of this class—in fact uncountably many subclasses with pairwise distinct generating functions. By an elementary counting argument, *some* of these proper subclasses must have non-rational (indeed, also non-algebraic and non-D-finite) generating functions.

Because Av(321) is not well-quasi-ordered, any result analogous to the one mentioned for 312-avoiding permutations (which are, to repeat, well-quasi-ordered) must be more discerning as to the subclasses considered. We develop a methodology for working with arbitrary subclasses of Av(321) and show how to apply it to two natural general families: subclasses defined by imposing finitely many additional forbidden patterns and subclasses that are well-quasi-ordered. Our main result shows that either of these conditions is sufficient to guarantee the rationality of generating functions.

For the rest of the introduction, we review the formal definitions of permutation containment and permutation classes. We generally represent permutations in one line notation as sequences of positive integers. We define the *length* of the permutation π, denoted $|\pi|$, to be the length of the corresponding sequence, i.e., the cardinality of the domain of π. Given permutations π and σ, we say that π *contains* σ, and write $\sigma \leq \pi$, if π has a subsequence $\pi(i_1)\cdots\pi(i_{|\sigma|})$ of the same length as σ that is *order isomorphic* to σ (i.e., $\pi(i_s) < \pi(i_t)$ if and only if $\sigma(s) < \sigma(t)$ for all $1 \leq s, t \leq |\sigma|$); otherwise, we say that π *avoids* σ. If π contains σ we also say that σ is a *subpermutation* of π particularly in contexts where we have a specific embedding (i.e., set of indices) in mind. Containment is a partial order on permutations. For example, $\pi = 251634$ contains $\sigma = 4123$, as can be seen by considering the subsequence $\pi(2)\pi(3)\pi(5)\pi(6) = 5134$. A collection of permutations $\mathcal{C}$ is a *permutation class* if it is closed downwards in this order; i.e., if $\pi \in \mathcal{C}$ and $\sigma \leq \pi$, then $\sigma \in \mathcal{C}$.

For any permutation class $\mathcal{C}$ there is a unique antichain B such that

$$\mathcal{C} = \mathrm{Av}(B) = \{\pi \,:\, \pi \text{ avoids all } \beta \in B\}.$$

This antichain, consisting of the minimal permutations *not* in $\mathcal{C}$, is called the *basis* of $\mathcal{C}$. If B happens to be finite, we say that $\mathcal{C}$ is *finitely based*. For non-negative integers n, we denote by $\mathcal{C}_n$ the set of permutations in $\mathcal{C}$ of length n, and refer to

$$\sum_n |\mathcal{C}_n| x^n = \sum_{\pi\in\mathcal{C}} x^{|\pi|}$$

as the *generating function* of $\mathcal{C}$. The goal of this paper is to establish the following.

Theorem 1.1. *If a proper subclass of the* 321*-avoiding permutations is finitely based or well-quasi-ordered then it has a rational generating function.*

In [14] Bousquet-Mélou writes

> "for almost all families of combinatorial objects with a rational [generating function], it is easy to foresee that there will be a bijection between these objects and words of a regular language".

In proving Theorem 1.1 we indeed adopt an approach via regular languages. We in fact encode permutations as words using several different encodings. We begin by introducing the *domino encoding* that records the relative positions of entries in pairs of adjacent cells in a staircase gridding. After that we combine this information and encode each 321-avoiding permutation as a word, say w, over the positive integers $\mathbb{P}$ satisfying the additional condition $w(i+1) \leq w(i)+1$ for all relevant indices i (throughout we denote by $w(i)$ the i^{th} letter of the word w). We then show that for any proper subclass, $\mathcal{C}$, of 321-avoiding permutations there is some positive integer c such that the encoding of every permutation in $\mathcal{C}$ avoids (as a subword) every shift of the word $(12\cdots c)^c$, i.e. all words $(i(i+1)\cdots(i+c-1))^c$ for $i \in \mathbb{P}$. The true key to our method is the *panel encoding* η_c, which translates languages not containing shifts of $(12\cdots c)^c$ to languages over *finite* alphabets. A careful analysis of the interplay between panel encodings, domino encodings, and the classical encodings by Dyck paths (from Figure 1) along with a technique called marking establishes the regularity of various images under η_c, and this completes the proof of Theorem 1.1.

We assume throughout that the reader has some familiarity with the basics of regular languages, as provided by Sakarovitch [28]; for a more combinatorial approach we refer the reader to Bousquet-Mélou [14] or Flajolet and Sedgewick [16, Section I.4 and Appendix A.7]. The notation used is mostly standard. Herein a *subword* of the word w is any subsequence of its entries while a *factor* is a contiguous subsequence. Given a set of letters X and a word w we denote by $w|^X$ the *projection* of w onto X, i.e., the subword of w formed by its letters in X. Finally, we denote the empty word by ϵ.

2. Staircase Griddings

A *staircase gridding* of a 321-avoiding permutation π is a partition of its entries into *cells* labelled by the positive integers satisfying four properties:

- the entries in each cell are increasing,
- for $i \geq 1$, all entries in the $(2i)^{\text{th}}$ cell lie to the right of those in the $(2i-1)^{\text{st}}$ cell,
- for $i \geq 1$, all entries in the $(2i+1)^{\text{st}}$ cell lie above those in the $(2i)^{\text{th}}$ cell, and
- if $j \geq i+2$ then all entries in the j^{th} cell lie above and to the right of those in the i^{th} cell.

Staircase griddings have been used extensively in the study of 321-avoiding permutations, for instance in [3, 7, 9, 17] and represent the fundamental objects of consideration here. We denote by $\pi^\sharp$ a particular staircase gridding of the 321-avoiding permutation π.

Every 321-avoiding permutation has at least one staircase gridding and indeed, we can identify a preferred staircase gridding of every such permutation: a staircase gridding of the 321-avoiding permutation π is *greedy* if the first cell contains as many entries as possible, and subject to this, the second cell contains as many entries as possible, and so on. Figure 3 provides an example of a greedy staircase gridding.

It is easy to construct greedy staircase griddings in the following iterative manner. The entries of the first cell are the maximum increasing prefix τ of π. Those of the second cell are then the maximum increasing sequence in $\pi \setminus \tau$ whose values form an initial segment of the values occurring in $\pi \setminus \tau$. Thereafter we continue alternately taking a maximum increasing prefix and then a maximum increasing sequence of values forming an initial segment of the values remaining.

The relative position of two entries in a 321-avoiding permutation π is completely determined by the numbers given to their cells in any staircase gridding, unless these numbers are consecutive. In the case of cells which lie next to each other horizontally we consider their entries as being *ordered from bottom to top*, and in the case of cells which lie next to each other vertically, from *left to right*. Observe that this gives us two orders on the entries of a given cell (except the first), but that the two orders in fact coincide. With this ordering in mind, we formulate two conditions that characterise greedy staircase griddings:

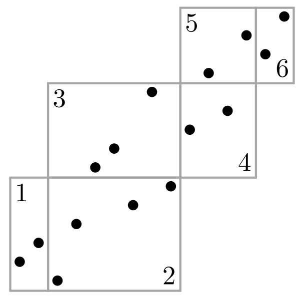


Figure 3: The greedy staircase gridding of the 321-avoiding permutation 2 3 1 4 7 8 5 11 6 9 12 10 14 13 15.

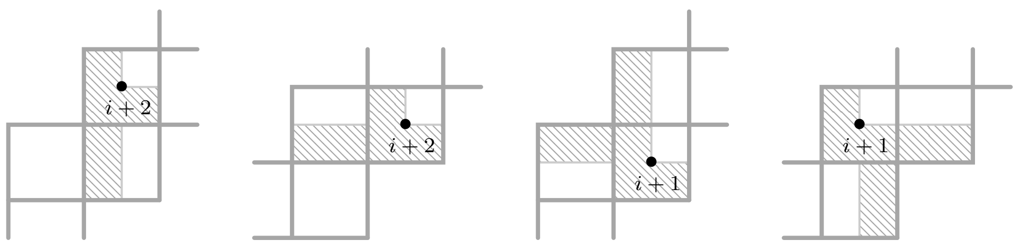


Figure 4: The four types (due to parity) of failures of (G1) and (G2). Here the hatched regions indicate positions where entries do not lie. Within the cell of the indicated entry these serve to identify it as the first entry of its cell. In the two rightmost pictures the hatched region in cell $i+2$ is empty because the gridding is assumed to satisfy (G1).

(G1) For all $i \geq 1$ the first entry in the $(i+1)^{\text{st}}$ cell occurs before all entries of the $(i+2)^{\text{nd}}$ cell.

(G2) For all $i \geq 1$ the first entry in the $(i+1)^{\text{st}}$ cell is followed (not necessarily immediately) by an entry of the i^{th} cell.

These restrictions, or rather how they can fail, are depicted in Figure 4. It is important for later to note that these conditions can be tested by inspecting only the first and last entries of each cell.

Proposition 2.1. *A staircase gridding is greedy if and only if it satisfies* (G1) *and* (G2)*.*

Proof. Let π be a 321-avoiding permutation, and consider first its greedy staircase gridding. If this gridding were to fail (G1) for some $i \geq 1$, then we see from the two leftmost pictures in Figure 4 that the first entry of the $(i+2)^{\text{nd}}$ cell could (and therefore, in a greedy gridding, would) have been placed instead in the i^{th} cell, a contradiction. On the other hand, if the gridding were to satisfy (G1) but fail (G2) for some $i \geq 1$ then we see from the two rightmost pictures in Figure 4 that the first entry of the $(i+1)^{\text{st}}$ cell would have been placed in the i^{th} cell, another contradiction.

Next consider a staircase gridding $\pi^\sharp$ of π that satisfies (G1) and (G2). The condition (G2) implies that the labels of the non-empty cells form an initial segment of $\mathbb{P}$ so we proceed inductively. By definition, the entries of the 1^{st} cell form an initial increasing segment of π so we need to show that it is the longest such segment. The next entry of π (reading left to right) must lie in the 2^{nd} cell because (G1) shows that the leftmost entry of the 2^{nd} cell lies to the left of all entries of the 3^{rd} cell.

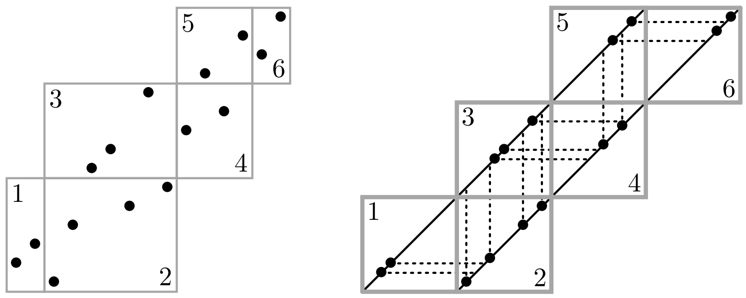


Figure 5: The greedy staircase gridding of the permutation 2 3 1 4 7 8 5 11 6 9 12 10 14 13 15 from Figure 3 and a drawing of this gridding on two parallel lines. The dotted lines in the picture on the right are included only to indicate relative positions.

Thus this entry is the first entry of the 2nd cell. By (G2) it lies below an entry of the 1st cell, and this implies that the entries of the 1st cell are a maximum initial increasing segment.

Let τ denote the contents of the 1st cell and consider the entries of the 2nd cell of π. By the third and fourth requirements for a staircase gridding, all entries of π not belonging to the first or second cells lie above those in the second cell. Thus the entries of the second cell form an increasing contiguous sequence by value in $\pi \setminus \tau$ and we must show that it is maximum. Consider the next smallest entry of $\pi \setminus \tau$ by value (if there is no such entry then we are done). As before, (G1) shows that this entry must lie in the 3rd cell, and thus must be the least entry of the 3rd cell. Again, (G2) implies that this entry lies to the left of an entry of the 2nd cell, and thus the contents of the 2nd cell are maximum.

To complete the argument we repeat the reasoning for the 1st and 2nd cells for odd cells and even cells respectively, with suitable modifications, basically referring throughout to the set of entries of π that belong to the remaining cells of $\pi^\sharp$. □

Staircase griddings have a pleasing geometric interpretation, as first observed by Waton in his thesis [32]. First we describe a general construction: given any figure in the plane and permutation π we say that π can be *drawn* on the figure if we can choose a set P consisting of n points in the figure, no two on a common horizontal or vertical line, label them 1 to n from bottom to top and then read them from left to right to obtain π. If this relationship holds between P and π we say that P and π are *order isomorphic*.

Suppose that we take our figure to consist of the two parallel rays $y = x$ and $y = x - 1$ for $y \geq 0$. From any staircase gridding of a 321-avoiding permutation π we can construct a drawing of π on these two parallel rays. First we add vertical and horizontal lines $x = i$ and $y = i$ for all natural numbers i, splitting the figure into cells. To draw π on this figure, take any staircase gridding of π and embed it cell by cell into the corresponding cells of the figure, making sure that the relative order between entries in adjacent cells is preserved. An example is shown in Figure 5.

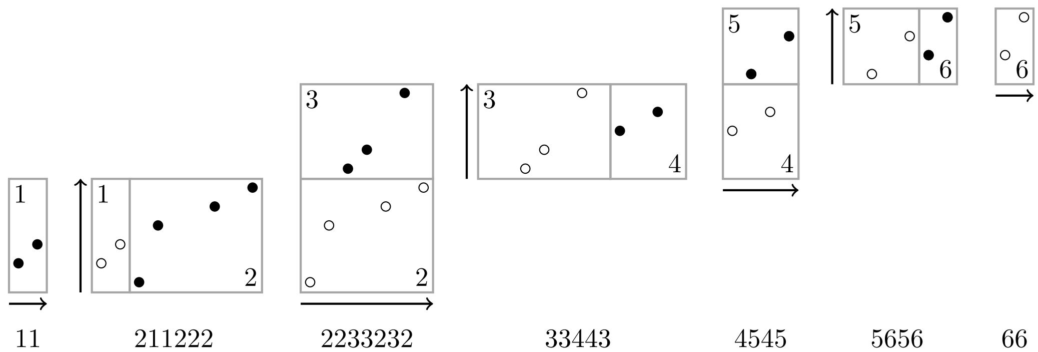


Figure 6: The domino factors (bottom row) corresponding to dominoes (top row) of the gridded permutation from Figure 3. The domino encoding of this permutation is therefore

$$\bullet\bullet\#\bullet\circ\circ\bullet\bullet\bullet\#\circ\circ\bullet\bullet\circ\bullet\circ\#\circ\circ\bullet\bullet\circ\#\circ\bullet\circ\bullet\#\circ\bullet\circ\bullet\#\circ\circ\#.$$

3. DOMINO AND OMNIBUS ENCODINGS

From any (not necessarily greedy) staircase gridding we construct dominoes. For each $i \geq 0$, the i^{th} *domino* consists of the entries of the staircase gridding in the i^{th} and $(i+1)^{\text{st}}$ cells. We then read the entries of this domino in the order specified before (left-to-right for vertically adjacent cells, and bottom-to-top for horizontally adjacent cells), recording the labels of their cells as the i^{th} *domino factor* d_i. Note that both the 0^{th} and final domino factors encode single cells. An example of dominoes and domino factors is shown in Figure 6.

We now translate the i^{th} domino factor d_i of the staircase gridded permutation $\pi^\sharp$ to the alphabet $\{\circ, \bullet\}$ by replacing occurrences of i by $\circ$ and occurrences of $i+1$ by $\bullet$, labeling the resulting word $d_i^\bullet$. The *domino encoding*, δ, of the gridded permutation $\pi^\sharp$ is then

$$\delta(\pi^\sharp) = d_0^\bullet \# d_1^\bullet \# \cdots \# d_m^\bullet \#,$$

where m is the last nonempty cell. Recall that the relative position of entries in cells j and k is determined by the cells themselves if $|k-j| \geq 2$. Therefore, as the domino factors completely determine the relative positions between entries of adjacent cells, the domino encoding is an injection (as a mapping from staircase gridded permutations to valid domino encodings). Note that the same definition of domino encodings can be applied to arbitrary words in $\mathbb{P}^*$.

We also derive a second encoding, the *omnibus encoding*, which again collects the domino factors of $\pi^\sharp$ into a single word but this time by interleaving them with each other. In this encoding, for which the alphabet is the positive integers, each entry corresponds to a single letter and the encoding contains every domino factor as a subword. Formally, this means that we insist that the omnibus encoding, w, of $\pi^\sharp$ satisfy the *projection condition*:

(PC) $w|^{\{i,i+1\}}$ is equal to the i^{th} domino factor of $\pi^\sharp$ for all i.

That is, when we look only at the letters i and $i+1$ of an omnibus encoding we recover the i^{th} domino factor d_i. This rule alone does not determine the encoding uniquely because it does not specify the order in which letters belonging to different domino factors should occur. In particular, if $|j-i| \geq 2$ then the letters i and j "commute" in the sense that replacing an ij factor by ji does not change the projections to domino factors. We choose to "prefer" letters of larger value moving to the left. It is easy to see that this is equivalent to stipulating that our encoding w satisfy the *small ascent condition*:

(SAC) $w(i+1) \leq w(i)+1$ for all relevant indices i.

The conditions (PC) and (SAC) together guarantee the uniqueness of the omnibus encoding (of gridded permutations). We prove this momentarily, after demonstrating how to compute the omnibus encoding from the domino factors for the example shown in Figure 6. Having written one domino factor d_i, in the next row we copy the occurrences of $i+1$, and then insert the occurrences of $i+2$ as far to the left as possible, subject to the requirement that the word in that row is d_{i+1}.

$$\begin{array}{ccccccccccccccccc}
d_0 & = & & 1 & 1 & & & & & & & & & & & & \\
d_1 & = & 2 & 1 & 1 & 2 & & & & & & & & & 2 & & 2 \\
d_2 & = & 2 & & & 2 & 3 & 3 & & & & & & & 2 & 3 & 2 \\
d_3 & = & & & & & 3 & 3 & 4 & & & 4 & & & & 3 & \\
d_4 & = & & & & & & & 4 & 5 & & 4 & 5 & & & & \\
d_5 & = & & & & & & & & 5 & 6 & & 5 & 6 & & & \\
d_6 & = & & & & & & & & & 6 & & & 6 & & & \\
\hline
 & & 2 & 1 & 1 & 2 & 3 & 3 & 4 & 5 & 6 & 4 & 5 & 6 & 2 & 3 & 2
\end{array}$$

By way of proving the uniqueness of the omnibus encoding, we establish that every word of positive integers satisfying the small ascent condition is uniquely determined by its projections to pairs of consecutive integers.

Proposition 3.1. *If the words $u, w \in \mathbb{P}^*$ both satisfy the small ascent condition, $u|^{\{1\}} = w|^{\{1\}}$, and $u|^{\{i,i+1\}} = w|^{\{i,i+1\}}$ for every positive integer i, then $u = w$.*

Proof. For a positive integer k, let $[k] = \{1, 2, \ldots, k\}$. We prove inductively that under the hypotheses of the proposition, we have $u|^{[i]} = w|^{[i]}$ for all $i \geq 1$. The hypotheses give the base case of $i = 1$. Suppose now that $u|^{[i]} = w|^{[i]}$ for some $i \geq 1$ and consider any occurrence of $i+1$ in $u|^{[i+1]}$. If this $i+1$ has any smaller elements to its left, then the rightmost such must equal i owing to the small ascent condition. Therefore $u|^{[i+1]}$ is formed from $u|^{[i]}$ by inserting all occurrences of $i+1$ correctly according to $u|^{\{i,i+1\}}$ and as far to the left as possible subject to this constraint. Since $w|^{[i+1]}$ is formed from $w|^{[i]}$ by the same rule and since both $u|^{[i]} = w|^{[i]}$ and $u|^{\{i,i+1\}} = w|^{\{i,i+1\}}$ it follows that $u|^{[i+1]} = w|^{[i+1]}$, completing the proof. $\square$

These facts allow us to define the *omnibus encoding*, ω from the set of all staircase gridded 321-avoiding permutation to $\mathbb{P}^*$ as the mapping sending $\pi^\sharp$ to the unique word satisfying both the (PC) and (SAC). We then define the two languages of interest,

$$\begin{aligned}
\mathcal{L}^\infty &= \{\omega(\pi^\sharp) : \pi^\sharp \text{ is a gridded 321-avoiding permutation}\} \text{ and} \\
\mathcal{G}^\infty &= \{\omega(\pi^\sharp) : \pi^\sharp \text{ is a greedily gridded 321-avoiding permutation}\}.
\end{aligned}$$

For most of the argument it is easier to ignore the greediness conditions and focus on $\mathcal{L}^\infty$, which has a simple alternative definition:

$$\mathcal{L}^\infty = \{w \in \mathbb{P}^* \,:\, w \text{ satisfies (SAC)}\}.$$

Translating the gridding conditions (G1) and (G2) to omnibus encodings, we immediately obtain the following characterisation of the language $\mathcal{G}^\infty$.

Observation 3.2. *The word $w \in \mathcal{L}^\infty$ lies in $\mathcal{G}^\infty$ if and only if it also satisfies the following two conditions:*

(ωG1) *For all $i \geq 1$, the first occurrence of $i+1$ occurs before all occurrences of $i+2$.*

(ωG2) *For all $i \geq 1$, the first occurrence of $i+1$ is followed (not necessarily immediately) by an occurrence of i.*

Given any word $w \in \mathcal{L}^\infty$, we define its i^{th} *domino factor* d_i to be $w|^{\{i,i+1\}}$, i.e., the subword of w made up of its letters equal to i and $i+1$. In this way, the domino factors of any gridded 321-avoiding permutation $\pi^\sharp$ are equal to the domino factors of its omnibus encoding $\omega(\pi^\sharp)$. In the same manner, we can define the *domino encoding* of any word $w \in \mathcal{L}^\infty$ as

$$\delta(w) = d_0^\bullet \# d_1^\bullet \# \cdots \# d_m^\bullet \#,$$

where m is the value of the largest letter in w.

Therefore given any omnibus encoding $w \in \mathcal{L}^\infty$, we can recover the domino factors (or, equivalently, the domino encoding) of the underlying gridded permutation and then, by our previous remarks, reconstruct this gridded permutation. In other words, ω is a bijection between the set of gridded 321-avoiding permutations and $\mathcal{L}^\infty$. By the same reasoning, ω is also a bijection between the set of greedily gridded 321-avoiding permutations and $\mathcal{G}^\infty$.

As every 321-avoiding permutation has a unique greedy staircase gridding, this shows that the number of words of length n in $\mathcal{G}^\infty$ is equal to the nth Catalan number. The authors asked on MathOverflow [31] for a simple bijection between (a variant of) this language and another Catalan family (other than staircase griddings). In response, Speyer [29] conjectured a link to the Catalan matroid of Ardila [10] that was subsequently proved by Stump [30] using Haglund's zeta map [18]. Mansour and Shattuck [24] have since provided several refinements of the enumeration, such as the number of words in the language with a specified number of occurrences of 1 and 2.

The domino encoding may appear at first to be superior to the omnibus encoding because the former is defined on the finite alphabet $\{\circ, \bullet, \#\}$ whereas the latter is defined on the infinite alphabet of positive integers. However, in the context of establishing a regularity result for subclasses, $\mathcal{C}$, of 321-avoiding permutations the domino encoding is of no immediate use. If $\mathcal{C}$ is not finite then it must contain arbitrarily long increasing sequences, and this already implies that the domino encodings of the greedy griddings of members of $\mathcal{C}$ do not form regular language, owing to the condition that the number of $\bullet$ symbols in the $\{\bullet, \circ\}$ factor preceding a punctuation mark must equal the number of $\circ$ symbols in the immediately following such factor. Nonetheless, as well as providing a foundation for the omnibus encoding, the domino encoding becomes useful again in the final stages of the proof of Theorem 1.1.

We say that the omnibus encoding is an *entry-to-entry mapping* because every letter of $\omega(\pi)$ corresponds to precisely one entry of π. The domino encoding is nearly an entry-to-entry mapping because each entry of π corresponds to precisely two non-punctuation letters of $\delta(\pi)$. We make frequent, though implicit, use of these correspondences.

The inverse of the omnibus encoding has a natural geometric interpretation, which can be viewed as an infinite version of the encodings defined in [2]. Following the notation there we denote the inverse of ω by $\varphi^\sharp$, which is a surjection from $\mathbb{P}^*$ to gridded 321-avoiding permutations, interpreted as equivalence classes of sets of points on the two parallel rays $y = x$ and $y = x - 1$ for $y \geq 0$ subdivided into cells by the vertical and horizontal lines $x = i$ and $y = i$ for all integers i.

Suppose that the word $w \in \mathbb{P}^*$ has length n and choose arbitrary real numbers $0 < d_1 < \cdots < d_n < 1$. For each $1 \leq i \leq n$, take p_i to be the point on the diagonal line segment in the cell numbered by $w(i)$ that is at infinity-norm distance d_i from the lower left corner of this cell. We define $\varphi^\sharp(w)$ to be the gridded permutation that is order isomorphic to the gridded set $\{p_1, p_2, \ldots, p_n\}$ of points in the plane and we further define $\varphi(w)$ to be the permutation obtained from $\varphi^\sharp(w)$ by "forgetting" the grid lines. It is routine to show that $\varphi^\sharp(w)$ does not depend on the particular choice of $d_1, \ldots, d_n$, and thus is well-defined. Given any two words $u, w \in \mathbb{P}^*$, it is clear from this construction that if u is a subword of w then $\varphi(u) \leq \varphi(w)$. Reframing this observation in terms of the omnibus encoding we obtain the following.

Observation 3.3. *Let $\sigma^\sharp$ and $\pi^\sharp$ be gridded 321-avoiding permutations. If $\omega(\sigma^\sharp)$ is a subword of $\omega(\pi^\sharp)$ then $\sigma \leq \pi$.*

4. Restricting to a Finite Alphabet

In order to appeal to the theory of formal languages we must translate the omnibus encoding to a finite alphabet. This—accomplished via the panel encoding—is the topic of the next section. Aside from restricting to a finite alphabet though, some other restriction is needed because Av(321) does not have a rational generating function. This section introduces a generic family of restrictions on the omnibus encodings in such a way that for any proper subclass of Av(321) one of the restrictions in the family is satisfied. This will subsequently be shown to be sufficient to enable encodings of finitely based and/or well-quasi-ordered subclasses into regular languages over finite alphabets.

Given a word $w \in \mathbb{P}^*$ its *shift by k* is defined by

$$w^{+k}(i) = w(i) + k$$

for all indices i. An *even shift* is a shift by an even integer. By the definition of φ, it follows immediately that $\varphi(w^{+2k}) = \varphi(w)$, so the image of φ is unaffected by even shifts. As a consequence of this fact and Observation 3.3, we obtain the following.

Observation 4.1. *Let π and σ be 321-avoiding permutations with staircase griddings $\pi^\sharp$ and $\sigma^\sharp$ respectively. If $\omega(\pi^\sharp)$ contains an even shift of $\omega(\sigma^\sharp)$ as a subsequence then π contains σ.*

Note that the converse of this observation does not hold—a simple example is given by the pair $\pi = 2314$, $\sigma = 123$. Letting $\pi^\sharp$ and $\sigma^\sharp$ denote the greedy griddings of these permutations we see that $\omega(\pi^\sharp) = 2112$ (see the centre of Figure 7) while $\omega(\sigma^\sharp) = 111$ so although σ is contained in π,

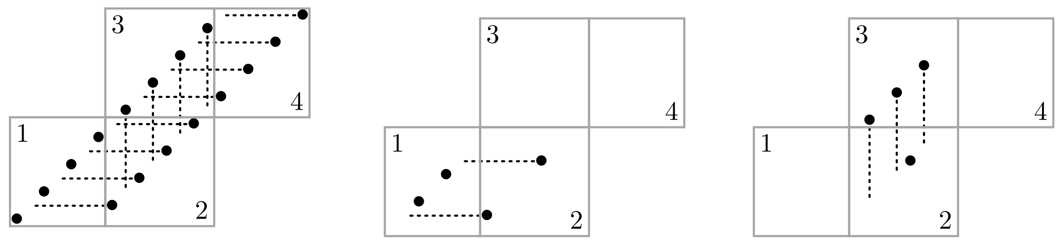


Figure 7: The drawing on the left shows plot of the gridded permutation $\pi^\sharp$ for which $\omega(\pi^\sharp) = (1234)^4$, or from the geometric viewpoint, $\varphi^\sharp((1234)^4)$. The drawings in the centre and right show two griddings of the permutation 2314, which are encoded, respectively, by 2112 and 3323.

$\omega(\pi^\sharp)$ contains no shift, let alone an even one, of $\omega(\sigma^\sharp)$. Note, however, that another gridding of 2314, shown on the right of Figure 7, leads to an encoding which does contain a shift of 111.

In the next two sections we focus on the languages

$$\mathcal{L}_c^\infty = \{w \in \mathbb{P}^* \,:\, w \text{ satisfies the small ascent condition and avoids all shifts of } (12\cdots c)^c\}.$$

This definition is justified by the following proposition. (Note that, as the proof shows, the condition on avoiding all shifts of $(12\cdots c)^c$ could be weakened, but we have no need to do so.)

Proposition 4.2. *For every proper subclass $\mathcal{C}$ of 321-avoiding permutations there is a positive integer c such that $\omega(\pi^\sharp) \in \mathcal{L}_c^\infty$ for all staircase griddings $\pi^\sharp$ of permutations $\pi \in \mathcal{C}$.*

Proof. Let β be any 321-avoiding permutation not belonging to $\mathcal{C}$ with greedy gridding $\beta^\sharp$ and set $c = |\beta| + 1$. Clearly $\omega(\beta^\sharp)$ is contained in $(12\cdots(c-1))^{c-1}$ and so no word of the form $\omega(\pi^\sharp)$ for $\pi \in \mathcal{C}$ may contain an even shift of $(12\cdots(c-1))^{c-1}$ by Observation 4.1. Moreover, the word $12\cdots c$ contains both $(12\cdots(c-1))^{+0}$ and $(12\cdots(c-1))^{+1}$, so any shift of $(12\cdots c)^{c-1}$ contains an even shift of $(12\cdots(c-1))^{c-1}$. Therefore no word of the form $\omega(\pi^\sharp)$ for $\pi \in \mathcal{C}$ may contain a shift of $(12\cdots c)^{c-1}$, proving the proposition. □

Though we work exclusively on the level of words for the next two sections, it is worth remarking that Proposition 4.2 shows that every proper subclass of Av(321) avoids the permutation encoded by $(12\cdots c)^c$ for some value of c. Stated from the geometric perspective, every 321-avoiding permutation is contained in $\varphi((12\cdots c)^c)$ for some value of c. Thus these permutations are *universal objects*[1] for Av(321). An example of one of these universal permutations is shown on the left of Figure 7.

[1] Universal objects for permutation classes are often called *super-patterns*. The typical problem is, given a class $\mathcal{C}$, determine the length of a shortest permutation containing all permutations in $\mathcal{C}_n$. Our universal object is not the shortest possible, as Miller [26] has found a universal permutation for the class of *all* permutations of length $\binom{n+1}{2}$, i.e., a permutation of this length containing all permutations of length n. No improvements worth mentioning are known for the class Av(321). For the class Av(231), Bannister, Cheng, Devanny, and Eppstein [12] have established an upper bound of $n^2/4 + \Theta(n)$.

5. THE PANEL ENCODING η_c

This section and the next focus solely on the language $\mathcal{L}_c^\infty$ and an encoding, η_c, which maps it to a language $\mathcal{L}_c^\eta$ over a finite alphabet. The encoding η_c is described in this section while the regularity of $\mathcal{L}_c^\eta$ is established in the next. Throughout, consider c to be a fixed positive integer and a word $w \in \mathcal{L}_c^\infty$ to be given. Further suppose that the maximum value of a letter in w is m.

The material in the remainder of this section is rather technical, so we begin with an overview of the general strategy. Consider the maximal factors of w not containing occurrences of symbol 1. With the exception of the factor preceding the first 1, all of them are immediately preceded by a 1. Some of those may contain an occurrence of c, and we designate those as *large*. Note that by the small ascent condition and the fact that these factors are all preceded by a 1, each large factor contains an occurrence of $23\cdots c$ which, together with the adjacent 1, yields an occurrence of $12\cdots c$. Since w avoids $(12\cdots c)^c$ there must be fewer than c large factors. The remaining factors are designated as *small*, except for the factor before the first 1 which is handled separately.

The idea of the encoding η_c is to first separate the small and large factors of w. The small factors form a word over $\{1,\ldots,c-1\}$ and this word is recorded essentially as is; the large factors are then processed recursively. In order to facilitate the reconstruction of w from its encoding, we need to record the places where the separation occurred. We achieve this by decorating 1s that are supposed to be followed by the (now removed) large factors, and the matching 2s at the start of these large factors. Since all letters of the large factors are greater than 1, we can reduce all of them by 1 and repeat the process. At each stage the maximum value remaining decreases by at least 1 (in fact exactly 1 if there is a large factor present), and so we eventually produce a sequence of (decorated) words over the alphabet $\{1,2,\ldots,c-1\}$. The encoding $\eta_c(w)$ is simply the concatenation of these words, separated by punctuation symbols.

Moving to the technical details, we aim to describe an injection $\eta_c : \mathcal{L}_c^\infty \to \Sigma^*$ where

$$\Sigma = \{1,2,\ldots,c-1\} \cup \{\overleftarrow{1}, \overrightarrow{1}, \overleftrightarrow{1}\} \cup \{\#\}.$$

We refer to the three symbols $\overleftarrow{1}$, $\overrightarrow{1}$, $\overleftrightarrow{1}$ as *decorated letters*, and in describing the construction also make use of one more decorated letter: $\overleftarrow{2}$. Specifically, we have

$$\eta_c(w) = p_0\#p_1\#p_2\#\cdots\#p_{m-1}\#$$

(recall that m is the maximum value of a letter in w) where each p_i does not contain the symbol $\#$. The words p_i are referred to as *panel words*. Each panel word corresponds to a subword of w. More specifically, p_k^{+k} is, after removal of the decorations from any letters, actually a subword of w, and together these subwords partition the letters of w. Therefore, ignoring the punctuation symbols, η_c is an entry-to-entry mapping. The careful reader may note that all panel words of index greater than $m-c+1$ are empty by construction; these are recorded (with punctuation) merely for the sake of consistency.

The construction is recursive: we extract the panel words from w in order, starting with p_0, so it is convenient to consider also a sequence of *remainder words* $r_0, r_1, \ldots, r_{m-1}$ that represent the as-yet-unencoded letters of w. Each word r_i is defined over the alphabet $\mathbb{P} \cup \{\overleftarrow{1}\}$.

The first step of the process is to set $r_0 = w$. Suppose that r_0 has k letters of value 1 and express it as

$$r_0 = r_{0,0}\ 1\ r_{0,1}\ 1\ r_{0,2}\ \cdots\ r_{0,k-1}\ 1\ r_{0,k},$$

so $r_{0,j} \in (\mathbb{P} \setminus \{1\})^*$ for all j. Let J denote the set of indices j between 1 and k such that $r_{0,j}$ contains a letter of value c (the large factors). It follows from the small ascent condition and the fact that it is preceded by a 1 that if $r_{0,j}$ contains a letter of value c (i.e., $j \in J$) then it contains the subword $23 \cdots c$, and so $|J| \leq c-1$ because $w \in \mathcal{L}_c^\infty$. Note that $r_{0,0}$ may also contain a letter of value c. These factors are precisely what we do *not* encode in the panel word p_0. Another consequence of the small ascent condition is that each non-empty word $r_{0,j}$ for $j \geq 1$ (and particularly all of those with $j \in J$) begins with a 2.

We now *decorate* $2|J|$ letters of r_0, producing an auxiliary word, t_0. We define t_0 via factors $t_{0,j}$ for $0 \leq j \leq k$. We first set $t_{0,j}$ equal to $r_{0,j}$ for all $j \notin J$. For $j \in J$, we know that $r_{0,j}$ is nonempty and so begins with a 2, and we set the corresponding $t_{0,j}$ equal to $r_{0,j}$ with its leftmost 2 adorned by a $\leftarrow$, turning it into a $\overleftarrow{2}$, which we call a *left letter*. Each 2 which is turned into a $\overleftarrow{2}$ in this process is immediately preceded by a 1 in r_0, which we call a *right letter* and denote by $\overrightarrow{1}$ in t_0. After performing these decorations, the auxiliary word t_0 can be written as

$$t_0 = t_{0,0}\ \ell_1\ t_{0,1}\ \ell_2\ t_{0,2}\ \cdots\ t_{0,k-1}\ \ell_k\ t_{0,k}$$

where

$$\ell_j = \begin{cases} \overleftarrow{1} & \text{if } j \in J \text{ and} \\ 1 & \text{if } j \notin J, \end{cases}$$

and $t_{0,j}$ begins with $\overleftarrow{2}$ if and only if $j \in J$.

We can now construct our first *panel word*. It is simply the concatenation of all factors $t_{0,j}$ with $j \notin J \cup \{0\}$ (the small factors) and all ℓ_i, retaining their order in t_0. To be precise, if we define

$$p_{0,j} = \begin{cases} \epsilon & \text{if } j \in J \text{ and} \\ t_{0,j} & \text{if } j \notin J, \end{cases}$$

then

$$p_0 = \ell_1\ p_{0,1}\ \ell_2\ p_{0,2}\ \cdots\ p_{0,k-1}\ \ell_k\ p_{0,k}.$$

The word p_0 is thus defined over the alphabet $\{1, 2, \ldots, c-1\} \cup \{\overrightarrow{1}\}$. Finally, we define r_1 to be the result of concatenating the remaining factors $t_{0,j}$ for $j \in J \cup \{0\}$ and then subtracting 1 from each letter (when we subtract 1 from a $\overleftarrow{2}$ we change it to a $\overleftarrow{1}$). Thus r_1 is defined over the alphabet $\mathbb{P} \cup \{\overleftarrow{1}\}$, and the maximum value of a symbol occurring in r_1 is $m-1$.

The decorations of the panel word and the remainder word specify the way to reassemble w from (p_0, r_1). To do so we first form r_1^{+1}. We divide this into factors each beginning with a $\overleftarrow{2}$ (except possibly the first factor). These are the factors $r_{0,j}$ for $j \in J \cup \{0\}$. If r_1^{+1} contains an initial factor that does not begin with $\overleftarrow{2}$ then we place this factor before p_0. Then proceeding from left to right we insert the first of the factors of r_1^{+1} that begin with $\overleftarrow{2}$ immediately after the first $\overrightarrow{1}$ of p_1, then the second factor immediately after the second $\overrightarrow{1}$ of p_1, and so on. Effectively, we "zip together" p_0 and r_1 using the arrows to mark the points where the two pieces should mesh with one another. We finish by removing the decorations.

There is only one change in subsequent iterations of this encoding. In constructing t_j, p_j and r_{j+1} from r_j for $j \geq 1$, we may wish to designate a $\overleftarrow{1}$ (a former left letter) as a right letter. If this situation arises, we simply turn the $\overleftarrow{1}$ into a $\overleftrightarrow{1}$. Thus after we have decorated r_j to form t_j, every decorated letter is either a $\overleftarrow{1}$ or occurs in a $\overrightarrow{1}\,\overleftarrow{2}$ or $\overleftrightarrow{1}\,\overleftarrow{2}$ factor. We call the resulting mapping $\psi : r_j \mapsto (p_j, r_{j+1})$ the *splitting mapping*.

It follows that, as claimed, every panel word is defined over the alphabet $\{1, 2, \cdots, c-1\} \cup \{\vec{1}, \overleftarrow{1}, \overleftrightarrow{1}\}$. Moreover, because the greatest letter in w has the value m, when we come to construct p_{m-1} from r_{m-1} all (if any) remaining letters are 1 (or a decorated version thereof) and thus after this stage we can guarantee that we have encoded all of w. As promised, our ultimate encoding consists of the concatenation of these panel words, separated by punctuation, $\eta_c(w) = p_0 \# p_1 \# \cdots \# p_{m-1} \#$.

We illustrate this process with a concrete example. Suppose that $c = 4$ and consider encoding the word

$$w = 2312312232345231233412123212343 \in \mathcal{L}_4^\infty.$$

In our first step, we set $r_0 = w$, divide it into factors, and add decorations (here and in what follows we underline the factors that remain in r_{j+1}). We call the resulting decorated word t_0. In our example, this step yields

$$t_0 = \underline{23}\ 123\ \vec{1}\ \underline{\overleftarrow{2}23234523}\ \vec{1}\ \underline{\overleftarrow{2}334}\ 121232\ \vec{1}\ \underline{\overleftarrow{2}343}.$$

We then form both p_0 and r_1 and repeat the process to form t_1, computing

$$\begin{array}{rclcrcl} p_0 & = & 123\ \vec{1}\ \vec{1}\ 121232\ \vec{1}, & \qquad & r_1 & = & 12\ \overleftarrow{1}12\ 1\ 234\ 12\ \overleftarrow{1}\ 223\ \overleftarrow{1}232, \\ & & & & t_1 & = & 12\ \overleftarrow{1}12\ \vec{1}\ \underline{\overleftarrow{2}34}\ 12\ \overleftrightarrow{1}\ \overleftarrow{2}23\ \overleftarrow{1}232. \end{array}$$

The list of panel words and remainders is completed by performing these operations four more times, in which we find

$$\begin{array}{rclcrcl} p_1 & = & 12\ \overleftarrow{1}12\ \vec{1}\ 12\ \overleftrightarrow{1}\ \overleftarrow{2}23\ \overleftarrow{1}232, & \qquad & r_2 & = & \overleftarrow{1}23, \\ & & & & t_2 & = & \overleftarrow{1}23, \\ p_2 & = & \overleftarrow{1}23, & & r_3 & = & \epsilon, \\ & & & & t_3 & = & \epsilon, \end{array}$$

and all of p_3, r_4, t_4, and p_4 are empty. Our encoding of w is the concatenation of the panel words p_0, p_1, p_2, p_3, and p_4 separated by punctuation,

$$\eta_4(w) = 123\vec{1}\vec{1}121232\vec{1}\#12\overleftarrow{1}12\vec{1}12\overleftrightarrow{1}\overleftarrow{2}23\overleftarrow{1}232\#\overleftarrow{1}23\#\#\#.$$

We define

$$\mathcal{L}_c^\eta = \{\eta_c(w) \ : \ w \in \mathcal{L}_c^\infty\}$$

to be the image of $\mathcal{L}_c^\infty$ under η_c. Except for punctuating letters, the panel encoding η_c is an entry-to-entry mapping. Moreover, the following bookkeeping result, which follows immediately from the definition of η_c, gives more detail on the entry-to-entry property of η_c.

Observation 5.1. *An entry of value j in the panel word p_i of $\eta_c(w)$ corresponds to an entry of value $i+j$ in w. Hence, every entry of the panel word p_i corresponds to a letter of value $i+1$, $i+2$, $\ldots$, or $i+c-1$ in w, while every letter i in w corresponds to an entry in one of the panel words p_{i-c+1}, p_{i-c+2}, $\ldots$, or p_{i-1}.*

It should be clear that η_c is injective, but as we shall need some properties of its inverse in what follows, we shall be a little more explicit. We begin by defining ψ^{-1}, the inverse of the splitting mapping. This is the mapping that "zips together" a panel word and a remainder word as described previously for the case of p_0 and r_1.

Suppose that p (to be thought of as the most recent panel word) is a word over $\mathbb{P} \cup \{\vec{1}, \overleftarrow{1}, \overleftrightarrow{1}\}$ and r (to be thought of as the most recent remainder word) is a word over $\mathbb{P} \cup \{\overleftarrow{1}\}$, and that the number of right letters in p equals the number of left letters in r (both are equal to k below). Now express p and r in the form

$$\begin{array}{rclcccccccccccccc} p & = & & q_0 & \ell_1 & & & q_1 & \ell_2 & & \cdots & & q_{k-1} & \ell_k & & & q_k, \\ r & = & s_0 & & & \overleftarrow{1} & s_1 & & & \overleftarrow{1} & \cdots & s_{k-1} & & & \overleftarrow{1} & s_k, & \end{array}$$

where each ℓ_j is a right letter (i.e., $\vec{1}$ or $\overleftrightarrow{1}$). The inverse of the splitting mapping is defined by

$$\psi^{-1}(p, r) \quad = \quad s_0^{+1} \quad q_0 \quad \dot{\ell}_1 \quad 2 \quad s_1^{+1} \quad q_1 \quad \dot{\ell}_2 \quad 2 \quad \cdots \quad s_{k-1}^{+1} \quad q_{k-1} \quad \dot{\ell}_k \quad 2 \quad s_k^{+1} \quad q_k,$$

where s_i^{+1} is the shift by 1 mapping applied to s_i and

$$\dot{\ell}_i = \begin{cases} 1 & \text{if } \ell_i = \vec{1}, \\ \overleftarrow{1} & \text{if } \ell_i = \overleftrightarrow{1} \end{cases}$$

is the mapping that removes right arrows.

Supposing that $p_0, p_1, \ldots, p_{m-1}$ are words over $\mathbb{P} \cup \{\vec{1}, \overleftarrow{1}, \overleftrightarrow{1}\}$ and that the number of right letters of p_i is equal to the number of left letters of p_{i+1} for $0 \le i < m-1$ we can define

$$\Psi^{-1}(p_0 \# p_1 \# \cdots \# p_{m-1} \#) = \psi^{-1}(p_0, \psi^{-1}(\ldots \psi^{-1}(p_{m-3}, \psi^{-1}(p_{m-2}, p_{m-1})) \ldots)).$$

For $w \in \mathcal{L}_c^\infty$, $\Psi^{-1}(\eta_c(w)) = w$, so η_c is indeed an injection on $\mathcal{L}_c^\infty$ (and in this context we often write η_c^{-1} in place of Ψ^{-1}).

There are several features of ψ^{-1} that are important to draw attention to. First, except for removing some decoration and incrementing r by one, ψ^{-1} does not change the subwords p and r at all; that is, absent decoration, p and r^{+1} occur as subwords in $\psi^{-1}(p, r)$. Second, undecorated letters play no significant role in the reassembly process performed by ψ^{-1}, in fact their only role is to be copied into the output (possibly after incrementation). Thus if we delete an undecorated letter from $\eta_c(w)$ and then apply Ψ^{-1} the result is w with the corresponding letter deleted.

The next result provides the interface that we need later to impose basis conditions on panel encodings.

Proposition 5.2. *Let u be a subword of $\eta_c(w)$ whose letters occur in the contiguous set of panel words p_i, p_{i+1}, …, p_{i+k}. The relative positions of the letters of w corresponding to those in u are determined by the subword of $\eta_c(w)$ consisting of the letters in u together with all decorated letters of the panel words p_i, p_{i+1}, …, p_{i+k} and the punctuation symbols $\#$ between them.*

Proof. Write $\eta_c(w) = p_0 \# p_1 \# \cdots \# p_{m-1} \#$ and consider the process of inverting the η_c mapping. Once we have formed a word containing all of the letters of u we may stop, so it suffices to compute

$$\psi^{-1}(p_i, \ldots \psi^{-1}(p_{i+k}, \psi^{-1}(p_{i+k+1}, \ldots \psi^{-1}(p_{m-2}, p_{m-1}) \ldots)) \ldots) = \psi^{-1}(p_i, \ldots \psi^{-1}(p_{i+k}, r) \ldots),$$

where $r = \psi^{-1}(p_{i+k+1}, \ldots \psi^{-1}(p_{m-2}, p_{m-1}) \ldots)$. In $\psi^{-1}(p_{i+k}, r)$, the letters corresponding to r have lost their decoration, and thus may be forgotten by our observation above. Thus it suffices to compute $\psi^{-1}(p_i, \ldots \psi^{-1}(p_{i+k-1}, p_{i+k}) \ldots)$. Applying our observation again, we may remove all undecorated letters not belonging to u from these panels without affecting the eventual order of the letters corresponding to u. What remains is the information specified in the statement of the proposition (the punctuation symbols serving to distinguish p_i through p_{i+k}). $\square$

6. THE REGULARITY OF $\mathcal{L}_c^\eta$

Our ultimate aim is to establish that various sublanguages of $\mathcal{L}_c^\eta$ (corresponding to finitely based or well-quasi-ordered subclasses of 321-avoiding permutations) are regular. We first establish that $\mathcal{L}_c^\eta$ itself is regular. The material in this section is also somewhat technical so we again provide an initial informal discussion. We seek to recognise whether a word over the alphabet $\{1, 2, \ldots, c-1\} \cup \{\overleftarrow{1}, \overrightarrow{1}, \overleftrightarrow{1}\} \cup \{\#\}$ belongs to $\mathcal{L}_c^\eta$. The basic idea is to identify several necessary conditions which are collectively sufficient. Then, if we verify that each individual necessary condition corresponds to a regular language, the closure of regular languages under the Boolean operations proves the result we want. Roughly speaking there are three such necessary conditions: a translation of the small ascent condition, consistency in left-right decorations between consecutive panel words (here the fact that the number of such decorations is bounded is critical), and that the number of panel words captures the maximum letter(s) properly, i.e., that the encoding is not terminated too early or too late.

We begin with a more detailed look at various properties of panel words, remainder words, and the encodings $\eta_c(w)$. Denote by left(p) and right(p) the number of left letters and right letters of a word p (occurrences of $\overleftrightarrow{1}$ contribute to both counts).

Consider the language of all remainder words $r = r_j$ which could arise in the process of encoding words from $\mathcal{L}_c^\infty$. This is a language over the alphabet $\mathbb{P} \cup \{\overleftarrow{1}\}$. For $j = 0$, r is an arbitrary element of $\mathcal{L}_c^\infty$; in particular $\mathcal{L}_c^\infty$ is contained within the language under consideration. Otherwise, r is obtained from an earlier remainder word by marking the left and right letters, concatenating the factor preceding the initial 1 with the factors from any $\overleftarrow{2}$ up to but not including the subsequent (marked or unmarked) 1 and then reducing the value of all letters by 1. It follows inductively that any such word r satisfies the following three conditions.

(R1) The undecorated copy of r (obtained by substituting 1 for every $\overleftarrow{1}$) belongs to $\mathcal{L}_c^\infty$.

(R2) The inequality left$(r) < c$ holds.

(R3) If left$(r) = k$ and $r = s_0 \; \overleftarrow{1} \; s_1 \; \overleftarrow{1} \; \cdots \; \overleftarrow{1} \; s_k$ then each factor s_i for $1 \leq i \leq k$ contains a letter of value $c - 1$.

Define $\mathcal{R}_c$ to be the language of all words over the alphabet $\mathbb{P} \cup \{\overleftarrow{1}\}$ satisfying (R1)–(R3).

Next we consider the language of all panel words $p = p_j$ that arise in encodings $\eta_c(w)$, and observe that every such p satisfies the following five conditions.

(P1) The undecorated copy of p satisfies the small ascent condition.

(P2) If p is non-empty then its first letter is 1, $\overleftarrow{1}$, $\overrightarrow{1}$, or $\overleftrightarrow{1}$.

(P3) The inequalities left$(p) < c$ and right$(p) < c$ hold.

(P4) Any letter immediately following a right letter of p is one of 1, $\overrightarrow{1}$, $\overleftarrow{1}$, or $\overleftrightarrow{1}$.

(P5) If left$(p) = k$ and $p = q_0 \; \ell_1 \; q_1 \; \ell_2 \; q_2 \; \cdots \; q_{k-1} \; \ell_k \; q_k$, where the ℓ_i are the left letters of p, then each factor $\ell_i q_i$ for $1 \leq i \leq k$ contains either an occurrence of $c - 1$ or a right letter.

Establishing the validity of these properties is fairly straightforward, so we limit ourselves to a few words of justification. For (P1) note that any panel word is a concatenation of factors beginning with 1 satisfying the small ascent condition, hence does so itself. The property (P3) follows from (R2), because the left letters of $p = p_j$ are inherited from the remainder word r_j, while the right letters are matched with the left letters of the remainder word r_{j+1}. For (P4), the factor following a right letter up to the next 1 is carried forward to the next remainder, so the next letter of a panel word must have value 1. Finally, for (P5), the left letters in $p = p_j$ correspond to the left letters of r_j. These, in turn, correspond to the distinguished occurrences of $12\cdots c$ in r_{j-1}: of each such occurrence, $\overleftarrow{2}3\cdots c$ is carried forward into r_j where it becomes $\overleftarrow{1}2\cdots(c-1)$. If the symbol $c-1$ does not make it from r_j into p, the reason is that it is carried forward into r_{j+1}, in which case a right letter remains in p to indicate the location of its removal.

We define $\mathcal{P}_c$ to be the set of words over the alphabet $\{1, 2, \ldots, c-1\} \cup \{\overleftarrow{1}, \overrightarrow{1}, \overleftrightarrow{1}\}$ satisfying (P1)–(P5). The language $\mathcal{P}_c$ is clearly regular.

At this point we need to note that the splitting mapping $\psi : r_j \mapsto (p_j, r_{j+1})$, as defined in Section 5, can actually be applied to all words satisfying (R1)–(R3). We abuse terminology and denote this extension to $\mathcal{R}_c$ also by ψ. Further, we recall the inverse, ψ^{-1}, of the splitting mapping defined in the previous section, and note that its definition can be extended verbatim to all pairs (p, r) with $p \in \mathcal{P}_c$, $r \in \mathcal{R}_c$ and $\mathrm{right}(p) = \mathrm{left}(r)$.

Proposition 6.1. *The extended mappings ψ and ψ^{-1} are mutually inverse bijections between $\mathcal{R}_c$ and $\{(p, r) \in \mathcal{P}_c \times \mathcal{R}_c : \mathrm{right}(p) = \mathrm{left}(r)\}$.*

Proof. If $s \in \mathcal{R}_c$ and $\psi(s) = (p, r)$, it is easy to see that $p \in \mathcal{P}_c$, $r \in \mathcal{R}_c$. Also, we have $\mathrm{right}(p) = \mathrm{left}(r)$, because at the stage when the letters of s are decorated, the newly decorated letters occur in adjacent pairs and indicate precisely the positions where the splitting into p and r occurs.

Now take an arbitrary pair $(p, r) \in \mathcal{P}_c \times \mathcal{R}_c$ with $\mathrm{right}(p) = \mathrm{left}(r) = k$ and set $s = \psi^{-1}(p, r)$. To establish that $s \in \mathcal{R}_c$ we observe that it must possess the following properties.

- It lies in $(\mathbb{P} \cup \{\overleftarrow{1}\})^*$ because all right arrows have been removed.
- The undecorated version of s satisfies the small ascent condition because the undecorated copies of p and r satisfy this condition by (P1) and (R1), and the factors inserted into p to form s create 12 factors at their left hand ends (by definition of ψ^{-1}) , and descents at their right hand ends (by (P4)).
- It satisfies $\mathrm{left}(s) < c$, because the left letters are inherited from those of p, which satisfies (P3).
- Each factor of s between two consecutive left letters contains an occurrence of $c - 1$, as does the suffix following the last left letter. This follows because left letters of s are inherited from those of p. Thus by (P5) either there is already an occurrence of $c-1$ in such a factor, or there was a right letter in the corresponding part of p. In the latter case, a factor of r beginning with $\overleftarrow{1}$ was inserted (and increased by 1) following such a right letter, and by (R3) this results in an occurrence of c. The small ascent condition then also guarantees an occurrence of $c - 1$.

Therefore s indeed satisfies (R1)–(R3), as required. Moreover, ψ^{-1} has been designed precisely so that the splitting mapping reverses it, which completes the proof. $\square$

We now turn to the language $\mathcal{L}_c^\eta$ of all η_c encodings of words from $\mathcal{L}_c^\infty$. A typical word $e \in \mathcal{L}_c^\eta$ may be written as $e = p_0\#p_1\#\cdots\#p_{m-1}\#$, where the p_j do not contain #, and satisfies the following conditions.

(L1) For all $0 \le j < m$ we have $p_j \in \mathcal{P}_c$.

(L2) For all $0 \le j < m-1$ we have $\mathrm{right}(p_j) = \mathrm{left}(p_{j+1}) < c$, and also $\mathrm{left}(p_0) = \mathrm{right}(p_{m-1}) = 0$.

(L3) There exists an index j such that the word p_j contains the letter $m-j$.

(L4) For all $0 \le j \le m-1$, no letters of value greater than $m-j$ occur in p_j.

Only the last two conditions require comment and they hold because if $e = \eta_c(w)$ then the number of panel words, m, in e is equal to the maximum value occurring in w. This value must be encoded in some panel, say p_j, where it is encoded as $m-j$ by Observation 5.1, satisfying (L3). No p_j can contain a letter of value greater than $m-j$ because this would correspond to a letter of value greater than m in w, showing that (L4) is satisfied.

Proposition 6.2. *The language $\mathcal{L}_c^\eta$ consists precisely of all words $e = p_0\#p_1\#\cdots\#p_{m-1}\#$ that satisfy* (L1)–(L4)*.*

Proof. Suppose that $e = p_0\#p_1\#\cdots\#p_{m-1}\#$ satisfies (L1)–(L4). Set $r_m = \epsilon$, and then for k from $m-1$ down to 0 let $r_k = \psi^{-1}(p_k, r_{k+1})$. It follows from Proposition 6.1 that, at each step, the conditions required for ψ^{-1} to be defined on the given arguments apply, and so we obtain a sequence $r_{m-1}, \dots, r_1, r_0$ of elements of $\mathcal{R}_c$. By (L2), the final word $r_0 = w$ does not contain any decorated letters, and so in fact $w \in \mathcal{L}_c^\infty$ by (R1). Furthermore, the conditions (L3) and (L4) imply, via a straightforward inductive argument, that the maximum value occurring in w is precisely m. It follows that the panel encoding $\eta_c(w)$ will contain precisely m panel factors. These panel factors are obtained starting from $w = r_0$ and successively applying the splitting mapping $m-1$ times. By Proposition 6.1, the sequence of the remainder words obtained will be precisely $r_0, r_1, \dots, r_{m-1}$, while the sequence of the panel words will be $p_0, \dots, p_{m-1}$, so that $e = \eta_c(w) \in \mathcal{L}_c^\eta$, completing the proof. □

We conclude this section with its main result.

Proposition 6.3. *For every positive integer c the language $\mathcal{L}_c^\eta$ is regular.*

Proof. We show that the languages defined by the individual conditions (L1)–(L4) are regular. The first follows easily because (L1) defines the language $(\mathcal{P}_c\#)^*$, which is regular because $\mathcal{P}_c$ is regular.

Condition (L2) also defines a regular language because of the bound on the values to be compared. Note that (L2) is violated exactly if there is some panel word containing more than c right letters, or some pair of consecutive panel words where the number of left letters in the second panel word is not the same as the number of right letters in the first. Such a violation is easily recognised by a non-deterministic automaton (i.e., the automaton we describe accepts all words which fail this condition). The automaton accepts (i.e. identifies a violation) if it detects any left letters in the first panel word p_0. Otherwise it idles until it reaches some punctuation symbol. Then it counts right letters in the next panel word, accepting if that count exceeds c. If it still has not accepted, the automaton remembers this count (which is bounded by c) and proceeds to count left letters in the following panel word, again accepting if that does not match the stored count of right letters (or if the stored count is non-zero and there is no following panel word). Once the word has been completely read, if it has not been accepted it is rejected. If a violation occurs, the input word is accepted by some computation of this automaton, while if no violation occurs, no computation accepts the input word. Thus the words satisfying (L2) are the complement of the language accepted by this automaton, and so form a regular language.

Finally note that conditions (L3) and (L4) only present non-vacuous restrictions for the final $c-1$ panel words since $p_j \in \mathcal{P}_c$ for all indices j. We verify both conditions with a common automaton which reads encodings from right to left (recall that the reverse of a regular language is also regular). This automaton records the set of letters occurring in each of the panel words $p_{m-1}, \ldots, p_{m-c+1}$. Since this is a bounded amount of information, it can be stored in a state, and each condition implied by (L3) and (L4) is tested by direct inspection of the recorded information (i.e. by designating the appropriate states as accepting). □

7. Marking, Transducing, and Greediness

We have established that there is a bijective correspondence between $\mathcal{L}_c^\infty$ and the regular language $\mathcal{L}_c^\eta = \eta_c(\mathcal{L}_c^\infty)$. However, $\mathcal{L}_c^\eta$ is not good enough for our counting purposes, because a permutation $\pi \in \mathrm{Av}(321)$ generally has several (and a variable number of) possible griddings, and it is the latter that are encoded in $\mathcal{L}_c^\eta$. We therefore need to pass to our distinguished, unique—i.e. greedy—griddings. In other words, we need to consider the set $\eta_c(\mathcal{L}_c^\infty \cap \mathcal{G}^\infty)$ and prove that it is regular. To do this we return to the domino encoding. In general, as noted previously, the domino encoding is not a suitable device for detecting regularity because of the consistency requirement between consecutive dominoes and the lack of bounds on the number of symbols in a domino. Fortunately, the properties we are interested in (initially, greediness; in the next section, finite bases; after that, well-quasi-order) depend only on a bounded number of letters per domino factor. Here we develop a technique, called *marking*, that allows us to focus on such bounded sets of letters.

In a *marked* permutation some of the entries, designated with overlines, are distinguished from the remaining entries. Generally the reason for adding marks to a permutation is to follow the marking with a test that identifies the presence or absence of some specific configuration among the marked elements. Notationally, marked permutations and sets of such permutations are indicated with overlines.

Because our encodings are entry-to-entry mappings or nearly so (in the case of the domino encoding which maps a single entry to two letters), it is easy to define marked versions of them (which we also distinguish with overlines): the encoding of a marked permutation is obtained by marking the letter(s) of the encoding that correspond to marked entries of the permutation. Essentially we double the size of the alphabet, introducing a marked version of each non-punctuation letter. For instance, the *marked omnibus encoding* $\overline{\omega}$ maps marked gridded permutations to words whose letters are either marked or unmarked positive integers. The *marked domino encoding* $\overline{\delta}$ similarly maps marked gridded permutations to $\{\circ, \bullet, \overline{\circ}, \overline{\bullet}, \#\}^*$.

We denote by $\overline{\mathcal{L}}_c^\eta$ the marked version of $\mathcal{L}_c^\eta$, i.e., the set of all marked words which would lie in $\mathcal{L}_c^\eta$ if the markings on their non-punctuation symbols were removed. Note that in these words letters can be both decorated (with arrows) and marked (with overlines). Fortunately, we have no need to actually depict this.

Typically we consider markings of gridded permutations such that a bounded number of entries in each cell are marked and then ask about the subpermutation formed by the marked entries. We begin with a simple example of the type of results we establish.

In Section 3, we defined domino factors of arbitrary words in $\mathbb{P}^*$. Here we extend this definition to arbitrary marked words in $\overline{\mathbb{P}}^*$, though we are interested only in the marked letters: given $\overline{w} \in \overline{\mathbb{P}}^*$, the i^{th} *domino factor corresponding to its marked letters* is defined as $\overline{d}_i = \overline{w}|^{\{\overline{i},\overline{i+1}\}}$. Note that unmarked letters do not occur in $\overline{d}_i$.

Proposition 7.1. *Let $\overline{w} \in \overline{\mathcal{L}}_c^{\infty}$. The i^{th} domino factor corresponding to the marked letters of $\overline{w}$ is completely determined by the subword of $\overline{\eta}_c(\overline{w})$ consisting of those letters in panel factors $\overline{p}_{i-c+1}$, $\overline{p}_{i-c+2}$, $\ldots$, $\overline{p}_i$ that are marked or decorated (or both), along with the punctuation symbols between them.*

Proof. By Observation 5.1, the letters i and $i+1$ (and their marked versions) may only be encoded in the panel words $\overline{p}_{i-c+1}, \overline{p}_{i-c+2}, \ldots, \overline{p}_i$. The result now follows immediately from Proposition 5.2. $\square$

In fact we need stronger results than that above. We want to translate one encoding into another, restricting to marked entries. For this we use transducers. A *transducer* is a finite-state automaton (not necessarily deterministic) that may produce output while reading. Thus given an input alphabet Σ and an output alphabet Γ, each transition of a transducer has both an input symbol $a \in \Sigma \cup \{\epsilon\}$ and an output symbol $b \in \Gamma \cup \{\epsilon\}$. If the transducer T has an accepting computation on reading w, then the output of that computation is the word formed by concatenating the output symbols associated with the transitions performed (in the same order as those transitions). No output is associated with non-accepting computations. Note that output is associated to a specific computation, so for non-deterministic transducers the same input word w may yield multiple outputs.

A simple and illustrative example is the transducer with input alphabet Σ and output alphabet $\overline{\Sigma}$ which marks precisely one letter of its input. This transducer can be defined using an underlying automaton defined by the following three properties.

- It has an initial non-accepting state that has transitions to itself whose input/output pairs are a/a for each $a \in \Sigma$.
- It has a second state, which is accepting, that also has transitions to itself whose input/output pairs are a/a for each $a \in \Sigma$.
- There are transitions from the first to the second state whose input/output pairs are $a/\overline{a}$ for each $a \in \Sigma$.

We use functional notation, so if T is a transducer and X is a set of words (of the appropriate alphabet for T) then $T(X)$ is the set of words output by T while reading the words of X (which could be empty if none of the words of X are accepted by the underlying automaton). As usual, when X is a singleton we generally omit set braces and write $T(w)$. We utilise the following basic facts about transducers.

- If X is a regular language and T is a transducer, then $T(X)$ is again a regular language.
- Conversely, if Y is a regular language then the preimage $T^{-1}(Y) = \{x : T(x) \cap Y \neq \emptyset\}$ is regular as well.
- The composition of two transducers is again a transducer.

For further details see, for example, Sakarovitch [28, Chapter IV].

For our next result we must make another definition. Given a marked word $\overline{w} \in \overline{\mathcal{L}}^{\infty}$, the *domino encoding of the word formed by its marked letters* is

$$\overline{d}_0^{\bullet} \# \overline{d}_1^{\bullet} \# \cdots \# \overline{d}_m^{\bullet} \#,$$

where m is the maximum value of a marked or unmarked letter of $\overline{w}$, each $\overline{d}_i$ is the i^{th} domino factor corresponding to the marked letters of $\overline{w}$ defined previously, and $\overline{d}_i^{\bullet}$ is the translation of $\overline{d}_i$ to the alphabet $\{\overline{\circ}, \overline{\bullet}\}$ formed by replacing $\overline{i}$ by $\overline{\circ}$ and $\overline{i+1}$ by $\overline{\bullet}$.

Proposition 7.2. *For every fixed integer k there is a transducer that, given the panel encoding $\overline{\eta}_c(\overline{w})$ of a marked word $\overline{w} \in \overline{\mathcal{L}}_c^{\infty}$ with at most k marked copies of each symbol, outputs the domino encoding of the word formed by its marked letters.*

Proof. Given a panel word, $\overline{p}$, its *stripped form* is the subword consisting of all marked or decorated letters. Since the bound on the number of marked copies of any symbol implies a bound on the number of marked entries in each panel word, and the number of decorated entries in a panel word is bounded in any case, there is a finite set of stripped panel words that can arise from panel encodings of $\overline{\eta}_c(\overline{w})$. We view this set as a new alphabet. We then transduce $\overline{\eta}_c(\overline{w})$ into the word over this alphabet determined by replacing each panel word by the single letter corresponding to its stripped form, deleting (i.e., not transcribing) the punctuation symbols as we proceed. We call the resulting word the *stripped form* of $\overline{\eta}_c(\overline{w})$.

Given an arbitrary alphabet Σ, a positive integer c, and a placeholder symbol $\cdot$ not in Σ, we form the alphabet $\Gamma = (\Sigma \cup \{\cdot\})^c$ and a transducer from Σ^* to Γ^* that maps $u \in \Sigma^*$ to a word v in Γ^* of the same length with $v(i) = (u(i-c+1), u(i-c+2), \ldots, u(i))$ (replacing references to symbols of negative index by $\cdot$). Applying this transducer to the stripped form of $\overline{\eta}_c(\overline{w})$ gives a word whose symbols correspond to the sequences of c consecutive stripped panel words of $\overline{\eta}_c(\overline{w})$. Proposition 7.1 shows that the stripped forms of the marked panel words $\overline{p}_{i-c}, \ldots, \overline{p}_{i-1}$ determine the i^{th} domino factor for the marked letters of $\overline{w}$, so one final transducer that replaces each such sequence by its corresponding domino factor completes the process. □

Up to this point we have been working with $\mathcal{L}_c^{\eta}$, a regular language that is in one-to-one correspondence with $\mathcal{L}_c^{\infty}$, the language of words $w \in \mathbb{P}^*$ that satisfy the small ascent condition and contain no shift of $(12 \cdots c)^c$. Recall that $\mathcal{G}^{\infty}$ is the image of the greedy griddings of 321-avoiding permutations under the omnibus encoding ω. We define two additional languages:

$$\mathcal{G}_c^{\infty} = \mathcal{G}^{\infty} \cap \mathcal{L}_c^{\infty} \quad \text{and} \quad \mathcal{G}_c^{\eta} = \eta_c(\mathcal{G}_c^{\infty}).$$

It is our principal goal in this section to prove that $\mathcal{G}_c^{\eta}$ is regular, i.e., that the panel encodings of omnibus encodings of greedy staircase griddings can be recognised by a finite automaton. By

Observation 3.2 and the results of the previous section, this is equivalent to showing that the set of η_c encodings of words in $\mathcal{L}_c^\infty$ that satisfy (ωG1) and (ωG2) can be recognised by a finite automaton. Note that these two conditions apply only to the first and last occurrence of each letter. Furthermore, the first (resp., last) occurrence of each letter in a word $w \in \mathcal{L}^\infty$ will also be the first (resp., last) occurrence of the corresponding letter in some panel word of $\eta_c(w)$. Therefore our first step is to describe a transducer which marks the first and last letter of each value in every panel word of $\eta_c(w)$.

Proposition 7.3. *There is a transducer that, given $w \in \mathcal{L}_c^\eta$, outputs a marked panel encoding $\overline{w}$ in which the first and last entries of each value in each panel word are marked.*

Proof. It suffices to define the operation of such a transducer on a single panel word—the full transducer can then be built by non-deterministically looping back to the initial state when a punctuation symbol is read. In turn it suffices to construct such a transducer for each individual value k of a letter from 1 through $c-1$ (since these can then be composed to give the required transducer). The transducer defined by the following properties performs this task.

- The initial state, start, is an accepting state.
- In any state the transducer transcribes all input that is not a k (that is, outputs the same symbol as the input symbol) and remains in the current state.
- When (or if) the transducer first encounters a k, it outputs $\overline{k}$ and enters either state seenfirst or seenlast (non-deterministically).
- In state seenfirst (which is non-accepting) if the transducer encounters a k it either transcribes it and remains in state seenfirst, or outputs $\overline{k}$ and enters state seenlast.
- In state seenlast (which is accepting) if the transducer encounters a k then it fails, resulting in no output (this can be implemented by way of a state fail which has no further transitions).

Note that in the case $k = 1$, some of the occurrences of 1 in the input word may be decorated with arrows—the transducer retains those arrows as well as possibly adding marking. □

Propositions 7.2 and 7.3 give us the machinery we need in order to verify compliance with conditions (ωG1) and (ωG2), allowing us to prove the main result of the section.

Proposition 7.4. *For every positive integer c, the language $\mathcal{G}_c^\eta$ is regular.*

Proof. Let T denote the composition of the transducers from Propositions 7.3 and 7.2. Thus given an encoding $w \in \mathcal{L}_c^\eta$, T first marks the first and last entry of each value in each panel and then outputs the domino encoding of the word formed by these marked letters. Note that T produces precisely one output for each $w \in \mathcal{L}_c^\eta$, so we denote this output by $T(w)$, temporarily neglecting our convention that transducers always output sets. Also note that $T(w)$ contains (in addition to other letters) the first and last occurrence of each letter of w. Therefore $T(w)$ provides enough information to allow us to decide whether w satisfies the conditions (ωG1) and (ωG2).

We claim that $\mathcal{G}_c^\eta$ is the intersection of $\mathcal{L}_c^\eta$ and $T^{-1}(\mathcal{R})$, where $\mathcal{R}$ is a regular language. Every domino in $T(w)$ has at most $2c-2$ occurrences of each letter (a first and last occurrence of the letter in all $c-1$ panel words it could be encoded in). Thus there is a finite set Δ of dominoes which occur in the domino encodings output by T. We may therefore consider Δ itself to be the output alphabet and ignore the punctuation symbols (which are superfluous at this point), so that $T(w) \in \Delta^*$ for all $w \in \mathcal{L}_c^\eta$.

Now we need to check whether w satisfies (ωG1) and (ωG2). These conditions translate to simple conditions on the dominoes of $T(w)$: each domino other than the first must begin with $\circ$, and each domino other than the first and last must contain the subword $\bullet\circ$. Let $\mathcal{R} \subseteq \Delta^*$ denote the language of domino encodings which satisfy these conditions. Clearly $\mathcal{R}$ is regular, and it follows that $\mathcal{G}_c^\eta = \mathcal{L}_c^\eta \cap T^{-1}(\mathcal{R})$, completing the proof. $\square$

8. Detecting Basis Elements

The results of the previous sections establish that, for each positive integer c, the set of 321-avoiding permutations such that the omnibus encodings of their greedy griddings do not contain any shift of $(12\ldots c)^c$ is in bijective correspondence with the regular language $\mathcal{G}_c^\eta$. We have also observed in Proposition 4.2 that for any proper subclass $\mathcal{C} \subsetneq \mathrm{Av}(321)$ there is a positive integer c such that $\omega(\pi^\sharp) \in \mathcal{G}_c^\infty$ for all greedy griddings $\pi^\sharp$ of permutations $\pi \in \mathcal{C}$ and hence the panel encodings of these omnibus encodings are contained in $\mathcal{G}_c^\eta$. To complete our goal of showing that any such *finitely based* class has a rational generating function, we need to show how to detect avoidance (or, equivalently, containment) of specified permutations within the panel encodings, while maintaining regularity.

The difficulty we are facing is that none of the three encodings we have used thus far—the omnibus encoding, its composition with the panel encoding, and the domino encoding—provide an easy way to test containment. To overcome this difficulty we resort again to the technique of marking, but this time we transduce the marked subpermutation to yet another encoding, namely the Dyck path encoding. This encoding—which was essentially described in the Introduction and illustrated on the right of Figure 1—consists of constructing a Dyck path whose outer corners lie just outside the left-to-right maxima of the permutation. We turn the resulting Dyck paths into words over the alphabet $\{\mathsf{u}, \mathsf{d}\}$ in the standard way. For instance, the Dyck path encoding of the permutation 31562487 depicted in Figure 1 is $\mathsf{u}^3\mathsf{d}^2\mathsf{u}^2\mathsf{dud}^3\mathsf{u}^2\mathsf{d}^2$.

Proposition 8.1. *For every fixed positive integer k there is a transducer that, given the domino encoding of a staircase gridding of a 321-avoiding permutation π with at most k entries per cell, outputs the Dyck path corresponding to π.*

Proof. As in the proof of Proposition 7.2 the bound on the number of entries per cell means that we may view the domino factors as letters themselves coming from a finite alphabet. In fact, borrowing another idea from the same proposition, we can view triples of consecutive translated domino factors $d^\bullet_{2i-1}$, $d^\bullet_{2i}$, $d^\bullet_{2i+1}$ (including padding at the beginning and end by empty domino factors) as individual letters. The reason for doing this is that we will show that we can compute the part of the Dyck path determined by the left-to-right maxima lying in the $2i^{\text{th}}$ and $(2i+1)^{\text{st}}$ cells from such a triple. Thus our transducer need only examine these triples in turn, and output the appropriate segment of a Dyck path for each one. This is illustrated in Figure 8.

To justify the claim we note that the information encapsulated in $d^\bullet_{2i-1}$, $d^\bullet_{2i}$, $d^\bullet_{2i+1}$ completely determines the relative values and positions of all entries in the $(2i-1)^{\text{st}}$ through $(2i+2)^{\text{nd}}$ cells. In particular it determines the left to right maxima in the $(2i)^{\text{th}}$ and $(2i+1)^{\text{st}}$ cells, and their relative

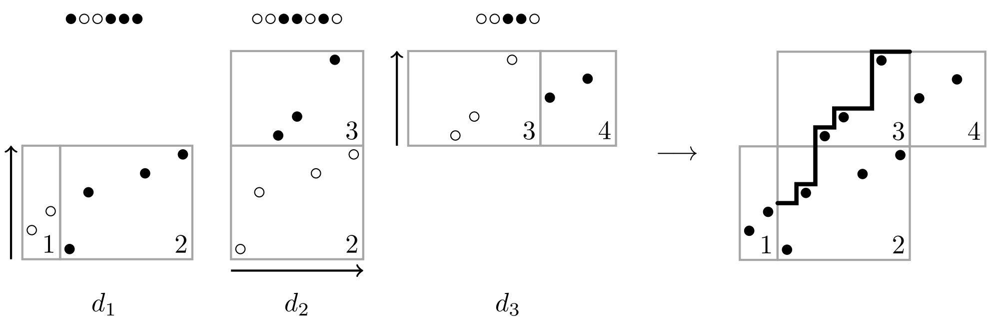


Figure 8: Upon reading the triple of domino factors shown in the top left, the transducer of Proposition 8.1 can compute the partial permutation shown on the right, and output the steps of the Dyck path passing through the 2^{nd} and 3^{rd} cells, $\mathsf{duduuududdduuudd}$.

positions with respect to the entries to the right and below them, all of which can be found in the $(2i)^{\text{th}}$ and $(2i+2)^{\text{nd}}$ cells. The final entry in the $(2i-1)^{\text{st}}$ cell (which is automatically a left to right maximum) indicates the entry point of the Dyck path into the $(2i)^{\text{th}}$ cell. (If the $(2i-1)^{\text{st}}$ cell is empty the path enters through the bottom left corner.) From the entry point, the path proceeds as dictated by the left to right maxima in the $(2i)^{\text{th}}$ and $(2i+1)^{\text{st}}$ cells and the entries to the right and below them. □

For any $\beta \in \mathrm{Av}(321)$ and positive integer c we now define $\mathcal{G}^{\eta}_{c,\geq\beta}$ to be the set of all encodings $\eta_c(\omega(\pi^\sharp))$ such that

- $\pi^\sharp$ is the greedy encoding of π,
- π contains β, and
- $\omega(\pi^\sharp)$ avoids all shifts of $(12\cdots c)^c$.

The transducer from our previous proposition shows that this is a regular language:

Proposition 8.2. *The language $\mathcal{G}^{\eta}_{c,\geq\beta}$ is regular.*

Proof. Let k denote the length of β. There is a non-deterministic transducer that takes words in $\mathcal{L}^{\eta}_c$ as input and outputs marked forms that contain exactly k marked letters. Denote by T the composition of that transducer and the one defined in Proposition 7.2 (which allows for up to k copies of each symbol) followed by the transducer described in Proposition 8.1. Further let X_β denote the singleton set whose only element is the word over the alphabet $\{\mathsf{u}, \mathsf{d}\}$ that represents the Dyck path corresponding to β.

Since T takes as input the panel encoding of the greedy gridding of a 321-avoiding permutation, marks exactly k letters, and outputs the Dyck path encoding of the marked letters, the panel encoding of some permutation π belongs to $T^{-1}(X_\beta) \cap \mathcal{G}^{\eta}_c$ if and only if β is contained in π. Thus $\mathcal{G}^{\eta}_{c,\geq\beta} = T^{-1}(X_\beta)$ and, being the preimage of a regular language (any singleton is regular) by a transducer, is itself regular. □

We have finally reached the point where we can prove the first half of our main result.

Proof of Theorem 1.1 (for finitely based subclasses). Suppose that the basis of a class $\mathcal{C}$ is the finite, nonempty, set B. Take any positive integer c such that $\omega(\pi^\sharp) \in \mathcal{G}_c^\infty$ for all greedy griddings $\pi^\sharp$ of permutations in $\mathcal{C}$ (the existence of such a value of c is guaranteed by Proposition 4.2). Then the set of panel encodings, $\mathcal{G}^\eta_{c,\mathcal{C}}$, of members of $\mathcal{C}$ is

$$\mathcal{G}^\eta_{c,\mathcal{C}} = \mathcal{G}^\eta_c \setminus \bigcup_{\beta \in B} \mathcal{G}^\eta_{c,\geq\beta}.$$

This is a regular language owing to Propositions 7.4 and 8.2 and the closure of the family of regular languages under Boolean operations. Therefore $\mathcal{C}$ is in one-to-one correspondence with a regular language. Moreover, if $\pi \in \mathcal{C}$ has length n then its image under the correspondence contains n non-punctuation symbols. The generating function of a regular language over commuting variables corresponding to its letters is a rational function and we can obtain the generating function for $\mathcal{C}$ from that for $\mathcal{G}^\eta_{c,\mathcal{C}}$ by replacing the variable corresponding to the punctuation symbol # by 1, and those variables corresponding to non-punctuation symbols by x, so the generating function of $\mathcal{C}$ is rational. $\square$

9. Well-Quasi-Ordered Subclasses

It remains to prove the second half of Theorem 1.1, namely that every well-quasi-ordered subclass of 321-avoiding permutations has a rational generating function. This proof breaks naturally into two parts. First we identify a necessary and sufficient condition for a subclass of Av(321) to be well-quasi-ordered. Then we show, using arguments similar to those in the preceding section, that this condition implies regularity of the corresponding languages. For the first part we identify a particular antichain $U \subseteq$ Av(321). Obviously, for a class $\mathcal{C} \subseteq$ Av(321), $\mathcal{C} \cap U$ must be finite. It happens that this condition is also sufficient. We begin with some preparatory remarks.

A permutation π is said to be *sum decomposable* if it can be written as a concatenation $\alpha\beta$ where every entry in the prefix α is smaller than every entry in the suffix β. If π has no non-trivial partition of this form then it is said to be *sum indecomposable*. We may in this way interpret an arbitrary permutation as a word over its sum indecomposable components (*sum components* for short).

Moving to a more general context, given a poset $(P, \leq)$, the *generalised subword order* on P^* is defined by $v \leq w$ if there are indices $1 \leq i_1 < i_2 < \cdots < i_{|v|} \leq |w|$ such that $v(j) \leq w(i_j)$ for all j. The following well-known result connects the well-quasi-ordering of P and P^*.

Higman's Lemma [19]. If $(P, \leq)$ is well-quasi-ordered then P^*, ordered by the subword order, is also well-quasi-ordered.

Returning to the context of permutations (and the containment order defined on them), Higman's Lemma easily implies the following result. (For more details we refer the reader to Atkinson, Murphy, and Ruškuc [11, Theorem 2.5].)

Proposition 9.1. *Let $\mathcal{C}$ be a permutation class. If the sum indecomposable members of $\mathcal{C}$ are well-quasi-ordered, then $\mathcal{C}$ is well-quasi-ordered.*

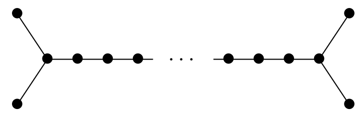


Figure 9: A double-ended fork.

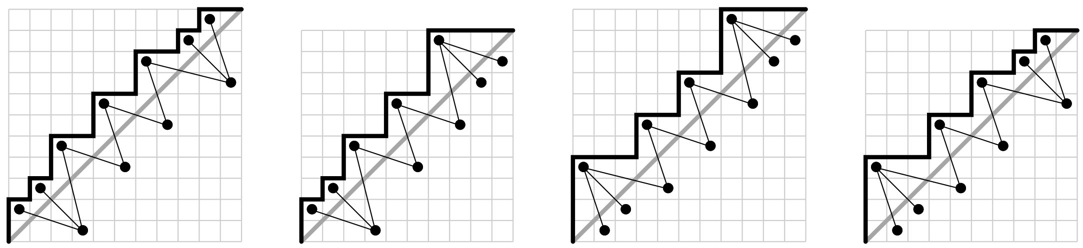


Figure 10: The different types of members of U, shown with both their inversion graphs and associated Dyck paths.

The identification of the antichain U requires a short digression related to a connection between permutations and graphs. Given a permutation π, the *inversion graph* corresponding to π is the graph G_π on the (unlabeled) vertices $\{(i, \pi(i))\}$ in which $(i, \pi(i))$ and $(j, \pi(j))$ are adjacent if they form an inversion, i.e., $i < j$ and $\pi(i) > \pi(j)$. As each entry of π corresponds to a vertex of G_π, we commit a slight abuse of language by referring (for example) to the degree of an entry of π when we mean the degree of the corresponding vertex of G_π. Note that the graph G_π is connected if and only if π is sum indecomposable.

If σ is a subpermutation of π, then the induced subgraph of G_π on the entries corresponding to a copy of σ is isomorphic to G_σ. Thus the image of a permutation class under the mapping $\pi \mapsto G_\pi$ is a class of inversion graphs closed under taking induced subgraphs. In particular, as occurrences of 321 in π correspond to triangles in G_π and no inversion graph may contain an induced cycle on 5 or more vertices, the 321-avoiding permutations correspond to bipartite inversion graphs. More importantly for our purposes, the inverse image of an antichain of graphs (in the induced subgraph ordering) is an antichain of permutations. Note incidentally that this is true even though the mapping $\pi \to G_\pi$ is not injective (in particular, $G_\pi \cong G_{\pi^{-1}}$ for all permutations π). These graphs have previously been studied in the context of well-quasi-order by Lozin and Mayhill [23], although we do not require their results here.

Let us consider permutations whose graphs are isomorphic to paths on $n \geq 4$ vertices. By direct construction it is easy to verify that there are precisely two such permutations of each length, which we call *increasing oscillations*:

$$\begin{array}{lll} 2416385\cdots n(n-3)(n-1), & 3152749\cdots(n-4)n(n-2) & \text{if } n \text{ is even, and} \\ 2416385\cdots(n-4)n(n-2), & 3152749\cdots n(n-3)(n-1) & \text{if } n \text{ is odd.} \end{array}$$

A *double-ended fork* is the graph formed from a path by adding four vertices of degree one, two adjacent to one end of the path and two adjacent to the other. An example is shown in Figure 9. It is clear that the set of double-ended forks is an antichain of graphs in the induced subgraph ordering.

Let U denote the set of all permutations π for which G_π is isomorphic to a double-ended fork. As in

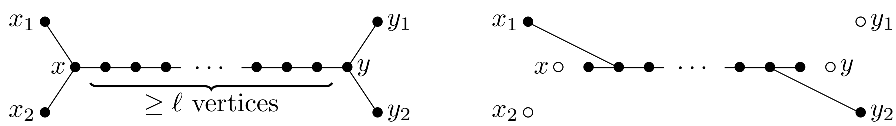


Figure 11: Two situations which arise in the proof of Proposition 9.2.

the case of increasing oscillations, direct construction shows that there are four slightly different types of members of U, depicted in Figure 10. By inspection $U \subseteq \text{Av}(321)$, which also follows because double-ended forks are bipartite. By our previous remarks, it follows that U forms an infinite antichain. In particular, every well-quasi-ordered subclass of Av(321) must have finite intersection with U. To establish the other direction, we begin with the following structural result.

Proposition 9.2. *If the subclass $\mathcal{C} \subseteq \text{Av}(321)$ has finite intersection with U then there is a number ℓ such that for all connected graphs G_π with $\pi \in \mathcal{C}$, the distance between any two vertices of degree three or greater is at most ℓ.*

Proof. Suppose that $\mathcal{C}$ contains no members of U of length $\ell + 2$ or longer (here length refers to the length of the permutation) for some $\ell \geq 4$ and choose $\pi \in \mathcal{C}$ to be an arbitrary sum indecomposable permutation.

Let x and y be two entries of π of degree three or greater and suppose to the contrary that the distance between these vertices is greater than ℓ, so there is a shortest path P in G_π between x and y with at least ℓ internal vertices. Because x and y each have degree at least three, x has neighbours $x_1 \neq x_2$ which do not lie on P and y has neighbours $y_1 \neq y_2$ which do not lie on P. Because the distance between x and y is at least $\ell \geq 4$, note that neither x_1 nor x_2 can be adjacent to y, y_1, or y_2 (and vice versa with x and y swapped). Also, because G_π does not contain a triangle, x_1 is not adjacent to x_2 and y_1 is not adjacent to y_2. If none of x_1, x_2, y_1, or y_2 are adjacent to any vertices of P other than x or y then $P \cup \{x_1, x_2, y_1, y_2\}$ is isomorphic to a double-ended fork on at least $\ell + 6$ vertices (as shown on the right of Figure 11), a contradiction.

On the other hand, if one or both of x_1 or x_2 were adjacent to another vertex of P then it could not be the vertex of P at distance one from x as this would create a triangle (a copy of 321 in π) and it also could not be a vertex of distance three or greater from x as this would contradict our choice of P (as a shortest path). Thus the only possibility would be the vertex of P at distance two from x, as shown on the right of Figure 11. An analogous analysis implies that if one or both of y_1 or y_2 were adjacent to another vertex of P then that vertex would have to be the vertex of distance two from y. In any case, as indicated on the right of Figure 11, we find an induced double-ended fork on at least $\ell + 2$ vertices, a contradiction which completes the proof. $\square$

We are now ready to prove that having finite intersection with U is a sufficient condition for a subclass of 321-avoiding permutations to be well-quasi-ordered. By Proposition 9.1, it suffices to consider the sum indecomposable members of our subclass. We then use Proposition 9.2 to show that these sum indecomposable permutations have severely constrained structure; in particular, we show that it implies that "most" of their entries are confined to a bounded number of cells. This characterisation is then shown to be sufficient for another appeal to Higman's Lemma, from which well-quasi-ordering follows.

Theorem 9.3. *A subclass $\mathcal{C} \subseteq \text{Av}(321)$ is well-quasi-ordered if and only if $\mathcal{C} \cap U$ is finite.*

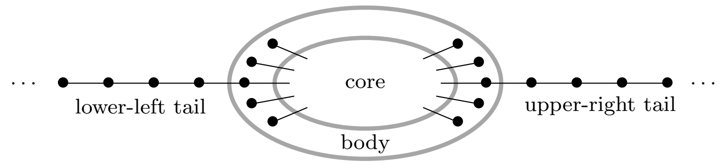


Figure 12: The core, body and tails of a 321-avoiding inversion graph.

Proof. By our previous remarks, it suffices to show that if $\mathcal{C} \cap U$ is finite for a subclass $\mathcal{C} \subseteq \mathrm{Av}(321)$ then the sum indecomposable permutations in $\mathcal{C}$ are well-quasi-ordered. To this end, suppose that $C \cap U$ is finite, choose a sum indecomposable permutation $\pi \in \mathcal{C}$, and fix a particular (not necessarily greedy) staircase gridding $\pi^\sharp$ of π. Thus every entry of π lies in some cell; we refer to the number of this cell as the *label* of the entry or corresponding vertex in G_π.

Because inversions in π can occur only between adjacent cells in the gridding, we conclude that the labels of adjacent vertices in G_π differ by precisely 1. In particular, the distance between two entries of π in G_π is bounded below by the difference of their labels. Thus by Proposition 9.2, all vertices in G_π of degree three or greater have labels in some bounded interval $\{i, i+1, \ldots, i+\ell\}$, where i is the least label of such a vertex (if no such vertices exist, choose $i = 0$) and ℓ depends only on $\mathcal{C}$. We refer to all entries of π in these cells as the *core* of π.

We aim to partition the entries of $\pi^\sharp$ into three groups: a body, comprising the core of $\pi^\sharp$ together with some of the entries from the adjacent cells at either end, a lower-left tail, and an upper-right tail. The two tails will comprise the entries of $\pi^\sharp$ to the southwest (respectively, northeast) of the core, and the graph induced by each tail will be shown to be a path.

To define this partition, first consider the entries outside the core in G_π. This set is naturally divided into two pieces: T_{SW}, consisting of entries belonging to cells of label less than i, and T_{NE}, consisting of entries belonging to cells of label greater than $i + \ell$. Since all vertices in these pieces have degree at most two and the graph G_π is connected, each consists of a disjoint union of paths. In fact, at most one of these paths in each piece can contain more than one vertex. Indeed, the vertices in two different paths within T_{SW}, say, would each correspond to entries of π forming a copy of 21, 231, 312, or an increasing oscillation. One of these would have to lie to the left and below the other (because the paths are disjoint), but then one can see that it cannot be connected to the core, and this contradicts the sum indecomposability of π.

Consequently, every vertex of G_π that does not correspond to an entry in the core either lies in one of two paths or is only adjacent to (at most two) vertices in the core. This latter collection of vertices must all lie in one of the two cells immediately adjacent to the cells that form the core, and we form the *body* of π by adding all these entries to the core (at which point the body is contained in at most $\ell + 3$ cells). The entries of T_{SW} which still lie outside the body now form a path in G_π. This path, if nonempty, must contain at least two vertices as otherwise it would already be included in the body. If the path is nonempty, we add the vertex of this path which is adjacent to the core to the body and call the remaining vertices the *lower-left tail.* We then perform the analogous operation on the entries of T_{NE} to form the *upper-right tail.* Note that the body is contained in at most $\ell + 3$ cells at the end of this process.

Our sum indecomposable permutation π now has a graph of the form shown in Figure 12 where each of the two tails is either absent or else contains at least one vertex outside the body which is adjacent to a vertex of degree two inside the body. Note also that it is possible in our gridding of π that some entries of the two tails can share cells with entries of the body, but this is of no consequence: they are included in the tail, and not in the body.

The subpermutation of π that makes up the body of π, together with the first point of each tail (i.e., the one adjacent to the body, if there is a tail) inherits a staircase gridding (which need not be greedy) from $\pi^\sharp$ in which it occupies not more than $\ell+3$ cells. This means that the body has a gridding into cells $1, 2, \ldots, \ell+3$ or $2, 3, \ldots, \ell+4$ depending on the parity of the first cell in the inherited gridding. Denote the omnibus encoding of this gridding of the body by w_π; this is a word over the alphabet $\{1, 2, \ldots, \ell+4\}$.

We now form a marked version of w_π. The lower tail of π has length $t^{\pi^\sharp}_{\mathrm{SW}} \geq 0$, while the upper tail has length $t^{\pi^\sharp}_{\mathrm{NE}} \geq 0$. If $t^{\pi^\sharp}_{\mathrm{SW}}$ (resp. $t^{\pi^\sharp}_{\mathrm{SE}}$) is non-zero, then there is a unique entry in the body which is adjacent to an entry of the lower (resp. upper) tail. We mark the letter of w_π which corresponds to this entry with an underline (resp. overline), and denote the resulting marked version of w_π by $\overline{w}_\pi$. The relative positions between all entries of the body and the two tails are now determined by $\overline{w}_\pi$, though the lengths of the tails are not captured in this word.

Let $\overline{\Sigma}$ be the extended alphabet consisting of the symbols $\{1, 2, \ldots, \ell+4\}$ together with over- and underlined versions of each. The discussion above defines an injective mapping from sum indecomposable permutations in $\mathcal{C}$ to $\overline{\Sigma}^* \times \mathbb{N} \times \mathbb{N}$ given by

$$\pi^\sharp \mapsto (\overline{w}_\pi, t^\pi_{\mathrm{SW}}, t^\pi_{\mathrm{NE}}).$$

Define an ordering on $\overline{\Sigma}^* \times \mathbb{N} \times \mathbb{N}$ by taking product of the subword ordering on $\overline{\Sigma}^*$ and the usual orderings on the two copies of $\mathbb{N}$. Because $\overline{\Sigma}^*$ is well-quasi-ordered by Higman's Lemma and the product of well-quasi-orders is again well-quasi-ordered, $\overline{\Sigma}^* \times \mathbb{N} \times \mathbb{N}$ is well-quasi-ordered. Moreover, if $(\overline{w}_\sigma, t^\sigma_{\mathrm{SW}}, t^\sigma_{\mathrm{NE}}) \leq (\overline{w}_\pi, t^\pi_{\mathrm{SW}}, t^\pi_{\mathrm{NE}})$ in this ordering then $\sigma \leq \pi$ as the comparability on the first coordinate implies that the body of σ embeds into the body of π in a way preserving the relative positions of the entries adjacent to the two tails (a consequence of Observation 3.3). The inequality of tail lengths then allows for the entire embedding of σ into π to be completed. Hence, with respect to subpermutation ordering, the sum indecomposable members of $\mathcal{C}$ are well-quasi-ordered, and so $\mathcal{C}$ is as well by Proposition 9.1. □

We now turn to the second half of the argument—that all well-quasi-ordered subclasses of Av(321) are encoded by regular languages. Guided by Theorem 9.3, we would like to check the involvement of sufficiently long members of U in a subclass $\mathcal{C}$ by considering the encodings $(\eta_c \circ \omega)(\pi^\sharp)$ of greedy griddings of members of $\mathcal{C}$ and an appropriate value of c. To achieve this, we resort once more to the Dyck path encodings. First, as indicated in Figure 10, it is easy to see that the Dyck path encodings of members of U form a regular language—outside of bounded prefixes and suffixes these words consist of repetitions of $\mathsf{u}^2\mathsf{d}^2$.

In fact we are interested in the encodings of sets $U_{\geq q}$ for $q \in \mathbb{P}$, consisting of permutations in U of length at least q. Noting that $U \setminus U_{\geq q}$ is finite for every value of q we obtain the following.

Proposition 9.4. *For any positive integer q, the language of Dyck paths corresponding to the members of $U_{\geq q}$ is regular.*

As demonstrated in Figure 13 it is impossible for a cell of a staircase gridding of an increasing oscillation to contain four or more entries. As every member of U is formed by adding two entries

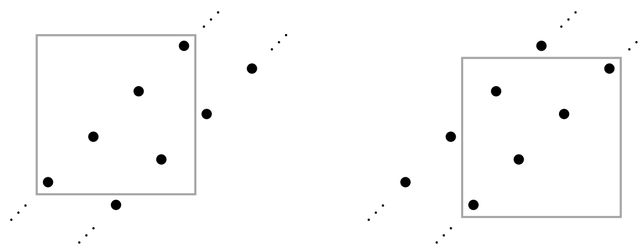


Figure 13: In any staircase gridding of an increasing oscillation, there can be at most three entries in a cell.

to an increasing oscillation, it follows that in every staircase gridding of a member of U each cell may contain at most five entries. In particular, if an element $\mu \in U$ occurs as a subpermutation of $\pi \in \mathrm{Av}(321)$ with greedy gridding $\pi^\sharp \in \mathcal{G}_c^\infty$, and if we mark the letters of $\eta_c(\omega(\pi^\sharp))$ corresponding to any one copy of μ in π, no more than $5(c-1)$ occurrences of each letter will be marked by Observation 5.1.

Proposition 9.5. *Let q be a positive integer and set $\mathcal{W}_q = \mathrm{Av}(321) \cap \mathrm{Av}(U_{\geq q})$. Further take c to be any positive integer such that the omnibus encodings of all greedy staircase griddings of members of $\mathcal{W}_q$ are contained in $\mathcal{G}_c^\infty$ (such a value of c is guaranteed to exist by Proposition 4.2). Then $\mathcal{G}^\eta_{c,\mathcal{W}_q}$, the set of panel encodings of members of $\mathcal{W}_q$, is regular.*

Proof. Combining Proposition 7.2 and Proposition 8.1 there is a transducer T that, when operating on panel encodings from $\mathcal{G}_c^\eta$, outputs the Dyck paths corresponding to subpermutations of the encoded permutation whose entries correspond to at most $5(c-1)$ copies of each symbol. The language of Dyck paths corresponding to the members of $U_{\geq q}$, say $\mathcal{D}$, is regular by Proposition 9.4. Finally, $\mathcal{G}^\eta_{c,\mathcal{W}_q}$ is the complement in $\mathcal{G}_c^\eta$ of the preimage under T of $\mathcal{D}$, and so is also regular. $\square$

We can now prove the second half of our main result.

Proof of Theorem 1.1 (for well-quasi-ordered subclasses). Using Theorem 9.3, choose a positive integer q such that $\mathcal{C}$ contains no element of $U_{\geq q}$, i.e., $\mathcal{C} \subseteq \mathcal{W}_q$, and choose c according to Proposition 4.2 so that the omnibus encodings all members of $\mathcal{W}_q$ are contained in $\mathcal{G}_c^\infty$. The minimal members of $\mathcal{W}_q \setminus \mathcal{C}$ form an antichain, say $B \subseteq \mathcal{W}_q$, which is finite because $\mathcal{W}_q$ is well-quasi-ordered. Thus we have

$$\mathcal{G}^\eta_{c,\mathcal{C}} = \mathcal{G}^\eta_{c,\mathcal{W}_q} \setminus \bigcup_{\beta \in B} \mathcal{G}^\eta_{c,\geq\beta}$$

and, as all parts of the right hand side are known to be regular (by Propositions 8.2 and 9.5) and B is finite, we may conclude that $\mathcal{G}^\eta_{c,\mathcal{C}}$ is regular. It follows that the generating function for $\mathcal{G}^\eta_{c,\mathcal{C}}$, which is equal to that of $\mathcal{C}$, is rational. $\square$

10. Conclusion

While we opened the paper by emphasising the differences between the two Catalan permutation classes defined by avoiding 312 and 321, respectively, our main result shows that they do share a remarkable property. Every finitely based or well-quasi-ordered proper subclass of either of these classes has a rational generating function. Of course, stating the result in this way obscures a serious difference: *all* subclasses of the 312-avoiding permutations are *both* finitely based and well-quasi-ordered.

One interested in actually computing these generating functions will notice an even more striking difference. While computing the enumeration of subclasses of 312-avoiding permutations is essentially trivial (as outlined in [1]), for subclasses of 321-avoiding permutations the enumeration method we have presented appears to be impractical.

Another context in which the differences between these classes are readily apparent is that of Wilf-equivalence. Two permutation classes $\mathcal{C}$ and $\mathcal{D}$ are said to be *Wilf-equivalent* if they are equinumerous, i.e., $|\mathcal{C}_n| = |\mathcal{D}_n|$ for all n. For classes defined by avoiding 312 and a single additional restriction, Albert and Bouvel [5] have provided a conjecturally complete classification of the Wilf-equivalences. However, while there are some enumerative coincidences among classes defined by avoiding 321 and a single additional restriction, empirically there does not appear to be anywhere near the same amount of collapse (into a small number of Wilf-equivalence classes). A related result was proved by Albert, Atkinson, Brignall, Ruškuc, Smith, and West [3], who gave some sufficient conditions for the classes of $\{321, \alpha\}$- and $\{321, \beta\}$-avoiding permutations to have the same exponential growth rate.

We believe that the techniques introduced in this work—especially the panel encoding of Section 5—will find many more applications. To introduce these we first observe that in the language of geometric grid classes [2, 8, 13], the 321-avoiding permutations form the grid class of the infinite matrix

$$\begin{pmatrix} & & & \iddots & \iddots \\ & & 1 & 1 & \\ & 1 & 1 & & \\ 1 & 1 & & & \end{pmatrix}.$$

This is equivalent to the observation, made at the end of Section 2, that the 321-avoiding permutations are precisely those that can be drawn on two parallel rays (see the first picture in Figure 14). While a great deal is known about geometric grid classes, the present work can be viewed as an initial attempt to extend that theory to infinite matrices (another initial attempt in this direction is [6]). One aspect of the infinite geometric grid class view of 321-avoiding permutations that seems particularly important is that the cells can be labelled so that cell i interacts only with cells $i-1$ and $i+1$, in the sense that the relative positions and values of any two entries in cells whose indices differ by more than one depend only on the indices of the cells, giving the class a "path-like" structure.

It would therefore be natural to attempt to extend the results established here to other infinite geometric grid classes possessing a similar structure. Two more examples are given by the second and third pictures shown in Figure 14.

The class corresponding to the second picture of Figure 14, which we call the *negative staircase*, demonstrates one reason why our techniques cannot be translated automatically to all path-like geometric grid classes. Indeed, while greedy staircase griddings are easy to describe for the 321-avoiding staircase, the issue is not so clear-cut for the negative staircase. To see this, consider the

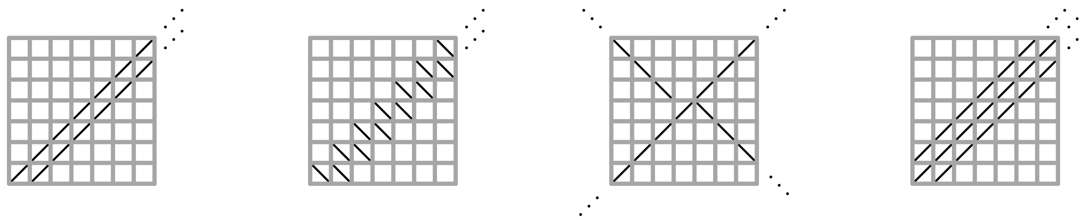
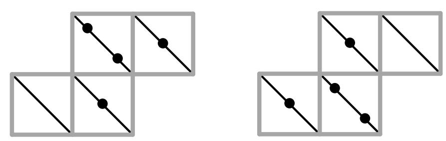


Figure 14: The 321-avoiding staircase, the negative staircase, an infinite spiral, and a thickened staircase.

permutations 4123 and 2341. Both of these permutations can be drawn on the negative staircase, as demonstrated below.



Moreover, up to shifting the choice of cells, the griddings shown above are the only negative staircase griddings of 4123 and 2341. The permutation 4123 shows that we cannot take the members of the first cell to consist of the maximum initial decreasing subsequence. On the other hand, 2341 shows that we cannot define greedy staircase griddings by the value either. Thus any definition of greedy negative staircase griddings would have to incorporate at least a slightly more global sense of the permutation to be gridded than was required for the 321-avoiding staircase.

In dealing with either the negative staircase class or the *infinite spiral class* (the third picture in Figure 14), one would also have to develop a replacement for the Dyck path encoding. However, we do not believe this step is, in and of itself, a major impediment, as the role of the Dyck path encoding is just a proxy for maintaining a set of requirements in finitely many states, and it seems clear that similar devices could be developed for other classes obtained from regular path-like structures.

Much more serious issues present themselves if we remove the path-like condition on the occupied cells; for instance, consider the class of permutations that can be drawn on the *thickened staircase* shown on the far right of Figure 14. This class is a proper subclass of the 4321-avoiding permutations and so to see that we cannot hope for a result like Theorem 1.1 in this context we need only note that this class contains the class of 321-avoiding permutations. On the language level, even if we could define the domino encoding in this setting, we could not impose the small ascent condition on the encodings of words describing members of this class, so their encodings would not lie in $\mathcal{L}^\infty$, and thus the panel encoding could not be applied.

Finally, an emerging topic of interest in the general study of permutation classes has been strong and broad rationality and algebraicity (see [4, 8]). While the presence of infinite antichains necessarily implies that a class has subclasses whose generating functions are not D-finite, we have shown that certain subclasses of the 321-avoiding permutations are nevertheless well-structured. To make this notion precise we say that a class is *broadly rational* if it and all of its finitely based subclasses have rational generating functions and/or *strongly rational* if this holds for *all* of its subclasses. Therefore Theorem 1.1 shows that all proper subclasses of the 321-avoiding permutations are broadly rational.

The same counting argument as above shows that every strongly rational class must be well-quasi-ordered. Thus Theorem 1.1 also implies the following.

Corollary 10.1. *A subclass of* Av(321) *is strongly rational if and only if it is well-quasi-ordered.*

This represents one more piece of evidence for the following conjecture (which is also supported by the results of [4]).

Conjecture 10.2. *A permutation class is strongly rational if and only if it is well-quasi-ordered and does not contain the class of* 312*-avoiding permutations or any symmetry of it.*

Acknowledgements. Significant inspiration for this research came from the work of Lozin [22], who proved that while the class of bipartite inversion graphs (the inversion graphs of 321-avoiding permutations) has unbounded clique-width, every proper subclass of this class has bounded clique-width. We are also grateful to Michael Engen and Jay Pantone for their numerous suggestions and corrections.

References

[1] Albert, M. H., and Atkinson, M. D. Simple permutations and pattern restricted permutations. *Discrete Math. 300*, 1-3 (2005), 1–15.

[2] Albert, M. H., Atkinson, M. D., Bouvel, M., Ruškuc, N., and Vatter, V. Geometric grid classes of permutations. *Trans. Amer. Math. Soc. 365*, 11 (2013), 5859–5881.

[3] Albert, M. H., Atkinson, M. D., Brignall, R., Ruškuc, N., Smith, R., and West, J. Growth rates for subclasses of Av(321). *Electron. J. Combin. 17* (2010), Paper 141, 16 pp.

[4] Albert, M. H., Atkinson, M. D., and Vatter, V. Subclasses of the separable permutations. *Bull. Lond. Math. Soc. 43* (2011), 859–870.

[5] Albert, M. H., and Bouvel, M. A general theory of Wilf-equivalence for Catalan structures. *Electron. J. Combin. 22*, 4 (2015), Paper 4.45, 29 pp.

[6] Albert, M. H., and Brignall, R. Enumerating indices of Schubert varieties defined by inclusions. *J. Combin. Theory Ser. A 123* (2014), 154–168.

[7] Albert, M. H., Lackner, M.-L., Lackner, M., and Vatter, V. The complexity of pattern matching for 321-avoiding and skew-merged permutations. *Discrete Math. Theor. Comput. Sci. 18*, 2 (2016), 17 pp.

[8] Albert, M. H., Ruškuc, N., and Vatter, V. Inflations of geometric grid classes of permutations. *Israel J. Math. 205*, 1 (2015), 73–108.

[9] Albert, M. H., and Vatter, V. Generating and enumerating 321-avoiding and skew-merged simple permutations. *Electron. J. Combin. 20*, 2 (2013), Paper 44, 11 pp.

[10] Ardila, F. The Catalan matroid. *J. Combin. Theory Ser. A 104*, 1 (2003), 49–62.

[11] Atkinson, M. D., Murphy, M. M., and Ruškuc, N. Partially well-ordered closed sets of permutations. *Order 19*, 2 (2002), 101–113.

[12] Bannister, M. J., Cheng, Z., Devanny, W. E., and Eppstein, D. Superpatterns and universal point sets. *J. Graph Algorithms Appl. 18*, 2 (2014), 177–209.

[13] Bevan, D. Growth rates of geometric grid classes of permutations. *Electron. J. Combin. 21*, 4 (2014), Paper 4.51, 17 pp.

[14] Bousquet-Mélou, M. Rational and algebraic series in combinatorial enumeration. In *International Congress of Mathematicians. Vol. III*, M. Sanz-Solé, J. Soria, J. L. Varona, and J. Verdera, Eds. Eur. Math. Soc., Zürich, Switzerland, 2006, pp. 789–826.

[15] Claesson, A., and Kitaev, S. Classification of bijections between 321- and 132-avoiding permutations. *Sém. Lothar. Combin. 60* (2008), Art. B60d, 30 pp.

[16] Flajolet, P., and Sedgewick, R. *Analytic Combinatorics.* Cambridge University Press, Cambridge, England, 2009.

[17] Guillemot, S., and Vialette, S. Pattern matching for 321-avoiding permutations. In *Algorithms and Computation, Proceedings of the Twentieth International Symposium (ISAAC)*, Y. Dong, D.-Z. Du, and O. Ibarra, Eds., vol. 5878 of *Lecture Notes in Comput. Sci.* Springer, Berlin, Germany, 2009, pp. 1064–1073.

[18] Haglund, J. *The q,t-Catalan numbers and the space of diagonal harmonics*, vol. 41 of *University Lecture Series.* American Mathematical Society, Providence, Rhode Island, 2008.

[19] Higman, G. Ordering by divisibility in abstract algebras. *Proc. London Math. Soc. (3) 2* (1952), 326–336.

[20] Knuth, D. E. *The Art of Computer Programming*, vol. 1. Addison-Wesley, Reading, Massachusetts, 1968.

[21] Krattenthaler, C. Permutations with restricted patterns and Dyck paths. *Adv. in Appl. Math. 27*, 2-3 (2001), 510–530.

[22] Lozin, V. Minimal classes of graphs of unbounded clique-width. *Ann. Comb. 15*, 4 (2011), 707–722.

[23] Lozin, V., and Mayhill, C. Canonical antichains of unit interval and bipartite permutation graphs. *Order 28*, 3 (2011), 513–522.

[24] Mansour, T., and Shattuck, M. Chebyshev polynomials and statistics on a new collection of words in the Catalan family. *J. Difference Equ. Appl. 20*, 11 (2014), 1568–1582.

[25] Mansour, T., and Vainshtein, A. Restricted 132-avoiding permutations. *Adv. in Appl. Math. 26*, 3 (2001), 258–269.

[26] Miller, A. Asymptotic bounds for permutations containing many different patterns. *J. Combin. Theory Ser. A 116*, 1 (2009), 92–108.

[27] Miner, S., and Pak, I. The shape of random pattern-avoiding permutations. *Adv. in Appl. Math. 55* (2014), 86–130.

[28] Sakarovitch, J. *Elements of Automata Theory.* Cambridge University Press, Cambridge, England, 2009.

[29] Speyer, D. A double grading of Catalan numbers. MathOverflow: `http://www.mathoverflow.net/questions/131809`, 2013.

[30] Stump, C. On a new collection of words in the Catalan family. *J. Integer Seq. 17*, 7 (2014), Article 14.7.1, 8 pp.

[31] Vatter, V. A family of words counted by the Catalan numbers. MathOverflow: `http://mathoverflow.net/q/131585`, 2013.

[32] Waton, S. *On Permutation Classes Defined by Token Passing Networks, Gridding Matrices and Pictures: Three Flavours of Involvement*. PhD thesis, University of St Andrews, 2007.