FINDING REGULAR INSERTION ENCODINGS FOR PERMUTATION CLASSES

VINCENT VATTER
Department of Mathematics
University of Florida
Gainesville, Florida USA

We describe a practical algorithm which computes the accepting automaton for the insertion encoding of a permutation class, whenever this insertion encoding forms a regular language. This algorithm is implemented in the accompanying Maple package INSENC, which can automatically compute the rational generating functions for such classes.

1. INTRODUCTION

Permutation classes, or restricted permutations, have received considerable attention over the past two decades, and during this time a great variety of techniques have been used to enumerate them. One of the most popular approaches, pioneered by Chung, Graham, Hoggatt, and Kleiman [4], employs generating trees. The permutation classes with finitely labeled generating trees were characterized in Vatter [15]. A more powerful technique based on formal languages and called the insertion encoding was later introduced by Albert, Linton, and Ruškuc [2]. While they characterized the classes that possess regular insertion encodings, naively employing their techniques requires the determinization of non-deterministic automata several times, and no implementation has been available. We study regular insertion encodings from a new point of view, essentially focusing on accepting automata instead of languages. This leads both to an implementation (the Maple package INSENC, available for download from the author's homepage) and to a new proof of the characterization of permutation classes with regular insertion encodings.

We begin with definitions. Two sequences of natural numbers are said to be *order isomorphic* if they have the same pairwise comparisons, so $9, 1, 6, 7, 2$ is order isomorphic to $5, 1, 3, 4, 2$. Every sequence w of natural numbers without repetition is order isomorphic

Date: August 18, 2011
Key words and phrases. insertion encoding, permutation class, rational generating function, regular language
AMS 2000 Subject Classification. 05A05, 05A15, 68Q20

to a unique permutation that we denote by $\mathrm{st}(w)$, so $\mathrm{st}(9,1,6,7,2) = 5,1,3,4,2$, which we shorten to 51342. We call $\mathrm{st}(w)$ the *standardization* of w. We further say that the permutation π *contains* the permutation β if π contains a subsequence that is order isomorphic to β, and in this case we write $\beta \leqslant \pi$. For example, 391867452 contains 51342, as can be seen by considering the subsequence 91672. If π does not contain β, then π is said to *avoid* β.

A *permutation class* is a lower order ideal in the containment ordering, meaning that if π is contained in a permutation in the class, then π itself lies in the class. Permutation classes can be specified in numerous ways, but we focus solely on the most common method, in which the minimal permutations *not* in the class are given (this set is called the *basis*). By the minimality condition, bases are necessarily *antichains*, meaning that no element of a basis is contained in another. Although there are infinite antichains of permutations (see Atkinson, Murphy, and Ruškuc [3] for constructions and references to earlier work), we restrict our attention to finitely based classes. Given a set of permutations B, we define $\mathrm{Av}(B)$ to be the set of permutations that avoid every permutation in B. Thus if $\mathcal{C}$ is a closed class with basis B then $\mathcal{C} = \mathrm{Av}(B)$, and for this reason the elements of a permutation class are often referred to as *restricted permutations*. We let $\mathcal{C}_n$ denote the set of permutations of length n in $\mathcal{C}$ and refer to $\sum |\mathcal{C}_n| x^n$ as the generating function of $\mathcal{C}$. All generating functions herein include the empty permutation of length 0.

2. FINITELY LABELED GENERATING TREES

In the generating tree approach to enumerating $\mathrm{Av}(B)$, the first step is to construct the *pattern-avoidance tree* $T(B)$ in which the children of the permutation $\pi \in \mathrm{Av}_{n-1}(B)$ are all permutations in $\mathrm{Av}_n(B)$ which can be formed by inserting n into π. The *active sites* of the permutation $\pi \in \mathrm{Av}_{n-1}(B)$ are defined as the indices i for which inserting n immediately before $\pi(i)$ produces a B-avoiding permutation (we also allow for n to be an active site if inserting n at the end of π produces a B-avoiding permutation). Thus every permutation has as many children in the pattern-avoidance tree as it has active sites. A *principal subtree* in $T(B)$ is a subtree consisting of a single permutation and all of its descendants.

If it happens that every principal subtree in the pattern-avoidance tree $T(B)$ belongs to one of a finite number of isomorphism classes, then the (rational) generating functions for $\mathrm{Av}(B)$ can be easily computed using the transfer matrix method (see Flajolet and Sedgewick [6, Section V.6]). Following [15], we say that the entry $\pi(i)$ of π is *generating-tree-reducible* (relative to B), or simply *GT-reducible*, if the principal subtree rooted at π and the principal subtree rooted at $\mathrm{st}(\pi - \pi(i))$ are isomorphic as rooted trees. (Here $\pi - \pi(i)$ denotes the sequence obtained by removing the entry $\pi(i)$ from π.) We can now state the main theorem of Vatter [15]:

Theorem 1 (Vatter [15])**.** *For a finite set B of permutations, the following are equivalent:*

(1) B contains both a child of an increasing permutation and a child of a decreasing permutation,

(2) there is an integer k such that no node of $T(B)$ has more than k children,

(3) every sufficiently long permutation in $\mathrm{Av}(B)$ *contains a GT-reducible entry,*

(4) $T(B)$ *has only finitely many isomorphism classes of principal subtrees (in other words,* $\mathrm{Av}(B)$ *has a* finitely labeled generating tree*).*

The implications (3) $\implies$ (4) and (4) $\implies$ (2) are trivial, while (1) $\iff$ (2) follows routinely from the Erdős-Szekeres Theorem [5]; the main content of the theorem is that (2) $\implies$ (3).

3. THE INSERTION ENCODING

Before describing the insertion encoding, we briefly review regular languages and finite automata. The classic results mentioned here are covered more comprehensively in many texts, for example, Flajolet and Sedgewick [6, Appendix A.7], so we give only the barest details.

A *deterministic finite automaton (DFA)* M over the alphabet Σ consists of a set S of *states*, one of which, s_0, is designated the *initial state*, a *transition function* $\delta : S \times \Sigma \to S$, and a subset $A \subseteq S$ designated as *accept states*. We denote this by $M = (S, \Sigma, s_0, A, \delta)$. It is useful to extend the definition of the transition function δ to a map $\delta : S \times \Sigma^* \to S$ in the obvious way. We say that the state t is *reachable* from the state s if there is a word $w \in \Sigma^*$ such that $\delta(s, w) = t$; otherwise, t is *unreachable* from s. The automaton M *accepts* the word $w \in \Sigma^*$ if $\delta(s_0, w)$ is an accept state. The set of all such words is the *language accepted* by the automaton, $\mathcal{L}(M)$.

A language that is accepted by a finite automaton (deterministic or not) is called *recognizable*. By Kleene's Theorem, the recognizable languages are precisely the *regular languages*, and for our purposes the reader may simply take this as the definition of regular languages. Regular languages have numerous pleasing properties, but we only need that they have rational generating functions which may be readily computed from their accepting automata.

Central to the insertion encoding is the notion of a configuration. Consider, for example, the permutation $\pi = 423615$. In the generating tree viewpoint, π is a descendant of 4231. In the insertion encoding viewpoint, we note that π is obtained from 4231 by inserting entries between the 3 and the 1 and after the 1. Thus we say that π evolves from the configuration $423\diamond1\diamond$. Formally, a *configuration* is a permutation together with zero or more $\diamond$ entries called *slots*, which may not be adjacent and must eventually be filled. Permutations correspond to the *slotless* configurations.

Given a configuration, there are four different ways to insert a new maximum entry m into the ith slot (we number slots from left to right) of a configuration. We may insert this maximum entry into the middle of the slot (replacing the $\diamond$ with $\diamond m\diamond$), to the left of the slot (replacing the $\diamond$ with $m\diamond$), to the right of the slot (replacing the $\diamond$ with $\diamond m$), or we may fill the slot (replacing the $\diamond$ with m). These four types of operations are denoted by $\mathbf{m}_i$, $\mathbf{l}_i$, $\mathbf{r}_i$, and $\mathbf{f}_i$, respectively. This gives a unique encoding of every permutation, called

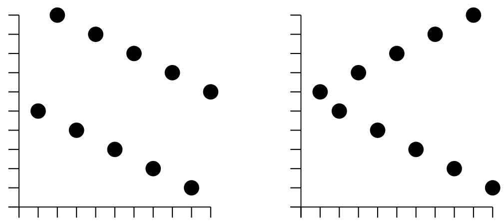

Figure 1: A vertical parallel alternation (left) and a vertical wedge alternation (right).

the *insertion encoding*[1]. For example, the insertion encoding of 423615 is $\mathbf{m}_1\mathbf{m}_1\mathbf{l}_2\mathbf{f}_1\mathbf{f}_2\mathbf{f}_1$. The insertion encoding of a permutation class $\mathrm{Av}(B)$ is the language consisting of the insertion encodings of every element of $\mathrm{Av}(B)$. We say that a configuration is *valid* for $\mathrm{Av}(B)$ if it can be filled in at least one way to produce a permutation in $\mathrm{Av}(B)$.

For any regular language $\mathcal{L}$, there is an integer k such that if w is a prefix of a word in $\mathcal{L}$, then w is a prefix of a word in $\mathcal{L}$ with at most k additional symbols[2]. Thus for the insertion encoding of $\mathrm{Av}(B)$ to be regular, there must be a bound on the number of slots in valid configurations for $\mathrm{Av}(B)$; if $\mathrm{Av}(B)$ satisfies this constraint, we call it *slot-bounded*. Thus $\mathrm{Av}(B)$ cannot contain arbitrarily long *vertical alternations*, which are permutations in which every even (resp., odd) indexed entry lies above every odd (resp., even) index entry. By the Erdős-Szekeres Theorem, every long vertical alternation contains a long vertical parallel alternation or a long vertical wedge alternation (see Figure 1), which makes it easy to check if the insertion encoding of a class needs only a finite alphabet[3]. This necessary condition is also sufficient:

Theorem 2 (Albert, Linton, and Ruškuc [2]). *For a finite set B of permutations, the following are equivalent:*

(1) $\mathrm{Av}(B)$ contains only finitely many vertical alternations,

(2) there is an integer k such that no valid configuration for $\mathrm{Av}(B)$ has more than k slots,

(4) the insertion encoding of $\mathrm{Av}(B)$ is regular.

4. OUR APPROACH TO THE INSERTION ENCODING

While Albert, Linton, and Ruškuc considered the insertion encoding from the viewpoint of formal languages, our approach parallels that of Theorem 1, and borrows terminology

[1]This encoding actually dates back at least to the work of Françon and Viennot [7], although instead of words, they encoded permutations with colored Motzkin paths.

[2]To see this, suppose that the accepting automaton, say $(S, \Sigma, s_0, A, \delta)$, for $\mathcal{L}$ has $k+1$ states. If there is a path from $\delta(s_0, w)$ to an accept state, then there is a path from $\delta(s_0, w)$ to an accept state which does not revisit any states, and thus has length at most k.

[3]More explicitly, $\mathrm{Av}(B)$ contains only finitely many vertical parallel alternations oriented as in Figure 1 if and only if B contains a member of $\mathrm{Av}(123, 3142, 3412)$, while it contains only finitely many vertical wedge alternations oriented as in Figure 1 if and only if B contains a member of $\mathrm{Av}(132, 312)$. One must also check the reversals of these classes, $\mathrm{Av}(321, 2143, 2413)$ and $\mathrm{Av}(213, 231)$, respectively.

from the minimization of DFAs.

Given a DFA $M = (S, \Sigma, s_0, A, \delta)$ over the alphabet Σ with no unreachable states, we say that $w \in \Sigma^*$ *distinguishes* between two states s and t of M if $\delta(s, w) \in \mathcal{L}(M)$ and $\delta(t, w) \notin \mathcal{L}(M)$, or vice versa. Two states are called *indistinguishable* if there is no word which distinguishes them. This defines an equivalence relation on the states of M; for a state $s \in S$, we let $[s]$ denote the equivalence class of all states which are indistinguishable from s. By the Myhill-Nerode Theorem, this equivalence relation describes the *minimal automation for* $\mathcal{L}(M)$, that is, the automaton with the minimum possible number of states which accepts the language $\mathcal{L}(M)$.

The Myhill-Nerode Theorem [12, 13]. *Let* $M = (S, \Sigma, s_0, A, \delta)$ *be a DFA with no states which are unreachable from* s_0*. The DFA* $\tilde{M} = (\tilde{S}, \Sigma, [s_0], \tilde{A}, \tilde{\delta})$ *where* $\tilde{S} = \{[s] : s \in S\}$, $\tilde{A} = \{[s] : s \in A\}$*, and* $\tilde{\delta}([s], w) = [sw]$ *is the minimum DFA for* $\mathcal{L}(M)$*.*

Suppose that we are given the basis B for a slot-bounded class $\mathrm{Av}(B)$, and that we would like to construct the accepting automaton for the insertion encoding of $\mathrm{Av}(B)$. We could build an infinite accepting automaton in which the states of the automata are the valid configurations for $\mathrm{Av}(B)$. The initial state would be $\diamond$, while the accept states would be the slotless configurations, and the transitions would be the obvious transitions given by inserting in the middle of a slot, to the left or right, or filling the slot.

In order to construct a finite automaton which accepts the insertion encoding of $\mathrm{Av}(B)$, we essentially minimize this infinite automaton, although we focus only on a special type of indistinguishable states. We say that the entry $c(i)$ in the configuration c is *insertion-encoding-reducible* (relative to B), or simply *IE-reducible* if c is indistinguishable from $\mathrm{st}(c - c(i))$, where here we have extended the notion of standardization to states in the obvious manner, e.g., $\mathrm{st}(9\diamond16\diamond72) = 5\diamond13\diamond42$. Note, trivially, that $c(i)$ will not be IE-reducible, if it is a slot or if it has slots to both of its sides, as then $c(i)$ and $\mathrm{st}(c - c(i))$ will have a different number of slots.

Proposition 3. *Let c be a valid configuration for* $\mathrm{Av}(B)$*, let b denote the length of the longest element of B, and let j denote the number of slots in c. If the entry $c(i)$ is neither a slot nor adjacent to two slots, then it is IE-reducible if and only if no word of length at most $b + j - 2$ distinguishes c and* $\mathrm{st}(c - c(i))$*.*

Proof. If the entry $c(i)$ is IE-reducible then c and $\mathrm{st}(c - c(i))$ are equivalent, so they are not distinguished by any words.

To prove the other direction, suppose then that the entry $c(i)$ is IE-irreducible, so c and $\mathrm{st}(c - c(i))$ are distinguished by some word, and choose w to be the shortest word which distinguishes between c and $\mathrm{st}(c - c(i))$. If $\delta(c, w)$ were an accept state then $\delta(\mathrm{st}(c - c(i)), w)$ would be as well, so $\delta(\mathrm{st}(c - c(i)), w)$ must be an accept state while $\delta(c, w)$ is not. This implies that $\delta(\mathrm{st}(c - c(i)), w)$ is a permutation (i.e., slotless configuration) in $\mathrm{Av}(B)$. As c and $\mathrm{st}(c - c(i))$ have the same number of slots (because $c(i)$ is neither a slot nor adjacent to two slots), $\delta(c, w)$ is also a permutation, but does *not* lie in $\mathrm{Av}(B)$. Let π denote the permutation $\delta(c, w)$ and choose some copy of a basis element $\beta \in B$ in π. Because $\mathrm{st}(\pi -$

$c(i)) = \delta(\mathrm{st}(c - c(i)), w) \in \mathrm{Av}(B)$, this copy of β consists of $c(i)$ together with at most $b-1$ other entries.

Now consider any permutation $\sigma \leqslant \pi$ which contains

- all entries in the underlying permutation of c,
- all entries from the chosen copy of β in π, and
- at least one entry per slot of c.

Clearly, every such σ evolves from c, or in other words, for any such σ there is some word v such that $\delta(c, v) = \sigma$. Also, $\sigma - c(i) \in \mathrm{Av}(B)$, so any such σ distinguishes between c and $c - c(i)$. To complete the proof, we need only show that there is such a σ which arises by inserting at most $b + j - 2$ entries into c, which follows readily. In the shortest possible choice of σ, we need to insert between 1 and $b - 1$ entries to contain the chosen copy of β, and then at most $j - 1$ entries to contain one entry per slot (because the entries from β must occupy at least one of the j slots of c). ☐

We are now ready to state and prove our strengthening of Theorem 2.

Theorem 2′. *For a finite set B of permutations, the following are equivalent:*

(1) $\mathrm{Av}(B)$ contains only finitely many vertical alternations,

(2) there is an integer k such that no valid configuration for $\mathrm{Av}(B)$ has more than k slots,

(3) every sufficiently long configuration contains an IE-reducible entry,

(4) the insertion encoding of $\mathrm{Av}(B)$ is regular.

As in Theorem 1, note that three of the implications in Theorem 2′ are trivial. We have already remarked that (4) $\Longrightarrow$ (2) and (2) $\Longleftrightarrow$ (1), while (3) $\Longrightarrow$ (4) because (3) implies that the insertion encoding for $\mathrm{Av}(B)$ has a finite accepting automaton. Only (2) $\Longrightarrow$ (3) remains.

Proof that (2) $\Longrightarrow$ (3) in Theorem 2′. Suppose that the longest element of B has length b. We are given that no valid configuration (for $\mathrm{Av}(B)$) has more than k slots, and must show that every sufficiently long configuration contains an IE-reducible entry.

Given a valid configuration c of length n with $j \leqslant k$ slots, let I denote the set of entries which are neither a slot nor adjacent to two slots. Note that since no valid configuration has more than k slots, all but a bounded number of entries lie in I.

If the entry $c(i) \in I$ is IE-irreducible then the proof of Proposition 3 shows that there is a word w of length at most $b + j - 2 \leqslant b + k - 2$ which distinguishes between c and $c(i)$. We say that w *witnesses* the IE-irreducibility of $c(i)$. As c and $c - c(i)$ have the same number of slots (because $c(i)$ was chosen from I), this implies that $\delta(c - c(i), w)$ is a permutation in $\mathrm{Av}(B)$ while $\delta(c, w)$ is also a permutation, but does not lie in $\mathrm{Av}(B)$. Every occurrence of an element of B in $\delta(c, w)$ must include the entry $c(i)$ and at least one entry not in c

(because c is a valid configuration), so no word may witness the IE-irreducibility of more than $b-1$ entries of I. Therefore, since there are a bounded number of possible witnesses (they can each have length at most $b+k-2$), each can witness at most $b-1$ entries of I, and I contains all but a bounded number of entries, for n sufficiently large, there must be at least one entry of I without a witness. This entry is IE-irreducible, completing the proof. □

It follows easily from the definitions that if $\pi(i)$ is GT-reducible (relative to B) for the permutation π, then in every configuration whose underlying permutation is π, the entry corresponding to $\pi(i)$ is IE-reducible. This verifies that every class with a finitely labeled generating tree also has a regular insertion encoding.

5. IMPLEMENTATION NOTES

Theorem 2′, and in particular Proposition 3, lead quickly to an algorithm for computing the accepting automaton, and therefore the rational generating function, for any class $\mathrm{Av}(B)$ with a finite basis and regular insertion encoding. To avoid trivialities, let us suppose that $1 \in \mathrm{Av}(B)$. We begin with a single state, labeled $\diamond$, in our partial automaton and a set $T = \{\diamond\}$ of states/configurations which we have yet to check for IE-reducibility.

At each stage until T is empty, we choose from T a configuration c of minimum possible length. We then check every entry $c(i)$ of c which is neither a slot nor adjacent to two slots for IE-reducibility, as described in Proposition 3 (this requires checking all words over the appropriate alphabet of length up to $b+j-2$, where j denotes the number of slots in c).

If no entry of c is IE-reducible, we add appropriate transitions from c to every valid state/configuration obtained by inserting a new maximum entry into c, and then we remove c from T and add each valid configuration obtained from it to this set. (Checking the validity of these configurations requires that we verify that they have a B-avoiding "filling".) For example, if $c = 2\diamond 1$ has no IE-reducible entry, then we add a transition labeled $\mathbf{m}_1$ to $2\diamond 3\diamond 1$ if this configuration is valid, a transition labeled $\mathbf{l}_1$ to $23\diamond 1$ if this configuration is valid, a transition labeled $\mathbf{r}_1$ to $2\diamond 31$ if this configuration is valid, and a transition labeled $\mathbf{f}_1$ to 231 if this configuration is valid.

Otherwise, c has an IE-reducible entry, say $c(i)$, so the state labeled c is indistinguishable from the state labeled $\mathrm{st}(c - c(i))$. This case involves a bit of subtlety because the configuration $\mathrm{st}(c - c(i))$ might itself have an IE-reducible entry, and because there is no guarantee that we have checked checked $\mathrm{st}(c - c(i))$ for IE-reducible elements up to this point. The simplest way to handle this situation is to search through our partial automaton, replacing productions which lead to the state labeled c by productions which lead to the state labeled $\mathrm{st}(c - c(i))$, to then remove c from T, and to add $\mathrm{st}(c - c(i))$ to T.

Theorem 2′ guarantees that this procedure will eventually terminate if $\mathrm{Av}(B)$ has a regular insertion encoding.

6. COUNTING SUM INDECOMPOSABLE PERMUTATIONS

The permutation π of length n is *sum indecomposable* (or, *connected*) if there is no integer $2 \leqslant i \leqslant n-1$ such that $\pi(\{1, 2, \ldots, i\}) = \{1, 2, \ldots, i\}$.

As we describe first, it is fairly easy to characterize the sum indecomposable permutations via the insertion encoding. The evolution of a sum *decomposable* permutation must contain a non-initial configuration whose only slot occurs at the end of the configuration. Conversely, every permutation which can be formed from such a configuration is sum decomposable. If $\mathrm{Av}(B)$ has a regular insertion encoding, then we know by Theorem 2 that there is a constant k such that no valid configuration for $\mathrm{Av}(B)$ has more than k slots. In order to recognize the sum *indecomposable* permutations in this class, we therefore need only to keep track of how many open slots a configuration has and whether the rightmost slot occurs at the end of the configuration, rejecting a permutation whenever its evolution includes a non-initial configuration whose only slot occurs at the end of the configuration. It follows from the closure properties of regular languages that the sum indecomposable permutations in a class with a regular insertion encoding also have a regular insertion encoding.

While this shows that the sum indecomposable permutations in a class $\mathrm{Av}(B)$ with a regular insertion encoding themselves have a regular insertion encoding, it describes a rather circuitous route to this encoding. Instead, a straight-forward adaptation of our approach leads directly to the accepting automaton for the insertion encoding of sum indecomposable permutations in $\mathrm{Av}(B)$. Let us say that the element $c(i)$ of the configuration c is *SIE-reducible* (relative to B) if it is IE-reducible and is not the rightmost entry of c. It follows from our proof of Theorem 2′ that every sufficiently long valid configuration for $\mathrm{Av}(B)$ has more than one IE-reducible entry, and so has at least one SIE-reducible entry. To construct the accepting automaton for the sum indecomposable permutations in $\mathrm{Av}(B)$, one therefore eliminates all configurations whose only slot occurs at the end, and identifies c and $\mathrm{st}(c - c(i))$ whenever $c(i)$ is SIE-reducible. This is also implemented in the package INSENC.

7. CONCLUSION

We have presented a new viewpoint of regular insertion encodings, which has led to Theorem 2′ and to an implementation in the Maple package INSENC, available from the author's homepage. We conclude with some results obtained from this package.

Recall that the three permutation class symmetries inverse ($\pi \mapsto \pi^{-1}$), reverse ($\pi \mapsto \pi(n) \cdots \pi(2)\pi(1)$), and complement ($\pi \mapsto (n+1-\pi(1))(n+1-\pi(2)) \cdots (n+1-\pi(n))$) generate the symmetries of the square. Given a set of permutation classes, it is therefore useful to divide them into symmetry classes. For example, the $\binom{4!}{2} = 276$ permutation classes with precisely two basis elements of length 4 fall into 56 distinct symmetry classes. Two permutation classes are further said to be *Wilf-equivalent* if they are equinumerous. Le [11] recently established that these 56 symmetry classes form 38 distinct Wilf classes.

Of these 38, 12 can be enumerated with regular insertion encodings. Of those 12, 10 have already been enumerated using finitely labeled generating trees, and their generating functions are reported in Vatter [15] (these generating functions were also computed by hand by Kremer and Shiu [10]). The 2 new generating functions are listed below.

Class	Generating function
$\mathrm{Av}(4321, 1324)$	$\frac{1-11x+56x^2-172x^3+357x^4-519x^5+554x^6-413x^7+217x^8-83x^9+20x^{10}-2x^{11}}{(1-x)^{12}}$
$\mathrm{Av}(4321, 3142)$	$\frac{(1-x)(1-3x)^2}{(1-2x)^2(1-4x+x^2)}$

From the generating function displayed above, it follows that for large n, the number of permutations in $\mathrm{Av}(4321, 1324)$ of length n is given by a polynomial. This is not a surprise, as it can be checked that this class meets the conditions of Huczynska and Vatter [8] or Albert, Atkinson, and Brignall [1] who, building on the work of Kaiser and Klazar [9], characterized the permutation classes of polynomial growth.

For a final example, Tenner [14] recently proved that the number of repeated letters in a reduced decomposition of the permutation π is at most the number of copies of 321 and 3412 in π, with equality if and only if

$$\pi \in \mathrm{Av}(4321, 34512, 45123, 35412, 43512, 45132, 45213, 53412, 45312, 45231).$$

The Maple package INSENC can automatically compute that the generating function for this class is

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

Acknowledgments: I thank Michael Albert for numerous helpful conversations, and for pointing out the work of Françon and Viennot [7].

REFERENCES

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

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

[3] ATKINSON, M. D., MURPHY, M. M., AND RUŠKUC, N. Partially well-ordered closed sets of permutations. *Order 19*, 2 (2002), 101–113.

[4] CHUNG, F. R. K., GRAHAM, R. L., HOGGATT, JR., V. E., AND KLEIMAN, M. The number of Baxter permutations. *J. Combin. Theory Ser. A 24*, 3 (1978), 382–394.

[5] ERDŐS, P., AND SZEKERES, G. A combinatorial problem in geometry. *Compos. Math. 2* (1935), 463–470.

[6] FLAJOLET, P., AND SEDGEWICK, R. *Analytic combinatorics*. Cambridge University Press, Cambridge, 2009.

[7] FRANÇON, J., AND VIENNOT, G. Permutations selon leurs pics, creux, doubles montées et double descentes, nombres d'Euler et nombres de Genocchi. *Discrete Math. 28*, 1 (1979), 21–35.

[8] HUCZYNSKA, S., AND VATTER, V. Grid classes and the Fibonacci dichotomy for restricted permutations. *Electron. J. Combin. 13* (2006), Research paper 54, 14 pp.

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

[10] KREMER, D., AND SHIU, W. C. Finite transition matrices for permutations avoiding pairs of length four patterns. *Discrete Math. 268*, 1-3 (2003), 171–183.

[11] LE, I. Wilf classes of pairs of permutations of length 4. *Electron. J. Combin. 12* (2005), Research Paper 25, 27 pp.

[12] MYHILL, J. Finite automata and the representation of events. *Wright Air Development Command Tech. Rep. 57-624* (1957), 112–137.

[13] NERODE, A. Linear automaton transformations. *Proc. Amer. Math. Soc. 9* (1958), 541–544.

[14] TENNER, B. E. Repetition in reduced decompositions. arXiv:1106.2839v1 [math.CO].

[15] VATTER, V. Finitely labeled generating trees and restricted permutations. *J. Symbolic Comput. 41*, 5 (2006), 559–572.