PU. M. A. Vol. **21** (2010), No. 2, pp. 137 – 160

Refining enumeration schemes to count according to the inversion number

Andrew M. Baxter*
Department of Mathematics,
Rutgers University,
email: baxter@math.rutgers.edu

(Received: February 15, 2010, and in revised form September 2, 2010)

Abstract. Enumeration schemes were developed by Zeilberger, Vatter, and Pudwell as automatable methods to enumerate the set of permutations of length n which avoid a set of patterns B, $S_n(B)$. In this paper we show how their schemes may be refined to enumerate $S_n(B)$ according to the inversion number of each permutation.

Mathematics Subject Classification(2000). 05A05, 05A15, 05A30, 68R15.
Keywords: pattern avoidance, automated discovery.

1 Introduction

For a word $\pi = \pi_1\pi_2\ldots\pi_n$, let red$(\pi)$ be the word obtained by replacing the i^{th} smallest letter(s) of π with i. For example, red$(413351) = 312241$. The word π of length n *contains* pattern σ, a word of length k, if there is some k-tuple $1 \leq i_1 < i_2 < \cdots < i_k \leq n$ such that red$(\pi_{i_1}\pi_{i_2}\ldots\pi_{i_k}) = \sigma$. If π does not contain σ, then π is said to *avoid* σ. Hence we see that $\pi = 34512$ contains the pattern 231 as formed by the letters $\pi_1\pi_3\pi_4 = 351$, but exhaustive checking shows it avoids 132. For a set of patterns B, π is said to avoid B if π avoids all $\sigma \in B$. We denote the subset of S_n which avoids B by $S_n(B)$, and its cardinality $s_n(B)$.

In the past two decades, several "general purpose" methods have developed to compute the number of permutations of a given length which avoid a set of patterns. These methods have been used to compute or to characterize $s_n(B)$ for large classes of sets B, as opposed to earlier results in the field which used arguments specific to a given B. Enumeration schemes, the topic of the present paper, were introduced by Zeilberger in [23] and improved by Vatter in [18]. Later Zeilberger improved Vatter's implementation in [24]. The greatest feature of these schemes is that they may be

*Partially supported by NSA award H98230-09-1-0014.

discovered automatically by a computer: the user need only input the set B (along with bounds to the computer search) and the computer will return an enumeration scheme (if one exists within the bounds of the search) which computes $s_n(B)$ in polynomial time. Furthermore the construction of the scheme itself is constant-time, so one has a polynomial-time algorithm before even constructing the scheme itself. It must be said, however, that not all B have finite (and thus computable by current methods) enumeration schemes — see [18] for examples. Pudwell extended these methods to compute the number of pattern-avoiding words in [15, 14], as well as the number of barred-pattern-avoiding permutations in [13].

Define the *inversion number of* π, $\mathrm{inv}(\pi)$, to be the number of pairs (i, j) such that $i < j$ but $\pi_i > \pi_j$. Such a pair is called an *inversion*. The inversion number gives a sense of sortedness of a permutation, indicating the number of adjacent transpositions $(i, i+1)$ necessary to sort π into $123 \ldots n$, and hence is a natural permutation statistic to compute. In this paper we refine enumeration schemes to compute

$$F(n, B, q) := \sum_{\pi \in S_n(B)} q^{\mathrm{inv}(\pi)},$$

as well as the analogous weight-enumerators for Pudwell's extensions in [13, 15]. The main results of this paper are Lemma 3.1, Corollary 3.2 and Lemma 4.2, as well as the accompanying Maple packages which implement these refinements.

For word $\pi \in [k]^n$, represent the trivial symmetries of reversal and complement by $\pi^r = \pi_n \pi_{n-1} \ldots, \pi_1$ and $\pi^c = (k+1-\pi_1)(k+1-\pi_2)\ldots(k+1-\pi_n)$, and let $B^r = \{\pi^r : \pi \in B\}$, $B^c = \{\pi^c : \pi \in B\}$. When B contains permutations only, let $B^{-1} = \{\pi^{-1} : \pi \in B\}$. These have the effect that $F(n, B^r, q) = F(n, B^c, q) = q^{\binom{n}{2}} F(n, B, q^{-1})$ since these operations change all inversions to non-inversions and vice versa, and $F(n, B^{-1}, q) = F(n, B, q)$ since inverting a permutation preserves its inversion number.

Barcucci, Del Lungo, Pergola, and Pinzani adapted generating trees to compute $F(n, B, q)$ for $B = \{321\}$, $B = \{321, 3\bar{1}42\}$, and $B = \{4231, 4132\}$ in [4]. Based on these results, Cheng, Eu, and Fu in [11] constructed a correspondence between $S_n(\{321\})$ and Catalan paths of length $2n$, where a permutation π with $\mathrm{inv}(\pi) = k$ maps to a Catalan path which contains a certain region with area k. In the same vein, Bandlow and Killpatrick in [3] constructed a similar inv-to-area bijection between $\pi \in S_n(\{312\})$ and Dyck paths, which was later extended with Egge in [2] to an inv-to-area bijection between $\pi \in S_n(\{4231, 4132\})$ and Schröder paths. Chen, Deng, and Yang in [10] also constructed an inv-to-area bijection between $S_n(\{321, 3\bar{1}42\})$ and Motzkin paths. The tools described in this paper can aid in similar endeavors to construct such bijections.

Due to their automatability, enumeration schemes provide a powerful tool to explore a set $S_n(B)$ for a given B. While techniques such as generating trees result in functional equations for generating functions, the discovery of simple generating tree rules requires considerable ingenuity. On the other hand, to use enumeration schemes one needs only to load the requisite Maple packages, enter the pattern set B and some search parameters, and wait for the computer to search for a scheme. If a scheme exists, then one can compute $s_n(B)$ quickly, getting results for, say, $n \leq 40$ in under five minutes on a personal laptop computer. The data may then be used as fodder to conjecture generating functions or even explicit formulas in n, or the algorithm itself may be considered "the answer" (as defined by Wilf in [21]). The refinements in the present paper yield the analogous results for the refinement according to inversion number, allowing one to compute $F(n, B, q)$ in polynomial time. Further, these refinements apply for any B with a finite scheme, so there is no loss in generality.

In section 2 we describe enumeration schemes for pattern-avoiding permutations, as they appear in [24]. In section 3 we describe how to adapt enumeration schemes to compute $F(n, B, q)$ for any B which has an enumeration scheme. In section 4 we summarize Pudwell's extensions to pattern-avoiding words and barred-pattern-avoiding permutations, as well the relevant adaptations which compute the refinement according to inv. Section 5 details the implementation of each adaptation in the Maple packages qVATTER, qmVATTER, and qbVATTER, and section 6 lists some of the results obtained from these packages.

2 Constructing Enumeration Schemes

Enumeration schemes are succinct encodings for a family of recurrence relations enumerating a family of sets. The enumerated sets are subsets of $S_n(B)$ determined by prefixes. For pattern $p \in S_k$, let $S_n(B)[p]$ be the set of permutations $\pi \in S_n(B)$ such that $\text{red}(\pi_1\pi_2 \dots \pi_k) = p$. In cases where we need further refinement call $C \in \binom{[n]}{k}$ a *prefix set*, and define $S_n(B)[p, C]$ to be those permutations in $S_n(B)[p]$ whose first k letters form C. In particular, if $C = \{c_1, \dots, c_k\}$ such that $c_i < c_{i+1}$ then if $\pi \in S_n(B)[p, C]$, then $\pi_i = c_{p_i}$ for $1 \leq i \leq k$. For example,

$$S_5(123)[21, \{3, 5\}] = \{53142, 53214, 53241, 53412, 53421\}.$$

Since we are interested in enumeration, it will be handy to have the notation $s_n(B)[p] := |S_n(B)[p]|$ and $s_n(B)[p, C] := |S_n(B)[p, C]|$.

By looking at the prefix of a permutation, one can identify likely "trouble spots" where forbidden patterns may appear. For example, suppose we wish to avoid the pattern 123. Then the presence of the patten 12 in the prefix indicates the potential for the whole permutation to contain a 123 pattern. In [18], $S_n(B)$ is partitioned according the inverse notion of the pattern formed by the smallest k letters in $\pi \in S_n(B)$.

Enumeration schemes take a divide-and-conquer approach to enumeration. Any $S_n(B)[p]$ for $p \in S_k$ may be partitioned into the family of sets $S_n(B)[p']$ for each of the *children* $p' \in S_{k+1}(B)[p]$. The sets indexed by these children are then counted as described below, and their sizes are totaled to obtain $s_n(B)[p]$. In the end we have counted $s_n(B)$, since $S_n(B) = S_n(B)[\epsilon] = S_n(B)[1]$, where ϵ is the empty (i.e., length-0) permutation.

For $p \in S_k$ a set $S_n(B)[p]$ fits into one of three cases:

(1) If $n = k$, then $S_n(B)[p]$ is either $\{p\}$ or $\emptyset$, depending on whether p avoids B.

(2) For each $C \in \binom{[n]}{k}$, one of the following happens:

   (2a) $S_n(B)[p, C]$ is empty, so $s_n(B)[p, C] = 0$

   (2b) $S_n(B)[p, C]$ is in bijection with some $S_{\hat{n}}(B)[\hat{p}, \hat{C}]$ for $\hat{n} < n$, so $s_n(B)[p, C] = s_{\hat{n}}(B)[\hat{p}, \hat{C}]$.

(3) $S_n(B)[p]$ must be partitioned further, so $s_n(B)[p] = \sum_{p' \in S_{k+1}(B)[p]} s_n(B)[p']$.

Case (1) provides the base cases for our recurrence. For case (2), if there is any $C \in \binom{[n]}{k}$ for which neither (2a) nor (2b) holds then we must divide $S_n(B)[p]$ as in case (3). For case (2a), the *gap vector criteria* for the given p identify which C yield empty $S_n(B)[p, C]$. Gap vector criteria are discussed in section 2.1. The bijection in (2b) is performed by removing certain of the first k letters of $\pi \in S_n(B)[p, C]$. Which letters may be removed depends on p and B, but not C. These letters are called *reversibly deletable*, and are discussed in section 2.2 below.

2.1 Gap Vectors

The motivation for gap vectors lies in the idea of "vertical space" (in the sense of the graph of a permuation) in a prefix set C. Sometimes the difference of the values of letters in the prefix is so great that a forbidden pattern *must* appear. To make this notion more precise, we follow our example above and count $S_n(123)[12]$. Observe that $S_n(123)[12, \{c_1, c_2\}]$ is empty if $c_1 < c_2 < n$, since otherwise if $\pi \in S_n(123)[12, \{c_1, c_2\}]$ then $\pi_i = n$ for some $i \geq 3$ and so $c_1 c_2 n$ forms a 123 pattern. Moreover the possibility for any $\pi_i > c_2$ for $i \geq 3$ prohibits the formation of any 123-avoiding permutation, so we must restrict the space above c_2.

More generally, consider $S_n(B)[p, C]$ where $C = \{c_1, c_2, \ldots, c_k\}$ for $c_i < c_{i+1}$. Let $c_0 = 0$ and $c_{k+1} = n + 1$, and form the $(k+1)$-vector $g = g(n, C)$ so that $g_i = c_i - c_{i-1} - 1$. Note that g_i counts the number of letters for any $\pi \in S_n(B)[p, C]$ which lie strictly between c_{i-1} and c_i, i.e. the size of vertical gap between c_{i-1} and c_i, and furthermore the letters filling this gap appear in the part of π following the prefix. In the example above, if $g(n, C) \geq \langle 0, 0, 1\rangle$ in the product order of $\mathbb{N}^3$ (i.e., component-wise), then $S_n(123)[12, C] = \emptyset$. We call $\langle 0, 0, 1\rangle$ a gap vector for the prefix 12. More generally we may make the following definition:

Definition 2.1 Given a set of forbidden patterns B and prefix p, then v is a *gap vector for prefix p with respect to B* if, for all n, $S_n(B)[p, C] = \emptyset$ for any C such that $g(n, C) \geq v$. When this happens, we say that C *satisfies* the *gap vector criterion* for v.

Hence $v = \langle 0, 0, 1\rangle$ is a gap vector for $p = 12$ with respect to $B = \{123\}$, and any prefix set $C = \{c_1, c_2\}$ with $c_1 < c_2 < n$ satisfies the gap vector condition for v.

It should be noted that this definition reverses the terminology of [18] to match that of [13, 14, 15, 24].

Observe that gap vectors for a given prefix $p \in S_k$ form an upper order ideal in $\mathbb{N}^{k+1}$, and that it suffices to determine only the minimal elements (which form a basis). For details on the discovery of gap vectors, and automating the process, see [18, 24]. The refinement discussed below does not depend on the construction itself, so those details may be safely omitted without sacrificing clarity.

2.2 Reversible Deletability

When C fails the gap vector criterion for all gap vectors v, we must rely on bijections with previously-computed $S_{\hat{n}}(B)[\hat{p}, \hat{C}]$. To continue our example above, consider $S_n(123)[12, \{c_1, n\}]$. Here $\{c_1, n\}$ fails all gap vector criteria since $\langle 0, 0, 1\rangle$ forms the basis for the ideal of gap vectors. However, any $\pi \in S_n[12, \{c_1, n\}]$ has $\pi_2 = n$, so we may form a bijection

$S_n(123)[12, \{c_1, n\}] \leftrightarrow S_{n-1}(123)[1, \{c_1\}]$ via the map $d_2 : \ \pi_1\pi_2 \ldots \pi_n \mapsto \text{red}(\pi_1\pi_3 \ldots \pi_n)$. The deletion of a letter always preserves pattern-avoidance properties, but reversing the process by adding a letter has the potential for creating a forbidden pattern. Here, however an n at the second index cannot possibly create a 123 so we may safely reverse the deletion.

More generally define the deletion $d_r(\pi)$ to be $\text{red}(\pi_1 \ldots \pi_{r-1}\pi_{r+1} \ldots \pi_n)$, that is, the permutation obtained by omitting the r^{th} letter of π and reducing. As for d_r's effect on $C = \{c_1 \ldots c_k\}$, $c_i < c_{i+1}$, define $d_{p,r}(C) = \{c_1, \ldots, c_{p_r-1}, c_{p_r+1}-1, \ldots, c_k-1\}$. Clearly if $\pi \in S_n(B)[p, C]$ then $d_r(\pi) \in S_{n-1}(B)[d_r(p), d_{p,r}(C)]$, since deleting a letter cannot cause the appearance of a pattern. However, for certain pairs (p, r), as in the case for $d_2 : S_n(123)[12, \{c_1, n\}] \rightarrow S_{n-1}(123)[1, \{c_1\}]$, this deletion operation is invertible. This brings us to the following definition:

Definition 2.2 The index r is *reversibly deletable with respect to* p *and* B if the map

$$d_r : S_n(B)[p, C] \rightarrow S_{n-1}(B)[d_r(p), d_{p,r}(C)]$$

is a bijection for any C failing the gap vector criterion for every gap vector of p with respect to B.

Vatter uses the term *enumeration scheme reducible* (or *ES-reducible*) for the same concept in [18].

Note that if $r < s$ are both reversibly deletable with respect to p and B, then the composition $d_r(d_s(\pi))$, the simultaneous deletion at both r and s, is still invertible. For a set R with $r_1 = \min R$, recursively define $d_R(\pi) = d_{r_1}(d_{R \setminus r_1}(\pi))$, and so $d_R(\pi)$ is the permutation obtained by omitting π_r for each $r \in R$ and reducing. Similarly let $d_{p,R}(C) = d_{d_{R\setminus r_1}(p), r_1}(d_{p,R\setminus r_1}(C))$. Making the deletions from right to left eases notation. We say R is *reversibly deletable with respect to* p *and* B if the map $d_R : S_n(B)[p, C] \rightarrow S_{n-|R|}(B)[d_R(p), d_{p,R}(C)]$ is a bijection for any C failing all gap vector criteria for p with respect to B. For permutations, R is reversibly deletable iff each $r \in R$ is reversibly deletable. This is not the case for other scenarios as we will see in section 4.

Vatter showed that identifying reversibly deletable indices is a finite process and thus subject to a computer search. Automating this process is discussed in [18, 23, 24]. As our refinement below is independent of how the enumeration schemes are constructed, we omit these details.

2.3 Formal Definition of Enumeration Schemes

Formally, an enumeration scheme, E, for $S_n(B)$ is a set of triples (p, G_p, R_p), where $p \in S_k$ is the prefix pattern, G_p is the basis of gap vectors associated with p, and R_p is the set of indices of reversibly deletable letters associated with p. Further, the following criteria must hold for any $(p, G_p, R_p) \in E$

1. $(\epsilon, \emptyset, \emptyset) \in E$.
2. If $R_p = \emptyset$, then $(p', G_{p'}, R_{p'}) \in E$ for every child p' of p.
3. If $R_p \neq \emptyset$, then $(\hat{p}, G_{\hat{p}}, R_{\hat{p}}) \in E$ for $\hat{p} = d_{R_p}(p)$.

One can then "read" the enumeration scheme E to compute $S_n(B)[p, C]$ according to the following rules:

1. If C passes the gap vector criteria for some $v \in G_p$, then $S_n(B)[p, C] = \emptyset$.
2. For each set C that fails the gap criteria for all $v \in G$ we have a bijection

$$d_{R_p} : S_n(B)[p, C] \leftrightarrow S_{n-|R_p|}(B)[d_{R_p}(p), d_{p,R_p}(C)]$$

(i.e. R_p is a reversibly deletable set of indices)

When combined with the obvious initial conditions (e.g. if $p \in S_n(B)$ and $C = \{1, \ldots, n\}$ then $S_n(B)[p, C] = \{p\}$), the enumeration scheme presents the recurrences for $s_n(B)[p, C]$, and hence $s_n(B)$.

To illustrate, consider the enumeration scheme for $S_n(123)$ as discussed in the following sections.

$$\{(\epsilon, \emptyset, \emptyset), (1, \emptyset, \emptyset), (12, \{\langle 0, 0, 1\rangle\}, \{2\}), (21, \emptyset, \{1\})\} \tag{1}$$

Since $R_\epsilon = \emptyset$, the first condition above requires the presence of $(1, G_1, R_1)$. Starting with the pattern 1 yields no additional information, so $R_1 = \emptyset$ and thus explaining the presence of $(12, G_{12}, R_{12})$ and $(21, G_{21}, R_{21})$. As discussed above, $\{\langle 0, 0, 1\rangle\}$ forms a basis for the gap vectors for 12 and whenever C fails this gap vector criteria the second letter is reversibly deletable. For the fourth entry in the scheme, suppose that $\pi \in S_n(\emptyset)[21]$ contains a 123 pattern involving the first letter, say $\pi_1 < \pi_i < \pi_j$ for $i < j$. Then since $\pi_2 < \pi_1$, $\pi_2 < \pi_i < \pi_j$ is another 123 pattern, we see that π_1 cannot be the deciding factor for whether π contains 123. Hence the index 1 is reversibly deletable, so $R_{21} = \{1\}$.

An enumeration scheme gives us a polynomial-time algorithm to compute $s_n(B)[p, C]$. We construct the system of recurrences based on the partitions and bijections above, along with base cases as given by the gap vector criteria and the trivial cases when $S_n(B)[p] = \{p\}$. For example, the above enumeration scheme is translated into the following system of recurrences:

$$\begin{aligned}
|S_n(123)| &= s_n(123)[\epsilon] \\
&= s_n(123)[1] \\
&= \sum_{i=1}^{n} s_n(123)[1, \{i\}] \\
s_n(123)[1, \{i\}] &= \sum_{j=1}^{i-1} s_n(123)[21, \{i > j\}] + \sum_{j=i+1}^{n} s_n(123)[12, \{i < j\}] \\
s_n(123)[12, \{i < j\}] &= \begin{cases} 0 & \text{if } n - j \geq 1 \\ s_n(123)[1, \{i\}] & \text{otherwise} \end{cases} \\
s_n(123)[21, \{i > j\}] &= s_n(123)[1, \{j\}]
\end{aligned}$$

Simplifying the above, we get:

$$s_n(123)[1, \{i\}] = \sum_{j=1}^{i} s_{n-1}(123)[1, \{j\}] \tag{2}$$

3 Refining Enumeration Schemes

Given an enumeration scheme E for $S_n(B)$ we now show how to re-interpret it to form a recurrence for $F(n, B, q)[p, C] = \sum_{\pi \in S_n(B)[p,C]} q^{\mathrm{inv}(\pi)}$. We begin with the initial conditions. As before when n is small, $S_n(B)[p]$ is either $\{p\}$, and hence $F(n, B, q)[p] = q^{\mathrm{inv}(p)}$, or $\emptyset$, in which case $F(n, B, q)[p] = 0$. The other initial conditions derive from the gap vector criteria. When C satisfies a gap vector criterion, $S_n(B)[p, C] = \emptyset$ and so $F(n, B, q)[p, C] = 0$. Clearly partitioning the set $S_n(B)[p]$ into $S_n(B)[p']$ has the same effect on $F(n, B, q)[p]$ as for $s_n(B)[p]$, that is, $F(n, B, q)[p] = \sum_{p'} F(n, B, q)[p']$. It remains to consider the bijections d_r and d_R formed by reversibly deletable indices. The effect that d_r has on the inversion number of a permutation, denoted $\delta_r(\pi)$, is described in the following lemma. Observe that the formulation for $\delta_r(\pi)$ below is given in terms of the prefix $\pi_1\pi_2 \ldots \pi_r$, making it suitable for prefix-based enumeration schemes.

LEMMA 3.1 *Let $\pi \in S_n$ and fix some index $1 \le r \le n$. Then*

$$\delta_r(\pi) := \operatorname{inv}(\pi) - \operatorname{inv}(d_r(\pi)) = (\pi_r - 1) + \sum_{i<r} \operatorname{sgn}(\pi_i - \pi_r) \qquad (3)$$

where $\operatorname{sgn}(x)$ *is the signum function*

$$\operatorname{sgn}(x) := \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ 1, & x > 0 \end{cases} . \qquad (4)$$

Proof. From the definition of inv we see that $\delta_r(\pi) = |\{i : i < r, \pi_i > \pi_r\}| + |\{i : r < i, \pi_r > \pi_i\}|$. This second set may be rewritten as $\{i : \pi_i < \pi_r\} \setminus \{i : i < r\}$. Hence we get that

$$\begin{aligned} \delta_r(\pi) &= |\{i : i < r, \pi_i > \pi_r\}| + |\{i : r < i, \pi_r > \pi_i\}| \\ &= |\{i : i < r, \pi_i > \pi_r\}| + |\{i : \pi_i < \pi_r\}| - |\{i : i < r, \pi_i < \pi_r\}| \end{aligned}$$

The formula follows from combining the first and third summands to get $\sum_{i<r} \operatorname{sgn}(\pi_i - \pi_r)$ and the fact that $|\{i : \pi_i < \pi_r\}| = \pi_r - 1$. □

When deleting multiple letters simultaneously, we make use of the following corollary.

COROLLARY 3.2 *Let $\pi \in S_n$ and $R \subseteq [n]$ so that $R = \{r_1, \dots, r_t\}$ for $r_i < r_{i+1}$. Then*

$$\delta_R(\pi) := \operatorname{inv}(\pi) - \operatorname{inv}(d_R(\pi)) = \sum_{r \in R} \delta_r(\pi) - \operatorname{inv}(\pi_{r_1}\pi_{r_2}\dots\pi_{r_t}) \qquad (5)$$

Proof. The deletion of each π_{r_i} causes the loss of $\delta_{r_i}(\pi)$ of the inversions in π. Inversions among the deleted letters, that is those of the form $\pi_{r_i} > \pi_{r_j}$ for $i < j$, are double-counted and thus we must subtract $\operatorname{inv}(\pi_{r_1}\pi_{r_2}\dots\pi_{r_t})$. □

Now consider a reversibly deletable set $R \subseteq \{1, \dots k\}$ for prefix $p \in S_k$ and prefix set C (with $c_i \in C$ such that $c_i < c_{i+1}$). In this case we have the bijection $d_R : S_n(B)[p, C] \leftrightarrow S_{n-|R|}(B)[d_R(p), d_{p,R}(C)]$. First observe that for $r \le k$, $\pi_r = c_{p_r}$ and that $\operatorname{sgn}(\pi_i - \pi_r) = \operatorname{sgn}(p_i - p_r)$ for $i < r$. Thus by Lemma 3.1 $\delta_r(\pi)$ is dependent only on r, p and C. Corollary 3.2

implies that $\delta_R(\pi)$ is constant over all $\pi \in S_n(B)[p, C]$. Hence it follows that $F(n, B, q)[p, C] = q^{\delta_R(p)} F(n - |R|, B, q)[d_R(p), d_{p,R}(C)]$.

To illustrate, we again consider $S_n(123)$ whose scheme is given in (1). As in the classical case, $F(n, 123, q) = \sum_{i=1}^{n} F(n, 123, q)[1, \{i\}]$. For any $\pi \in S_n(123)[21, \{i > j\}]$ we get $\delta_{\{1\}}(\pi) = i-1$, and for any $\pi \in S_n(123)[12, \{i, n\}]$ we get $\delta_{\{2\}}(\pi) = n - 2$. The enumeration scheme leads directly to the following refinement of equation 2:

$$F(n, 123, q)[1, \{i\}] = \sum_{j=1}^{i-1} q^{i-1} \cdot F(n - 1, 123, q)[1, \{j\}] + q^{n-2} \cdot F(n - 1, 123, q)[1, \{i\}]$$

The generating functions $F(n, 321, q) = q^{\binom{n}{2}} F(n, 123, q^{-1})$ were studied via generating trees in Barcucci et al. in [4], as well as Cheng et al. in [11].

4 Extensions

4.1 Words avoiding patterns without repeated letters

We now turn our attention towards Pudwell's extensions of enumeration schemes, the first of which is for words avoiding patterns without repeated letters as discussed in [15]. Let $\nu_i(w)$ represent the number of copies of i in w, and $\nu(w) := \langle \nu_1(w), \nu_2(w), \ldots \rangle \in \mathbb{N}^{\mathbb{N}}$. For alphabet vector $\mathbf{a} \in \mathbb{N}^{\mathbb{N}}$ with finitely many nonzero entries, we may define $S_{\mathbf{a}}$ to be all words w so that $\nu(w) = \mathbf{a}$. Let $m(\mathbf{a})$ be the index of the largest nonzero entry in $\mathbf{a}$, i.e., the largest letter appearing in any $w \in S_{\mathbf{a}}$.

Recall that the definitions of containment and avoidance are given above in terms of words, so we may define $S_{\mathbf{a}}(B)$ to be the subset of $S_{\mathbf{a}}$ avoiding all patterns in B. We will restrict our attention to sets B containing only permutations. Let $s_{\mathbf{a}}(B) = |S_{\mathbf{a}}(B)|$ and $F(\mathbf{a}, B, q) := \sum_{w \in S_{\mathbf{a}}(B)} q^{\mathrm{inv}(w)}$. Questions of enumerating words avoiding permutations first appeared in the Ph.D. thesis of Burstein, [7].

As in the case for permutations, we will partition $S_{\mathbf{a}}(B)$ according to prefixes. For a prefix p of length k, we define $S_{\mathbf{a}}(B)[p]$ to be those words $w \in S_{\mathbf{a}}$ such that $\mathrm{red}(w_1 w_2 \ldots w_k) = p$. We refine this further for a prefix multiset C, letting $S_{\mathbf{a}}(B)[p, C]$ be the set of words $w \in S_{\mathbf{a}}(B)[p]$ such that the multiset $\{w_1, \ldots, w_k\} = C$. As in the case above, Pudwell's schemes compute $s_{\mathbf{a}}(B)[p, C] := |S_{\mathbf{a}}(B)[p, C]|$, while our refinement will compute

$F(\mathbf{a}, B, q)[p, C] := \sum_{w \in S_{\mathbf{a}}(B)[p,C]} q^{\mathrm{inv}(w)}$. Again, each prefix p leads to one of three cases:

(1) $S_{\mathbf{a}}(B)[p]$ is $\{p\}$ or $\emptyset$

(2) For each prefix multiset C, one of the following happens:

(2a) $S_{\mathbf{a}}(B)[p, C]$ is empty

(2b) $S_{\mathbf{a}}(B)[p, C]$ is in bijection with some $S_{\hat{\mathbf{a}}}(B)[\hat{p}, \hat{C}]$.

(3) $S_{\mathbf{a}}(B)[p]$ must be partitioned further, so $s_{\mathbf{a}}(B)[p] = \sum_{p'} s_{\mathbf{a}}(B)[p']$, where the sum runs over all words p' of length $k+1$ such that $\mathrm{red}(p'_1 \ldots p'_k) = p$.

As before, case (1) provides base cases for the recurrences, while case (2a) is handled by an analogue of gap vectors, and the bijections in (2b) are deletion maps d_r.

4.2 Gap Vectors

Again, gap vectors are restrictions on the amount of vertical space which may appear between letters in the prefix of a permutation. Consider prefix p with k letters and prefix multiset $C = \{c_1, c_2, \ldots, c_k\}$ for $c_i \leq c_{i+1}$. Let $c_0 = 1$ and $c_{k+1} = m(\mathbf{a})$, and form the $(k+1)$-vector $h = h(\mathbf{a}, C)$ so that $h_i = c_i - c_{i-1}$. Then we may again define an analogue of gap vectors for words by finding those C such that $S_{\mathbf{a}}(B)[p, C]$ is necessarily empty.

Definition 4.1 Given a set of forbidden patterns B and prefix p, then v is a *word gap vector* for prefix p if, for all $\mathbf{a}$, $S_{\mathbf{a}}(B)[p, C] = \emptyset$ for any C such that $h(\mathbf{a}, C) \geq v$. When this happens, we say that C satisfies the *word gap vector criterion* for v.

Observe the change in interpretation of the h_i compared to the g_i in the case of pattern-avoiding permutations. Here h_i for $2 \leq i \leq k$ represents *one more than* the number of letters between c_{i-1} and c_i, or $h_i = 0$ when $c_{i-1} = c_i$, while h_1 and h_{k+1} are still the number of letters lying below c_1 and above c_k, respectively. While this convention seems odd, it allows for neater computation elsewhere. In particular if $v = \langle v_1, v_2, \ldots, v_{k+1} \rangle$ is a gap vector for p, then $p_1 \ldots p_{t-1} p_{t+1} \ldots p_k$ has the gap vector $v = \langle v_1, v_2, \ldots, v_t + v_{t+1}, \ldots v_{k+1} \rangle$. This is of great use when computing the schemes, although once the scheme is computed either convention would suffice. This alternate convention is presented here for the sake of clarifying the output from the Maple package mVATTER and its refinement qmVATTER.

As before, the word gap vectors form an upper ideal in $\mathbb{N}^{k+1}$ and it suffices to find a finite basis. The details of their automated discovery are given in [15] and will not be repeated here.

4.3 Reversible Deletability

The deletion operator d_r is defined as for permutations, where $d_r(w) = \text{red}(w_1 \dots w_{r-1} w_{r+1} \dots w_n)$. Define $d_{p,r}(\mathbf{a}) = \mathbf{a} - \nu(p_r)$ and $d_{p,r}(C) = \{c_1, \dots, c_{p_r-1}, c_{p_r+1}, \dots, c_k\}$.

Again some maps $d_r : S_{\mathbf{a}}(B)[p, C] \to S_{d_{p,r}(\mathbf{a})}(B)[d_r(p), d_{p,r}(C)]$ are bijections for all C which fail all word gap vector criteria for p with respect to B. In this case r is called *reversibly deletable*, as before. Unlike before, however, if r and s are both reversibly deletable, the set $\{r, s\}$ may not be. For an example consider $w \in S_{\mathbf{a}}(\{123\})[11, \{1, 1\}]$, so w has the form $11w_3w_4 \dots w_n$ where each $w_i \geq w_{i+1}$ for $i \geq 3$. Here both d_1 and d_2 are bijections to $S_{\mathbf{a}-\nu(1)}(\{123\})(1, \{1\})$, however $d_{\{1,2\}}$ is not surjective since $S_{\mathbf{a}-\nu(11)}(\{123\})[\epsilon, \emptyset]$ contains words other than the nonincreasing word. It is not necessary to perform deletions d_R for $|R| > 1$, however, so long as the scheme contains $(p', G_{p'}, R_{p'})$ for all patterns p' contained in p. Since schemes may always be constructed as to contain all subpatterns, we need not consider simultaneous deletions. When there are several reversibly deletable indices, we use the convention to delete only the first.

The maps d_r do affect the number of inversions appearing, but act analogously to Lemma 3.1.

LEMMA 4.2 *Let $w \in S_{\mathbf{a}}$ and fix some index $1 \leq r \leq n$. Then*

$$\delta_r(w) := \text{inv}(w) - \text{inv}\,(d_r(w)) = (\nu_1(w) + \cdots \nu_{w_r-1}(w)) + \sum_{i<r} \text{sgn}(w_i - w_r) \tag{6}$$

where $\text{sgn}(x)$ *is the signum function.*

The proof for Lemma 3.1 applies for this lemma essentially unchanged.

Observe that once again δ_r remains constant over all $w \in S_{\mathbf{a}}(B)[p, C]$. Hence we have refined Pudwell's enumeration schemes for permutation-avoiding words to compute

$$F(\mathbf{a}, B, q)[p, C] := \sum_{w \in S_{\mathbf{a}}(B)[p,C]} q^{\text{inv}(w)}$$

via a system of recurrences. For an example, consider $F(\mathbf{a}, \{123\}, q)$, which has the enumeration scheme

$$\{(\epsilon, \emptyset, \emptyset), (1, \emptyset, \emptyset), (21, \emptyset, \{1\}), (11, \emptyset, \{1, 2\}), (12, \{\langle 0, 1, 1\rangle\}, \{2\})\}$$

This can be translated into the following set of recurrences.

$$\begin{aligned}
F(\mathbf{a}, B, q) &= F(\mathbf{a}, B, q)[\epsilon] = F(\mathbf{a}, B, q)[1] \\
&= \sum_{i=1}^{m(\mathbf{a})} F(\mathbf{a}, B, q)[1, \{i\}] \\
F(\mathbf{a}, B, q)[1, \{i\}] &= \sum_{j=1}^{i-1} F(\mathbf{a}, B, q)[21, \{i, j\}] + \\
&\qquad F(\mathbf{a}, B, q)[11, \{i, i\}] + \sum_{j=i+1}^{m(\mathbf{a})} F(\mathbf{a}, B, q)[12, \{i, j\}] \\
F(\mathbf{a}, B, q)[21, \{i, j\}] &= q^{\mathbf{a}_1 + \cdots + \mathbf{a}_{i-1}} F(\mathbf{a} - \nu(i), B, q)[1, \{j\}] \\
F(\mathbf{a}, B, q)[11, \{i, i\}] &= q^{\mathbf{a}_1 + \cdots + \mathbf{a}_{i-1}} F(\mathbf{a} - \nu(i), B, q)[1, \{i\}] \\
F(\mathbf{a}, B, q)[12, \{i, j\}] &= \begin{cases} 0 & j - i \geq 1 \ \& \\ & n - j \geq 1 \\ q^{\mathbf{a}_1 + \cdots + \mathbf{a}_{j-1} - 1} F(\mathbf{a} - \nu(j), B, q)[1, \{i\}] & \text{otherwise} \end{cases}
\end{aligned}$$

Using this recurrence and implementation discussed below, it takes less than a minute on a two-year-old commercial laptop for Maple to compute the first 12 terms of the sequence given by $F(\mathbf{a}, \{123\}, q)$ where $\mathbf{a}$ represents the alphabet vector with 2 copies of each letter $1, \ldots, n$. The first 4 terms are given in Table 1, and more may be viewed at this paper's website on the author's homepage (see Section 5).

n	$F(\nu(112233\ldots nn), \{123\}, q)$
1	1
2	$1 + q + 2\,q^2 + q^3 + q^4$
3	$3\,q^4 + 3\,q^5 + 6\,q^6 + 7\,q^7 + 9\,q^8 + 7\,q^9 + 5\,q^{10} + 2\,q^{11} + q^{12}$
4	$q^8 + q^9 + 4\,q^{10} + 6\,q^{11} + 15\,q^{12} + 18\,q^{13} + 28\,q^{14} + 35\,q^{15} +$ $+ 44\,q^{16} + 47\,q^{17} + 49\,q^{18} + 42\,q^{19} + 31\,q^{20} + 18\,q^{21} +$ $+ 9\,q^{22} + 3\,q^{23} + q^{24}$

Table 1: $F(\mathbf{a}, \{123\}, q)$ for $\mathbf{a} = \nu(112233\ldots nn)$

4.4 Permutations avoiding barred patterns

Pudwell also extended enumeration schemes to count permutations avoiding barred patterns in [13]. Barred pattern avoidance is a variation on classical pattern avoidance, first occuring in the characterization of 2-stack sortable permutations in [20]. Recently barred patterns have appeared in a variety of other contexts, such as in [5, 6, 8, 22]. Define $\bar{S_k} := S_k \times \{0,1\}^k$, where $(\sigma, v) \in \bar{S_k}$ can be written $\sigma_1 \dots \sigma_k$ such that σ_i has a bar over it iff $v_i = 1$. Denote $\underline{\sigma}$ the permutation formed by reducing the subpermutation of σ formed by the unbarred letters, and let $\bar{\sigma}$ denote the permutation formed by σ disregarding bars. A permutation π is said to avoid σ as a barred pattern if every copy of $\underline{\sigma}$ may be extended to form the pattern $\bar{\sigma}$. Put another way, π avoids $\underline{\sigma}$ except as part of $\bar{\sigma}$. As before, we let B be a set of barred patterns and denote $S_n(B)$ the subset of S_n avoiding all patterns in B. For an example $S_n(\{\bar{1}32\})$ is the set of permutations so that any $21 = \text{red}(32)$ pattern is part of a larger 132, i.e., any pair of letters $\pi_i > \pi_j$ for $i < j$ is preceded by some letter less than π_j, so $132 \in S_3(\{\bar{1}32\})$ while $321 \notin S_3(\{\bar{1}32\})$.

These schemes again partition according to prefixes, so define $S_n(B)[p]$ and $S_n(B)[p, C]$ as before. We again meet three possiblies for a given prefix $p \in S_k$:

(1) $S_n(B)[p]$ is $\{p\}$ or $\emptyset$.

(2) For each $C \in \binom{[n]}{k}$, one of the following happens:

(2a) $S_n(B)[p, C] = \emptyset$

(2b) $S_n(B)[p, C]$ is in bijection with some $S_{\hat{n}}(B)[\hat{p}, \hat{C}]$ for $\hat{n} < n$

(3) $S_n(B)[p]$ must be partitioned further, so
$s_n(B)[p] = \sum_{p' \in S_{k+1}(B)[p]} s_n(B)[p']$.

As before, case (1) applies only if p has length n and depends on whether p avoids B. There are additional base cases, however, which are hidden in the statement for case (2). In the case of classical pattern avoidance, if we knew that $S_n(B)[p] = \emptyset$ then it followed that $S_{n+1}(B)[p] = \emptyset$ since the addition of an extra letter could not cause the avoidance of a pattern. For barred patterns, however, that added letter may be exactly the barred letter we require to avoid a barred pattern. Hence for certain small n, it is possible that $S_n(B)[p] = \emptyset$ while $S_{n+k}(B)[p] \neq \emptyset$ for some $k > 0$. These prefixes p are identified by *stop points*, maximal values $s(p)$ such that $S_n(B)[p] = \emptyset$

for any $n \leq s(p)$. Stop points are built into the schemes for barred patterns, which have a structure as a set of 4-tuples $(p, G_p, R_p, s(p))$ instead of triples (p, G_p, R_p). These additional base cases are trivial to incorporate into our refinement, since they imply additional cases when $F(n, B, q)[p] = 0$.

When case (2) applies and stop points are not to blame, the subcase (2a) is indicated by gap vector criteria — their application remains unchanged from the description in section 2.1. Similarly, the bijections in subcase (2b) are performed via the deletion map d_r, so Lemma 3.1 applies as written. The discovery of these gap vectors and reversibly deletable indices meet additional wrinkles as explained in [13], but again the discovery is not our present concern. Hence we have refined enumeration schemes to compute $F(n, B, q)$ for pattern sets B containing barred patterns.

5 Implementation in Maple

To accompany their papers, Zeilberger wrote WILF for [23], Vatter wrote WILFPLUS for [18], and Zeilberger wrote VATTER for [24]. Pudwell continued this tradition, writing mVATTER to accompany the multiset permutation results in [15] and bVATTER to accompany the barred pattern results in [13]. Here we continue that tradition, presenting three suites of additional Maple procedures: qVATTER, qmVATTER, and qbVATTER to implement the refinements above for each of the schemes implemented by VATTER, mVATTER, and bVATTER, respectively. To use qVATTER, first load VATTER (using the 'read' command and directing Maple to the appropriate directory) followed by qVATTER. Similarly qmVATTER requires procedures from mVATTER and qbVATTER requires procedures from bVATTER. Tables 2, 3, and 4 summarize the primary procedures in each package. All files, along with dozens of pre-computed examples of $F(n, B, q)$ and $F(\mathbf{a}, B, q)$, may be downloaded from the author's homepage. Further comments on syntax are presented upon loading the packages successfully.

It should be noted the scheme-construction algorithms implemented follow the algorithms described by Zeilberger in [24] and as a result contain the parameter "GapLimit." To find the minimal gap vectors G_p for $p \in S_k$, the algorithm considers all vectors $g \in \mathbb{Z}^{k+1}$ such that $\sum_i g_i \leq$ GapLimit. Vatter showed this parameter is not strictly necessary, as GapLimit may be set to one less than the length of the longest pattern appearing in B without loss of generality. Setting the GapLimit to this level lengthens computation time significantly and is often unnecessary. For example, all minimal gap vectors g appearing in the scheme for $B = \{12345\}$ have $\sum_i g_i = 1$. This scheme has depth 7, so setting GapLimit=4 takes considerably more computing time. Using an artificially low GapLimit can speed up the runtime at the risk of missing vectors, and so this parameter is included.

qWilf(n,B,q)	A brute-force computation of $F(n, B, q)$. This is based on the procedure Wilf(n,B) in VATTER, which recursively constructs the set $S_n(B)$.
SchemeFast(B, Depth, GapLimit)	Attempts to compute an enumeration scheme for pattern set B, limiting the search to prefixes of length Depth and gap vectors g such that $\sum g_i \leq$ GapLimit. If no such scheme exists, it returns FAIL. This procedure is contained in VATTER.
qMiklos(E, Prefix, GapVector, q)	Computes $F(n, B, q)$[Prefix, C] using the scheme E, where C is the prefix set such that $g(n, C)$=GapVector. To compute $F(n, B, q)$, let Prefix=[] and GapVector=[n].
qSeqS(E, N, q)	Computes $F(n, B, q)$ for $n = 1, 2, \ldots, N$, where E is the scheme for B.

Table 2: Primary procedures in qVATTER

qmWilf($\mathbf{a}$,B,q)	Computes $F(\mathbf{a}, B, q)$ via brute force, based on mWilf($\mathbf{a}$,B) from mVATTER which recursively constructs the set $S_{\mathbf{a}}(B)$.
SchemeF(B, Depth, GapLimit)	Attempts to compute an enumeration scheme for pattern set B, limiting the search to prefixes of length Depth and gap vectors g such that $\sum g_i \leq$ GapLimit. If no such scheme exists, it returns FAIL. This procedure is contained in mVATTER.
qMiklosA(E, Prefix, Remaining,q)	Computes $F(\mathbf{a}, B, q)[p, C]$ where E is the enumeration scheme for B, Prefix is p, and Remaining is the frequency vector for the letters lying outside of the prefix.
qSeqS11(E,r,N,q)	Computes the sequence $F(\mathbf{a}, B, q)$ where $\mathbf{a}$ is the alphabet vector for r copies of each letter $1, \ldots, n$ as n ranges from 1 to N.
qMiklosTot(E,k,n,q)	Computes $\sum\limits_{w \in [k]^n(B)} q^{\mathrm{inv}(w)}$, the distribution over all words avoiding B of length n with letters at most k, where E is the enumeration scheme for B.
qSeqSkn(E,k,n,q)	Computes the sequence given by qMiklosTot(E,k,N,q) for $n = 1, 2, \ldots, N$.

Table 3: Primary procedures in qmVATTER

qWilf(n,B,q)	A brute-force computation of $F(n,B,q)$. This is based on the procedure Wilf(n,B) in bVATTER, which recursively constructs the set $S_n(B)$.
SchemeFast(B, Depth, GapLimit)	Attempts to compute an enumeration scheme for pattern set B, limiting the search to prefixes of length Depth and gap vectors g such that $\sum g_i \leq$ GapLimit. If no such scheme exists, it returns FAIL. This procedure is contained in bVATTER.
qMiklos(E, Prefix, GapVector, q)	Computes $F(n,B,q)[\text{Prefix},C]$ using the scheme E, where C is the prefix set such that $g(n,C)$=GapVector. To compute $F(n,B,q)$, let Prefix=[] and GapVector=[n].
qSeqS(E, N, q)	Computes $F(n,B,q)$ for $n=1,2,\dots,N$, where E is the scheme for B.

Table 4: Primary procedures in qbVATTER

6 Applications

6.1 Exploring $S_n(B)$

As an example of how schemes can help us explore $S_n(B)$ and form conjectures, we consider the family of patterns $B_k = \{312, k(k-1)\dots 21\}$. Chow and West showed these have generating functions related to Chebyshev polynomials in [9], and Vatter showed in [18] that each of these B_k's has an enumeration scheme of depth 2 and exhibits the scheme in tree form. In our notation, this is the scheme

$$\{(\emptyset,\emptyset,\emptyset),(1,\emptyset,\emptyset),(12,\emptyset,\{1\}),(21,\{\langle 0,1,0\rangle,\langle k-2,0,0\rangle\},\{1\})\}$$

This yields the following family of recurrences for $F(n,B_k,q)[p,C]$:

$$\begin{aligned} F(n,B_k,q) &= \sum_{i=1}^{n} F(n,B_k,q)[1,\{i\}] \\ F(n,B_k,q)[1,\{i\}] &= \sum_{j=1}^{i-1} F(n,B_k,q)[21,\{i>j\}] + \\ &\quad \sum_{j=i+1}^{n} F(n,B_k,q)[12,\{i<j\}] \\ F(n,B_k,q)[12,\{i<j\}] &= q^{i-1}F(n-1,B_k,q)[1,\{j-1\}] \end{aligned}$$

$$F(n, B_k, q)[21, \{i > j\}] = \begin{cases} 0, & \text{if } i - j > 1 \text{ or} \\ & j > k-2 \\ q^j F(n-1, B_k, q)[1, \{i-1\}], & \text{otherwise} \end{cases}$$

It can be seen from [9] that the generating functions

$$f_k(x) = \sum_{x \geq 0} s_n(B_k) x^n$$

satisfy the relation

$$f_k(x) = \frac{1}{1 - x\, f_{k-1}(x)}.$$

A similiar relationship holds for $f_k(x,q) = \sum_{x\geq 0} F(n, B_k, q)x^n$. With a combination of the above recurrences and the Maple package gfun, a suite of tools to work with generating functions, we may quickly conjecture the generating functions $f_k(x,q)$ for small k. After a bit of tinkering and luck we may conjecture the following proposition, which when combined with the obvious initial condition $f_1(x,q) = 1$ allows us to compute $f_k(x,q)$ for any k.

Proposition 6.1 *If* $f_k(x,q) = \sum_{n\geq 0} F(n, B_k, q)x^n$, *then*

$$f_k(x,q) = \frac{1}{1 - x\, f_{k-1}(qx, q)}.$$

Proof. We will follow the method of weight enumerators, as used in [16]. Define the weight function $W_{(x,q)}(\pi) = x^{\ell(\pi)} q^{\mathrm{inv}(\pi)}$, where $\ell(\pi)$ represents the length of π. Note $W_{(x,q)}(\epsilon) = 1$. We will weight-count $S_\infty(B_k) := \bigcup_{n\geq 0} S_n(B_k)$, where

$$W_{(x,q)}(S_\infty(B_k)) := \sum_{\pi \in S_\infty(B_k)} W(\pi).$$

Observe $f_k(x,q) = W_{(x,q)}(S_\infty(B_k))$. We wish to prove that $W_{(x,q)}(S_\infty(B_k)) = 1 + x\, W_{(qx,q)}(S_\infty(B_{k-1})) \cdot W_{(x,q)}(S_\infty(B_k))$.

Any $\pi \in S_\infty(B_k)$ of positive length must contain a 1, say $\pi_i = 1$, and so we may decompose π into $\pi^{(1)}\, 1\, \pi^{(2)}$, where $\mathrm{red}(\pi^{(1)}) \in S_\infty(B_{k-1})$ and $\mathrm{red}(\pi^{(2)}) \in S_\infty(B_k)$. Clearly $\ell(\pi) = \ell(\pi^{(1)}) + \ell(\pi^{(2)}) + 1$. Now π avoids 312, so we know that every letter in $\pi^{(1)}$ lies below every letter in $\pi^{(2)}$ otherwise the $\pi_i = 1$ would play the role of "1" in a 312 pattern. From this we determine that $\text{inv}(\pi) = \text{inv}(\pi^{(1)}) + \text{inv}(\pi^{(2)}) + \ell(\pi^{(1)})$, where the last term is the number of inversions involving the 1 of π. Combining these observations we get:

$$\begin{aligned} W_{(x,q)}(\pi) &= W_{(x,q)}(\pi^{(1)}\,1\,\pi^{(2)}) \\ &= x^{\ell(\pi^{(1)}\,1\,\pi^{(2)})} q^{\text{inv}(\pi^{(1)}\,1\,\pi^{(2)})} \\ &= x\, x^{\ell(\pi^{(1)})}\, x^{\ell(\pi^{(2)})}\, q^{\text{inv}(\pi^{(1)})}\, q^{\text{inv}(\pi^{(2)})}\, q^{\ell(\pi^{(1)})} \\ &= x\, W_{(qx,x)}(\pi^{(1)})\, W_{(q,x)}(\pi^{(2)}) \end{aligned}$$

Summing over all nonempty $\pi \in S_\infty(B_k)$, we get the relation

$$W_{(x,q)}(S_\infty(B_k)) - 1 = xW_{(qx,q)}(S_\infty(B_{k-1})) \cdot W_{(x,q)}(S_\infty(B_k)),$$

from which our proposition follows. □

6.2 Symmetry Applications

We may also consider symmetry questions which were studied by Simion and Schmidt in [17]. Define $E_n(B)$ to be the set of even permutations avoiding B and $O_n(B)$ to be the set of odd permutations avoiding B. Let $e_n(B) = |E_n(B)|$ and $o_n(B) = |O_n(B)|$. Then since permutation π is even if and only if $\text{inv}(\pi)$ is even, we see that $F(n, B, -1) = e_n - o_n$. Of course if we have already gone through the trouble to find the scheme, we can just as easily compute $e_n = (F(n, B, -1) + s_n(B))/2$ and $o_n = (F(n, B, -1) - s_n(B))/2$.

We call a set of patterns B *evenly-split* if $e_n(B) = o_n(B)$ for all $n \geq 2$, i.e., $F(n, B, -1) = 0$. An obvious necessary condition is that $|B \cap E_k| = |B \cap O_k|$ for all k, since otherwise $e_k(B) \neq o_k(B)$. This is not sufficient, however, as shown by $B = \{1234, 1324\}$ where $e_7(B) = o_7(B) - 1 = 918$ (data for $n \leq 10$ do suggest that $e_{2n}(B) = o_{2n}(B)$). From exploring various pairs $\{\sigma, \tau\} \subseteq S_4$, we were able to observe and then prove the following theorem.

THEOREM 6.2 *Let $\sigma \in S_k$ and $\hat{\sigma}$ be the σ with the first two letters transposed, i.e. $\sigma_2\sigma_1\sigma_3\ldots\sigma_k$. Then $B = \{\sigma, \hat{\sigma}\}$ is evenly-split.*

Proof. Consider the map transposing the first two letters of a permutation, $\tau : \pi_1\pi_2\ldots\pi_n \mapsto \pi_2\pi_1\ldots\pi_n$. This is an involution $E_n \to O_n$ which we will see preserves the B-avoiding property. Suppose otherwise so that for $\pi \in S_n(B)$, $\hat{\pi} = \tau(\pi)$ contains σ or $\hat{\sigma}$. If $\hat{\pi}_{i_1}\hat{\pi}_{i_2}\ldots\hat{\pi}_{i_k}$ is a copy of σ, then the first two letters of $\hat{\pi}$ must be involved in an copy of σ or else π would contain

σ as well. If both are involved in a copy of σ, however, then $\pi_{i_1}\pi_{i_2}\ldots\pi_{i_k}$ is a copy of $\hat{\sigma}$. Hence τ preserves B-avoidance and reverses sign. □

We of course have similar symmetric criteria by tranposing the *last* two letters, the *smallest* two letters, and the *largest* two letters. This theorem is not a full characterization, as there are B of size 4 which are evenly-split (for example, $\{123, 132, 213, 231\}$). Even among pairs, $e_n(\{1234, 1432\}) = o_n(\{1234, 1432\})$ for $2 \leq n \leq 20$ and so it appears $\{1234, 1432\}$ is evenly-split without satisfying this criterion (nor any of its symmetries). It should also be said that transposing two adjacent letters in the middle of a permutation does not lead to an evenly-split pair, as demonstrated above by $\{1234, 1324\}$.

We close this section with a remark regarding multiset permutations avoiding $B = \{321, 312\}$. These have the enumeration scheme

$$\{(\epsilon, \emptyset, \emptyset), (1, \{\langle 2, 0\rangle\}, \{1\})\}.$$

This implies

$$F(\mathbf{a}, B, q) = F(\mathbf{a} - \nu(1), B, q) + q^{\mathbf{a}_1} F(\mathbf{a} - \nu(2), B, q). \tag{7}$$

However, for $w \in S_{\langle a_1, a_2, a_3 \ldots\rangle}$, let $\phi(w)$ be the word in $S_{\langle a_2, a_1, a_3 \ldots\rangle}$ obtained by changing each 1 in w to a 2 and each 2 to a 1. When we restrict the domain to B-avoiding words, observe that ϕ is a bijective involution $S_{\langle a_1, a_2, a_3 \ldots\rangle}(B) \leftrightarrow S_{\langle a_2, a_1, a_3 \ldots\rangle}(B)$. If 321 did appear in $\phi(w)$, then if "21" were played by a literal 21 in $\phi(w)$ then w must have contained a corresponding 312. If only the "1" of a 321 were played by a 1, then the "2" must be played by some letter of $\phi(w)$ which is at least 3, so a 2 may also play the "1". If the "1" were played by a 2, then obviously it may be played by a 1 instead. The analogous arguments apply should 312 appear in $\phi(w)$ as well. Hence w avoids B if and only if $\phi(w)$ avoids B.

Next consider the effect of ϕ on $\mathrm{inv}(w)$. Since only the 1's and 2's change places, it can easily be seen that $\mathrm{inv}(\phi(w)) - \mathrm{inv}(w) = a_1a_2 - 2\mathrm{inv}_{12}(w)$ where $\mathrm{inv}_{12}(w)$ denotes the number of pairs $i < j$ such that $w_i = 2$ and $w_j = 1$. It follows that

$$F(\langle a_1, a_2, a_3, \ldots\rangle, B, -1) = (-1)^{a_1a_2} F(\langle a_2, a_1, a_3, \ldots\rangle, B, -1)$$

We return to the recurrence in (7) and specialize to $q = -1$, $\mathbf{a}_1 = \mathbf{a}_2 = a$

to see:

$$\begin{aligned}F(\langle a,a,a_3,\ldots\rangle,B,-1) =&F(\langle a-1,a,a_3,\ldots\rangle,B,-1)+\\ &(-1)^a F(\langle a,a-1,a_3,\ldots\rangle,B,-1)\\ =&(-1)^{(a-1)a}F(\langle a,a-1,a_3,\ldots\rangle,B,-1)+\\ &(-1)^a F(\langle a,a-1,a_3,\ldots\rangle,B,-1)\\ =&(1+(-1)^a)F(\langle a,a-1,a_3,\ldots\rangle,B,-1)\end{aligned}$$

Hence we get the following proposition:

PROPOSITION 6.3 *If* $\mathbf{a}_1 = \mathbf{a}_2$ *are odd then* $F(\mathbf{a}, \{321, 312\}, -1) = 0$.

7 Conclusions and Future Work

In [14] Pudwell used a similar notion of enumeration schemes to count words avoiding patterns with repeated letters. These schemes, however, partition the $S_{\mathbf{a}}(B)$ according to the locations of copies of the letters 1 through k rather than the first k letters. Further, the schemes hinge on a bijection other than d_R, instead deleting all copies of a certain letter. Hence Lemma 3.1 is inapplicable. The similarity in process, however, indicates a refinement according to inv is still possible.

One application not mentioned in the previous section is exploring asymptotic distributions of inv over pattern-avoiding sets. It is well known, see for example [12], that inv is asymptotically normal over all of S_n. With the tools above, one may now explore the asymptotic distributions of inv over a set of restricted permutations $S_n(B)$. With a fast way to compute $F(n, B, q)$, the techniques of [25] could apply very nicely.

These refinements are subject to the same pitfalls as the classical case, and may be used to compute $F(n, B, q)$ only when there exist classical schemes to compute $s_n(B)$ or $s_{\mathbf{a}}(B)$. For example, of the 7 trivial equivalence classes of single patterns of length 4, only two have known finite enumeration schemes. Hence there is work to be done to extend the notion of schemes to handle additional pattern sets B.

Different refinements may also be possible. For example, one may consider different permutation statistics, say the the major index $\mathrm{maj}(\pi) := \sum_{i:\pi_i>\pi_{i+1}} i$, and then compute $\sum_{\pi\in S_n(B)} q^{\mathrm{maj}(\pi)}$. Perhaps most suitable to these methods would be the function $(c-ba)(\pi) = |\{(i,j) : i < j, \pi_i > \pi_j > \pi_{j+1}\}|$, or any of the other dashed pattern functions with a single dash, or combinations of such functions, as presented in [1]. These may be considered a restricted form of inv, where we count only certain of the pairs $i < j$ and $\pi_i > \pi_j$. The methods above do not apply as simply as the additional $q^{\delta_r(p)}$ factor, but the ideas of schemes should certainly be fruitful in computing the distributions of these other statistics over $S_n(B)$.

8 Acknowledgements

The author would like to thank Lara Pudwell and Vince Vatter for helpful discussions, as well as the anonymous referees. The author also thanks his advisor Doron Zeilberger for suggesting this avenue of inquiry, as well as the organizers of Permutation Patterns 2009 for giving him the chance to present this work.

References

[1] E. Babson and E. Steingrímsson, *Generalized permutation patterns and a classification of the Mahonian statistics*, Sém. Lothar. Combin., 44 (2000) Article B44b (18 pp.).

[2] J. Bandlow, E. Egge and K. Killpatrick, *A weight-preserving bijection between Schröder paths and Schröder permutations*, Ann. Comb., 6 (2002) 235–248.

[3] J. Bandlow and K. Killpatrick, *An area-to-inv bijection between Dyck paths and 312-avoiding permutations*, Electron. J. Combin., 8 (2001) Article #R40 (16 pp).

[4] E. Barcucci, A. Del Lungo, E. Pergola and R. Pinzani, *Some permutations with forbidden subsequences and their inversion number*, Discrete Math., 234 (2001) 1–15.

[5] M. Bousquet-Mélou and S. Butler, *Forest-like permutations*, Ann. Comb., 11 (2007) 335–354.

[6] M. Bousquet-Mélou, A. Claesson, M. Dukes and S. Kitaev, *(2+2)-free posets, ascent sequences and pattern avoiding permutations*, J. Combin. Theory Ser. A, 117 (2010) 884–909.

[7] A. Burstein, *Enumeration of words with forbidden patterns*, University of Pennsylvania, 1998.

[8] A. Burstein and I. Lankham, *Combinatorics of patience sorting piles*, Sém. Lothar. Combin., 54A (2005-2007) Article B54Ab (19 pp.).

[9] T. Chow and J. West, *Forbidden subsequences and Chebyshev polynomials*, Discrete Math., 204 (1999) 119–128.

[10] W. Y. C. Chen, Y. P. Deng and L. L. M. Yang, *Motzkin paths and reduced decompositions for permutations with forbidden patterns*, Electron. J. Combin., 9 (2002/2003) Article #R15 (13 pp).

[11] S. E. Cheng, S. P. Eu and T. S. Fu, *Area of Catalan paths on a checkerboard*, European J. Combin., 28 (2007) 1331–1344.

[12] W. Feller, *An Introduction to Probability Theory and Its Applications*, John Wiley and sons, 1957.

[13] L. Pudwell, *Enumeration schemes for permutations avoiding barred patterns*, Electron. J. Combin., 17 (2010) Article #R29 (27 pp).

[14] L. Pudwell, *Enumeration schemes for words avoiding patterns with repeated letters*, Integers, 8 (2008) Article A40 (19 pp).

[15] L. Pudwell, *Enumeration schemes for words avoiding permutations*, London Mathematical Society Lecture Note Series, 376, Cambridge University Press, 2010, 193–211.

[16] A. Robertson, H. S. Wilf and D. Zeilberger, *Permutation patterns and continued fractions*, Electron. J. Combin., 6 (1999) Article #R38 (6 pp).

[17] R. Simion and F. W. Schmidt, *Restricted permutations*, European J. Combin., 6 (1985) 383–406.

[18] V. Vatter, *Enumeration schemes for restricted permutations*, Combin. Probab. Comput., 17 (2008) 137–159.

[19] J. West, *Generating trees and the Catalan and Schröder numbers*, Discrete Math., 146 (1995) 247–262.

[20] J. West, *Permutations with forbidden subsequences and stack-sortable permutations*, MIT, 1990.

[21] H. S. Wilf, *What is an answer?*, Amer. Math. Monthly, 89 (1982) 289–292.

[22] A. Woo and A. Yong, *When is a Schubert variety Gorenstein?*, Adv. Math., 207 (2006) 205–220.

[23] D. Zeilberger, *Enumeration schemes and, more importantly, their automatic generation*, Ann. Comb., 2 (1998) 185–195.

[24] D. Zeilberger, *On Vince Vatter's brilliant extension of Doron Zeilberger's enumeration schemes for Herb Wilf's classes*, Published in The Personal Journal of Ekhad and Zeilberger, 2006.

[25] D. Zeilberger, *The Automatic Central Limit Theorems Generator (and Much More!)*, Advances in Combinatorial Mathematics: Proceedings of the Waterloo Workshop in Computer Algebra 2008 in honor of Georgy P. Egorychev, Springer-Verlag, 2009, 165–174.