On the Effective and Automatic Enumeration of Polynomial Permutation Classes

Cheyne Homberger and Vincent Vatter*
Department of Mathematics
University of Florida
Gainesville, Florida USA

We describe an algorithm, implemented in Python, which can enumerate any permutation class with polynomial enumeration from a structural description of the class. In particular, this allows us to find formulas for the number of permutations of length n which can be obtained by a finite number of block sorting operations (e.g., reversals, block transpositions, cut-and-paste moves).

1. Introduction

The "Fibonacci Dichotomy" of Kaiser and Klazar [15] was one of the first general results on the enumeration of permutation classes. It states that if there are fewer permutations of length n in a permutation class than the nth Fibonacci number, for any n, then the enumeration of the class is given by a polynomial for sufficiently large n. Since the Fibonacci Dichotomy was established for permutation classes, Balogh, Bollobás, and Morris [6] showed that it extends to the (more general) context of ordered graphs, while other proofs of the Fibonacci Dichotomy for permutations have been given by Huczynska and Vatter [14] and Albert, Atkinson, and Brignall [3].

While much of the focus on this strand of research has shifted to the consideration of larger classes (see Bollobás [7] and Klazar [16] for surveys), we return to consider two open questions about polynomial classes.

- **Question 1.1.** Given a structural description of a polynomial permutation class, how can we enumerate it?
- **Question 1.2.** Which polynomials can occur as enumerations of polynomial classes?

We view a satisfactory answer to Question 1.1 as a prerequisite for the investigation of Question 1.2, and thus our focus in this paper is on enumerating polynomial classes. Our answer of Question 1.1 also has applications to the study of genome rearrangements, as discussed in Section 3.

*The second author was partially supported by the NSA Young Investigator Grant H98230-12-1-0207.
Date: August 23, 2013
AMS 2000 Subject Classification. Primary: 05A15; Secondary: 05A05

The permutation π of length n contains the permutation σ of length k (written $\sigma \leqslant \pi$) if π has a subsequence of length k which is order isomorphic to σ. For example, $\pi = 391867452$ (written in list, or one-line notation) contains $\sigma = 51342$, as can be seen by considering the subsequence 91672 $(= \pi(2)\pi(3)\pi(5)\pi(6)\pi(9))$. A *permutation class*, or simply *class*, is a downset in this subpermutation order; thus if $\mathcal{C}$ is a class, $\pi \in \mathcal{C}$, and $\sigma \leqslant \pi$, then $\sigma \in \mathcal{C}$.

While there are many ways to specify a class, two are particularly relevant to this problem. One is by the class' *basis*, the minimal permutations *not* in the class. Another is by some structural description of the class. We adopt a structural approach to the specification of classes, the details of which will be described briefly.

We should mention that there are several established approaches which could, *theoretically*, be used to enumerate polynomial classes, but these each has drawbacks.

- Polynomial classes are contained in "geometric grid classes", so they fall under the purview of the results of Albert, Atkinson, Bouvel, Ruškuc, and Vatter [2]. However, their proofs are nonconstructive. Indeed, our work can be viewed as illuminating some preliminary obstacles which an algorithmic approach to geometric grid classes would have to overcome.

- Polynomial classes contain only finitely many "simple permutations", so the methods of Albert and Atkinson [1] (or the refinements introduced by Brignall, Huczynska, and Vatter [8]) could be used to compute their generating functions. However, this method has yet to be implemented, and applying it would require us to first determine the basis of the class is question.

- Polynomial classes can be enumerated using the insertion encoding of Albert, Linton, and Ruškuc [4] (which is implemented in the Maple package INSENC described in Vatter [18]). However, this method also requires the basis of the class.

Before describing our approach we must first describe the structure of polynomial classes. An *interval* in the permutation π is a sequence of contiguous entries whose values form an interval of natural numbers. A *monotone interval* is then an interval in which the entries are monotone (increasing or decreasing). Given a permutation σ of length m and nonempty permutations $\alpha_1, \ldots, \alpha_m$, the *inflation* of σ by $\alpha_1, \ldots, \alpha_m$ is the permutation $\pi = \sigma[\alpha_1, \ldots, \alpha_m]$ obtained by replacing each entry $\sigma(i)$ by an interval that is order isomorphic to α_i. For example,

$$3142[1, 321, 1, 12] = 6\ 321\ 7\ 45.$$

Going against tradition, in this work we *allow inflations by the empty permutation*.

The polynomial classes can be, roughly, described as those for which the entries of every member of the class can be partitioned into a finite number of monotone intervals, which are related to each other in one of a finite number of ways. To describe this more concretely, let us say that a *peg permutation* is a permutation where each entry is decorated with a $+$, $-$, or $\bullet$, such as

$$\tilde{\rho} = 3^{\bullet}1^{-}4^{\bullet}2^{+}$$

As demonstrated above, we decorate peg permutations with tildes; in this context, ρ denotes for us the underlying (non-pegged) permutation, 3142 in this example. The *grid class* of the peg permutation $\tilde{\rho}$, $\mathrm{Grid}(\tilde{\rho})$ is the set of all permutations which may be obtained by inflating ρ by monotone

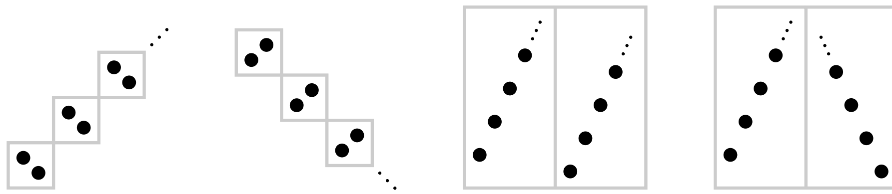

Figure 1: The two permutations shown on the left are the obstructions which prevent a class from being “monotone griddable”. The two permutations on the right (and all symmetries of them, eight in total) are the obstructions which prevent a monotone griddable class from being a polynomial class.

intervals of type determined by the signs of $\tilde{\rho}$: $\rho(i)$ may be inflated by an increasing (resp., decreasing) interval if $\tilde{\rho}(i)$ is decorated with a $+$ (resp., $-$) while it may only be inflated by a single entry or the empty permutation if $\tilde{\rho}(i)$ is dotted. Thus if $\pi \in \mathrm{Grid}(\tilde{\rho})$ then its entries can be partitioned into monotone intervals which are compatible with $\tilde{\rho}$; if this partition is denoted P then we refer to P as a $\tilde{\rho}$-partition of π and the pair (π, P) as a $\tilde{\rho}$-partitioned permutation.

Given a set $\tilde{G}$ of peg permutations, we denote the union of their corresponding grid classes by

$$\mathrm{Grid}(\tilde{G}) = \bigcup_{\tilde{\rho}\in\tilde{G}} \mathrm{Grid}(\tilde{\rho}).$$

As the next result shows, our goal is to enumerate such classes.

Theorem 1.3 (The combination of [14, Corollary 3.4] and [2, Theorem 10.3])**.** *For a permutation class $\mathcal{C}$ the following are equivalent:*

(1) $|\mathcal{C}_n|$ is given by a polynomial for all sufficiently large n,

(2) $|\mathcal{C}_n| < F_n$ for some n,

(3) $\mathcal{C}$ does not contain arbitrary long permutations of any of the forms described in Figure 1, and

(4) $\mathcal{C} = \mathrm{Grid}(\tilde{G})$ for a finite set $\tilde{G}$ of peg permutations.

2. THE ALGORITHM

We need a few prerequisites before our algorithm can be described. First we define a partial order on peg permutations. Given peg permutations $\tilde{\tau}$ and $\tilde{\rho}$ of lengths k and n, respectively, $\tilde{\tau} \leqslant \tilde{\rho}$ if there are indices $1 \leqslant i_1 < i_2 < \cdots < i_k \leqslant n$ such that $\rho(i_1)\rho(i_2)\cdots\rho(i_k)$ is order isomorphic to τ and for each j, $\tilde{\tau}(j)$ is decorated with a

$$\begin{cases} + \text{ or } \bullet & \text{if } \tilde{\rho}(i_j) \text{ is decorated with a } +, \\ - \text{ or } \bullet & \text{if } \tilde{\rho}(i_j) \text{ is decorated with a } -, \text{ or} \\ \quad\bullet & \text{if } \tilde{\rho}(i_j) \text{ is dotted.} \end{cases}$$

In other words, in order to obtain a smaller element in this *peg permutation order*, one can change signs to dots and delete entries. Note that $\mathrm{Grid}(\tilde{\tau}) \subseteq \mathrm{Grid}(\tilde{\rho})$ whenever $\tilde{\tau} \leqslant \tilde{\rho}$, but the reverse implication is not true; for example, $\mathrm{Grid}(1^\bullet 2^\bullet) \subseteq \mathrm{Grid}(1^+)$, but $1^\bullet 2^\bullet \nleqslant 1^+$.

In addition, we extend the notion of intervals to peg permutations in the trivial way, by ignoring decoration; thus the intervals of $\tilde{\rho}$ are the same as the intervals of ρ, although they carry their decoration from $\tilde{\rho}$. We must also say something about monotone intervals of (unpegged) permutations:

Proposition 2.1. *If two monotone intervals of a permutation intersect, then their union is also a monotone interval.*

Proof. Suppose that two monotone intervals intersect. By symmetry, we may assume that one of them is increasing. By considering the various cases, it is clear that the other monotone interval must also be increasing, and that their union is also an increasing interval. ☐

We say that the $\tilde{\rho}$-partitioned permutation (π, P) *fills* the peg permutation $\tilde{\rho}$ if each part of P corresponding to a signed entry in $\tilde{\rho}$ contains at least two entries, and each part of P corresponding to a dotted entry contains precisely one entry. Clearly every peg permutation has a unique minimum filling permutation, in which each part corresponding to a signed entry contains precisely two entries, and each part corresponding to a dotted entry contains precisely one entry. We further say that the set $\tilde{G}$ of peg permutations is *complete* if every $\pi \in \mathrm{Grid}(\tilde{G})$ fills some $\tilde{\rho} \in \tilde{G}$. It is not difficult to construct complete sets of peg permutations, as we observe below without proof.

Proposition 2.2. *Every downset in the peg permutation order is complete.*

Given a set of peg permutations, the first step of our algorithm is to complete it, by computing its downward closure.

Proposition 2.1 shows every permutation π has a unique coarsest partition into monotone intervals. In other words, for each π there is a unique peg permutation $\tilde{\rho}$ such that π is $\tilde{\rho}$-griddable, but not $\tilde{\tau}$-griddable for any $\tilde{\tau} < \tilde{\rho}$. In particular, this implies that, for this $\tilde{\rho}$, $\mathrm{Grid}(\tilde{\rho}) \neq \mathrm{Grid}(\tilde{\tau})$ for all $\tilde{\tau} < \tilde{\rho}$. We call a peg permutation $\tilde{\rho}$ with this property *compact*, i.e., $\tilde{\rho}$ is compact if $\mathrm{Grid}(\tilde{\tau}) \neq \mathrm{Grid}(\tilde{\rho})$ for all $\tilde{\tau} < \tilde{\rho}$. For example, $2^{\bullet}1^{-}$ is not compact because $1^{-} < 2^{-}1^{\bullet}$ and $\mathrm{Grid}(2^{-}1^{\bullet}) = \mathrm{Grid}(1^{-})$.

Our next result ties the definitions of compactness and filling together.

Proposition 2.3. *For a peg permutation $\tilde{\rho}$, the following conditions are equivalent:*

(1) $\tilde{\rho}$ is compact,

(2) $\tilde{\rho}$ does not have an interval order isomorphic to $1^{+}2^{+}$, $1^{+}2^{\bullet}$, $1^{\bullet}2^{+}$, or symmetrically, to $2^{-}1^{-}$, $2^{-}1^{\bullet}$, $2^{\bullet}1^{-}$, and

(3) every permutation which fills $\tilde{\rho}$ has a unique $\tilde{\rho}$-partition.

Proof. First we show that (1) and (2) are equivalent. It is clear that (1) implies (2), so it remains only to show that (2) implies (1). Suppose to the contrary that $\tilde{\rho}$ is not compact, so $\mathrm{Grid}(\tilde{\rho}) = \mathrm{Grid}(\tilde{\tau})$ for some $\tilde{\tau} < \tilde{\rho}$. Let π be any permutation that fills $\tilde{\rho}$ and suppose that P is its $\tilde{\rho}$-partition and P' is its $\tilde{\tau}$-partition. Then, because $\tilde{\tau} < \tilde{\rho}$ and π fills $\tilde{\rho}$, there must be a part of P' which intersects two parts of P. Thus these two parts of P together form a monotone interval of π by Proposition 2.1, and this implies that $\tilde{\rho}$ must contain one of the intervals listed in (2).

Now we show that (3) is equivalent to (2). If (2) fails to hold (so $\tilde{\rho}$ contains one of the specified intervals) then it is clear that some permutations which fill $\tilde{\rho}$ have nonunique $\tilde{\rho}$-partitions, so (3) implies (2). Suppose then that the permutation π fills $\tilde{\rho}$ but has two different $\tilde{\rho}$-partitions, P and P'. As in the proof that (1) is equivalent to (2), there must be two parts of one of these two partitions which together form a monotone interval, but this again implies that $\tilde{\rho}$ contains one of the intervals listed in (2), completing the proof. ☐

We say that the set $\tilde{G}$ of peg permutations is *compact* if every peg permutation it contains is compact. Note that we do not lose any permutations if we remove the non-compact peg permutations from a complete set $\tilde{G}$ to obtain a set $\tilde{G}' \subseteq \tilde{G}$. For $\tilde{\rho} \in \tilde{G}'$, if $\pi \in \mathrm{Grid}(\tilde{\rho})$ then π has a unique $\tilde{\rho}$-partition, but this does not necessarily imply that π doesn't fill some other $\tilde{\tau} \in \tilde{G}'$. For example, 2341 fills both $2^\bullet 3^\bullet 4^\bullet 1^\bullet$ and $2^+1^\bullet$. To address this problem, we say that a compact peg permutation $\tilde{\rho}$ is *clean* if $\mathrm{Grid}(\tilde{\rho}) \nsubseteq \mathrm{Grid}(\tilde{\tau})$ for any *shorter* peg permutation $\tilde{\tau}$. We say that the set $\tilde{G}$ of peg permutations is *clean* if each of them is clean.

Proposition 2.4. *The compact peg permutation $\tilde{\rho}$ is clean if and only if it does not have an interval order isomorphic to $1^\bullet 2^\bullet$ or $2^\bullet 1^\bullet$.*

Proof. If $\tilde{\rho}$ contains an interval order isomorphic to $1^\bullet 2^\bullet$ or $2^\bullet 1^\bullet$ then let $\tilde{\tau}$ denote the peg permutation obtained by contracting this interval to a single entry decorated with the appropriate sign; clearly $\mathrm{Grid}(\tilde{\rho}) \subseteq \mathrm{Grid}(\tilde{\tau})$.

Otherwise suppose that $\mathrm{Grid}(\tilde{\rho}) \subseteq \mathrm{Grid}(\tilde{\tau})$ where $\tilde{\tau}$ is shorter than $\tilde{\rho}$ and let π be any permutation which fills $\tilde{\rho}$. In any $\tilde{\tau}$-partition of π there must be a monotone interval formed from entries in different parts of any $\tilde{\rho}$-partition of π. Because $\tilde{\rho}$ is compact, we see from our previous proposition that $\tilde{\rho}$ must have contain a $1^\bullet 2^\bullet$ or $2^\bullet 1^\bullet$ interval, as desired. ☐

Given a complete and compact set $\tilde{G}$, it is not in general possible to obtain a clean set $\tilde{G}'$ such that $\mathrm{Grid}(\tilde{G}') = \mathrm{Grid}(\tilde{G})$: to return to our previous example, if $2^\bullet 3^\bullet 4^\bullet 1^\bullet \in \tilde{G}$ but $2^+1^\bullet \notin \tilde{G}$ then we cannot simply remove $2^\bullet 3^\bullet 4^\bullet 1^\bullet$ from $\tilde{G}$ as we would lose permutations in doing so (and we cannot counteract this by adding $2^+1^\bullet$ to $\tilde{G}$ as then we would gain permutations). We address this problem shortly.

Now suppose that we are given a set $\tilde{G}$ of peg permutations and wish to enumerate $\mathrm{Grid}(\tilde{G})$. By Proposition 2.2, we may assume that $\tilde{G}$ is complete (in the algorithm, this amounts to a preprocessing step). Then, by Proposition 2.3 (2) we may assume that $\tilde{G}$ is compact (another preprocessing step). We may describe the elements of $\mathrm{Grid}(\tilde{G})$ by specifying a peg permutation together with a vector indicating the lengths of the monotone intervals (though $\mathrm{Grid}(\tilde{G})$ may not be in bijection with the set of such pairs because $\tilde{G}$ need not be clean).

To describe this precisely, we need notation for integer vectors. For us, $\mathbb{P}^m$ denotes the set of all m-tuples of positive integers, and given $\vec{v} \in \mathbb{P}^m$, the sum of its entries is denoted $\|\vec{v}\| = \sum v(i)$. Given two vectors $\vec{v}, \vec{w} \in \mathbb{P}^m$, we write $\vec{v} \leqslant \vec{w}$ and say that $\vec{v}$ is contained in $\vec{w}$ if $\vec{v}(i) \leqslant \vec{w}(i)$ for all i. If $\vec{v} \nleqslant \vec{w}$, we say that $\vec{w}$ *avoids* $\vec{v}$. It is worth noting that $\mathbb{P}^m$ forms a lattice under this partial order with join given by pairwise maximum,

$$\vec{v} \vee \vec{w} = (\max\{v(1), w(1)\}, \ldots, \max\{v(m), w(m)\}).$$

Downsets of vectors under this order are the vector analogues of permutation classes, and thus we call them *vector classes*. As with permutation classes, every vector class has a unique *basis* B

consisting of the minimal vectors not in the class. Unlike permutation classes, however, bases of vector classes are necessarily finite (by Higman's Theorem [13]).

Now let $\tilde{\rho}$ be a compact peg permutation of length m and $\vec{v} \in \mathbb{P}^m$. We define the *inflation* of $\tilde{\rho}$ by $\vec{v}$, denoted $\tilde{\rho}[\vec{v}]$, to be the permutation in $\mathrm{Grid}(\tilde{\rho})$ obtained by inflating $\tilde{\rho}(i)$ by a monotone interval of length $v(i)$, with the direction of the run determined by the decoration of $\tilde{\rho}(i)$. For example, $2^+1^+[(4,2)] = 3456\ 12$. If $\vec{v} \leqslant \vec{w}$ then it is clear that $\tilde{\rho}[\vec{v}] \leqslant \tilde{\rho}[\vec{w}]$; by Proposition 2.3 (3), it follows that if $\tilde{\rho}$ is compact and $\tilde{\rho}[\vec{v}]$ fills $\tilde{\rho}$ then the reverse is true: $\tilde{\rho}[\vec{v}] \leqslant \tilde{\rho}[\vec{w}]$ implies that $\vec{v} \leqslant \vec{w}$.

Suppose that $\mathcal{V}$ and $\mathcal{W}$ are vector classes with bases $B_{\mathcal{V}}$ and $B_{\mathcal{W}}$, respectively. It is obvious that the basis of $\mathcal{V} \cap \mathcal{W}$ is the set of minimal elements in $B_{\mathcal{V}} \cup B_{\mathcal{W}}$. Computing bases for unions is slightly less transparent. If $B_{\mathcal{V}}$ and $B_{\mathcal{W}}$ are both singletons, consisting of $\vec{v}$ and $\vec{w}$, respectively, say, then the basis of $\mathcal{V} \cup \mathcal{W}$ is $\vec{v} \vee \vec{w}$. Therefore we see that for general bases,

$$\begin{aligned}
\mathcal{V} \cup \mathcal{W} &= \left(\bigcap_{\vec{v} \in B_{\mathcal{V}}} \{\vec{v}\text{-avoiding vectors}\} \right) \bigcup \left(\bigcap_{\vec{w} \in B_{\mathcal{W}}} \{\vec{w}\text{-avoiding vectors}\} \right), \\
&= \bigcap_{\substack{\vec{v} \in B_{\mathcal{V}}, \\ \vec{w} \in B_{\mathcal{W}}}} \{\vec{v}\text{-avoiding vectors}\} \cup \{\vec{w}\text{-avoiding vectors}\}, \\
&= \bigcap_{\substack{\vec{v} \in B_{\mathcal{V}}, \\ \vec{w} \in B_{\mathcal{W}}}} \{\vec{v} \vee \vec{w}\text{-avoiding vectors}\},
\end{aligned}$$

so the basis for $\mathcal{V} \cup \mathcal{W}$ consists of the minimal elements in the set

$$\{\vec{v} \vee \vec{w} \ : \ \vec{v} \in B_{\mathcal{V}} \text{ and } \vec{w} \in B_{\mathcal{W}}\}.$$

The class of vectors $\vec{v}$ such that $\tilde{\rho}[\vec{v}] \in \mathrm{Grid}(\tilde{\rho})$ is therefore

$$\mathcal{M}_{\tilde{\rho}} = \{\vec{v} \ : \ v(i) \leqslant 1 \text{ if } \tilde{\rho}(i) \text{ is dotted}\}.$$

Given a peg permutation $\tilde{\rho}$ and any vector class $\mathcal{V} \subseteq \mathcal{M}_{\tilde{\rho}}$ we now define a *cross-section*, denoted by $\mathrm{Grid}(\tilde{\rho}, \mathcal{V})$, to be the set of permutations of the form $\tilde{\rho}[\vec{v}]$ for $\vec{v} \in \mathcal{V}$ which fill $\tilde{\rho}$; these are the permutations of the form $\tilde{\rho}[\vec{v}]$ for vectors $\vec{v} \in \mathcal{V}$ which contain the vector $\vec{m}_{\tilde{\rho}}$ defined by

$$\vec{m}_{\tilde{\rho}}(i) = \begin{cases} 2 & \text{if } \tilde{\rho}(i) \text{ is signed, or,} \\ 1 & \text{if } \tilde{\rho}(i) \text{ is dotted.} \end{cases}$$

We are now in position to "clean" $\tilde{G}$. Proposition 2.4 shows that cleaning $\tilde{\rho}$ amounts to contracting monotone intervals of dotted entries. Thus for every peg permutation $\tilde{\rho}$ there is a clean peg permutation $\tilde{\tau}$ and a downset $\mathcal{V}$ such that the peg permutations which fill $\tilde{\rho}$ are precisely the cross-section $\mathrm{Grid}(\tilde{\tau}, \mathcal{V})$. Note that two different peg permutations in $\tilde{G}$ might clean to the same peg permutation, and thus in this step we may need take unions of the associated vector classes.

Therefore, given a complete and compact set $\tilde{G}$ of peg permutations, we can compute a complete, compact, and clean set $\tilde{G}'$ and associate to each $\tilde{\rho} \in \tilde{G}'$ a vector class $\mathcal{V}_{\tilde{\rho}}$ such that $\mathrm{Grid}(\tilde{G})$ is in bijection with the disjoint union of cross-sections

$$\bigcup_{\tilde{\rho} \in \tilde{G}'} \mathrm{Grid}(\tilde{\rho}, \mathcal{V}_{\tilde{\rho}}).$$

After all this preprocessing, the enumeration problem is essentially trivial. Consider a cross-section $\mathrm{Grid}(\tilde{\rho}, \mathcal{V})$ where the basis of $\mathcal{V}$ is $B_{\mathcal{V}}$ and let $s(\tilde{\rho})$ denote the number of signed entries of $\tilde{\rho}$. The generating function for $\mathrm{Grid}(\tilde{\rho}, \mathcal{V})$ is given by inclusion-exclusion (based on how many basis elements a given vector contains):

$$\sum_{B \subseteq B_{\mathcal{V}}} (-1)^{|B|} \frac{x^{\|\vec{m}_{\tilde{\rho}} \vee (\bigvee B)\|}}{(1-x)^{s(\tilde{\rho})}},$$

where $\bigvee B$ denotes the join of all vectors in B (and $\bigvee \varnothing$ is the all-1 vector).

3. GENOME REARRANGEMENT

The genes in a chromosonal genome may be thought of a discrete blocks of DNA, and thus labeled from 1 to n along the genome. During the process of evolution, the genes in a genome of one species might be rearranged via one or several operations and then appear in a different order (given by a permutation π) in the genome of a different species. By studying the number of operations required to transform the identity permutation into π we may therefore get an estimate of how many mutations occured in the evolution of the second species from the first. There are several different operations of interest, which we briefly survey in what follows. Of particular interest to the results of this paper is that in all of these operations, the class of permutations which can be obtained in at most k operations from the identity is a polynomial permutation class, and its structural description, as $\mathrm{Grid}(\tilde{G})$ for a set $\tilde{G}$ of peg permutations, is routine to compute. Thus using the Python package which implements the approach described in the previous section, we are able to automatically compute the polynomials enumerating these classes. The majority of these enumerations were not previously in the OEIS [17]— the new sequences are those numbered A228392–A228401.

All of the operations we survey are based on the notion of a *block*, which is a contiguous sequence of entries. The block transposition operation was introduced by Bafner and Pevzner [5]. In a single *block transposition* one is allowed to interchange two adjacent blocks of a permutation. Thus we may change

$$\pi(1)\cdots\pi(i-1)\,\boxed{\pi(i)\cdots\pi(j-1)}\,\boxed{\pi(j)\cdots\pi(k-1)}\,\pi(k)\cdots\pi(n)$$

into

$$\pi(1)\cdots\pi(i-1)\,\boxed{\pi(j)\cdots\pi(k-1)}\,\boxed{\pi(i)\cdots\pi(j-1)}\,\pi(k)\cdots\pi(n).$$

In the language of grid classes, the set of permutations which can be generated by a single block transposition from the identity is $\mathrm{Grid}(1^+3^+2^+4^+)$. Below we include the data for permutations which can be generated from the identity with 3 or fewer block transpositions. Note that the polynomials given are only valid for sufficiently large n.

<table>
<tr><th>$k \backslash n$</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>OEIS [17] reference</th></tr>
<tr><td>1</td><td>1</td><td>2</td><td>5</td><td>11</td><td>21</td><td>36</td><td>57</td><td>85</td><td>121</td><td>166</td><td>A000292</td></tr>
<tr><td></td><td colspan="10">$\binom{n}{0}+\binom{n}{2}+\binom{n}{3}$</td><td></td></tr>
<tr><td>2</td><td>1</td><td>2</td><td>6</td><td>23</td><td>89</td><td>295</td><td>827</td><td>2017</td><td>4405</td><td>8812</td><td>A228392</td></tr>
<tr><td></td><td colspan="10">$\binom{n}{0}+\binom{n}{2}+2\binom{n}{3}+8\binom{n}{4}+18\binom{n}{5}+11\binom{n}{6}$</td><td></td></tr>
<tr><td>3</td><td>1</td><td>2</td><td>6</td><td>24</td><td>120</td><td>675</td><td>3527</td><td>15484</td><td>56917</td><td>179719</td><td>A228393</td></tr>
<tr><td></td><td colspan="10">$\binom{n}{0}+\binom{n}{2}+2\binom{n}{3}+9\binom{n}{4}+44\binom{n}{5}+220\binom{n}{6}+656\binom{n}{7}+841\binom{n}{8}+369\binom{n}{9}$</td><td></td></tr>
</table>

A *prefix block transposition* is a special case of a block transposition in which the blocks must be at the beginning of the permutation. This method of rearrangement was first studied by Dias and Meidanis [11]. The data for permutations which can be generated from the identity by 3 or fewer prefix block transpositions is below; again, the polynomials are only valid for sufficiently large n.

$k \backslash n$	1	2	3	4	5	6	7	8	9	10	OEIS [17] reference
1	1	2	4	7	11	16	22	29	37	46	A000124
						$\binom{n}{0}+\binom{n}{2}$					
2	1	2	6	21	61	146	302	561	961	1546	A228394
						$\binom{n}{0}+\binom{n}{2}+2\binom{n}{3}+6\binom{n}{4}$					
3	1	2	6	24	116	521	1877	5531	13939	31156	A228395
						$\binom{n}{0}+\binom{n}{2}+2\binom{n}{3}+9\binom{n}{4}+40\binom{n}{5}+90\binom{n}{6}$					

A *reversal* reverses one block in a permutation, thus transforming

$$\pi(1)\cdots\pi(i-1)\boxed{\pi(i)\cdots\pi(j-1)}\,\pi(j)\cdots\pi(n)$$

into

$$\pi(1)\cdots\pi(i-1)\boxed{\pi(j-1)\cdots\pi(i)}\,\pi(j)\cdots\pi(n).$$

Hence the class of permutations which can be sorted by a single reversal is $\mathrm{Grid}(1^{+}2^{-}3^{+})$. This method of rearrangement was first introduced by Watterson, Ewens, Hall, and Morgan [19]. Below is our data for this operation.

$k \backslash n$	1	2	3	4	5	6	7	8	9	10	OEIS [17] reference
1	1	2	4	7	11	16	22	29	37	46	A000124
						$\binom{n}{0}+\binom{n}{2}$					
2	1	2	6	22	63	145	288	516	857	1343	A228396
						$8\binom{n}{0}-3\binom{n}{1}+\binom{n}{2}+4\binom{n}{3}$					
3	1	2	6	24	118	534	1851	5158	12264	25943	A228397
						$318\binom{n}{0}-214\binom{n}{1}+131\binom{n}{2}-61\binom{n}{3}+20\binom{n}{4}+70\binom{n}{5}+35\binom{n}{6}$					

By restricting reversals to initial segments of a permutation we obtain the *prefix reversal* operation, which was introduced under the name *pancake sorting* by "Harry Dweighter" (actually, Jacob E. Goodman) as a *Monthly* problem [12].

$k \backslash n$	1	2	3	4	5	6	7	8	9	10	OEIS [17] reference
1	1	2	3	4	5	6	7	8	9	10	A000027
						$\binom{n}{1}$					
2	1	2	5	10	17	26	37	50	65	82	A002522
						$2\binom{n}{0}-1\binom{n}{1}+2\binom{n}{2}$					
3	1	2	6	21	52	105	186	301	456	657	A228398
						$-3\binom{n}{0}+3\binom{n}{1}-2\binom{n}{2}+6\binom{n}{3}$					

The cut-and-paste operation is a generalization of both the reversal operation and the block transposition operation. A single *cut-and-paste* move consists of moving a single block of the permutation anywhere else in the permutation, with the option of reversing this block at the same time.

Cut-and-paste sorting was introduced by Cranston, Sudborough, and West [10].

$k \backslash n$	1	2	3	4	5	6	7	8	9	10	OEIS [17] reference
1	1	2	6	16	35	66	112	176	261	370	A060354
	$\binom{n}{1} + 3\binom{n}{3}$										
2	1	2	6	24	120	577	2208	6768	17469	39603	A228399
	$-18\binom{n}{0} + 45\binom{n}{1} - 61\binom{n}{2} + 70\binom{n}{3} - 53\binom{n}{4} + 88\binom{n}{5} + 107\binom{n}{6}$										
3	1	2	6	24	120	720	5040	36757	223898	1055479	A228400
	$508264\binom{n}{0} - 280036\binom{n}{1} + 140012\binom{n}{2} - 57622\binom{n}{3} + 13839\binom{n}{4}$ $+4136\binom{n}{5} - 5368\binom{n}{6} + 531\binom{n}{7} + 21125\binom{n}{8} + 12615\binom{n}{9}$										

Finally, the *block interchange* operation is similar to the block transposition operation except that in this operation we are allowed to interchange any two blocks. This was first studied by Christie [9]. The size of the peg permutations involved in this case grows so fast that we were only able to compute the first two enumerations.

$k \backslash n$	1	2	3	4	5	6	7	8	9	10	OEIS [17] reference
1	1	2	6	16	36	71	127	211	331	496	A145126
	$\binom{n}{0} + \binom{n}{2} + 2\binom{n}{3} + \binom{n}{4}$										
2	1	2	6	24	120	540	1996	6196	16732	40459	A228401
	$\binom{n}{0} + \binom{n}{2} + 2\binom{n}{3} + 9\binom{n}{4} + 44\binom{n}{5} + 85\binom{n}{6} + 70\binom{n}{7} + 21\binom{n}{8}$										

The Python code used to perform these computations is available at

`https://github.com/cheyneh/polypermclass`.

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.*, in press.

[3] ALBERT, M. H., ATKINSON, M. D., AND BRIGNALL, R. Permutation classes of polynomial growth. *Ann. Comb. 11*, 3–4 (2007), 249–264.

[4] ALBERT, M. H., LINTON, S., AND RUŠKUC, N. The insertion encoding of permutations. *Electron. J. Combin. 12*, 1 (2005), Research paper 47, 31 pp.

[5] BAFNA, V., AND PEVZNER, P. A. Sorting by transpositions. *SIAM J. Discrete Math. 11*, 2 (1998), 224–240.

[6] BALOGH, J., BOLLOBÁS, B., AND MORRIS, R. Hereditary properties of ordered graphs. In *Topics in discrete mathematics*, M. Klazar, J. Kratochvíl, M. Loebl, J. Matoušek, R. Thomas, and P. Valtr, Eds., vol. 26 of *Algorithms Combin.* Springer, Berlin, 2006, pp. 179–213.

[7] BOLLOBÁS, B. Hereditary and monotone properties of combinatorial structures. In *Surveys in Combinatorics 2007*, A. Hilton and J. Talbot, Eds., no. 346 in London Mathematical Society Lecture Note Series. Cambridge University Press, 2007, pp. 1–39.

[8] BRIGNALL, R., HUCZYNSKA, S., AND VATTER, V. Simple permutations and algebraic generating functions. *J. Combin. Theory Ser. A 115*, 3 (2008), 423–441.

[9] CHRISTIE, D. A. Sorting permutations by block-interchanges. *Inform. Process. Lett. 60*, 4 (1996), 165–169.

[10] CRANSTON, D. W., SUDBOROUGH, I. H., AND WEST, D. B. Short proofs for cut-and-paste sorting of permutations. *Discrete Math. 307*, 22 (2007), 2866–2870.

[11] DIAS, Z., AND MEIDANIS, J. Sorting by prefix transpositions. In *Proceedings of the 9th International Symposium on String Processing and Information Retrieval* (London, UK, UK, 2002), SPIRE 2002, Springer-Verlag, pp. 65–76.

[12] DWEIGHTER, H. Elementary problems and solutions, problem E2569. *Amer. Math. Monthly 82* (1975), 1010.

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

[14] HUCZYNSKA, S., AND VATTER, V. Grid classes and the Fibonacci dichotomy for restricted permutations. *Electron. J. Combin. 13* (2006), R54, 14 pp.

[15] KAISER, T., AND KLAZAR, M. On growth rates of closed permutation classes. *Electron. J. Combin. 9*, 2 (2003), Research paper 10, 20 pp.

[16] KLAZAR, M. Overview of some general results in combinatorial enumeration. In *Permutation Patterns* (2010), S. Linton, N. Ruškuc, and V. Vatter, Eds., vol. 376 of *London Mathematical Society Lecture Note Series*, Cambridge University Press, pp. 3–40.

[17] THE ON-LINE ENCYCLOPEDIA OF INTEGER SEQUENCES. Published electronically at http://oeis.org/.

[18] VATTER, V. Finding regular insertion encodings for permutation classes. *J. Symb. Comput. 47* (2012), 259–265.

[19] WATTERSON, G. A., EWENS, W. J., HALL, T. E., AND MORGAN, A. The chromosome inversion problem. *Journal of Theoretical Biology*, 99 (1982), 1–7.