ENUMERATION SCHEMES FOR PATTERN-AVOIDING WORDS AND PERMUTATIONS

BY LARA KRISTIN PUDWELL

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Mathematics

Written under the direction of

Doron Zeilberger

and approved by

New Brunswick, New Jersey

May, 2008

ABSTRACT OF THE DISSERTATION

Enumeration Schemes for Pattern-Avoiding Words and Permutations

by Lara Kristin Pudwell

Dissertation Director: Doron Zeilberger

Let $p = p_1 \cdots p_n \in S_n$ and $q = q_1 \cdots q_m \in S_m$. We say that p contains q as a pattern if there are indices $1 \leq i_1 < \cdots < i_m \leq n$ such that $p_{i_j} < p_{i_k} \iff q_i < q_k$; otherwise, p avoids q. The study of pattern avoidance in permutations is well studied from a variety of perspectives. This thesis is concerned with two generalizations of this pattern avoidance problem. The first generalization is that of pattern avoidance in words (where p and q may have repeated letters). The second is that of barred permutation patterns (where p avoids q unless q is part of an instance of an even larger pattern in p). In both cases, we seek to find universal methods that count words (resp. permutations) avoiding a particular set of patterns, and automate these methods to achieve new enumeration results.

Acknowledgements

I am incredibly grateful to my advisor, Doron Zeilberger, for his guidance and encouragement. His influence on me has been tremendous, and I could not ask for a better dissertation mentor.

I am indebted to Neil Sloane for numerous insightful comments and suggestions which have improved this work significantly. I am also grateful to József Beck and Vladimir Retakh for their valuable time and assistance. I very much appreciate each of their contributions to this thesis.

I owe many thanks to fellow graduate students Andrew Baxter, Emilie Hogan, Elizabeth Kupin, and Eric Rowland for their careful reading, comments, and patience during the writing process.

Finally, I am thankful to my parents, Rick and Kris Pudwell, and my brother Ryan for their unending support.

This work would not have been possible without you all.

Table of Contents

List of Tables

List of Figures

Chapter 1

A Survey of Related Literature

In this thesis, we use several techniques to study pattern avoidance in words and barred pattern avoidance in permutations. Both of these topics are less well-understood generalizations of the notion of pattern avoidance in permutations. In this chapter, we seek to set an appropriate context by recalling standard results about pattern avoidance in permutations. We also recall the ground-breaking results of Regev and Burstein for pattern-avoiding words, and the results of Callan for barred pattern avoidance. Finally, as this thesis is concerned with *universal* techniques for enumerating these pattern-avoiding objects, the chapter concludes with a summary of the four major universal techniques for pattern avoidance in permutations.

1.1 Pattern Avoidance in Permutations

Definition 1. *Given a string of numbers $s = s_1 \cdots s_n$, the reduction of s, denoted $red(s)$, is the string obtained by replacing the ith smallest letter(s) of s with i.*

For example, $red(15487864) = 13265642$ because in 15487864 the smallest letter is 1, the second smallest letter is 4, the third smallest letter is 5, etc., and so in the reduction 4 is replaced with 2, 5 is replaced with 3, etc. This definition gives rise to a generalized notion of one string being contained in another string, namely:

Definition 2. *Given strings s of length n and t of length m, we say that s* contains *t as a pattern if there there exist indices $1 \leq i_1 < \cdots < 1_m \leq n$ such that $red(s_{i_1} \cdots s_{i_m}) = t$. Otherwise, s* avoids *t as a pattern.*

This notion is well studied for the case where both s and t are permutations.

It is a straightforward exercise to fix a permutation s and list all permutations t that are contained in s (or similarly that s avoids). However, it is a much more interesting question to study the following object:

$$S_n(Q) = \{\pi \in S_n \mid \pi \text{ avoids } q \text{ for all } q \in Q\}.$$

More specifically, we are interested in finding a closed form formula, generating function, or recurrence enumerating $|S_n(Q)|$ for various sets of patterns Q.

The ground-breaking paper in the area of permutation patterns is that of Simion and Schmidt in 1985 [31], and in the past 25 years this research area has become increasingly popular. However, while the terminology is relatively new, the ideas behind pattern avoidance have been around much longer. For example, consider the following well-known result of Erdős and Szekeres:

Theorem 1. *(Erdős and Szekeres, 1935 [16]) Given $a, b \in \mathbb{N}$, $n = ab + 1$ and $x = x_1, \ldots, x_n$ a sequence of n real numbers. Then x either contains a monotonically increasing subsequence of length $a + 1$ or a monotonic decreasing subsequence of length $b + 1$.*

Rephrased in our terminology, this theorem says that

$$|S_n(\{1 \cdots (a+1), (b+1) \cdots 1\})| = 0 \text{ if } n \geq ab + 1.$$

Before we return to further enumeration results, we first discuss some particularly celebrated asymptotic results for the pattern avoidance problem.

1.1.1 Asymptotic Results

While we are primarily interested in finding $|S_n(Q)|$ exactly, a number of asymptotic results exist. The most famous of these is the Stanley–Wilf Conjecture, which supposed that

Conjecture 1. *(Stanley and Wilf, 1980 [5]) Given a permutation pattern q, there exists a constant c_q such that for all positive integers n, we have*

$$|S_n(\{q\})| \leq c_q^n.$$

While an exponential bound may still seem rather large, this is a significant bound compared to the total number of permutations of length n, which grows as $n!$.

The Stanley–Wilf Conjecture is now known to be true, although no direct proof is known. In 2000, Martin Klazar showed that the Füredi–Hajnal Conjecture implies the Stanley–Wilf Conjecture [21]. The Füredi–Hajnal Conjecture, which deals with pattern avoidance in 0-1 matrices, states:

Conjecture 2. *(Füredi and Hajnal, 1992 [18]) Let P be a permutation matrix and let $f(n, P)$ be the maximum number of 1s in an $n \times n$ P-avoiding 0-1 matrix. Then there exists a constant c_P so that*

$$f(n, P) \leq c_P n.$$

The Füredi–Hajnal Conjecture was proved in a remarkably beautiful and simple way by Marcus and Tardos in 2004 [26].

The study of Stanley–Wilf limits (the optimal constant c_q given in the Stanley–Wilf Conjecture) is currently receiving much attention. Notably, the constant c_q constructed in the proof of the conjecture is generally a severe over-estimate and can be much improved. The first optimal result in this area is that of Regev for monotone patterns:

Theorem 2. *(Regev, 1981 [30]) For all k and n, $|S_n(\{12\cdots(k+1)\})|$ asymptotically equals $\lambda_k \dfrac{(k)^{2n}}{n^{(k^2-1)/2}}$, where $\lambda_k = \dfrac{k^{k^2/2}\prod_{j=1}^{k-1} j!}{(2\pi)^{(k-1)/2}2^{(k^2-1)/2}}$.*

We now turn our attention to two key applications of pattern avoidance before considering explicit enumeration results for various sets $S_n(Q)$.

1.1.2 Sorting

Donald Knuth [23] first generated interest in the study of pattern avoidance in permutations by demonstrating a connection to sorting through an arrangement of stacks and queues. In particular, the set of permutations that can be sorted through a system of stacks and queues may be characterized in terms of pattern avoidance.

As a reminder, a stack is a first-in last-out data structure, and a queue is a last-in last-out data structure. In either case, one may always add or remove precisely one element to the stack or queue. The type of structure tells us what element we are allowed to remove.

We say a permutation π is sortable through a system of stacks and queues if we can obtain the identity permutation, $12\cdots n$, by passing π through the system of stacks and queues exactly once, making decisions in such a way as to not make the permutation more disorderly. See Figure 1.1 below for the example of sorting the permutation 132 through a single stack:

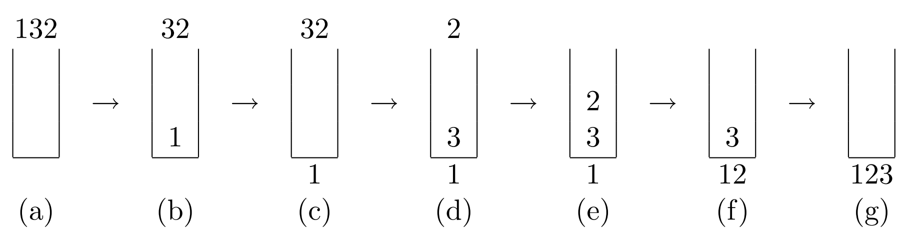


Figure 1.1: Sorting the permutation 132 through a single stack

Note that from state (b) to state (c) we have purposely taken 1 out of the stack before inserting 3, otherwise, we would create an inversion (3 before 1) that was not in the original permutation. Similarly between state (d) and state (e) we have put 2 in the stack before removing 3, otherwise we would introduce an inversion (3 before 2) that was not present before. Sorting through a queue is similar, except numbers are removed from the opposite end of the queue than they were inserted.

Now, as a first simple characterization, the only permutation that is sortable through a single queue is the identity permutation.

On the other hand, in the case of stacks:

Theorem 3. *(Knuth, 1973 [23]) A permutation is sortable through a single stack if and only if it avoids the pattern* 231*.*

For the case of parallel queues or stacks, we have the following results:

Theorem 4. *(Tarjan, 1972 [34]) A permutation is sortable through l parallel queues if and only if it avoids the pattern $(l+1)l\cdots 1$.*

Theorem 5. *(Tarjan, 1972 [34]) A permutation is sortable through l parallel stacks if and only if it avoids the pattern* $23\cdots(l+2)1$.

Clearly, the language of pattern avoidance is very powerful for these characterizations. We will return to sorting applications when we discuss barred patterns in section 1.3.

1.1.3 Schubert varieties

In the 19th century, H. Schubert developed the techniques of enumerative calculus to find the number of points, lines, planes, etc. that obey certain intersection properties and other geometric conditions. His original work, while far-reaching and extremely beautiful, was largely based on intuition rather than rigor. Thus, Hilbert's 15th problem called for building a rigorous foundation for Schubert's enumerative calculus [22]. One object that appears in Schubert calculus the Schubert variety.

Recall that the Grassmannian $G(n,k)$ is the set of all k-dimensional subspaces of an n-dimensional vector space over a field K. Schubert varieties are a class of subvarieties of the Grassmannian. More precisely, given integers $1 \leq a_1 < \cdots < a_k \leq n$, the Schubert variety $\Omega(a_1,\ldots,a_k)$ is the collection of all points of $G(n,k)$ representing k-dimensional subspaces W with the property $\dim_K(W \cap \langle e_{n-a_i+1},\ldots,e_n\rangle) \geq i$ for $1 \leq i \leq m$, where $e_1,\ldots e_n$ are the standard basis vectors of n-dimensional space.

Schubert varieties are one of the most-studied classes of singular algebraic varieties, and it is useful to be able to say whether a Schubert variety is singular or smooth. The most relevant result for our discussion is the following:

Theorem 6. *(Lakshmibai and Sandhya, 1990 [24]) Let B be a Borel subgroup of $SL_n(\mathbb{C})$, w an element of the symmetric group S_n, and X_w the Schubert variety associated to w in the flag manifold $SL_n(\mathbb{C})/B$. Then X_w is smooth if and only if w avoids the patterns* 4231 *and* 3412.

While delving further into the language of projective algebraic geometry is beyond the scope of this introduction, this theorem serves to illustrate yet another application with connections to our definition of pattern avoidance. This notion will also be revisited when we discuss barred patterns in section 1.3.

1.1.4 Symmetries

We may think of a permutation in S_n as a function from $[n] = \{1, \ldots, n\}$ to $[n]$. For example, we may graph the permutation 14253 as in Figure 1.2.

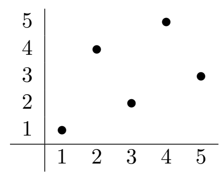


Figure 1.2: The graph of the permutation 14253

Since the graph of a permutation is necessarily on an $n \times n$ square we may use the natural symmetries of the square to determine some useful relationships between various sets $S_n(Q)$.

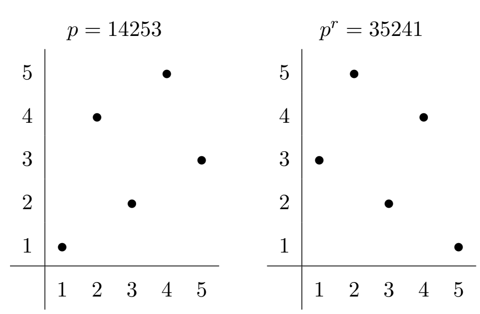
We consider three symmetries that are natural both for the square and in the language of permutations.

Definition 3. *Let $p = p_1 \cdots p_n \in S_n$. Then:*

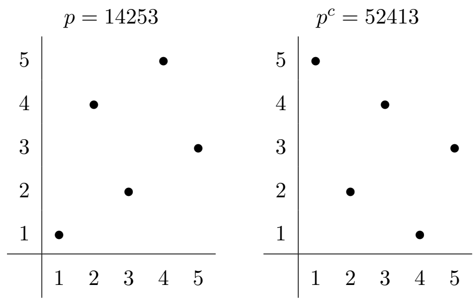
- $p^r = p_n \cdots p_1$ *(reversal)*
- $p^c = p_1^c \cdots p_n^c$ *where* $p_i^c = n + 1 - p_i$ *for* $1 \leq i \leq n$ *(complement)*
- $p^{-1} = p_1^{-1} \cdots p_n^{-1}$ *where* $p_i^{-1} = j$ *if and only if* $p_j = i$ *(inverse)*

We note that:

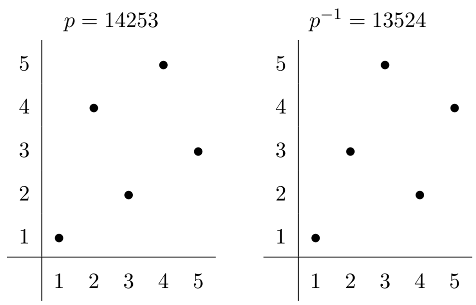
- Reversal corresponds to flipping the graph of p over the vertical line of symmetry:



- Complement corresponds to flipping the graph of p over the horizontal line of symmetry:



- Inverse corresponds to flipping the graph of p over the main diagonal line of symmetry:



These lead to the natural symmetries:

$$\begin{aligned} p \text{ avoids } q &\iff p^r \text{ avoids } q^r, \\ &\iff p^c \text{ avoids } q^c, \\ &\iff p^{-1} \text{ avoids } q^{-1}, \end{aligned}$$

and moreover $|S_n(Q)| = |S_n(Q^r)| = |S_n(Q^c)| = \left|S_n(Q^{-1})\right|$, where Q^* is the set obtained by applying the operation $*$ to all patterns in the set Q. By repeatedly applying the operations of reverse, complement, and inverse, which generate the symmetries of the square, we see that we can partition sets of patterns into equivalence classes up to size 8 that will necessarily have the same enumeration. Two pattern sets Q and Q' that yield the same sequence $\{|S_n(Q)|\}_{n\geq 0}$ are said to be *Wilf-equivalent.*

There are other relations besides those given by the symmetries of the square that give Wilf-equivalent classes of patterns. For example:

Theorem 7. *(West, 1990 [37]) Let* $A = a_3 \cdots a_l$ *be a permutation of* $3 \cdots l$. *Then* $|S_n(\{12A\})| = |S_n(\{21A\})|$ *for all* $n \geq 0$, *i.e.* $\{12A\}$ *and* $\{21A\}$ *are Wilf-equivalent.*

Further, West conjectured (this was later proved on a case-by-case basis for each of $t = 2$, $t = 3$, and $t \geq 4$):

Theorem 8. *Let* $I_t = 12 \cdots t$, $J_t = t(t-1) \cdots 1$, *and let* $A = a_{t+1} \cdots a_l$ *be any permutation of* $(t+1) \cdots l$. *Then* $|S_n(\{I_t A\})| = |S_n(\{J_t A\})|$ *for all* $n \geq 0$, *i.e.* $\{I_t A\}$ *and* $\{J_t A\}$ *are Wilf-equivalent.*

1.1.5 Enumeration Results

The symmetries and equivalences of the previous section will guide us as we seek to comprehensively enumerate Q-avoiding permutations for various Q.

We first consider results for when Q contains exactly one pattern:

- Length 1:
  $|S_n(\{1\})| = 0$ for $n \geq 1$ since any permutation with at least one letter contains a 1 pattern.

- Length 2:
  $|S_n(\{12\})| = 1$ for $n \geq 1$, which counts the strictly decreasing permutations. Also, since $21 = 12^r$ we have $|S_n(\{21\})| = 1$.

- Length 3:
  From the symmetries of the square, we have $|S_n(\{123\})| = |S_n(\{321\})|$ and $|S_n(\{132\})| = |S_n(\{231\})| = |S_n(\{213\})| = |S_n(\{312\})|$. Simion and Schmidt [31] provided a bijection between $\{123\}$-avoiding permutations and $\{132\}$-avoiding permutations, and moreover showed that $|S_n(\{\pi_3\})| = C_n = \frac{\binom{2n}{n}}{n+1}$ where π_3 is any permutation of length 3, and C_n denotes the nth Catalan number.

For length 4 and greater, less comprehensive results are known. From the symmetries of the square and other non-trivial equivalences, patterns of length 4 fall into three distinct categories, which are typically represented by the patterns 1342, 1234, and 1324.

For each of these we have the following:

Theorem 9. *(Cori, Jacquard, and Schaeffer, 1997 [13])*

$$\sum_{n\geq 0} |S_n(\{1342\})|\, x^n = \frac{32x}{-8x^2+20x+1-(1-8x)^{3/2}},$$

and thus

$$|S_n(\{1342\})| = (-1)^{n-1}\frac{7n^2-3n-2}{2} + 3\sum_{i=2}^{n}(-1)^{n-i}2^{i+1}\frac{(2i-4)!}{i!(i-2)!}\binom{n-i+2}{2} \sim 8^n$$

This result corresponds to sequence A022558 in the Online Encyclopedia of Integer Sequences. It is significant because 1342 is one of only two permutations of length > 3 for which there is an *exact* formula for $|S_n(\{q\})|$. The other permutation is:

Theorem 10. *(Gessel, 1990 [19])*

$$|S_n(\{1234\})| = \frac{1}{(n+1)^2(n+2)}\sum_{k=0}^{n}\binom{2k}{k}\binom{n+1}{k+1}\binom{n+2}{k+1} \sim 9^n$$

This formula corresponds to sequence A117158. In 2006, Elizalde found a remarkable exponential generating function for these permutations:

Theorem 11. *(Elizalde, 2006 [15])*

$$\sum_{n\geq 0} |S_n(\{1234\})| \frac{x^n}{n!} = \frac{2}{\cos x - \sin x + e^{-x}}$$

For the remaining class of $\{1324\}$-avoiding permutations, Marinov and Radoičić [27] found a recurrence based on the generating tree for this permutation class, and have used it to compute $|S_n(\{1324\})|$ for $n \leq 20$, given in A113228. No exact enumeration or generating function is known; however, it can be shown that $|S_n(\{1324\})|$ is asymptotically strictly greater than 9^n. There are no known exact enumeration results for $S_n(\{q\})$ where q has length > 4.

For permutations avoiding more than one pattern, much work has been done; many cases can be solved using the method of finitely labeled generating trees or the method of enumeration schemes, both discussed in section 1.4. Vatter has programmed both of these techniques in `Maple`, and the results can be found in an online webbook [14].

Now that we have exhausted enumeration results for pattern-avoiding permutations, we continue with a a generalization of this concept that is the focus of Chapters 2, 3, and 4.

1.2 Pattern Avoidance in Words

1.2.1 Definitions

We maintain the same definitions of reduction and containment as for permutations, but are now interested in words, where a word is a string of n letters chosen from the alphabet $[k] = \{1, \dots k\}$. The key difference is that while a permutation of length n uses *all* letters from 1 to n without repetition, a word may have repeated letters, and may not necessarily use all letters in the alphabet. We write $W_n(k, Q)$ for the set of words in $[k]^n$ avoiding pattern set Q. Clearly, $|W_n(k, \{\})| = k^n$. For example, the complete set of words in $[2]^3$ is $W_3(2, \{\}) = \{111, 112, 121, 211, 122, 212, 221, 222\}$.

We may be interested in sets of words with a specific number of occurrences of each letter. For example, the set of words with one 1, two 2s, and one 3 is {1322, 1232, 1223, 2132, 2123, 2213, 3122, 3212, 3221, 2312, 2321, 2231}. Given fixed alphabet size k, an *alphabet vector* or *frequency vector* is a vector v of length k of non-negative integers. Given a word $w \in [k]^n$, w has frequency vector $[a_1, \dots a_k]$ where a_i is the number of is in w for $1 \leq i \leq k$. We denote the weight of a frequency vector v by $||v|| = \sum_{i=0}^{n} a_i$. If $w \in [k]^n$ and w has frequency vector v, then $||v|| = n$.

Now, the key object with which we are concerned is:

Definition 4.

$$A_{[a_1,\dots,a_k]}(Q) := \left\{ w \in [k]^{\sum a_i} \;\middle|\; \begin{array}{l} w \textit{ has } a_i \textit{ i's}, \\ w \textit{ avoids } q \textit{ for every } q \in Q \end{array} \right\}.$$

We have two immediate observations about $A_{[a_1,\dots,a_k]}(Q)$:

- If $a_i = 0$, then $\left|A_{[a_1,\dots,a_k]}(Q)\right| = \left|A_{[a_1,\dots,a_{i-1},a_{i+1},\dots,a_k]}(Q)\right|$.
- $\left|A_{[\underbrace{1,\dots,1}_{n}]}(Q)\right| = |S_n(Q)|$.

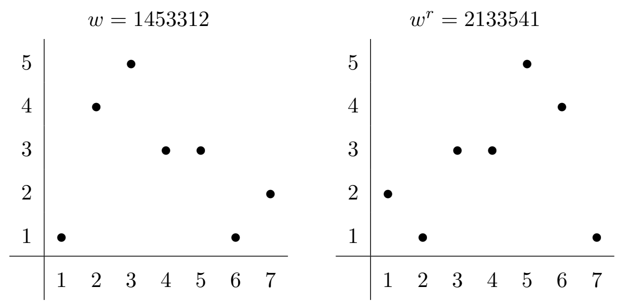
By the first observation, we may assume that $a_i > 0$ for $1 \leq i \leq k$. By the second observation, we see that pattern avoidance in permutations is merely a special case of pattern avoidance in words.

1.2.2 Symmetries

The symmetries of reverse and complement for permutations generalize to pattern avoidance in words if we apply the appropriate operations to their alphabet vectors as well.

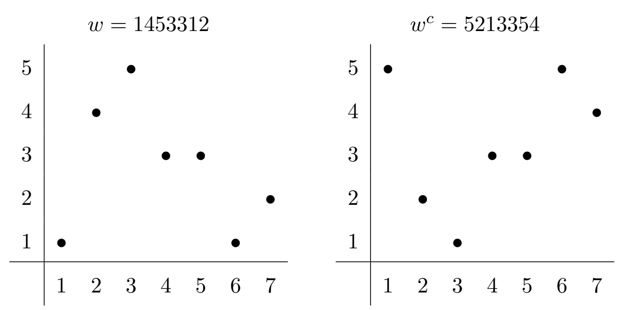
For example:

- Reverse corresponds to flipping the graph of w over the vertical line of symmetry:



  That is if w has alphabet vector $[a_1, \ldots, a_k]$ and avoids q, then w^r has alphabet vector $[a_1, \ldots, a_k]$ and avoids q^r.

- Complement corresponds to flipping the graph of w over the horizontal line of symmetry:



  That is if w has alphabet vector $[a_1, \ldots, a_k]$ and avoids q, then w^c has alphabet vector $[a_k, \ldots, a_1]$ and avoids q^c.

However, since words may have repeated letters, the graph of a word is no longer a square but a rectangle, and we lose the symmetry of inverses. Thus, the trivial equivalence classes given by the symmetries of the rectangle partition pattern sets into equivalence classes of size at most 4.

1.2.3 Asymptotic and Enumeration Results

The problem of pattern avoidance in words is more general than the problem of pattern avoidance in permutations, and thus more difficult; however some results are known.

As for permutations, Regev has computed asymptotics for the number of words avoiding the monotone pattern, more precisely:

Theorem 12. *(Regev, 1981 [30])*

$$|W_n(k, \{12\cdots(l+1)\})| \sim \frac{0!1!2!\cdots(l-1)!}{(k-l)!\cdots(k-1)!}\left(\frac{1}{l}\right)^{l(k-l)} n^{l(k-l)} l^n$$

We focus on generating function results for $W_n(k, Q)$ in Chapter 2. For words with specified alphabet vectors, more is known. In his Ph.D. Thesis, Burstein [7] uses generating function techniques to find the number of words avoiding any set of permutation patterns of length 3. He also discusses various additional symmetries beyond those of reverse and complement. In 2001, Albert, Aldred, Atkinson, Handley, and Holton [2] reformulated the results for patterns of length 3 as exact formulas rather than generating functions. Beyond these results little is known about words avoiding permutations. Finding recurrences for words avoiding permutations is the focus of Chapter 3.

For the case of words avoiding patterns with repeated letters, little is known beyond the work of Burstein and Mansour [6] for patterns with at most two distinct letters, and of Heubach and Mansour [20] for words avoiding patterns of length 3. These words are the focus of Chapter 4.

1.3 Barred Pattern Avoidance in Permutations

1.3.1 Definitions

Finally, we consider another generalization of the notion of pattern avoidance in permutations.

Let $q' \in S_m$, $b \in \{0,1\}^m$. The *barred permutation* q is the permutation obtained by copying the entries of q' and putting a bar over q'_i if and only if $b_i = 1$. Write $\overline{S_m}$ for the set of all barred permutations of length m. For example, the complete set of barred permutations of length 2 is $\{12, \bar{1}2, 1\bar{2}, \overline{12}, 21, \bar{2}1, 2\bar{1}, \overline{21}\}$.

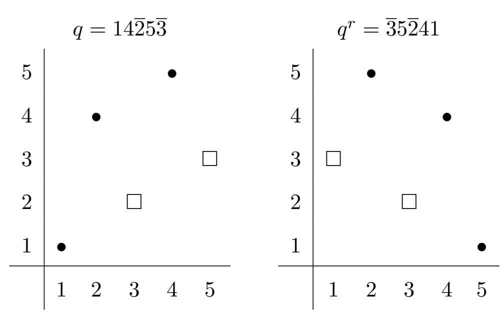
Let $p \in S_n$, $q \in \overline{S_m}$. Given q, let $\overline{q}$ be the permutation formed by ignoring the bars in q, and let $\underline{q}$ be the permutation formed by deleting all barred letters of q and reducing the remaining (unbarred) letters. We say p *contains* q as a barred pattern if every instance of $\underline{q}$ in p is part of an instance of $\overline{q}$ in p. In this case, we say every instance of $\underline{q}$ *extends to* an instance of $\overline{q}$. For example if $q = \bar{1}32$, we have $\overline{q} = 132$ and $\underline{q} = red(32) = 21$. p avoids q if and only if every decreasing pair of numbers in p has a smaller number preceding them.

Most notably, if *every* letter of $q \in \overline{S_n}$ is barred (that is $\underline{q} = \emptyset$), then $S_n(\{q\})$ is the set of permutations *containing* q, since avoiding q means every instance of the empty subpermutation extends to a copy of q. Therefore, in some sense, barred pattern avoidance bridges the gap from pattern avoidance to pattern containment, with a number of intermediate cases.

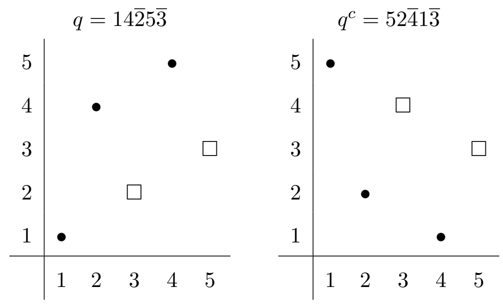
1.3.2 Symmetries

The same symmetries of the square apply to barred patterns with the caveat that bars are moved in the appropriate way. In the graphs below, we will use $\bullet$ to denote an unbarred element and $\square$ to denote a barred element.

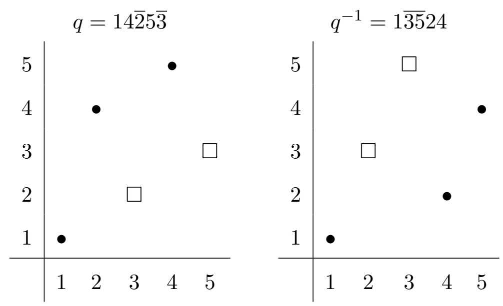
- Reversal corresponds to flipping the graph of q over the vertical line of symmetry:



- Complement corresponds to flipping the graph of q over the horizontal line of symmetry:



- Inverse corresponds to flipping the graph of q over the main diagonal line of symmetry:



By applying the natural operations to the bars in a pattern as well, we maintain the relationship

$$|S_n(Q)| = |S_n(Q^r)| = |S_n(Q^c)| = \left|S_n(Q^{-1})\right|$$

for barred patterns just as for unbarred patterns.

1.3.3 Applications

Barred pattern avoidance also appears in several interesting applications.

Recall that the number of permutations that can be sorted through a single stack is characterized in terms of pattern avoidance.

Consider the permutations that are not sortable by passing through a stack once, but can be sorted by passing through the stack a second time. These permutations are called $2-$stack sortable. For example 231 is not stack sortable because sorting it once through a stack yields 213. However 213 is stack sortable, so 231 is $2-$stack sortable. We have the following characterization:

Theorem 13. *(West, 1990 [37]) A permutation is 2-stack sortable if and only if it avoids* 2341 *and* $3\bar{5}241$*.*

The number of 2-stack sortable permutations was first computed by Doron Zeilberger, who solved a degree-9 functional equation to obtain:

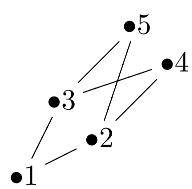
Theorem 14. *(Zeilberger, 1992 [41]) The number of 2-stack sortable permutations of length n is* $\frac{2(3n)!}{(n+1)!(2n+1)!)}$.

This formula produces OEIS sequence A000139. A characterization of k-stack sortable permutations for $k \geq 3$ does not yet exist.

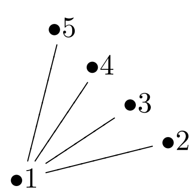
Beyond stack sortability, barred pattern avoidance characterizes at least two other combinatorial objects.

Consider a permutation $p = p_1 \cdots p_n \in S_n$. We can draw G_p, the combinatorial graph of p, on vertex set $[n]$ where $i \sim j$ if and only if both (i) $i < j$ and $p_i < p_j$ and (ii) there is no $i < k < j$ with $p_i < p_k < p_j$.

For example the graph G_p of $p = 13254$ is



p is called *forest-like* if G_p is a forest, that is if G_p has no cycles. Since there is a cycle between the vertices $3 \to 4 \to 2 \to 5 \to 3$, we see that 13254 is *not* forest-like, but the permutation $p = 15432$ is, since its graph G_p is:



Theorem 15. *(Butler and Bousquet-Melou, 2006 [9]) A permutation is forest-like if and only if it avoids the patterns* 1324 *and* $21\bar{3}54$. *Moreover, the generating function for these permutations is*

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

The number of such permutations is given in OEIS sequence A111053.

The pattern set $\{1324, 21\bar{3}54\}$ is also interesting because it is part of another useful characterization for Schubert varieties.

A variety is *locally factorial* if the local ring at every point is a unique factorization domain. Otherwise stated, a Schubert variety is locally factorial if it is smooth except at a closed set of points that has particularly nice properties. Again, our intent is not to delve deeply into the language of algebraic geometry but to illustrate connections with pattern avoidance. We have:

Theorem 16. *(Woo and Yong, 2006 [40]) A Schubert variety X_w is locally factorial if and only if w avoids the patterns* 1324 *and* $21\bar{3}54$.

1.3.4 Enumeration Results

Beyond the special cases of barred pattern avoidance relevant to the above applications, little is known beyond the work of Callan. He has completely enumerated permutations avoiding a single pattern of length 4 with one bar [10], and has dealt with the special case of $\{3\bar{5}241\}$-avoiding permutations [11]. In Chapter 5, we present a universal technique for counting permutations that avoid barred patterns and give the first comprehensive study of permutations that avoid barred patterns of length ≤ 5 with any number of bars.

1.4 Universal Enumeration Techniques for Pattern-Avoiding Permutations

Typically, by nature of the problem, techniques for the enumeration of pattern-avoiding objects depend on the pattern(s) being avoided. This leads to many beautiful results that give intuition about particular forbidden patterns, but that are usually not readily generalizable. As a result, several attempts have been made to compute the size of classes of pattern-avoiding permutations using techniques that are independent of the set of forbidden patterns. These universal techniques are finitely labeled generating trees, insertion encoding, substitution decomposition, and enumeration schemes. The sequel will primarily be concerned with this last method of enumeration schemes.

1.4.1 Finitely Labeled Generating Trees

The technique of generating trees has been used to enumerate a number of combinatorial objects. It was introduced for the enumeration of pattern-avoiding permutations by Chung, Graham, Hoggatt, and Kleiman [12], used extensively by West [39, 38], and automated by Vatter [35].

According to Vatter, "A generating tree is a rooted, labeled tree such that the labels of the children of each node are determined by the label of that node. Thus, a generating tree may be given by specifying the label of the root and a set of *succession rules*."

For example, a complete ternary tree may be given by:

$$\mathit{Root} : (1)$$

$$\mathit{Rule} : (1) \rightsquigarrow (1)(1)(1).$$

Notice that the above generating tree uses only one label, yet it is an infinite tree.

Given a set of forbidden patterns Q, we also consider a *pattern avoidance tree* $T(Q)$ that has nodes $\{S_n(Q)\}_{n\geq 1}$ where $\pi \in S_n(Q)$ is a child of $\pi' \in S_{n-1}(Q)$ if π can be obtained by inserting n into π'.

For example, the first 4 levels of $T(\{123, 231\})$ are given in Figure 1.3.

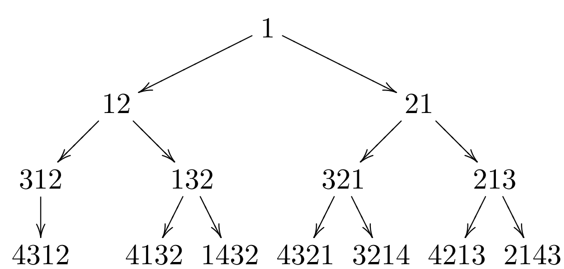


Figure 1.3: The pattern avoidance tree $T(\{123, 231\})$

Given a pattern avoidance tree $T(Q)$, our goal is to find an isomorphic generating tree with finitely many labels. Once we have found such a finitely labeled generating tree, there are standard techniques, such as the transfer matrix method [33], for computing the generating function for Q-avoiding permutations. Further, it is guaranteed to be a rational function. Vatter showed:

Theorem 17. *(Vatter, 2006 [35]) Let Q be a finite set of patterns. The pattern avoidance tree $T(Q)$ is isomorphic to a finitely labeled generating tree if and only if Q contains both a child of an increasing permutation and a child of a decreasing permutation.*

Further, he exhibited an algorithm that is guaranteed to succeed whenever the above condition is satisfied. He observes that his algorithm for finitely labeled generating trees is a special case of the method of enumeration schemes, described later in this chapter; however, the technique of finitely labeled generating trees guarantees the easy computation of a rational generating function, whereas enumeration schemes do not.

For the case of pattern-avoiding words, Bernini, Ferrari, and Pinzani [4] have made use of generating trees to enumerate words avoiding specific classes of patterns of length 3, but nothing is yet known for more general cases.

1.4.2 Insertion Encoding

The method of insertion encoding was introduced by Albert, Linton, and Ruškuc [3]. Generally, insertion encoding builds on the machinery of regular languages.

The key idea is that every permutation can be generated from the empty permutation by successive insertions of a new maximum element. This observation was already used to construct generating trees in the previous section, but will now be used in a new way. During the construction of a permutation, we may insert the maximum at any of a number of *active sites*, denoted by $\diamond$. Moreover, upon inserting a new maximum letter into an active site, it may do precisely one of the following:

- Fill the active site, i.e. replace $\diamond$ with n.
- Be inserted to the left of the active site, i.e. replace $\diamond$ with $n\diamond$.
- Be inserted to the right of the active site, i.e. replace $\diamond$ with $\diamond n$.
- Be inserted in the middle of the active site, i.e. replace $\diamond$ with $\diamond n \diamond$.

Further, we allow only active sites where a new maximum element *will* eventually be inserted. This gives a unique construction of any permutation.

For example, the permutation 152436 is generated in the following way:

$$\diamond$$

$$1\diamond$$

$$1\diamond 2\diamond$$

$$1\diamond 2\diamond 3\diamond$$

$$1\diamond 243\diamond$$

$$15243\diamond$$

$$152436$$

If the insertion encoding of a class of pattern-avoiding permutations forms a regular language, then there are standard techniques to compute the rational generating function that enumerates elements the class. The following definition is necessary for the characterization of those permutation classes which have regular insertion encodings:

A *vertical alternation* is a permutation π of length $2n$ such that either

$$\pi(1), \pi(3), \ldots, \pi(2n-1) < \pi(2), \pi(4), \ldots, \pi(2n)$$

or

$$\pi(1), \pi(3), \ldots, \pi(2n-1) > \pi(2), \pi(4), \ldots, \pi(2n).$$

Finally, we have:

Theorem 18. *(Albert, Linton, and Ruškuc, 2005 [3]) The insertion encoding of $S_n(Q)$, where Q is a finite set of patterns, forms a regular language if and only if the class contains only finitely many vertical alternations.*

Notice that this theorem considers vertical alternations in the entire set $\{S_n(Q)\}_{n\geq 1}$ rather than just in the pattern set Q.

Brändén and Mansour [6] have demonstrated a method using automata to count pattern-avoiding words. Their results, however, require fixing the alphabet size a priori.

1.4.3 Substitution Decomposition

A block or interval of a permutation π is a contiguous sequence of indices $[a, b]$ such that the images $\{\pi(i) : i \in [a, b]\}$ are also contiguous. For example, the permutation 28635417 has a block composed of 6354. Every permutation has n trivial blocks of length 1, and one trivial block of length n (itself). A permutation π is simple if these are the only blocks of π.

Non-simple pattern-avoiding permutations admit a clear recursive structure (by insertion of smaller pattern-avoiding permutations as blocks of larger ones), which leads to the following theorem:

Theorem 19. *(Albert and Atkinson, 2005 [1]) A permutation class with only finitely many simple permutations has an algebraic generating function.*

The first standard example of substitution decomposition is for $\{132\}$-avoiding permutations. By determining the location i of the maximum element of π, we note that all elements before position i must be strictly larger than all elements after position i, giving the recursive structure: $|S_n(\{132\})| = \sum_{i=1}^{n} |S_{i-1}(\{132\})| \cdot |S_{n-i}(\{132\})|$, which is satisfied by the Catalan numbers.

1.4.4 Enumeration Schemes

Finally, we consider the notion of enumeration schemes which will be extended throughout the following chapters. Generally, an enumeration scheme is a recurrence developed using a divide-and-conquer algorithm. Rather than looking for appropriate notation to describe and organize pattern-avoiding permutations as in the previous techniques, enumeration schemes introduce new sets and objects into consideration to accomplish more difficult enumerations.

Enumeration schemes for pattern-avoiding permutations were first introduced by Zeilberger [42], but then improved and reformulated by Vatter [36]. Later, Zeilberger rewrote and programmed Vatter's improved enumeration schemes in his original terminology [43].

Zeilberger's original enumeration schemes rely on two key definitions: refinement and reversibly deletable elements. Vatter later introduced the notion of gap vectors, which greatly increased the success rate of the schemes. He also took advantage of one of the symmetries of the square to reformulate Zeilberger's notation. Below, we introduce each of these three concepts in Zeilberger's terminology.

Since the set $S_n(Q)$ may be complicated, we partition it into disjoint subsets and seek to find recurrences between these subsets. Then, if $S_n(Q) = \cup_{i=0}^{n} A_i$ we have $|S_n(Q)| = \sum_{i=0}^{n} |A_i|$.

Zeilberger partitions $S_n(Q)$ based on the pattern formed by the initial letters of the permutations in the set. The reduction of the initial l letters of a permutation is called its prefix. For example, 14325 has prefixes 1, 12, 132, 1432, 14325. We have:

$$S_n\left(Q; p_1 \cdots p_l\right) := \quad \{p \in S_n \mid p \text{ avoids } Q,\ p \text{ has prefix } p_1 \cdots p_l\}.$$

Sometimes it is necessary to specify not only the pattern formed by the initial letters of a permutation, but also explicitly which letters form that prefix. Thus we also have:

$$S_n\left(Q; \begin{matrix} p_1 \cdots p_l \\ i_1 \cdots i_l \end{matrix}\right) := \left\{ p \in S_n \;\middle|\; \begin{array}{l} p \text{ avoids } Q, \\ p \text{ has prefix } p_1 \cdots p_l, \text{ and} \\ i_1, \ldots, i_l \text{ are the first } l \text{ letters of } p \end{array} \right\}.$$

We observe that $S_n(Q) = S_n(Q; 1) = S_n(Q; 12) \cup S_n(Q; 21)$, etc.

Now that we have a way to partition $S_n(Q)$, we explore ways to find recurrences between the resulting subsets.

Definition 5. *Given a prefix $p = p_1 \cdots p_t$, position r is* reversibly deletable *if every possible bad pattern involving p_r implies the existence of another bad pattern without p_r.*

For example, if $Q = \{123\}$ and $p = 21$, we have that p_1 is reversibly deletable. The only way to involve p_1 in a bad 123 pattern is if we have $p_1 < a < b$, where a and b are letters appearing later in the 123-containing permutation. But prefix 21 means that $p_1 > p_2$, so $p_1 < a < b$ as well. That is, every bad pattern involving p_1 implies the existence of a forbidden 123 pattern without p_1.

There is always a natural embedding

$$S_n \begin{pmatrix} p_1 \cdots p_l \\ i_1 \cdots i_l \end{pmatrix} \to S_{n-1} \begin{pmatrix} p_1 \cdots \hat{p_r} \cdots p_l \\ i_1 \cdots \hat{i_r} \cdots i_l \end{pmatrix},$$

where $\hat{i}$ means to delete i. Namely, delete the rth letter of each permutation and reduce.

However, if p_r is reversibly deletable, and the role of p_r is played by letter j, then

$$\left| S_n \begin{pmatrix} p_1 \cdots p_l \\ i_1 \cdots i_l \end{pmatrix} \right| = \left| S_{n-1} \begin{pmatrix} p_1 \cdots \hat{p_r} \cdots p_l \\ i_1 \cdots \hat{i_r} \cdots i_l \end{pmatrix} \right|,$$

thus giving us the desired recurrence.

Our current definitions give the foundation of a method to derive recurrences for pattern-avoiding permutations, but are often not enough. For example, when $Q = \{123\}$, the prefix 12 does not have a reversibly deletable element.

Thus we introduce the notion of gap vectors. These vectors will greatly reduce the number of cases we must check to decide if a letter is reversibly deletable, and increase the success rate of our schemes.

Definition 6. *Given a prefix p of length l, a spacing vector is a vector in $\mathbb{N}^{l+1}$.*

Given a set of forbidden patterns Q, prefix p and spacing vector v, let $s_1 \cdots s_l$ be the permutation obtained by sorting p. Then, $S_n(Q; p; v)$ is the set of permutations of length n, avoiding Q, beginning with prefix p, and with at least v_1 letters smaller than s_1, v_j letters that are greater than s_{j-1} and smaller than s_j, and v_{l+1} letters that are greater than s_l.

We say that v is a gap vector *for (Q, p) if $|S_n(Q; p; v)| = 0$ for all $n \geq l$.*

For example, $v = \langle 0, 0, 1 \rangle$ is a gap vector for $Q = \{123\}$, $p = 12$. Sorting p, we get $s_1 s_2 = 12$. Now $v_3 = 1$ implies that there is a letter a after the prefix that is larger than both letters in the prefix. However $12a$ forms a forbidden 123 pattern, so there are no permutations of length n avoiding Q with prefix p and spacing v. Thus, we know that if a permutation avoids $\{123\}$ and begins with an increasing pair of letters, the second letter must necessarily be the largest letter of the permutation. In our earlier notation, that is:

$$\left| S_n\left(\{123\}\,;\begin{matrix}1\ 2\\ \\ i\ j\end{matrix}\right)\right| = 0 \text{ for } j < n$$

$$\left| S_n\left(\{123\}\,;\begin{matrix}1\ 2\\ \\ i\ n\end{matrix}\right)\right| = \left| S_{n-1}\left(\{123\}\,;\begin{matrix}1\\ \\ i\end{matrix}\right)\right|.$$

Now we have all the tools necessary for the main definition of this section:

Definition 7. *An* enumeration scheme *is a set of triples* $[p_i, R_i, G_i]$ *such that for each triple:*

- p_i *is a reduced prefix of length* n*.*
- R_i *a subset of* $\{1,\ldots,n\}$*, the reversibly deletable elements of* p_i*.*
- G_i *is a set of vectors of length* $n+1$*, the set of gap vectors corresponding to* (Q, p_i)*.*

  and

- *Either* R_i *is non-empty or all refinements of* p_i *are also in the scheme.*

The final condition guarantees that the enumeration scheme encodes a recurrence counting the elements of $S_n(Q)$. If R_i is empty, then we must further divide the set $S_n(Q;p_i)$, in our divide-and-conquer algorithm. If R_i is non-empty, then we have a recurrence to count the elements of the set $S_n(Q;p_i)$.

For the case of $\{123\}$-avoiding permutations, we have seen that for $p = 21$, p_1 is reversibly deletable, and for $p = 12$, we have gap vector $\langle 0,0,1\rangle$ which gives that p_2 is reversibly deletable. So the corresponding enumeration scheme for $Q = \{123\}$ is:

$$\{[\emptyset,\emptyset,\emptyset],[1,\emptyset,\emptyset],[21,\{1\},\emptyset],[12,\{2\},\{\langle 0,0,1\rangle\}]\}$$

From this, we get the recurrence:

$$\begin{aligned}|S_n(Q)| &= \sum_{i=1}^{n}\left| S_n\left(Q;\begin{matrix}1\\ \\ i\end{matrix}\right)\right| \\ &= \sum_{i=1}^{n}\left(\sum_{h=1}^{i-1}\left| S_n\left(Q;\begin{matrix}2 & 1\\ \\ i & h\end{matrix}\right)\right| + \sum_{j=i+1}^{n}\left| S_n\left(Q;\begin{matrix}1 & 2\\ \\ i & j\end{matrix}\right)\right|\right)\end{aligned}$$

$$= \sum_{i=1}^{n} \left(\sum_{h=1}^{i-1} \left| S_{n-1} \left(Q; \begin{matrix} 1 \\ h \end{matrix} \right) \right| + \left| S_n \left(Q; \begin{matrix} 1 & 2 \\ i & n \end{matrix} \right) \right| \right)$$

$$= \sum_{i=1}^{n} \left(\sum_{h=1}^{i-1} \left| S_{n-1} \left(Q; \begin{matrix} 1 \\ h \end{matrix} \right) \right| + \left| S_{n-1} \left(Q; \begin{matrix} 1 \\ i \end{matrix} \right) \right| \right)$$

$$= \sum_{i=1}^{n} \left(\sum_{h=1}^{i} \left| S_{n-1} \left(Q; \begin{matrix} 1 \\ h \end{matrix} \right) \right| \right)$$

From this, we can compute

$$\left| S_n \left(Q; \begin{matrix} 1 \\ i \end{matrix} \right) \right| - \left| S_n \left(Q; \begin{matrix} 1 \\ i-1 \end{matrix} \right) \right| = \left| S_{n-1} \left(Q; \begin{matrix} 1 \\ i \end{matrix} \right) \right|,$$

and together with the initial conditions of

$$\left| S_n \left(Q; \begin{matrix} 1 \\ 1 \end{matrix} \right) \right| = 1 \text{ and } \left| S_n \left(Q; \begin{matrix} 1 \\ n+1 \end{matrix} \right) \right| = 0,$$

we get the unique solution $\left| S_n \left(Q; \begin{matrix} 1 \\ i \end{matrix} \right) \right| = \binom{n+i-2}{n-1} - \binom{n+i-2}{n}$, which in turn gives $|S_n(Q)| = \frac{\binom{2n}{n}}{n+1}$, as desired.

This example of a scheme for $\{123\}$-avoiding permutations is fairly simple, and there are other methods that will solve it. In general, however, enumeration schemes can handle more complex pattern-avoiding situations and give results in cases where no other method works. In many cases, enumeration schemes can find a recurrence to compute the size permutation classes for which there is no easily computable closed form or generating function enumeration.

It is this notion of enumeration schemes that will be extended in Chapters 3, 4, and 5 for the cases of words avoiding permutations, words avoiding patterns with repeated letters, and permutations avoiding barred patterns, respectively.

Chapter 2

Generating Functions for Pattern-Avoiding Words

In this chapter, I describe an algorithm to find the generating function for the number of pattern-avoiding words in an alphabet of fixed size, and give a bijective proof of the generating function for $\{123, 132\}$-avoiding words.

2.1 Definitions

As in Chapter 1, let $[k]^n$ denote the set of words of length n in the alphabet $\{1, \ldots, k\}$, and let $w = w_1 \cdots w_n \in [k]^n$. The *reduction* of w, denoted by $red(w)$, is the unique word of length n obtained by replacing the i^{th} smallest entries of w with i for each i.

For $w \in [k]^n$ and $q \in [k]^m$, we say that w *contains* q if there exist $1 \leq i_1 < i_2 < \cdots < i_m \leq n$ so that $red(w_{i_1} \cdots w_{i_m}) = q$. Otherwise w *avoids* q.

It is an easy exercise to fix a word w and list all patterns of length m that w avoids. However, it is a much more difficult question to fix a pattern (or set of patterns) Q and enumerate all words avoiding Q. In later chapters, we will be interested in counting pattern-avoiding words with specific alphabet distributions, but in this chapter we focus on a simpler task: Finding a generating function for the number of Q-avoiding words of length n in the alphabet $\{1, \ldots, k\}$.

Most of the work already done in this area depends on the pattern(s) to be avoided. That is, typically, one chooses a pattern set, and then uses the structure of those forbidden patterns to compute the number of permutations or words avoiding Q. However, we would like to accomplish this in a way that is programmable and independent of the pattern set Q.

2.2 Automated Generating Functions

First, we consider several simple cases. Let $f_{k,Q}$ be the generating function for words in the alphabet $[k] = \{1, \ldots, k\}$ avoiding pattern set Q. That is, if $W_n(k, Q)$ is the set of words of length n in the alphabet $[k]$ which avoid Q, then $f_{k,Q}(x) = \sum_{n \geq 0} |W_n(k, Q)|\, x^n$. We have:

1. If $1 \in Q$, then $f_{k,Q}(x) = 1$ (only the empty word avoids the pattern 1).
2. If $Q = \{\}$, then $f_{k,Q}(x) = \frac{1}{1-kx}$ (the generating function for *all* words in $[k]^n$, $n \geq 0$).
3. Let m be the minimum number of letters used in a pattern in Q. If $k < m$, then $f_{k,Q}(x) = \frac{1}{1-kx}$ (That is, if k is smaller than the minimum number of letters required to form a bad pattern, then *all* words in the alphabet $[k]$ avoid Q).

Now that these cases have been taken care of, we move to the more complex situation that Q is a non-trivial set of patterns and k is sufficiently large that a forbidden pattern may occur in words in the alphabet $[k]$.

Our work is made simpler with the following observation: If we have a fixed alphabet size k, there are a finite number of sets of letters that can compose the forbidden pattern q. Namely, if q uses m letters, there are $\binom{k}{m}$ different choices of letters to compose q.

For example, in the alphabet [4], we can create the pattern 121 in 6 ways: 121, 131, 141, 232, 242, and 343.

So, we consider the particular generating function $g_{[k],S}(x)$, which counts all words in the alphabet $[k] = \{1, \ldots, k\}$ avoiding the specific strings in S. We have $f_{k,Q}(x) = g_{[k],S}(x)$ where S is the set of all strings in $[k]$ which form patterns in Q. Observe that $g_{\{1,\ldots,i-1,i+1,\ldots,k\},S}(x) = g_{[k-1],S^*}(x)$ by decreasing all letters greater than i in the alphabet $\{1, \ldots, i-1, i+1, \ldots, k\}$ and in the strings of S.

Let $del(S, a)$ be the set of strings s' such that either (i) $s' \in S$ and s' does not begin with a, or (ii) $as' \in S$. So, $|del(S, a)| = |S|$. We now have the following easy observation:

Proposition 1.

$$g_{[k],S}(x) = 1 + x \sum_{a \in [k]} g_{[k],del(S,a)}(x)$$

Proof. It suffices to note that 1 counts the empty permutation, and each term in the sum counts Q-avoiding permutations beginning with the letter a. □

Our earlier observations will now come into play.

- By (1), if $del(S,a)$ contains a string of length 1, say l, we may remove that letter from the alphabet and then remove that string from $del(S,a)$, and compute instead $g_{[k]\setminus\{l\},del(S,a)\setminus\{l\}}(x)$. Let L be the set of all such l.
- By (2), if $del(S,a) = \{\}$, we may add $\frac{1}{1-kx}$ in place of $g_{[k],del(S,a)}(x)$.
- By (3), if there exists $q \in del(S,a)$ involving letters not in alphabet $[k] \setminus L$, we may remove q from $del(S,a)$.

Proposition 2. *Repeated application of Proposition 1 terminates after finitely many iterations to give an algebraic equation for $g_{[k],S}(x)$ in terms of itself, x, and k.*

Proof. Notice that after each iteration, we have $g_{[k],S}(x)$ as a sum of itself and terms of type $g_{[k'],S'}(x)$ with $k' < k$ and/or $|S'| < |S|$. Each of these $g_{[k'],S'}(x)$ can be written as a sum of itself and even smaller $g_{[k''],S''}(x)$, until we finally reach a $g_{[k^*],S^*}(x)$ which can be written in terms of x and k as given by the above observations.

Now, we may solve for this smallest $g_{[k^*],S^*}(x)$ and then substitute to find each successively larger $g_{[k'],S'}(x)$ until we have an algebraic equation for $g_{[k],S}(x)$ itself. □

We now have an automated method for finding a generating function for words of length n avoiding pattern set Q in fixed alphabet $[k]$. The Maple code for this process can be found on the author's webpage at:

`http://www.math.rutgers.edu/~lpudwell/maple/new_words`

A parallel observation helps us to find the multivariate generating function for pattern-avoiding words, namely:

Proposition 3.

$$g_{[k],S}(x_1,\ldots,x_k) = 1 + \sum_{a\in[k]} x_a g_{[k],del(S,a)}(x_1,\ldots,x_k)$$

The Maple code for this process can be found at:

`http://www.math.rutgers.edu/~lpudwell/maple/mwords`

2.3 Results

While the multivariate generating function encodes more information, the data given by it for specific k is often more unruly, so we focus on results given by experimentation with the univariate generating function.

The above process for finding generating functions has the advantage that it is *guaranteed* to work regardless of the pattern(s) being avoided. However, even the univariate generating function may become complicated rather quickly. For example, the pattern 1324 is particularly difficult to grapple with. It is the smallest pattern for which there is no closed form formula or generating function for the number of permutations of length n which avoid it. With this automated technique, we see that the generating function for the number of words avoiding $\{1324\}$ on an alphabet of size $k = 1,\ldots,6$ is:

$$f_{[1],\{1324\}} = \frac{1}{1-x}$$

$$f_{[2],\{1324\}} = \frac{1}{1-2x}$$

$$f_{[3],\{1324\}} = \frac{1}{1-3x}$$

$$f_{[4],\{1324\}} = \frac{20x^3-22x^2+8x-1}{(3x-1)^4}$$

$$f_{[5],\{1324\}} = \frac{1274x^7-3185x^6+3410x^5-2030x^4+727x^3-157x^2+19x-1}{(3x-1)^8}$$

$$f_{[6],\{1324\}} = \frac{P(x)}{(3x-1)^{12}}$$

where $P(x) = 84458x^{11} - 327677x^{10} + 578062x^9 - 612281x^8 + 432820x^7 - 214509x^6 + 76104x^5 - 19344x^4 + 3456x^3 - 414x^2 + 30x - 1$.

Although there is a clear conjecture for the denominator of these functions, the numerator grows quickly and in a seemingly unpredictable way. Although this technique does not allow us to count $\{1324\}$-avoiding words with a specific number of occurrences of each letter, it gives us a beginning tool for study of this class of words that is largely not understood.

Sometimes, however, we are able to find a sequence of generating functions for specific alphabet sizes that lends itself well to a more general result. We consider $f_{[k],\{123,132\}}$ with various values of k:

$$f_{[1],\{123,132\}} = \frac{1}{1-x}$$
$$f_{[2],\{123,132\}} = \frac{1}{1-2x}$$
$$f_{[3],\{123,132\}} = \frac{1-2x+2x^2}{(2x-1)^2(x-1)}$$
$$f_{[4],\{123,132\}} = \frac{1-2x+4x^2}{(2x-1)^3}$$
$$f_{[5],\{123,132\}} = \frac{1-4x+12x^2-16x^3+8x^4}{(2x-1)^4(x-1)}$$

Experimentation with these generating functions leads to the conjecture:

$$f_{[k],\{123,132\}} = \frac{1}{2(1-x)(1-2x)^{k-1}} + \frac{1}{2(1-x)}.$$

We confirm this with an explicit bijection in the following section.

2.4 A Bijective Proof

In his thesis [7], Burstein used analytic methods to find the generating function for words on an alphabet of size k which simultaneously avoid the patterns 123 and 132. In this section, I give a new bijective proof of this result via a bijection between $\{123, 132\}$-avoiding permutations, and 2-colored decreasing sequences. There is a similar bijection between $\{231, 213\}$-avoiding words and 2-colored increasing sequences. The composition of these two maps yields a new bijection between the two sets of restricted words which were first shown to be equinumerous by Mansour [25] in 2005. We will show:

Theorem 20. *[7] Let $W_n(k, \{123, 132\})$ be the set of words in $[k]^n$ $(k \geq 0)$ which simultaneously avoid the patterns 123 and 132, and let $F_k(x) = \sum_{n\geq 0} |W_n(k, \{123, 132\})|\, x^n$. Then*

$$F_k(x) = \frac{1}{2(1-x)(1-2x)^{k-1}} + \frac{1}{2(1-x)}.$$

First, notice that for $k \leq 2$, *all* words in $[k]^n$ avoid 123 and 132, so $F_0(x) = 1$, $F_1(x) = \frac{1}{1-x}$, and $F_2(x) = \frac{1}{1-2x}$, which are all of the desired form. Now, we turn our attention to general k, and consider the following proposition.

Proposition 4. *The number of $\{123, 132\}$-avoiding words on $[k]^n$ with at least one k is $\sum_{i=0}^{n-1} \binom{k-2+i}{k-2} 2^i$.*

Proof. We prove the proposition with a bijection.

By the standard "stars and bars" computation, the number of decreasing sequences of length i in the alphabet $[k-1]$ is $\binom{k-2+i}{k-2}$. Thus, $\sum_{i=0}^{n-1} \binom{k-2+i}{k-2} 2^i$ computes the number of 2-colored decreasing sequences in $[k-1]$ of length at most $n-1$.

Now we exhibit a bijection from the set of 2-colored decreasing sequences on $[k-1]$ of length at most $n-1$ to $\{123, 132\}$-avoiding words in $[k]^n$ with at least one k.

Algorithm 1.

Input: $s = (s_1, ..., s_m)$, a 2-colored decreasing sequence in $[k-1]$ of length m $(< n)$.

Output: $w \in [k]^n$, a $\{123, 132\}$-avoiding word with at least one k.

max:=k

w:=k

for i from 1 to m do

  l:=length(w)

  if color(s_i)=R then

    w:=$w_1, ..., w_l, s_i$

  if color(s_i)=L, $s_{i-1} > s_i$, and color(s_{i-1})=R then

    w:=$w_1, ..., w_{l-1}, s_i, w_l$

    max:=s_{i-1}

  if color(s_i)=L and there does not exist $j < i$ so that $s_j = s_i$ and *color(s_j)=R then*

w:=$w_1, ..., w_{l-1}, s_i, w_l$

if color(s_i)=L and there exists $j < i$ so that $s_j = s_i$ and color(s_j)=R then

w:=$w_1, ..., w_{l-1}, max, w_l$

If length(w) $< n$ then

$w := w, max^{n-l}$.

Return w.

Example 1. *In general, the 2-coloring of the sequence s instructs us whether to insert s_i to the right (R) or the left (L) of the last letter of w with a little added bookkeeping. Let $k = 3$ and $n = 5$.*

Input: (2(R),1(L),1(R),1(L))

First, initialize $w = 3$, $max = 3$.

Since $s_1 = 2$ and color(s_1) $= R$, $w = 32$, $max = 3$.

Since $s_2 = 1$, color(s_2) $= L$, $s_1 > s_2$, and color(s_1)=R, $w = 312$, $max = 2$.

Since $s_3 = 1$ and color(s_3) $= R$, $w = 3121$, $max = 2$.

Since $s_4 = 1$, color(s_4) $= L$, and $s_3 = s_4$ with color(s_3)=R, $w = 31221$, $max = 2$.

Since all entries of s have been exhausted and w has length n, w is the desired $\{123, 132\}$-avoiding word on $[3]^5$ with at least one 3.

We will show that Algorithm 1 maps to the correct set after exhibiting its inverse. Appealing to the general intuition that Algorithm 1 places an R-colored (resp. L-colored) entry of a decreasing sequence to the right (resp. left) of the last entry of the word, an inverse map should reverse this process. All letters to the left of the first k are colored with L. After this, the inverse map finds the largest possible uncolored entry to add to the decreasing sequence and labels it R, labeling everything smaller and to the left of this entry with an L. Consider the following:

Algorithm 2.

Input: $w \in [k]^n$, a $\{123, 132\}$-avoiding word with at least one k.

Output: D, a decreasing sequence on $[k-1]$ of length m ($< n$) and C, a sequence of colors of the same length.

C:=the empty word

D:=the empty word

i:=1

while $w_i \neq k$

   D:=D, w_i*,*   *C:=C,L,*   *i:=i+1*

while there are uncolored entries of w

   m:=min(max(uncolored entries of w),last entry of D, k-1).

   curr := the position of the first uncolored occurrence of m.

   D:=D, w_{curr}*,*   *C:=C, R*

   $w_{j_1}, ..., w_{j_c}$ *:= the uncolored elements of w to the left of curr*

   (other than the first occurrence of k)

     for i from 1 to c

       If $w_{j_i} \leq w_{curr}$ *then*

         D:=D, w_j*,*   *C:=C, L*

       If $w_{j_i} > w_{curr}$ *then*

         D:=D, w_{curr}*,*   *C:=C, L*

If all elements of w have been colored, or all remaining entries of w are larger than the last entry of D, then return [D,C].

Example 2. *Let* $k = 3$ *and* $w = 31221$*.*
There are no entries to the left of the first 3, so $m := \min\{\max\{1, 2, 2, 1\}, 2\} = 2$*.*
w_3 *is the first uncolored occurrence of m, and* w_2 *is the only uncolored element besides* $w_1 = k$ *to the left of* w_3*. Thus, D=[2,1], C=[R,L].*
Now, $m := \min\{\max\{2, 1\}, 1, 2\} = 1$*.*
w_5 *is the first uncolored occurrence of m and* w_4 *is the only uncolored element to the left of* w_5*. Thus, D=[2,1,1,1], C=[R,L,R,L].*

We have exhibited an inverse map for Algorithm 1. Now, we check that Algorithm 1 always maps 2-colored decreasing sequences to $\{123, 132\}$-avoiding words with at least one k. It is clear from the algorithm that w contains at least one k.

To see that w is $\{132\}$-avoiding, recall that s is a decreasing sequence and that the algorithm inserts each entry of s into w as either the last, or next to last entry of w. Find the smallest i such that the word formed from $1, \dots s_i$ avoids 132 but insertion of s_{i+1} creates a 132 pattern. Consider the case that s_i and s_{i+1} form the 32 part of a 132 pattern. Then, some entry s_k, $k < i$ must play the role of 1 in the 132 pattern, which contradicts the fact that s is decreasing. Otherwise, s_{i+1} plays the role of the 1 in the 132 pattern, which implies that s_{i+1} was inserted earlier than the next to last element of w, a contradiction.

To see that w is $\{123\}$-avoiding, notice that the only way for two entries of s to be inserted into w in increasing order is if $s_i > s_{i+1}$, and color(s_{i+1}) = L, color(s_i) = R. However, in this case, the algorithm reassigns $max := s_i$, and no entry larger than max is inserted into w during the rest of the algorithm. Thus it is impossible to create a 123 pattern.

Now, it is clear that Algorithms 1 and 2 provide a bijection between the set of 2-colored decreasing sequences on $[k-1]$ of length at most $n-1$ and $\{123, 132\}$-avoiding words on $[k]^n$ with at least one k.

□

From this proposition, we are in a position to prove the following lemma. Then, by induction, we can prove the theorem.

Lemma 1. *[25] Let $W_n^*(k, \{123, 132\})$ be the set of words in $[k]^n$ which simultaneously avoid the patterns 123 and 132 and contain at least one k, and let $G_k(x)$ be the generating function for these words, so $G_k(x) = \sum_{n\geq 0} |W_n^*(k, \{123, 132\})| \, x^n = F_k(x) - F_{k-1}(x)$. Then, $G_k(x) = \frac{x}{(1-x)(1-2x)^{k-1}}$.*

Proof. The coefficient of x^n in $G_k(x)$ is the number of $\{123, 132\}$-avoiding words in $[k]^n$ with at least one k. By the proposition this is $\sum_{i=0}^{n-1} \binom{k-2+i}{k-2} 2^i$.

$$G_k(x) = \sum_{n\geq 0} |W_n^*(k, \{123, 132\})| \, x^n = \sum_{n\geq 0} \sum_{i=0}^{n-1} \binom{k-2+i}{k-2} 2^i x^n$$
$$= \frac{x}{1-x} \sum_{n\geq 0} \binom{k-2+n}{k-2} (2x)^n = \frac{x}{1-x} \frac{1}{(1-2x)^{k-1}} = \frac{x}{(1-x)(1-2x)^{k-1}}$$

□

2.5 Concluding Remarks

Although the generating function for $\{123, 132\}$-avoiding words is especially nice, and leads to a combinatorial enumeration, this process of automatically computing generating functions for specific k has the advantage that it will succeed for *any* pattern set Q, and allow for exploration of words avoiding even the most unruly patterns.

In the sequel, we will consider methods that work for arbitrary alphabet size. While those results are more general, they also work only if the pattern set is sufficiently nice. Thus, this most general technique is in some sense also the most broadly successful to gain an initial perspective on pattern-avoiding words.

Chapter 3

Enumeration Schemes for Words Avoiding Permutations

3.1 Background

In this chapter, we are concerned with finding recurrences to count words with specific alphabet vectors which avoid permutations as patterns. We first recall some key notation from Chapter 1.

A *frequency vector* or *alphabet vector* is a vector $\mathbf{a} = [a_1, \ldots, a_k]$ such that $k \geq 1$ and $a_i \geq 0$ for $1 \leq i \leq k$. Let $\|\mathbf{a}\| = \sum_{i=1}^{k} a_i$. Then, given a frequency vector $\mathbf{a}$ and a set of reduced words Q in $[k]^m$ for some $m > 0$, we define

$$A_{\mathbf{a}}(Q) := \{w \in [k]^{\|\mathbf{a}\|} \mid w \text{ avoids } q \text{ for every } q \in Q, w \text{ has } a_i\ i\text{'s for } 1 \leq i \leq k\}.$$

Notice that if $a_1 = \cdots = a_k = 1$, we reduce to the case of counting pattern-avoiding permutations. Also note that if $a_i = 0$ for some i, then we have $A_{\mathbf{a}}(Q) = A_{\mathbf{a}'}(Q)$, where $\mathbf{a}' = [a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_k]$. Thus, we may assume that $a_i > 0$ for $1 \leq i \leq k$. When the set of patterns Q is clear from context, we may simply write $A_{\mathbf{a}}$.

In Chapter 1, I briefly described Zeilberger's method of prefix enumeration schemes to count pattern-avoiding permutations. This was later improved by Vatter. In this chapter, I extend the notion of Zeilberger and Vatter's schemes to enumerate words avoiding sets of permutations (i.e. the patterns in Q have no repeated letters, but w may), and detail the success rate of this method.

3.2 Refinement

Assume we want to enumerate the elements of a set $A(n)$. If we cannot find a closed form formula for $|A(n)|$, then ideally, we want to find a recurrence only involving n. Unfortunately, this is not always possible.

If we cannot find a direct recurrence, following Zeilberger, we introduce the notion of *refinement* as follows. Decompose $A(n)$ as $A(n) = \bigcup_{i \in I} B(n,i)$, so that the $B(n,i)$ are disjoint. Then, if we can find a recurrence for each $B(n,i)$ in terms of the other $B(n,i)$ and $A(n)$, we have a recursive formula for $A(n)$ as well. If not, then refine each $B(n,i)$ as the disjoint union $B(n,i) = \bigcup_{j \in J} C(n,i,j)$, and repeat.

In the case of words, we will use *reduced prefixes* as our refinement parameter. Let $w \in [k]^n$, $w = w_1 \cdots w_n$. The *i-prefix* of w is $red(w_1 \cdots w_i)$, $1 \le i \le n$. For example, if $w = 152243$, the 1-prefix of w is 1, the 2-prefix of w is 12, the 3-prefix of w is 132, the 4-prefix of w is 1322, the 5-prefix of w is 14223, and the 6-prefix of w is 152243. In particular, the n-prefix of w is $red(w)$.

Now, to allow us to talk about sets, we introduce the following notation:

$$A_{\mathbf{a}}\left(Q; p_1 \cdots p_l\right) := \{w \in [k]^{\|\mathbf{a}\|} \mid w \text{ avoids } Q, w \text{ has } l-\text{prefix } p_1 \cdots p_l\}$$

$$A_{\mathbf{a}}\left(Q; \begin{matrix} p_1 \cdots p_l \\ i_1 \cdots i_l \end{matrix}\right) := \left\{ w \in [k]^{\|\mathbf{a}\|} \;\middle|\; \begin{matrix} w \text{ avoids } Q \\ w \text{ has } l-\text{prefix } p_1 \cdots p_l \\ w = i_1 \cdots i_l w_{l+1} \cdots w_n \end{matrix} \right\}$$

For example, $A_{[1,2,1]}\left(\{\}; 21\right) = \{3122, 3212, 3221, 2123, 2132\}$.

$$A_{[2,1,2]}\left(\{\}; \begin{matrix} 121 \\ 131 \end{matrix}\right) = \{13123, 13132\}.$$

$$A_{[2,1,1,1,1]}\left(\{\}; \begin{matrix} 132 \\ 154 \end{matrix}\right) = \{154123, 154132, 154312, 154321, 15432132, 154231\}.$$

Thus, for any pattern set Q, we have

$$\begin{aligned} A_{\mathbf{a}}(Q; \emptyset) &= A_{\mathbf{a}}(Q; 1) \\ &= A_{\mathbf{a}}(Q; 12) \cup A_{\mathbf{a}}(Q; 11) \cup A_{\mathbf{a}}(Q; 21) \\ &= (A_{\mathbf{a}}(Q; 231) \cup A_{\mathbf{a}}(Q; 121) \cup A_{\mathbf{a}}(Q; 132) \cup A_{\mathbf{a}}(Q; 122) \cup A_{\mathbf{a}}(Q; 123)) \\ &\quad \cup (A_{\mathbf{a}}(Q; 221) \cup A_{\mathbf{a}}(Q; 111) \cup A_{\mathbf{a}}(Q; 112)) \\ &\quad \cup (A_{\mathbf{a}}(Q; 321) \cup A_{\mathbf{a}}(Q; 211) \cup A_{\mathbf{a}}(Q; 312) \cup A_{\mathbf{a}}(Q; 212) \cup A_{\mathbf{a}}(Q; 213)), \text{ etc.} \end{aligned}$$

Finally, for ease of notation, we make the following definition:

Definition 8. *Given a prefix p of length l, the set of* refinements *of p is the set of all prefixes of length $l+1$ whose l-prefix is p.*

For example, the set of refinements of 1 is $\{11, 12, 21\}$. The set of refinements of 11 is $\{221, 111, 112\}$. The set of refinements of 12 is $\{231, 121, 132, 122, 123\}$. This simplifies our notation to the following:

$A_{\mathbf{a}}(Q; p) = \bigcup_{r \in \{\text{refinements of } p\}} A_{\mathbf{a}}(Q; r).$

Graphically, we may represent refinement using a directed tree with root $\emptyset$, where the children of each vertex are its refinements. For example, the first three levels of the refinement above may be encoded as in Figure 3.1. Each vertex labeled with prefix p represents the set $A_{\mathbf{a}}(Q; p)$. Then, to count the elements of $A_{\mathbf{a}}(Q)$ is is enough to count the elements of the sets represented by the leaves of the tree.

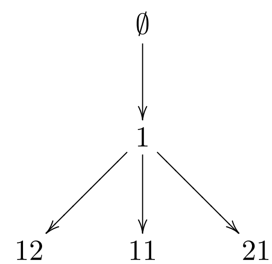


Figure 3.1: Refinement for an arbitrary pattern set

Now that we have developed a way to partition $A_{\mathbf{a}}(Q)$ into disjoint subsets, we investigate methods to find recurrences between these subsets.

3.3 Reversibly Deletable Elements

Following Zeilberger, we have the following:

Definition 9. *Given a set of forbidden patterns Q, $p = p_1 \cdots p_l$ an l-prefix, and $1 \leq t \leq l$, we say that p_t is* reversibly deletable *if every instance of a forbidden pattern q which involves p_t in a word with prefix p implies the presence of an instance of a forbidden pattern without p_t.*

For example, let $Q = \{123\}$ and $p = 21$. Then $w = ij \cdots$ with $i > j$. p_1 is reversibly deletable since the only way for $p_1 = i$ to be involved in a 123 pattern is if

$w = ij \cdots a \cdots b \cdots$ with $i < a < b$. But since $i > j$, we have $j < a < b$ as well so jab forms a 123 pattern without using position p_1.

Now, if p_t is reversibly deletable, we have the following recurrence, where $\hat{p}_t$ (resp. $\hat{i}_t$) indicates that the letter p_t (resp. i_t) has been deleted:

$$\left| A_{\mathbf{a}} \left(Q; \begin{matrix} p_1 \cdots p_l \\ i_1 \cdots i_l \end{matrix} \right) \right| = \left| A_{[a_1, \ldots, a_t - 1, \ldots, a_k]} \left(Q; \begin{matrix} p_1 \cdots \hat{p}_t \cdots p_l \\ i_1 \cdots \hat{i}_t \cdots i_l \end{matrix} \right) \right|$$

That is, deleting and replacing p_t provides a bijection between the two sets above.

It should be noted that in Zeilberger's original schemes for permutations, if positions t and s are both reversibly deletable, then

$$\left| A_{\mathbf{a}} \left(Q; \begin{matrix} p_1 \cdots p_l \\ i_1 \cdots i_l \end{matrix} \right) \right| = \left| A_{[a_1, \ldots, a_t - 1, \ldots, a_s - 1, \ldots, a_k]} \left(Q; \begin{matrix} p_1 \cdots \hat{p}_t \cdots \hat{p}_s \cdots p_l \\ i_1 \cdots \hat{i}_t \cdots \hat{i}_s \cdots i_l \end{matrix} \right) \right|$$

However, in the case of pattern-avoiding words, this is no longer true. Consider for example $q = 123$, $p = 11$. Both p_1 and p_2 are reversibly deletable independently, but not together. In a later section, we will revisit the question of when reversibly deletable letters in the same prefix can be deleted at the same time.

Graphically, we represent a reversibly deletable element p_t with a dotted arrow from p to $red(p_1 \cdots \hat{p}_t \cdots p_l)$ labelled d_t, where d_t stands for the deletion of position t. For example, in the case of $Q = \{123\}$ above, we saw that prefix $p = 12$ has position 1 reversibly deletable. By a similar argument, when $Q = \{123\}$ and $p = 11$, p_1 is also reversibly deletable. We encode this information in Figure 3.2.

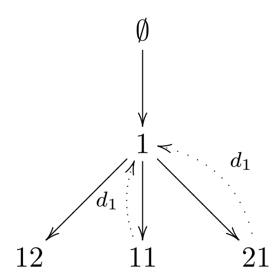


Figure 3.2: Reversibly deletable elements in the scheme for $Q = \{123\}$

3.4 Gap Vectors

Thus far, knowing only a prefix and a forbidden pattern are enough to determine reversibly deletable positions. However, there are instances when this is not the case. Consider for example $q = 123$ and $p = 12$. With our current definition neither position of the prefix is reversibly deletable. However, observe that if $w \in [k]^n, w = ij\cdots$ with $i < j < k$, then k eventually appears in the word w. Thus, $w = ij\cdots k\cdots$ and ijk is a 123 pattern. Every word with prefix 12 where the letter playing the role of "2" is less than k has a 123 pattern, so we know that in any $\{123\}$-avoiding word with prefix 12, the second letter is necessarily k. Since this is the largest letter in the alphabet, it cannot be involved in a 123 pattern, so $p_2 = k$ is trivially reversibly deletable.

To help determine the reversibly deletable positions in these more sophisticated cases, we introduce the following:

Definition 10. *Given a pattern set Q, a prefix $p = p_1 \cdots p_l$, and letters $i_1 \cdots i_l$ comprising the prefix p, let $s_1 \leq \cdots \leq s_l$ such that $\{s_1, \ldots, s_l\} = \{i_1, \ldots, i_l\}$. We say that $\boldsymbol{g} = \langle g_1, \ldots, g_{l+1}\rangle$ is a gap vector for $[Q, p]$ if there are no words avoiding Q, with prefix p and with $s_1 - 1 \geq g_1$, $s_j - s_{j-1} \geq g_j$ $(2 \leq j \leq l)$, and $k - s_l \geq g_{l+1}$.*

For example, in the case of $Q = \{123\}$ and $p = 12$ above, $\mathbf{g} = \langle 0, 1, 1\rangle$ is a gap vector since the set of all words in $[k]^n$ where $w = i_1 i_2 \cdots$, and $i_1 - 1 \geq 0$, $i_2 - i_1 \geq 1$, and $k - i_2 \geq 1$, (i.e. all words that begin with an increasing pair where $i_2 < k$) is empty. (Otherwise, $w = i_1 i_2 \cdots k \cdots$ contains a 123 pattern, namely $i_1 i_2 k$.)

This definition may at first seem awkward in that the the entries at the beginning and end of a vector have a slightly different meaning from the interior entries. An interior 0 denotes a repeated letter in the prefix, while an interior 1 denotes two necessarily adjacent letters. In the convention of Zeilberger and Vatter, we would have used -1 and 0 respectively instead. This change gives one advantage of notation. If prefix $p = p_1 \cdots p_l$ has gap vector $\langle g_1, \ldots, g_{l+1}\rangle$, then prefix $p_1 \cdots \hat{p_t} \cdots p_l$ has gap vector $\langle g_1, \ldots, g_t + g_{t+1}, \ldots, g_{l+1}\rangle$. With the notation of 0s and -1s, further adjustment would need to be made.

Notice that as in Vatter's schemes, if $\mathbf{g}$ and $\mathbf{h}$ are vectors of length l, $g_i < h_i$ for all

$1 \leq i \leq l$, and $\mathbf{g}$ is a gap vector for some $[Q, p]$, then so is $\mathbf{h}$. In other words, the set of gap vectors for a given pattern and prefix form an upper ideal in the poset of vectors in $\mathbb{N}^l$, so we can find a finite basis of gap vectors for $[Q, p]$ by choosing the minimal elements of this ideal.

Graphically, since gap vectors are associated with a particular prefix p, and there is a finite basis of gap vectors for any $[Q, p]$, we write the set of basis gap vectors for $[Q, p]$ below p. For example, if $Q = \{123\}$, we encode the gap vector for prefix $p = 12$ as in Figure 3.3.

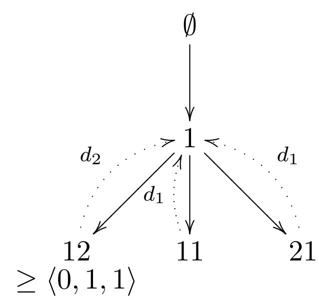


Figure 3.3: Gap vectors in the scheme for $Q = \{123\}$

3.5 Enumeration Schemes for Words

We now come to the main definition of this chapter, identical to the definition of scheme for permutations. An *abstract enumeration scheme* $\mathbb{S}$ is a set of triples $[p, G, R]$ where p is a reduced prefix of length l, G is a (possibly empty) set of vectors of length $l + 1$ and R is a subset of $\{1, \ldots, l\}$. If d is the maximum length of a prefix p in $\mathbb{S}$, we say that $\mathbb{S}$ is a scheme of depth d.

Such an enumeration scheme is said to be a *concrete enumeration scheme* if for all triples in $\mathbb{S}$, either R is non-empty or all refinements of p are also in $\mathbb{S}$. Once we have such an enumeration scheme, it can be considered as an encoding of a system of recurrences. The simplest example of such a scheme is

$$\mathbb{S} = \{\ [\emptyset, \{\}, \{\}],\ [1, \{\}, \{1\}]\ \}$$

For prefix $\emptyset$, all refinements, i.e. $\{1\}$, belong to $\mathbb{S}$. For prefix 1, $R \neq \emptyset$.

In fact, this is the scheme for counting all words in $[k]^n$. First note that it is equivalent to count all words or to count all words beginning with a 1 pattern. For words beginning with a 1 pattern, the "1" is trivially reversibly deletable (there is no forbidden pattern to avoid). This gives the following recurrence:

$$|A_{\mathbf{a}}(\{\};\emptyset)| = |A_{\mathbf{a}}(\{\};1)| = \sum_{i=1}^{k} \left| A_{\mathbf{a}}\left(\{\}; \begin{matrix} 1 \\ i \end{matrix}\right) \right| = \sum_{i=1}^{k} \left| A_{[a_1,\ldots,a_i-1,\ldots,a_k]}(\{\};\emptyset) \right|,$$

$$\left| A_{[a_1]}(\{\}) \right| = 1.$$

As expected, this gives the unique solution $|A_{\mathbf{a}}(\{\};\emptyset)| = \binom{\|\mathbf{a}\|}{a_1,\ldots,a_k}$.

We have now developed all the necessary tools to completely automatically find concrete enumeration schemes to count pattern-avoiding words in the following way:

1. Initialize $\mathbb{S} := \{[\emptyset, \{\}, \{\}]\}$.

2. Let $P = \{$refinements of all prefixes in $\mathbb{S}$ with no reversibly deletable elements$\}$

3. For each prefix $p \in P$, find its set G_p of gap vectors.

4. For each pair $[p, G_p]$, find the set R_p of all reversibly deletable elements, and let and $\mathbb{S}_2 = \cup_{p \in P}\{[p, G_p, R_p]\}$.

5. If $R_p \neq \{\}$ for all triples in $\mathbb{S}_2$, then return $\mathbb{S} \cup \mathbb{S}_2$. Otherwise let $\mathbb{S} = \mathbb{S} \cup \mathbb{S}_2$, and return to step 2.

It is clear that steps 1, 2, and 5 can be done completely automatically. In the following sections, we will prove that steps 3 and 4 can be done completely rigorously and automatically as well.

Recall from section 1.2, that, as in the case of permutations, the operations of reversal and complement are involutions on the set of words in $[k]^n$ with some useful properties. Namely,

$$A_{[a_1,\ldots,a_k]}(Q) = A_{[a_1,\ldots,a_k]}(Q^r),$$

$$A_{[a_1,\ldots,a_k]}(Q) = A_{[a_k,\ldots,a_1]}(Q^c),$$

where Q^* denotes applying operation $*$ to all elements of Q.

If we are unable to automatically find a scheme for $A_{\mathbf{a}}(Q)$ directly, we may use these natural symmetries, or Wilf equivalences, on patterns to find an equivalent scheme.

3.6 Finding Gap Vectors Automatically and Rigorously

Recall that for a fixed set of patterns Q and a prefix p of length l, $\mathbf{g}$ is a gap vector if there are no words avoiding Q with prefix p and spacing given by $\mathbf{g}$. Thus, to study gap vectors we consider the following sets:

$$A(Q,p,\mathbf{g}) = \left\{ \begin{array}{l} w \in [1+\|\mathbf{g}\|]^n, \\ n \geq l \end{array} \;\middle|\; \begin{array}{c} w \text{ avoids } Q, w \text{ has prefix } p, \\ \{s_1,\ldots,s_l\} = \{w_1,\ldots,w_l\} \text{ with } s_1 \leq \cdots \leq s_l, \\ \text{and } s_1 = g_1+1, s_j = s_{j-1}+g_j \ (j>1) \end{array} \right\},$$

where $\|\mathbf{g}\| = \sum_i g_i$, the weight of $\mathbf{g}$. Thus, $A(Q,p,\mathbf{g})$ is the set of all Q-avoiding words with alphabet size $k = 1+\|\mathbf{g}\|$ whose first l elements form a p pattern composed of the letters $g_1+1, g_2+g_1+1, \ldots, g_l+\cdots+g_1+1$.

Not all pairs $(p,\mathbf{g})$ result in a non-empty set $A(Q,p,\mathbf{g})$. If the set $A(Q,p,\mathbf{g})$ is empty, then $\mathbf{g}$ is a gap vector.

Denote $G(p) := \{\mathbf{g} \in \mathbb{N}^{l+1} | A(Q,p,\mathbf{g}) \neq \emptyset \text{ for all } n \geq l\}$. The set $\mathbb{N}^{l+1} \setminus G(p)$ is the same as the set of all gap vectors that was introduced previously. We observed before that the set of gap vectors is an upper ideal in $\mathbb{N}^{l+1}$. Since $\mathbb{N}^{l+1}$ is partially well-ordered, we may define the set of gap vectors in terms of a basis by specifying the minimal elements not in $G(p)$. We are guaranteed, by the poset structure of $\mathbb{N}^{l+1}$ under product order, that this basis is finite.

Now that we are concerned with determining a *finite* set of vectors, two questions remain: (1) How can we determine all gap vectors of a particular weight?, and (2) What is the maximum weight of a gap vector in the basis?

First, following Zeilberger, we may find all gap vectors of a specific norm $k = \|\mathbf{g}\|$ in the following way. Intuitively, a gap vector $\mathbf{g}$ specifies the relative spacing of the initial entries of a word beginning with prefix p. Consider prefix $p = p_1 \ldots p_l$, sorted and reduced to be $s = s_1 \cdots s_l$ and potential gap vector $\mathbf{g} = \langle g_1, \ldots, g_{l+1} \rangle$. This means that there are g_1 entries smaller than s_1, $max\{0, g_{i+1} - 1\}$ entries between s_i and s_{i+1} (for $2 \leq i \leq l-1$), and finally g_{l+1} entries larger than s_l.

Let $\frac{1}{g_1+1}, \frac{2}{g_1+1}, \ldots, \frac{g_1}{g_1+1}$ be the g_1 elements smaller than s_1.

Let $s_i + \frac{1}{g_{i+1}+1}, s_i + \frac{2}{g_{i+1}+1}, \ldots, s_i + \frac{max\{0, g_{i+1}\}}{g_{i+1}+1}$ be the elements between s_i and s_{i+1}, $(2 \leq i \leq l-1)$.

Let $s_l + \frac{1}{g_{l+1}+1}, \ldots, s_l + \frac{g_{l+1}}{g_{l+1}+1}$ be the g_{l+1} elements larger than s_l.

Extending the definition of reduction to fractional elements, we may consider all words of length $l + \|\mathbf{g}\|$ which begin with s and end with some permutation of the set of fractional letters above. There are $\leq (g_1 + \cdots + g_{l+1})!$ such possibilities. If *each and every* one of these words contains an element of Q, then we know that $\mathbf{g}$ is a gap vector for prefix p since the set of words beginning with p, avoiding Q, and obeying the gap conditions imposed by $\mathbf{g}$ is the empty set.

Now, we have a rigorous way to find all gap vectors of a specific weight, but the question remains: what is the maximum weight of elements in the (finite) basis of gap vectors guaranteed above?

First, it should help to remember how gap vectors are used. The notion of gap vector was introduced to help determine when a particular letter of a word prefix is reversibly deletable. We revisit this concept more rigorously.

For any $r \in [l]$, the set $A(Q, p, \mathbf{g})$ embeds naturally (remove the entry (p_r) and reduce) into $A(Q, d_r(p), d_r(\mathbf{g}))$ where $d_r(p)$ is obtained by deleting the rth entry of p and reducing. $d_r(\mathbf{g})$ is obtained by sorting p, and finding the index i corresponding to p_r, then letting $d_r(\mathbf{g}) = \langle g_1, \ldots, g_{i-1}, g_i + g_{i+1}, g_{i+2}, \ldots, g_{l+1} \rangle$.

Sometimes this embedding of $A(Q, p, \mathbf{g})$ into $A(Q, d_r(p), d_r(\mathbf{g}))$ is a bijection. If this is true for all gap vectors $\mathbf{g}$ that obey $G(p)$, that is, this embedding is a bijection whenever the set $A(Q, p, \mathbf{g})$ is non-empty, then we say that p_r is *reversibly deletable* for p with respect to Q. Notice that this equivalent to the notion of reversibly deletable introduced previously.

Adapting notation from Vatter, we have the following proposition, which puts a bound on the maximum weight of gap vectors to check before declaring an element to be reversibly deletable.

Proposition 1. *The entry p_r of the prefix p is reversibly deletable if and only if*

$$|A(Q,p,\boldsymbol{g})| = |A(Q,d_r(p),d_r(\boldsymbol{g}))|$$

for all $\boldsymbol{g} \in G(p)$ with $\|\boldsymbol{g}\| \leq \|Q\|_\infty + l - 2$, where $\|Q\|_\infty$ denotes the maximum length of a pattern in Q, and l is the length of prefix p.

Proof. If p_r is reversibly deletable then the claim follows by definition. To prove the converse, suppose that p_r is not reversibly deletable. We trivially have that

$$|A(Q,d_r(p),d_r(\mathbf{g}))| \geq |A(Q,p,\mathbf{g})|,$$

and since p_r is not reversibly deletable, we now have

$$|A(Q,d_r(p),d_r(\mathbf{g}))| > |A(Q,p,\mathbf{g})|$$

for some $\mathbf{g} \in G(p)$. Pick $\mathbf{g} \in G(p)$ and $p^* \in A(Q,d_r(p),d_r(\mathbf{g}))$ so that p^* cannot be obtained from a word in $A(Q,p,\mathbf{g})$ by removing p_r and reducing.

Now, form the Q-containing word p' by incrementing every entry of p^* that is at least p_r by 1 and inserting p_r into position r. p' is the word that would have mapped to p^*, except that p' contains a pattern $\rho \in Q$, and thus is in $A(\emptyset,p,\mathbf{g}) \setminus A(Q,p,\mathbf{g})$.

Now, pick a specific occurrence of $\rho \in Q$ that is contained in p'. Since $p^* = red(p' - p(r))$ avoids Q, this occurrence of ρ must include the entry p_r. Let p'' be the reduction of the subsequence of p' formed by all entries that are either in the chosen occurence of ρ or in prefix p (or both). p'' is now a word of length $\leq \|Q\|_\infty + l - 1$. Since all gap vectors $\mathbf{g}$ have $\|g\| = k - 1$ where k is the size of the alphabet, we have that p'' lies in $A(\emptyset,p,\mathbf{h})$ for some $\mathbf{h}$ with $\|\mathbf{h}\| \leq \|Q\|_\infty + l - 2$. On the other hand, $red(p'' - p(r))$ avoids Q, so that $|A(Q,d_r(p),\mathbf{h})| > |A(Q,d_r(p),\mathbf{h})|$, as desired. $\square$

Although not as sharp as the original bound of $\|\mathbf{g}\| \leq \|Q\|_\infty - 1$ given by Vatter for pattern-avoiding *permutations*, this still gives a bound on the maximum weight of basis vectors for the set of gap vectors that only increases linearly with the depth of the enumeration scheme. Now we have found a completely rigorous way to compute a basis for all gap vectors corresponding to a given prefix p. Finally, we turn our attention to the notion of reversibly deletable elements.

3.7 Finding Reversibly Deletable Elements Rigorously

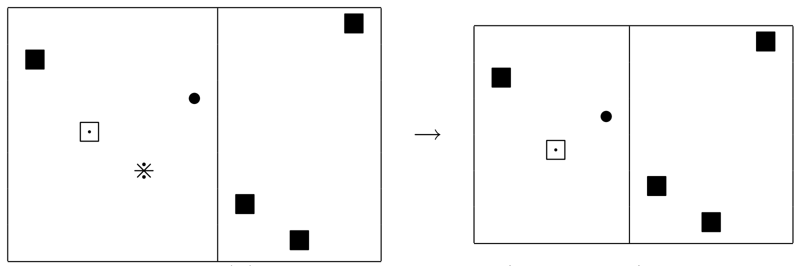
By definition, to show that p_r is reversibly deletable, we must show that every conceivable forbidden pattern involving p_r implies the presence of another forbidden pattern not involving p_r. For example, in the case of $Q = \{1234\}$ and $p = 123$, we first compute that $\mathbf{g} = \langle 0, 1, 1, 1\rangle$ is a basis for $G(p)$ with respect to p, thus $p_3 = k$, the largest letter in the word. Since the third (and largest) letter cannot be in a 1234 pattern, p_3 is trivially reversibly deletable.

As a more instructive example, consider $p = 4213$ and $Q = \{43215\}$ and check if p_3 is reversibly deletable. To check, note that the only ways that $p_3 =$"1" can participate in a 43215 pattern is if we have (1) **4213abc**, where $c >$"4"$>$"1"$> a > b$, (2) 4**213abc** where $c >$"2"$>$"1"$> a > b$, or (3) **4213ab** where $b >$"4"$>$"2"$>$"1"$> a$.

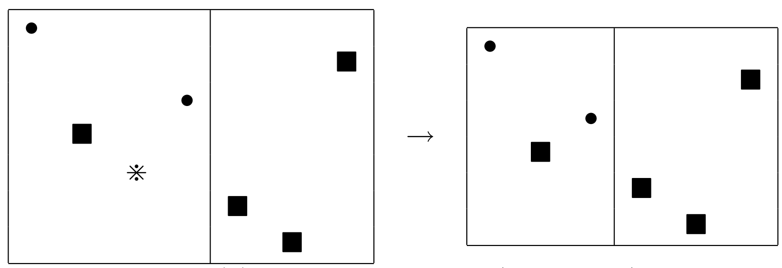
Consider the first case. If this happens, then we have 2 letters smaller than "1", and one letter larger than "4", i.e. our word has the gap condition $\langle 2, 0, 0, 0, 1\rangle$. In this case, form the word $4213abc$ and delete the "1", to obtain $423abc$. Then, $43abc$ forms a 43215 pattern, so p_3 is ok.

Now, consider the case where $21abc$ is our 43215 pattern. Again, we have 2 letters smaller than "1", but our final letter is bigger than "2", so we may have any of the following gap vectors: $\langle 2, 0, 1, 0, 0\rangle$ (that is, "2"$< c <$"3"), $\langle 2, 0, 0, 1, 0\rangle$ (that is, "3"$< c <$"4"), or $\langle 2, 0, 0, 0, 1\rangle$ (that is, $c >$"4"). Again, we must test all 3 cases, to check for implied instances of 43215. For gap $\langle 2, 0, 1, 0, 0\rangle$, we have $4213abc$ with $b < a <$"1"$<$"2"$< c <$"3"$<$"4", so $423abc$ reduces to 635214. There is no implied 43215 pattern, so p_3 is not reversibly deletable.

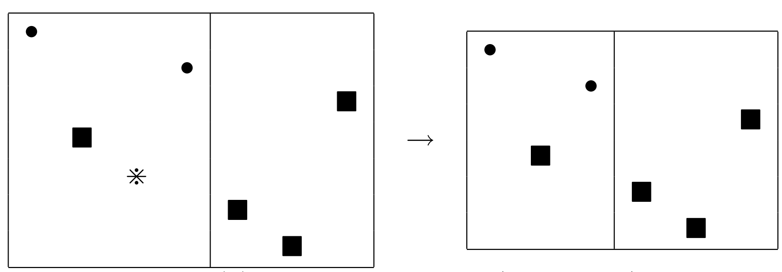
The graphs of these permutations may give more intuition for the situation at hand. The graphs of $4213abc$ and $423abc$ for each of cases (1), (2), and (3) are given in Figure 3.4, where ⋇ highlights $p_3 =$"1" to be deleted, ■ highlights the other letters in a forbidden 43215 pattern that uses p_3, and ⊡ highlights a letter in $4213abc$ that, together with the letters labeled ■, forms a 43215 pattern without using p_3 (if such a letter exists).



Case (1) $p = 4213$ and $v = \langle 2, 0, 0, 0, 1 \rangle$



Case (2) $p = 4213$ and $v = \langle 2, 0, 0, 1, 0 \rangle$



Case (3) $p = 4213$ and $v = \langle 2, 0, 1, 0, 0 \rangle$

Figure 3.4: An example of finding a reversibly deletable element

These examples give the general idea for how to test if a position p_r is reversibly deletable:

1. List all possible bad patterns involving p_r.

2. For each possible bad pattern involving p_r, list all gap spacings the pattern may have with respect to the prefix p.

3. If each gap spacing of the bad pattern implies an instance of a forbidden pattern after p_r has been removed, then p_r is reversibly deletable. Otherwise, it is not.

Furthermore, if there are non-trivial gap vectors, we may rule out many of the above cases in our computation because the gap vectors imply that the set of all such words with no bad pattern is empty.

Now that we have shown how to completely automatically determine the set of all gap vectors, and the set of reversibly deletable entries of a given prefix, we revisit the notion of independence of reversibly deletable elements. We showed earlier that if both p_r and p_s are reversibly deletable entries of prefix p, we cannot necessarily delete both p_r and p_s. We now show an important case where elements *may* be deleted simultaneously.

Proposition 2. *Let $\mathbb{S}$ be a concrete enumeration scheme, let p be a prefix in $\mathbb{S}$ and let p_r, p_s be reversibly deletable elements of p. If neither $d_r(p)$ nor $d_s(p)$ is a member of S, and p_s is reversibly deletable for some i-prefix of $d_r(p)$ in S, then p_r and p_s may be deleted simultaneously.*

We will denote this embedding (deleting p_r and p_s simultaneously) as $d_{r,s}$.

Proof. Suppose that $p, r, s, \mathbb{S}$ are as above. Since $d_r(p)$ is not in $\mathbb{S}$, there must be some i-prefix p^* of $d_r(p)$ in $\mathbb{S}$ with a reversibly deletable element p^*_j. Since p^*_j is reversibly deletable for p^*, it is also reversibly deletable for $d_r(p)$ (which begins with p^*), therefore this position is also reversibly deletable. □

In short, this proposition shows that while we may not always be able to delete more than one prefix entry at a time, when it is necessary to obtain a prefix in scheme $\mathbb{S}$, it can always be done.

We have now shown how to completely rigorously find all components of a concrete enumeration scheme for pattern-avoiding words.

3.8 The Maple Package `mVATTER`

The above algorithm has been programmed in the Maple package `mVATTER`, available from the author's website: `http://www.math.rutgers.edu/~lpudwell/maple.html`. The main functions are `SchemeF`, `MiklosA`, `MiklosTot`, and `SipurF`.

`SchemeF` inputs a set of patterns Q and a maximum depth scheme to search for, and outputs a concrete enumeration scheme for words avoiding Q of the specified maximum depth. `SchemeF` also makes use of the natural symmetries of pattern-avoiding words: reversal and complement. If it cannot find a scheme for a set of patterns, it tries to find a scheme for a symmetry-equivalent pattern set and returns that scheme instead.

`MiklosA` inputs a scheme, a prefix, and an alphabet vector v and returns the number of words obeying the scheme and the vector, having that prefix. To count *all* words with a specific alphabet vector v avoiding a specific set of patterns Q, try `MiklosA( SchemeF( Q, SchemeDepth), [], v)`.

`MiklosTot` inputs a scheme, and positive integers k and n, and outputs the total number of words in $[k]^n$ obeying the scheme.

`SipurF` inputs a list $[L]$, a maximum scheme depth, an integer r, and a list of length r. It outputs all information about schemes for words avoiding one pattern of each length in L. For example, `SipurF([3],2,4,[10,8,6,6])` outputs all information about words avoiding one permutation pattern of length 3. It will output the first 10 terms in the sequence of the number of permutations (1 copy of each letter) avoiding a pattern, the first 8 terms in the sequence of the number of words with exactly 2 copies of each letter, and the first 6 terms in the sequences with exactly 3 or 4 copies of each letter.

`SipurF` has been run on $[L]$ for various lists of the form $[3^a, 4^b]$, and the output is available from the author's website.

3.9 A Collection of Failures

Although this notion of enumeration schemes for words is successful for many sets of forbidden patterns, there is more to be done.

There are many cases where enumeration schemes of Vatter and Zeilberger fail. Unfortunately, these schemes for words avoiding permutation patterns will necessarily fail whenever Zeilberger and Vatter's schemes fail for permutations avoiding the same patterns. Namely, the chain of prefixes with no reversibly deletable elements from the permutation class enumeration scheme will still have no reversibly deletable elements for words, since there are even more possibilities for a bad pattern to occur.

Further, this technique only succeeds for words avoiding permutations (i.e., the patterns to be avoided have precisely one copy of each letter). The key observation is that gap vectors keep track of spacing, but they do not keep track of frequency. More precisely:

Proposition 3. *If $Q = \{\rho\}$ where ρ has a repeated letter, then there is no finite enumeration scheme for words avoiding Q.*

Proof. To show that there is no finite enumeration scheme, we must exhibit a chain of prefixes which have no reversibly deletable elements with respect to Q. Consider the structure of ρ. We have $\rho = q_1 l q_2 l q_3$, where l is the first repeated letter of ρ. Thus, $q_1 l q_2$ is a permutation.

First we consider a simple case. Suppose that $q_1 = \emptyset$. Then, consider the chain of prefixes of the form $p_i = 1 \dots i$. Consider an occurence of ρ beginning with j, $1 \leq j \leq i$. Since j is the first repeated letter in forbidden pattern ρ, there is no other letter in p_i which can take its place. Thus we have an infinite chain of prefixes with no reversibly deletable element.

Now, suppose that $|q_1| \geq 1$ and the final letter of q_1 is $> l$. For $1 \leq i \leq |q_1|$, we let $p_i = (q_1)_1 \cdots (q_1)_i$. Now, for $i > |q_1|$, let $d = i - |q_1|$, and make the following construction:

$$(q_1^*)_i = \begin{cases} (q_1)_i, & \text{if } (q_1)_i < l; \\ (q_1)_i + d & \text{if } (q_1)_i > l; \end{cases}$$

Then, $p_i = (p_i)_1 \cdots (p_i)_i$ where

$$(p_i)_j = \begin{cases} (q_1^*)_i, & \text{if } j \leq |q_1^*|; \\ l + (j - (|q_1^*| + 1)) & \text{if } j > |q_1^*|; \end{cases}$$

In essence, for large i, $p_i = q_1^* l^*$, where l has been replaced by increasing sequence l^*, and all entries of q_1 greater than l are incremented accordingly.

Viewing a prefix as a function from $\{1, \ldots, i\}$ to $\{1, \ldots, i\}$, the prefixes p_i of length $|q_1| + 1$ and $|q_1| + 4$ are displayed in Figure 3.5 as an example.

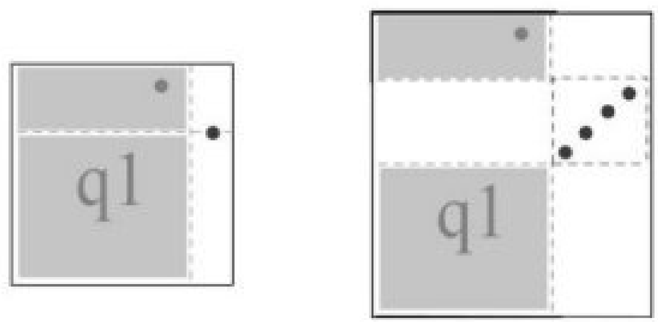


Figure 3.5: Constructing prefixes without reversibly deletable elements

Now, consider the occurrence of forbidden pattern ρ that uses element j of the monotone run at the end of p_i as l. Since this is the first repeated letter in the pattern, no matter how ρ occurs in the word, the role of l must be played by j. Thus there are no reversibly deletable entries of p_i.

For the remaining case: $|q_1| \geq 1$ and the final letter of q_1 is smaller than l, repeat the construction above, but with a decreasing run instead of an increasing run at the end of p_i for large i. Again, for each letter in p_i, we can pick an occurrence of ρ that demonstrates that letter is not reversibly deletable.

□

This shortcoming raises the question whether there is yet another way to extend schemes. Recall that in this paper, we have modified Zeilberger's original schemes which use prefixes for refinement. On the other hand, Vatter took symmetries and refined by the patterns formed by the smallest entries of a permutation. In the study of restricted permutations, these two notions are equivalent, but for words, this is no longer true. Indeed, in Vatter's notation, if we refine by adding one letter at a time, the repeated letters in words cause $1 \rightarrow 11 \rightarrow 111 \rightarrow \cdots$ to often be an infinite chain in schemes for pattern-avoiding words. Ideally we would like to find schemes that do not depend on the alphabet size or on specifying frequency of letters. One way to circumvent this difficulty is to refine words by adding multiple letters at a time. These results will be given in Chapter 4.

3.10 Examples and Successes

Despite the holes for progress discussed in Section 3.9, prefix enumeration schemes for words have a reasonable success rate, especially when avoiding *sets* of permutation patterns. This method is 100% successful when avoiding sets of patterns of length 3, and enjoys a fairly high success rate when avoiding sets of 3 or more patterns simultanously. Some of the nicer results are displayed below.

As described above, we draw an enumeration scheme as a directed graph, where the vertices are prefixes. A solid arrow goes from a prefix to any of its refinements. A dotted arrow, denoting reversibly deletable elements, goes from a prefix to one of its i-prefixes, and is labelled by the corresponding deletion map. If there are any gap vectors for a given prefix, the basis for those gap vectors is written below that prefix.

Many of the enumeration schemes for permutations carry over to enumerating words almost directly. Some simple examples include the scheme for counting all words, and the scheme for counting words avoiding the pattern 12.

$A_{\mathbf{a}}(\emptyset)$

$$\begin{tikzcd} \emptyset \arrow[d] \\ 1 \arrow[u, dotted, bend left, "d_1"] \end{tikzcd}$$

$A_{\mathbf{a}}(12)$

$$\begin{tikzcd} \emptyset \arrow[d] \\ 1 \arrow[u, dotted, bend left, "d_1"] \\ \geq (0,1) \end{tikzcd}$$

Figure 3.6: The schemes for $A_{\mathbf{a}}(\emptyset)$ and $A_{\mathbf{a}}(12)$

The first non-trivial examples are schemes for avoiding one pattern of length 3. These schemes are nearly identical to the permutation schemes, only with the 11 prefix now included. The symmetry of these schemes gives an alternate explanation that $A_{\mathbf{a}}(123)$ and $A_{\mathbf{a}}(132)$ are Wilf-equivalent for words as well as for permutations. The schemes for words avoiding 123 and 132 are found in Figure 3.7.

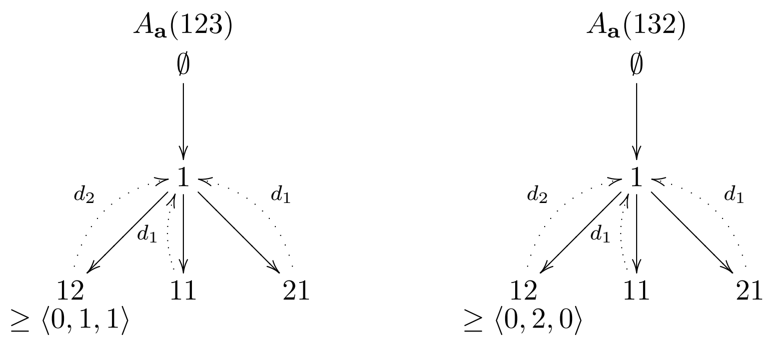


Figure 3.7: The schemes for $A_{\mathbf{a}}(123)$ and $A_{\mathbf{a}}(132)$

A more interesting example is seen in Figure 3.8.

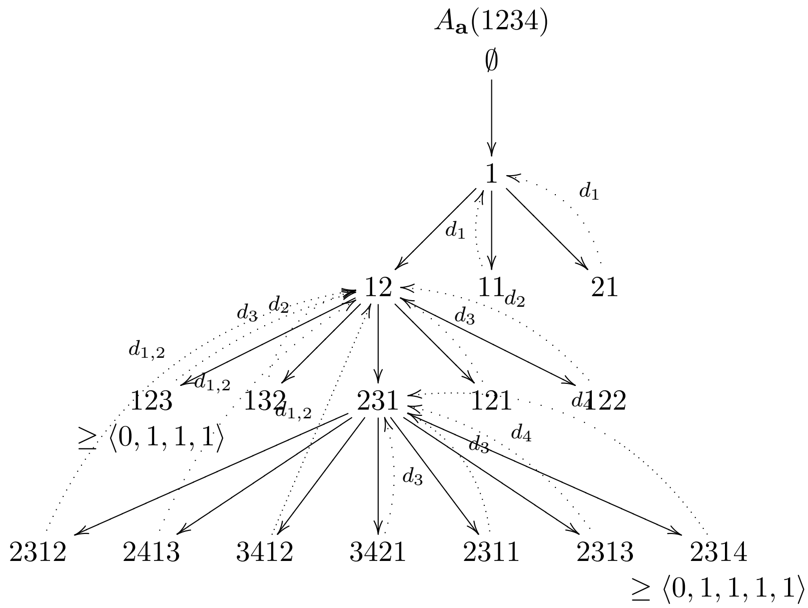


Figure 3.8: The scheme for $A_{\mathbf{a}}(1234)$

We conclude this chapter with statistics comparing the success rate of Vatter and Zeilberger's schemes for permutations versus the success rate of schemes for words. As discussed above, the current success of word schemes is bounded above by the success of permutation schemes. We consider success rate to be the percentage of trivial Wilf classes of patterns whose elements can be enumerated via schemes. Notice that there are fewer Wilf classes when enumerating permutations, since the operations of reverse, complement, and inverse all give trivial equivalences, while in the case of words, inverses no longer exist. Finally, we use the notation of the program `SipurF`, described above.

For example, avoiding the list $[3, 4]$ means to avoid one pattern of length 3 and one pattern of length 4. It is important to note that a pattern set that is not counted as successful in the following table does not necessarily indicate that there does not exist a scheme for that pattern set, but rather that it may have a scheme of greater depth than we have asked the computer to search.

Pattern Lengths	**Permutation Scheme Success Rate**	**Word Scheme Success Rate**
[2]	1/1 (100%)	1/1 (100%)
[2,3]	1/1 (100%)	1/1 (100%)
[2,4]	1/1 (100%)	1/1 (100%)
[3]	2/2 (100%)	2/2 (100%)
[3,3]	5/5 (100%)	6/6 (100%)
[3,3,3]	5/5 (100%)	6/6 (100%)
[3,3,3,3]	5/5 (100%)	6/6 (100%)
[3,3,3,3,3]	2/2 (100%)	2/2 (100%)
[4]	2/7 (28.6%)	2/8 (25%)
[3,4]	17/18 (94.4%)	9/24 (37.5%)
[3,3,4]	23/23 (100%)	27/31 (87.1%)
[3,3,3,4]	16/16 (100%)	20/20 (100%)
[3,3,3,3,4]	6/6 (100%)	6/6 (100%)
[3,3,3,3,3,4]	1/1 (100%)	1/1 (100%)
[4,4]	29/56 (51.8%)	$\geq$ 9/84 ($\geq$ 10.7%)
[3,4,4]	92/92 (100%)	38/146 (26%)
[3,3,4,4]	68/68 (100%)	89/103 (86.4%)
[3,3,3,4,4]	23/23 (100%)	29/29 (100%)
[3,3,3,3,4,4]	3/3 (100%)	3/3 (100%)

Table 3.1: Success rate of permutation schemes versus word schemes

We continue our discussion of pattern-avoiding words in Chapter 4, where we consider words avoiding patterns with repeated letters.

Chapter 4

Enumeration Schemes for Patterns with Repeated Letters

In the previous chapter, we adapted the method of enumeration schemes to count words that avoid permutation patterns. In this chapter, we modify the enumeration scheme paradigm further to count words that avoid patterns with repeated letters.

As before, we are concerned with the enumeration of

$$A_{[a_1,\ldots,a_k]}(Q) = \left\{ w \in [k]^{\|\mathbf{a}\|} | w \text{ avoids } Q, w \text{ has } a_i \ i\text{'s}, 1 \leq i \leq k \right\},$$

only now in the case when Q contains patterns with repeated letters. First, I will review the limitations of Zeilberger and Vatter's enumeration schemes when they are extended to the case of pattern-avoiding words. Then, I will introduce a new notion of schemes for words. I will use this notion to find recurrences counting words avoiding *any* pattern of length 3. The main result is that this new kind of enumeration scheme is guaranteed to work when enumerating words that avoid *any* monotone pattern of arbitrary length.

4.1 Old Schemes for Permutations

In Chapter 3, we extended Zeilberger's notion of prefix scheme to pattern-avoiding words.

Suppose we would like to enumerate the elements of a set $A(n)$. If we cannot find a closed-form formula for $A(n)$, ideally, we could find a recurrence which depends only on n. However, this is not always possible. Following Zeilberger, we introduce the notion of refinement. Namely, parameterize $A(n) = \bigcup_{i \in I} B(n, i)$ for some parameter i, so that $A(n)$ is a disjoint union of the $B(n, i)$'s. If we can then find a recurrence for each $B(n, i)$ in terms of the $A(n)$'s and the $B(n, i)$'s, we then have a formula for $A(n)$. If not, continue by parameterizing each $B(n, i) = \bigcup_{j \in J} C(n, i, j)$.

Zeilberger's schemes for pattern-avoiding permutations refine by looking at prefixes. That is if $A(n)$ is the set of words whose first l letters reduce to p, then $B(n,i)$ is the set of words whose first $l+1$ letters reduce to some longer prefix $p' = p_1 \cdots p_l \cdot i$ and such that $red(p_1 \cdots p_l) = p$.

For example, given prefix p, let $A_{\mathbf{a}}(Q;p)$ be the set of Q-avoiding words with alphabet vector $\mathbf{a} = [a_1, \ldots, a_k]$ whose first $|p|$ letters reduce to p. We have:

$$A_{\mathbf{a}}(Q;\emptyset) = A_{\mathbf{a}}(Q;1) = A_{\mathbf{a}}(Q;12) \cup A_{\mathbf{a}}(Q;11) \cup A_{\mathbf{a}}(Q;21), \text{ etc.}$$

Furthermore, we can deduce recurrences by knowing the prefix of a word. For example, we have seen that if a $\{123\}$-avoiding word has prefix 21, the "2" can be deleted, because any possible way for the "2" to be involved in a bad pattern implies that the "1" is also in a bad pattern. Therefore, if the role of "2" is played by the letter j, we have $\left| A_{[a_1,\ldots,a_j,\ldots,a_k]}\left(\{123\}; \begin{array}{c} 21 \\ ji \end{array}\right)\right| = \left| A_{[a_1,\ldots,a_j-1,\ldots,a_k]}\left(\{123\}; \begin{array}{c} 1 \\ i \end{array}\right)\right|$.

While this method of finding recurrences with refinement based on prefixes has a reasonable success rate for words avoiding permutations, it was shown in Chapter 3 that it necessarily fails if the pattern to be avoided has a repeated letter. Thus, we turn to a symmetric approach introduced by Vatter [36].

4.2 Old Schemes for Permutations: a Symmetry

Vatter's schemes for permutations take a symmetry of Zeilberger's approach and look at the patterns formed by the i smallest letters in a permutation instead of the initial i letters.

In his case, the notation $A_{\mathbf{a}}(Q;p)$ denotes the set of Q-avoiding words with alphabet vector $\mathbf{a} = [a_1, \ldots, a_k]$ whose *smallest* $|p|$ letters form pattern p. For example, $A_{\mathbf{a}}(Q;132)$ denotes the set of words where 1 appears before 3, which appears before 2. Still, we have:

$$A_{\mathbf{a}}(Q;\emptyset) = A_{\mathbf{a}}(Q;1) = A_{\mathbf{a}}(Q;12) \cup A_{\mathbf{a}}(Q;11) \cup A_{\mathbf{a}}(Q;21), \text{ etc.}$$

The logic for finding recurrences is similar. For example, if we wish to avoid the pattern 123, and consider the set $A_{[a_1,\ldots,a_k]}(\{123\};21)$, we know that if the letter 1 is involved in a 123 pattern, then the letter 2 must also be involved in a 123 pattern. Then we have $\left|A_{[a_1,\ldots,a_k]}(\{123\};21)\right| = \left|A_{[a_1-1,a_2,\ldots,a_k]}(\{123\};2)\right| = \left|A_{[a_2,\ldots,a_k]}(\{123\};1)\right|$, where the last equality is because Vatter considers the special case of *permutations*.

In essence, Vatter takes inverses of the permutations in Zeilberger's schemes, and as the inverse map provides an involution on the set of all permutations, this is an equivalent construction.

For words, however, the inverse map no longer exists. Enumerating words by considering the pattern formed by the i smallest letters is no longer as straightforward. In general, to count pattern-avoiding words, the chain of prefixes of smallest letters $1 \to 11 \to 111 \to \cdots$ forms an infinite chain of subsets of $A_{\mathbf{a}}(Q)$ without recurrences.

Clearly, while both Zeilberger's and Vatter's schemes are effective for enumerating permutations, they have their drawbacks when extended to words. Thus, we must make more significant modifications.

4.3 New Schemes

Vatter's approach can be modified in the following way. Instead of looking at the patterns formed by the i smallest letters in a word by adding *one* letter at a time, refine by successively adding *all* copies of the smallest letter at once. As we will see, this introduces new parameters into the enumeration scheme, but it allows the enumeration of words which were unable to be counted by previous methods. For convenience, we consider the sets $\mathcal{A}_s(Q,\mathbf{a})$, where $\mathcal{A}_\emptyset(Q,\mathbf{a}) = A_{\mathbf{a}}(Q)$. If the set of patterns Q is clear from context, we may write only $\mathcal{A}_s(\mathbf{a})$. The function of the subscript s will become clear in section 4.4.

For example, let $\mathcal{A}_\emptyset(\mathbf{a})$ be the set of words with alphabet vector $\mathbf{a} = [a_1,\ldots,a_k]$ avoiding 112. Then, we can refine $\mathcal{A}_\emptyset(\mathbf{a}) = \mathcal{A}_{11}(\mathbf{a}) \cup \mathcal{A}_1(\mathbf{a})$, where $\mathcal{A}_{11}(\mathbf{a}) = \{$ the set of words with at least two 1s $\}$, and $\mathcal{A}_1(\mathbf{a}) = \{$the set of words with only one 1$\}$. That is, $\mathcal{A}_{11}(\mathbf{a})$ is the set of words with enough 1s to be the start of a forbidden 112 pattern, and $\mathcal{A}_1(\mathbf{a})$ is the set of words without enough 1s to start a forbidden 112 pattern involving the letter 1.

Essentially, instead of tracking the patterns formed by the initial letters of the word (as in Zeilberger's method) or the patterns formed by the smallest letters of the word (as in Vatter's method), we begin with the empty word, and successively insert all copies of a letter at once. After each insertion of new letters, we keep track of the maximal subpattern of a forbidden 112 pattern. More explicitly, begin with an empty word and insert all a_1 1s. Keep track of the earliest 11 pattern and insert all a_2 2s. Keep track of the new first 11 pattern, and insert all a_3 3s. Repeat this process until all $a_1 + \cdots + a_k$ letters have been inserted into the word.

Here, we introduce a revised notion of scheme.

Definition 11. *Let $\mathbb{S}$ be a set of triples $[\mathcal{A}_i, C_i, R_i]$ where*

- *$\mathcal{A}_i$ is a set, possibly with extra parameters distinguishing elements of $\mathcal{A}_i$.*
- *C_i is a set of pairs $[P_{i,1}, P_{i,2}]$ where each $P_{i,1}$ is a set of $\mathcal{A}_j$'s with $j \geq i$, and the $P_{i,2}$'s are disjoint conditions on the parameters of $\mathcal{A}_i$.*
- *R_i is a linear combination of the $|\mathcal{A}_j|, j \leq i$, possibly with coefficients depending on the parameters of $\mathcal{A}_i$.*

We say that $\mathbb{S}$ is an enumeration scheme *if for each triple $[\mathcal{A}_i, C_i, R_i]$ in $\mathbb{S}$, exactly one of C_i or R_i is non-empty.*

Notice that such a scheme can be considered as an encoding for a system of recurrences. Namely, C_i are the children of $\mathcal{A}_i$, so $|\mathcal{A}_i| = \sum_{c \in C_i} |c|$, and R_i is a recurrence for $|\mathcal{A}_i|$ in terms of the sizes of earlier sets, so $|\mathcal{A}_i| = R_i$.

A simple example is the following:

$$\{[\mathcal{A}_\emptyset([a_1, \ldots, a_k]), \{[\mathcal{A}_1([a_1, \ldots, a_k]), (a_1 = 1)], [\mathcal{A}_{11}([a_1, \ldots, a_k]), (a_1 > 1)]\}, \emptyset],$$

$$\left[\mathcal{A}_1([a_1, \ldots, a_k]), \emptyset, \binom{a_2 + \cdots + a_k + 1}{1} \cdot |\mathcal{A}_\emptyset([a_2, \ldots, a_k])|\right],$$

$$[\mathcal{A}_{11}([a_1, \ldots, a_k]), \emptyset, 0]\}$$

This scheme can be interpreted in the following way: Let $\mathcal{A}_{\emptyset}([a_1,\ldots,a_k])$ be the set of all words avoiding 11, $\mathcal{A}_1([a_1,\ldots,a_k])$ the set of all words avoiding 11 with $a_1 = 1$, and $\mathcal{A}_{11}([a_1,\ldots,a_k])$ the set of all words avoiding 11 where $a_1 > 1$.

If $a_1 = \cdots = a_k = 1$, we have

$$
\begin{aligned}
|\mathcal{A}_{\emptyset}([a_1,\ldots,a_k])| &= |\mathcal{A}_1([a_1,\ldots,a_k])| \\
&= \binom{a_2+\cdots+a_k+1}{1}|\mathcal{A}_{\emptyset}([a_2,\ldots,a_k])| \\
&= \binom{a_2+\cdots+a_k+1}{1}\cdot|\mathcal{A}_1([a_2,\ldots,a_k])| \\
&= \binom{a_2+\cdots+a_k+1}{1}\cdot\binom{a_3+\cdots+a_k+1}{1}|\mathcal{A}_{\emptyset}([a_3,\ldots,a_k])| \\
&= \cdots \\
&= \binom{a_2+\cdots+a_k+1}{1}\cdot\binom{a_3+\cdots+a_k+1}{1}\cdots\binom{a_{k-1}+a_k+1}{1}\cdot\binom{a_k+1}{1} = k!.
\end{aligned}
$$

Otherwise, let j be the smallest integer for which $a_j > 1$, and let $(k)_j = k\cdot(k-1)\cdots(k-j+1)$. Then, we have:

$$
|\mathcal{A}_{\emptyset}([a_1,\ldots,a_j,\ldots,a_k])| = \cdots = (k)_{j-1}\cdot|\mathcal{A}_{\emptyset}([a_j,\ldots,a_k])|
$$

$$
= (k)_{j-1}\cdot|\mathcal{A}_{11}([a_j,\ldots,a_k])| = (k)_{j-1}\cdot 0 = 0
$$

Both cases are as expected.

4.4 Finding Schemes

In Vatter and Zeilberger's schemes, we found recurrences by looking for letters that were *reversibly deletable*, that is, letters that could be deleted from and reinserted into a pattern-avoiding word without creating a new forbidden pattern. Now we make the notion of *reversibly deletable* more general.

Definition 12. *Let $w \in [k]^n$ be an arbitrary word and $q \in [k]^m$ a forbidden pattern written in reduced form. Let $s_i(q)$ denote the substring of q formed by the letters $\leq i$, and set $s_0(q) = \epsilon$, the empty pattern. We say that w is* i-critical with respect to q *($i \geq 0$) if w contains a copy of $s_i(q)$ but avoids $s_{i+1}(q)$.*

For example, let $w = 1431532231$, and let $q = 12324$. Then $s_1(q) = 1$, $s_2(q) = 122$, $s_3(q) = 1232$, and $s_4(q) = 12324$. w is 3-critical, since it contains $s_1(q)$, $s_2(q)$, and $s_3(q)$ as patterns while it avoids the pattern 12324.

Now, we have a more rigorous way to produce a scheme in the sense of Section 4.3. Given a forbidden pattern $q \in [k]^m$,

- The $\mathcal{A}_i$s are the sets of words that are i-critical for $0 \leq i \leq k-1$, plus $\mathcal{A}_k$ (the set of words containing q) and $\mathcal{A}_\emptyset$ (the set of ALL words avoiding q). $\mathcal{A}_i$ may include parameters to track the location of a copy of $s_i(q)$.

- If $\mathcal{A}_i$ is the set of i-critical words, then C_i consists of the pairs

  $$[\mathcal{A}_i, (\text{conditions to insert new letters while keeping an } i\text{-critical word } i\text{-critical})]$$

  and

  $$\left[\mathcal{A}_{i+1}, \begin{pmatrix} \text{conditions to insert new letters so that an } i\text{-critical word} \\ \text{becomes } (i+1)\text{-critical} \end{pmatrix}\right]$$

- R_i results from a case by case analysis of the structure of i-critical words. Namely, if there are letters in an i-critical word that cannot possibly be involved in a forbidden pattern, they may be deleted. Also, the parameters of $\mathcal{A}_i$ that keep track of the location of a copy of $s_i(q)$ within a given word may be adjusted.

As before, the operations of complement and reversal are involutions on the set of words in $[k]^n$ with the following useful properties:

$$\mathcal{A}_\emptyset(Q; [a_1, \ldots, a_k]) = A_{[a_1,\ldots,a_k]}(Q) = A_{[a_1,\ldots,a_k]}(Q^r) = \mathcal{A}_\emptyset(Q^r; [a_1, \ldots, a_k])$$

$$\mathcal{A}_\emptyset(Q; [a_1, \ldots, a_k]) = A_{[a_1,\ldots,a_k]}(Q) = A_{[a_k,\ldots,a_1]}(Q^c) = \mathcal{A}_\emptyset(Q^c; [a_k, \ldots, a_1])$$

In the following sections, we illustrate the power of this method by finding recurrences to count the elements of $A_{\mathbf{a}}(\{q\}) = \mathcal{A}_\emptyset(\{q\}; \mathbf{a})$, where q is any pattern of length 3.

As a final comment on the $\mathcal{A}_s$ notation, $\mathcal{A}_s(Q; [a_1, \ldots, a_l])$ denotes the set of Q-avoiding words which are s critical when restricted to the alphabet $\{1, \ldots, i\}$, but with

a_1 i+1's, a_2 (i+2)'s, ..., a_l (i+l)'s . Thus, in this chapter the vector $[a_1, \ldots, a_l]$ is not an alphabet vector in the sense of Chapter 3, but rather only contains the information for the frequency of letters $> i$ for some i. We may refer to such a vector as the *upper alphabet vector*. For each of patterns below, we consider letters $\{1, \ldots, i\}$ "old letters" and (i+1) as the "new letter" to be inserted.

4.5 Avoiding the pattern 111

The simplest pattern of length 3 is 111. The scheme for $\mathcal{A}_{\mathbf{a}}(111)$ is very similar to the scheme for $\mathcal{A}_{\mathbf{a}}(11)$ given in Section 4.3 . Notice that $s_1(111) = 111$, so a word is 1-critical if it contains 3 copies of the same letter.

Let $w \in \mathcal{A}_\emptyset([a_1, \ldots, a_k])$ be an arbitrary $\{111\}$-avoiding word in $[k]^{||a||}$. Either $a_1 \leq 2$, i.e. w is 0-critical, or $a_1 \geq 3$, in which case w is 1-critical. If a word is 0-critical, the letters in it cannot possibly be part of a bad pattern after subsequent insertion of larger letters, so they may be inserted anywhere in the word, i.e. $|\mathcal{A}_1([a_1, \ldots, a_k])| = \binom{a_1+\cdots+a_k}{a_1} \cdot |\mathcal{A}_\emptyset([a_2, \ldots, a_k])|$. We represent the situation graphically as in Figure 4.1.

$$\begin{tikzcd}
 & \mathcal{A}_\emptyset \arrow[dl, "(a_1\leq 2)"'] \arrow[dr, "(a_1\geq 3)"] & \\
\mathcal{A}_0 \arrow[ur, dotted, bend left, "\binom{a_1+\cdots+a_k}{a_1}"] & & \mathcal{A}_1 = 0
\end{tikzcd}$$

Figure 4.1: The scheme for $\mathcal{A}_{\mathbf{a}}(111)$

The nodes in this graph are the sets $\mathcal{A}_i$, the solid lines go from $\mathcal{A}_i$ to the sets in C_i, and are labelled with the corresponding conditions. A labelled dotted arrow contains the information of R_i

Or, in the more familiar scheme notation, we have:

$$\{[\mathcal{A}_\emptyset([a_1, \ldots, a_k]), \{[\mathcal{A}_0([a_1, \ldots, a_k]), (a_1 \leq 2)], [\mathcal{A}_1([a_1, \ldots, a_k]), (a_1 \geq 3)]\}, \emptyset],$$

$$[\mathcal{A}_0([a_1, \ldots, a_k]), \emptyset, \binom{a_1 + \cdots + a_k}{a_1} \cdot |\mathcal{A}_\emptyset([a_2, \ldots, a_k])|],$$

$$[\mathcal{A}_1([a_1, \ldots, a_k]), \emptyset, 0]\}$$

We can read this scheme to obtain the following system of recurrences:

If $a_i \leq 2$ for all i; then,

$$|\mathcal{A}_\emptyset([a_1,\ldots,a_k])| = |\mathcal{A}_0([a_1,\ldots,a_k])| = \binom{a_1+\cdots+a_k}{a_1} \cdot |\mathcal{A}_\emptyset([a_2,\ldots,a_k])|$$
$$= \cdots = \binom{a_1+\cdots+a_k}{a_1} \cdots \binom{a_{k-1}+\cdots+a_k}{a_{k-1}} \cdot \binom{a_k}{a_k}$$
$$= \binom{a_1+\cdots+a_k}{a_1,\ldots,a_k} = |\{\text{all words with alphabet vector } \mathbf{a}\}|$$

Otherwise, let j be the smallest integer for which $a_j \geq 3$. Then,

$$|\mathcal{A}_\emptyset([a_1,\ldots,a_k])| = \binom{a_1+\cdots+a_k}{a_1} \cdots \binom{a_{j-1}+\cdots+a_k}{a_{j-1}} \cdot |\mathcal{A}_\emptyset([a_j,\ldots,a_k])|$$
$$= \binom{a_1+\cdots+a_k}{a_1} \cdots \binom{a_{j-1}+\cdots+a_k}{a_{j-1}} \cdot |\mathcal{A}_1([a_j,\ldots,a_k])|$$
$$= \binom{a_1+\cdots+a_k}{a_1} \cdots \binom{a_{j-1}+\cdots+a_k}{a_{j-1}} \cdot 0 = 0$$

4.6 Avoiding the pattern 112

Now, we turn to the case of avoiding patterns of length 3 with 2 distinct letters. Taking into account the symmetries of complement and reversal, once we can count the elements of $A_{\mathbf{a}}(112)$, we may also count the elements of $A_{\mathbf{a}}(211)$, $A_{\mathbf{a}}(122)$, and $A_{\mathbf{a}}(221)$.

Consider the case of $\{112\}$-avoiding words in more detail. As before, let $\mathcal{A}_\emptyset$ be the set of all words avoiding 112, and let $\mathcal{A}_0$ and $\mathcal{A}_1$ be the sets of 0-critical and 1-critical words respectively. That is, $\mathcal{A}_0$ denotes words without a repeated letter, and $\mathcal{A}_1$ denotes words with a 11 pattern but no 112 pattern when considering the letters $\{1,\ldots,i\}$.

We still write $\mathcal{A}_\emptyset([a_1,\ldots,a_k])$, and $\mathcal{A}_0([a_1,\ldots,a_k])$ to denote words with a particular alphabet vector, but for $\mathcal{A}_1$, we write $\mathcal{A}_1([a_i,\ldots,a_k],j)$, where j is the position of last letter of the first 11 pattern formed by the letters $1,2,\ldots,i-1$ already in the word.

We have the following trivial base cases: If $k=1$, then $\mathcal{A}_0([a_1]) = 1$ (since there is only one word with an alphabet vector $[a_1]$ and it avoids 112), and $\mathcal{A}_1([a_i],j) = \binom{j-1+a_i}{a_i}$ (since any larger letter inserted after the repeated letter in position j forms a 112 forbidden pattern).

Now, consider what happens when $k > 1$. In general, we start with the empty word, and insert all a_1 copies of 1 into the word. Next, we insert all a_2 copies of 2 into the word. At each iteration, the word composed of all letters $1, 2, \ldots, i$ is called the "old word", and the word composed of $1, 2, \ldots, i, i+1$, immediately after all copies of the letter $i+1$ have been inserted is called the "new word".

If $k > 1$ and $a_1 = 1$, there is no way for a single smallest letter to be part of a 112 pattern, so we may find the number of words with alphabet vector $[a_2, \ldots, a_k]$, and insert the smallest letter anywhere. Thus, $|\mathcal{A}_0([a_1, \ldots, a_k])| = \binom{a_2+\cdots+a_k+1}{1} \cdot |\mathcal{A}_\emptyset([a_2, \ldots, a_k])|$.

If $k > 1$ and $a_1 > 1$, the first repeated letter in a string of identical letters $\underbrace{1 \ldots 1}_{a_1}$ is in position 2, so we have $|\mathcal{A}_\emptyset([a_1, \ldots a_k])| = |\mathcal{A}_1([a_2, \ldots, a_k], 2)|$.

Now, we move on to considering the sets $\mathcal{A}_1([a_1, \ldots, a_k], j)$.

If $k > 1$ and $a_1 = 1$, we may not insert the new (larger) letter after position j. There are j choices for where to insert this letter into the word before position j. Moreover, inserting this letter in the beginning of the word moves the first repeated letter to position $j+1$ (see Figure 4.2). Thus, we have $|\mathcal{A}_1([a_1, \ldots, a_k], j)| = j \cdot |\mathcal{A}_1([a_2, \ldots, a_k], j+1)|$.

	$j-1$ old letters	1st repeated letter
position		j

(1) Old word, before inserting a_1 (i+1)'s

	$j-1$ old letters $+1$ new letter	1st repeated letter
position		$j+1$

(2) New word, after inserting a_1 (i+1)'s

Figure 4.2: Constructing elements of $\mathcal{A}_1(\{112\}; [a_1, \ldots, a_k], j)$ when $a_1 = 1$

If $k > 1$ and $a_1 > 1$, again we know that none of the a_1 (larger) letters to be inserted may appear after position j. Since there are at least two identical letters to insert before position j, the new first repeated letter will be one of the newly inserted letters. Thus, let the new first repeated letter be in position l (as in Figure 4.3). There are $l-2$ old letters and 1 new letter before position l, and there are $(j-1)-(l-2)$ old letters and a_1-2 old letters between position l and the old first repeated letter in position j. Thus, summing over all possibilities for position l, $|\mathcal{A}_1([a_1,\ldots,a_k],j)| = \sum_{l=2}^{j+1}(l-1)\binom{(i-1)+(l-2)+(a_1-2)}{(a_1-2)}|\mathcal{A}_1([a_2,\ldots,a_k],j)|$.

	$(j-1)$ old letters	1st repeated letter
position		j

(1) Old word, before inserting a_1 (i+1)'s

	$l-2$ old letters $+1$ new letter	i+1	$(j-1)-(l-2)$ old letters $+(a_1-2)$ new letters	1st repeated letter
position		l		$j+a_1$

(2) New word, after inserting a_1 (i+1)'s

Figure 4.3: Constructing elements of $\mathcal{A}_1(\{112\};[a_1,\ldots,a_k],j)$ when $a_1>1$

Graphically, we have

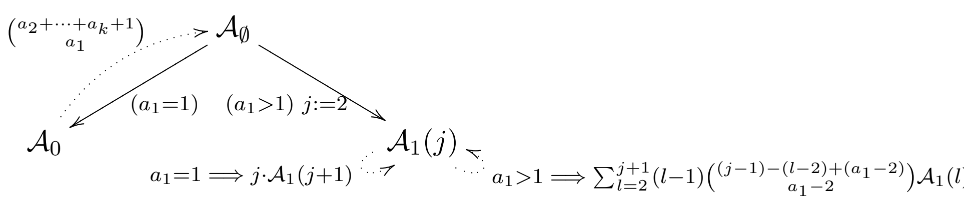


Figure 4.4: The scheme for $\mathcal{A}_{\mathbf{a}}(112)$

Which is the same as:

$$|\mathcal{A}_\emptyset([a_1,\ldots,a_k])| = \begin{cases} 1 & k=1 \\ \binom{a_2+\cdots+a_k+1}{1}|\mathcal{A}_\emptyset([a_2,\ldots,a_k])| & k>1, a_1=1 \\ |\mathcal{A}_1([a_2,\ldots,a_k],2)| & k>1, a_1>1 \end{cases}$$

$$|\mathcal{A}_1([a_1,\ldots,a_k],j)| = \begin{cases} \binom{j-1+a_1}{a_1} & k=1 \\ j\cdot|\mathcal{A}_1([a_2,\ldots,a_k],j+1)| & a_1=1 \\ \sum_{l=2}^{j+1}(l-1)\binom{(j-1)-(l-2)+(a_1-2)}{a_1-2}|\mathcal{A}_1([a_2,\ldots,a_k],l)| & a_1>1 \end{cases}$$

This recurrence is uniquely satisfied by

$$|\mathcal{A}_\emptyset([a_1,\ldots,a_k])| = \prod_{i=2}^{k}(a_i+\cdots+a_k+1)$$

This is a new proof of a result given by Heubach and Mansour [20]. More significantly it can be easily generalized, as we will see in section 4.10.

4.7 Avoiding the pattern 121

To complete our characterization of words which avoid a pattern of length 3 with at most 2 letters, it remains to compute $|A_{\mathbf{a}}(121)|$ (which will allow us to find $|A_{\mathbf{a}}(212)|$).

We can do this easily by adding a new parameter. The algorithm remains the same. Begin with an empty word, and insert all copies of the smallest letter. Then, consider how many ways to insert the next smallest letter without inducing a forbidden pattern, keeping track of the maximal bad pattern formed. Since 121 is not a monotone pattern, however, it no longer suffices to keep track of the earliest 11 pattern in 1-critical words.

More specifically, when we consider all copies of a letter l in a 121-avoiding word, we know that there can be no larger letters between the first l and the last l in the word. Thus, this first l, last l, and all letters in between act as if they were only one letter. Instead of parameterizing our scheme in terms of locations of letters, it suffices to keep track of the number of such "blocks" of letters in the word. Notice that a word may be either 0-critical or 1-critical from our previous notation and still have any number of blocks. Thus, for 1-critical words, we define the following:

$$\mathcal{A}_1(\mathbf{a},i) := \left\{ \begin{array}{c} \text{1-critical words with upper alphabet vector } \mathbf{a} \\ \text{and exactly } i \text{ blocks of letters} \end{array} \right\}$$

so $\mathcal{A}_\emptyset([a_1, \ldots, a_k]) := \{\mathit{all}\ 121\text{-avoiding words}\} = \mathcal{A}_1([a_2, \ldots, a_k], 1)$. i.e., a word consisting of a_1 1s consists of a single block.

Now, consider a word with i blocks. We may not insert new letters into the middle of a block, but we may insert letters anywhere between blocks. Moreover, if the new letters are not all adjacent, the first new letter and the last new letter form the beginning and end of a new block.

For example, suppose the current $\{121\}$-avoiding word is 33211222 and we wish to insert two 4s. The current word has 2 blocks: 33, and 211222. So we may put the 4s together: 4433211222, 3344211222 or 3321122244, or we may separate them 4334211222, 4332112224, 3342112224.

In general, suppose we have a word $w \in \mathcal{A}([a_1, \ldots, a_{l-1}])$ with i blocks, and we wish to insert a_l copies of the letter l. The position of the first new l and the last new l determine a new block. Suppose that between these two new letters there were b old blocks. Then there are $\binom{(a_l-2)+(b)}{b}$ ways to arrange the other new letters inside this new block. Moreover this turned b blocks into 1 block for a net loss of $b-1$ blocks. So if j is the new number of blocks after letter insertions, there are $j = i-(b-1)$ blocks, i.e. $b = i+1-j$. Now there are j ways to pick which consecutive b blocks will become one single new block, so there are $j \cdot \binom{(a_l-2)+(i+1-j)}{(i+1-j)}$ ways to get j blocks from i blocks by inserting a_l letters.

This is represented graphically in Figure 4.5, and can be written as:

$$\begin{tikzcd}
\emptyset \arrow[d, "(i:=1)"] \\
\mathcal{A}_1(i) \arrow[loop below, dotted, "\sum_{j=1}^{i+1} j\cdot\binom{(a_1-2)+(i+1-j)}{(i+1-j)}\cdot\mathcal{A}_1(j)"]
\end{tikzcd}$$

Figure 4.5: The scheme for $\mathcal{A}_{\mathbf{a}}(121)$

$$|\mathcal{A}_1([a_1,\ldots,a_k],i)| = \sum_{j=1}^{i+1} j \cdot \binom{(a_1-2)+(i+1-j)}{(i+1-j)} |\mathcal{A}_1([a_2,\ldots,a_k],j)|$$

Together with the base cases of $\mathcal{A}_\emptyset([]) = 1$ for the empty word, and $\mathcal{A}_1([],i) = 1$, we have a recurrence completely counting all words avoiding the pattern 121 which yields the unique solution:

$$|\mathcal{A}_\emptyset([a_1,\ldots,a_k])| = \prod_{i=2}^{k}(a_i + \cdots + a_k + 1)$$

This result was also given by Heubach and Mansour [20], but was shown in a different way.

4.8 Avoiding the pattern 213

We now move on to words avoiding patterns of length 3 with 3 letters, i.e. words avoiding permutations. This case can be taken care of by prefix schemes, as noted in Chapter 3, but for the sake of completeness, we describe an alternate enumeration using the method of this chapter.

Again, by the symmetries of complement and reversal, an enumeration scheme for $A_{\mathbf{a}}(213)$ allows us to count the elements of $A_{\mathbf{a}}(312)$, $A_{\mathbf{a}}(132)$, and $A_{\mathbf{a}}(231)$.

We return to our original notation of i-critical words, and add a few parameters.

Let:

$$\mathcal{A}_\emptyset([a_1,\ldots,a_k]) = \{\text{all } \{213\}\text{-avoiding words}\}$$

$$\mathcal{A}_1([a_1,\ldots,a_k],p) = \left\{ \begin{array}{c} \text{1-critical } \{213\}\text{-avoiding words with} \\ p \text{ letters after the leftmost 1 pattern} \end{array} \right\}$$

$$\mathcal{A}_2([a_1,\ldots,a_k],p) = \left\{ \begin{array}{c} \text{2-critical } \{213\}\text{-avoiding words with} \\ \text{leftmost 21 pattern ending in position } p \end{array} \right\}$$

Trivially, by inserting a_1 identical letters into the empty word, we have

$$|\mathcal{A}_\emptyset([a_1,\ldots,a_k])| = |\mathcal{A}_1([a_2,\ldots,a_k],a_1-1)|\,.$$

Now consider an arbitrary 1-critical word. Since this word contains a 1 pattern, but not a 21 pattern, all letters must be in increasing order.

Consider a generic member of $\mathcal{A}_1([a_1, \ldots, a_k], p)$, as in Figure 4.6(1). When we insert a_1 new letters into this word, either (a) we do not create a new 21 pattern (i.e. all new letters are appended to the end of the word), as in Figure 4.6(2), or (b), we do create a new 21 pattern, and keep track of where the leftmost such pattern ends, as in Figure 4.6(3).

	"1"	p old letters
position	1	

(1) Generic member of $\mathcal{A}_1([a_1, \ldots, a_k], p)$

	"1" (old letter)	p old letters	a_1 new letters
position	1		

(2) Case a: no new 21 pattern

	j old letters	"2" (new letter)	$l-1$ new letters	"1" (old letter)	$p-j$ old letters $+a_1-l$ new letters
position		$j+1$		$j+l+1$	

(3) Case b: new 21 pattern induced

Figure 4.6: Constructing elements of $\mathcal{A}_1(\{213\}, [a_1, \ldots, a_k], p)$

Thus, $|\mathcal{A}_1([a_1, \ldots, a_k], p)| =$

$$|\mathcal{A}_1([a_2, \ldots, a_k], p + a_1)| + \sum_{l=1}^{a_1} \sum_{j=0}^{p} \binom{(p-j) + (a_1 - l)}{a_1 - l} \cdot |\mathcal{A}_2([a_2, \ldots, a_k], j + l + 1)|.$$

Finally, consider all 2-critical words, that is, words that contain a 21 pattern, but not a 213 pattern.

Say that the leftmost 21 ends in position p. Then no new (larger) letters may be inserted after position p without creating a forbidden 213 pattern. Again, either (a) the letter that plays the role of 1 in the current leftmost 21 pattern stays the same, as in Figure 4.7(1), or (b) the newly inserted letters create an earlier 21 pattern, as in Figure 4.7(2).

Thus,

$$|\mathcal{A}_2([a_1, \ldots, a_k], p)| = |\mathcal{A}_2([a_2, \ldots, a_k], p + a_1)| +$$

	$p-1$ old letters	a_1 new letters	"1"
position			$p+a_1$

(1) Case a: 2-critical word with same leftmost 21 pattern

	j old letters	"2" (new letter)	$l-1$ new letters	"1" (old letter)	$(p-1)-(j+1)$ old letters $+a_1-l$ new letters	old "1"
position		$j+1$		$j+l+1$		

(2) Case b: 2-critical word with new leftmost 21 pattern

Figure 4.7: Constructing elements of $\mathcal{A}_2(\{213\},[a_1,\ldots,a_k],p)$

$$\sum_{l=1}^{a_1}\sum_{j=0}^{p-2}\binom{(p-1)-(j+1)+(a_1-l)}{a_1-l}\left|\mathcal{A}_2([a_2,\ldots,a_k],j+l+1)\right|.$$

We can represent this in the more compact graphical notation, as in Figure 4.8.

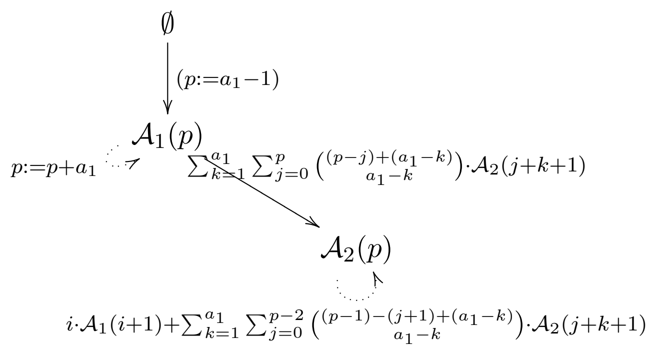


Figure 4.8: The scheme for $\mathcal{A}_{\mathbf{a}}(213)$

Although this recurrence does not readily yield a nice closed form formula as in the previous examples, we have now deduced a system of recurrences that completely enumerates all words avoiding 213. This is an alternative way to enumerate these words which were first counted by Burstein [7], and later using the prefix schemes of Chapter 3.

4.9 Avoiding the pattern 123

To complete our classification of words avoiding patterns of length 3, we examine words avoiding the permutation 123. The analysis of $A_{\mathbf{a}}(123)$ turns out to be very similar to the analysis of $A_{\mathbf{a}}(213)$.

Let:

$$\mathcal{A}_{\emptyset}([a_1,\ldots,a_k]) = \{\text{all } \{123\}\text{-avoiding words}\}$$

$$\mathcal{A}_1([a_1,\ldots,a_k],p) = \left\{ \begin{array}{c} \text{1-critical } \{123\}\text{-avoiding words with} \\ p \text{ letters after the leftmost 1 pattern} \end{array} \right\}$$

$$\mathcal{A}_2([a_1,\ldots,a_k],p) = \left\{ \begin{array}{c} \text{2-critical words with leftmost 12 pattern} \\ \text{ending in position } p \end{array} \right\}$$

Trivially, by inserting a_1 identical letters into the empty word, we have

$$\mathcal{A}_{\emptyset}([a_1,\ldots,a_k]) = \mathcal{A}_1([a_2,\ldots,a_k],a_1-1).$$

Now consider an arbitrary 1-critical word. Since this word contains a 1 pattern, but not a 12 pattern, all letters must be in decreasing order, as in Figure 4.9(1). Notice, that we keep the leftmost 1 pattern separate for further analysis.

When we insert a_1 new letters into this word, either (a) we do not create a new 12 pattern (i.e. all new letters are appended to the beginning of the word), as in Figure 4.9(2), or, (b), we do create a new 12 pattern, and keep track of where leftmost such pattern ends, as in Figure 4.9(3).

Thus, $|\mathcal{A}_1([a_1,\ldots,a_k],p)| =$

$$|\mathcal{A}_1([a_2,\ldots,a_k],p+a_1)| + \sum_{l=1}^{a_1}\sum_{j=0}^{p}\binom{(p-j)+(a_1-l)}{a_1-l}\cdot|\mathcal{A}_2([a_2,\ldots,a_k],j+l+1)|\,.$$

Finally, consider all 2-critical words, that is, words that contain a 12 pattern, but not a 123 pattern. A generic 2-critical word is shown in Figure 4.10(1).

Say that the leftmost 12 ends in position p. Then no letters may be inserted after position p without creating a forbidden 123 pattern. Again, either (a) the letter that plays the role of 2 in the current leftmost 12 pattern stays the same, as in Figure 4.10(2), or (b) the newly inserted letters create an earlier 12 pattern, as in Figure 4.10(3).

	"1"	p old letters
position	1	

(1) Generic member of $\mathcal{A}_1([a_1, \ldots, a_k], p)$

	"1" (new letter)	$(a_1 - 1)$ new letters	old "1"	p old letters
position	1		$a_1 + 1$	

(2) Case a: no new 12 pattern

	$l - 1$ new letters	"1" (old letter)	j old letters	"2" new letter	$p - j$ old letters $+a_1 - l$ new letters
position		l		$j + l + 1$	

(3) Case b: new 12 pattern is induced

Figure 4.9: Constructing elements of $\mathcal{A}_1(\{123\}, [a_1, \ldots, a_k], p)$

Thus,

$$|\mathcal{A}_2([a_1, \ldots, a_k], p)| = |\mathcal{A}_2([a_2, \ldots, a_k], p + a_1)| +$$
$$\sum_{l=1}^{a_1} \sum_{j=0}^{p-2} \binom{(p-1) - (j+1) + (a_1 - l)}{a_1 - l} \cdot |\mathcal{A}_2([a_2, \ldots, a_k], j + l + 1)| \, .$$

This can be represented using the more compact graphical notation as in Figure 4.11.

The fact that $A_{\mathbf{a}}(123)$ and $A_{\mathbf{a}}(132)$ are equinumerous, as noted by Burstein [7] and by Chapter 3, can be illustrated in another new way via these identical schemes.

Now that we have used the new definition of enumeration scheme to count words that avoid any pattern of length 3, we turn to the main theorem of this paper.

4.10 Avoiding Monotone Patterns

Up to this point, the results of this paper have previously been shown using other methods. As the case by case analysis involved in finding schemes for pattern-avoiding words seems to be quite tedious at times, one may wonder what the utility of the present method is.

To date, there has been no infinite family of classes of pattern-avoiding words (or permutations) which have been shown to each have a finite enumeration scheme. This

	$p-1$ old letters including "1"	"2" (from leftmost 12 pattern)	remaining old letters
position		p	

(1) generic 2-critical word

	a_1 new letters	$p-1$ old letters	"2" (old letter)
position			$p+a_1$

(2) Case a: same leftmost 12 pattern

	$l-1$ new letters	"1" (old letter)	j old letters	"2" (new letter)	$(p-1)-(j+1)$ old letters $+a_1-l$ new letters	old "2"
position		l		$j+l+1$		

(c) Case b: earlier leftmost 12 pattern is induced

Figure 4.10: Constructing elements of $\mathcal{A}_2(\{123\},[a_1,\ldots,a_k],p)$

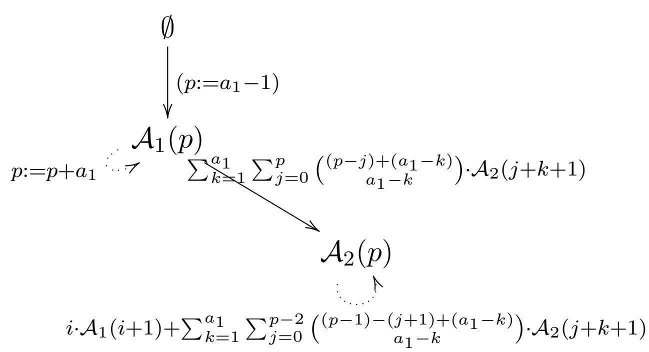


Figure 4.11: The scheme for $\mathcal{A}_{\mathbf{a}}(123)$

new kind of scheme has the advantage that there provably exists a scheme for words avoiding *any* monotone pattern.

For ease of notation, consider the monotone pattern $p = 1^{b_1}\cdots m^{b_m}$. Let $\mathcal{A}_\emptyset([a_1,\ldots,a_k])$ be the set of all words avoiding p with frequency vector $[a_1,\ldots,a_k]$. As before, $\mathcal{A}_i([a,\ldots,a_k])$ is the set of i-critical words with respect to p.

If $a_1 < b_1$, then we have $|\mathcal{A}_\emptyset([a_1,\ldots,a_k])| = \binom{a_2+\cdots+a_k+1}{a_1}|\mathcal{A}_\emptyset([a_2,\ldots,a_k])|$, and also $|\mathcal{A}_i([])| = 1$ for any i.

For $\mathcal{A}_i$, we also keep track of the positions of the end of the leftmost 1^{b_1}, $1^{b_1}2^{b_2}$, $1^{b_1}2^{b_2}3^{b_3}$, $\ldots$, $1^{b_1}2^{b_2}\cdots i^{b_i}$ patterns.

Now, to find a recurrence equation for each $\mathcal{A}_i$, one must only complete a case by case counting exercise, as in the examples above. That is, consider the cases:

- the insertion of new (larger) letters does not affect any of the existing 1^{b_1}, $1^{b_1}2^{b_2}$, $1^{b_1}2^{b_2}3^{b_3}$, ..., $1^{b_1}2^{b_2}\cdots i^{b_i}$ patterns
- the new letters create a new leftmost 1^{b_1} pattern, but do not affect any of the $1^{b_1}2^{b_2}$, ..., $1^{b_1}2^{b_2}\cdots i^{b_i}$ patterns
- the new letters create a new leftmost $1^{b_1}2^{b_2}$ pattern, but do not affect any of the $1^{b_1}2^{b_2}3^{b_3}$, ..., $1^{b_1}2^{b_2}\cdots i^{b_i}$ patterns
- the new letters create a new leftmost $1^{b_1}2^{b_2}3^{b_3}$ pattern, but do not affect any of the $1^{b_1}2^{b_2}3^{b_3}4^{b_4}$,..., $1^{b_1}2^{b_2}\cdots i^{b_i}$ patterns
- ...
- the new letters create new leftmost $1^{b_1}2^{b_2}$, $1^{b_1}2^{b_2}3^{b_3}$, ..., $1^{b_1}2^{b_2}\cdots i^{b_i}$ patterns
- the new letters create a $1^{b_1}2^{b_2}\cdots(i+1)^{b_{i+1}}$ pattern, causing the word to be $(i+1)$-critical.

Although the counting and notation may get quite hairy, there are no added subtleties: to count words avoiding a monotone pattern, one must only take sums of combinations of the sets as shown above.

Theorem 21. *The set of words avoiding the monotone pattern $1^{b_1}\cdots m^{b_m}$ has a scheme consisting of at most m triples $[\mathcal{A}_i, C_i, R_i]$.*

Proof. For each $\mathcal{A}_i$ above, $\mathcal{A}_i$ can clearly be written as a combination of $\mathcal{A}_i$s (either adding new letters does not create an $(i+1)$-critical pattern) and $\mathcal{A}_{i+1}$s (adding new letters creates at worst an $(i+1)$-critical pattern). Since $\mathcal{A}_i = 0$ for $i \geq m$, the scheme ends with state $\mathcal{A}_{m-1}$, which has no new children (because words in $\mathcal{A}_m$ contain pattern p) thus giving a scheme of m triples. $\square$

It should be noted that this is the first method that *guarantees* a way to compute the size of each member in a non-trivial family of classes of pattern-avoiding words of arbitrary length. It should also be noted that the case of enumerating permutations which avoid a monotone pattern is a special case of this theorem, given by setting $a_1 = \cdots = a_k = b_1 = \cdots = b_m = 1$.

4.11 Final Comments

The new version of enumeration schemes described in this chapter gives a way to count words avoiding *any* pattern of length up to 3. Further, the simple structure of monotone patterns makes it easy to track occurrences of subpatterns. Thus, words avoiding *any* monotone pattern are *guaranteed* to be counted by this method.

Chapter 5

Enumeration Schemes for Permutations Avoiding Barred Patterns

5.1 Introduction

In this chapter, we give a comprehensive list of enumeration results for permutations that avoid barred patterns. Then, we modify the notion of enumeration schemes to count permutations avoiding barred patterns. We first recall several definitions from Chapter 1.

Let $q' \in S_m$, $b \in \{0,1\}^m$. The *barred permutation* q is the permutation obtained by copying the entries of q' and putting a bar over q'_i if and only if $b_i = 1$. Write $\overline{S_m}$ for the set of all barred permutations of length m.

Let $p \in S_n$, $q \in \overline{S_n}$. Given q, let $\overline{q}$ be the permutation formed by all numbers of q, with or without bars, and let $\underline{q}$ be the permutation formed by deleting all barred letters of q and then reducing all the remaining (unbarred) letters. We say p *contains* q as a barred pattern if every instance of $\underline{q}$ in p is part of an instance of $\overline{q}$ in p. In this case, we may say every instance of $\underline{q}$ *extends to* an instance of $\overline{q}$.

We we have seen, this variation of pattern avoidance also appears in several interesting applications.

- A permutation is 2-stack sortable if and only if it avoids 2341 and $3\overline{5}241$ [37].
- A permutation is forest-like if and only if it avoids the patterns 1324 and $21\overline{3}54$ [9]. These patterns also characterize locally factorial Schubert varieties [40].

Beyond the special cases of barred pattern avoidance relevant to these applications, little is known beyond the work of Callan, where he completely enumerates permutations avoiding a single pattern of length 4 with one bar [10], and deals with the special case of $\{3\bar{5}241\}$-avoiding permutations [11]. The goal of this chapter is to consider the problem of barred pattern avoidance in a more general and comprehensive context. We consider barred permutations of any length and with any number of bars. Several preliminary results are given, and we completely characterize permutations avoiding a barred pattern of length ≤ 5 before we modify the method of prefix enumeration schemes to the case of barred pattern avoidance, and discuss its success rate.

5.2 Enumeration

Before we consider results for specific sets of barred patterns, we derive a series of useful Lemmas.

Lemma 2. *Let $q \in \overline{S_m}$, such that* every *number of q is barred. Then $S_n(\{q\})$ is the set of permutations that contain $\overline{q}$.*

Proof. Notice that $\underline{q} = \emptyset$. That is for p to avoid q every instance of the empty permutation must be a part of a copy of $\overline{q}$, i.e. p contains $\overline{q}$. □

More specifically, this lemma illustrates that, in some sense, barred pattern avoidance bridges the gap from standard pattern avoidance (no bars) to standard pattern containment (all possible bars), with a range of intermediate cases. However, as the following Lemmas illustrate, a number of these intermediate cases may also be equivalent to standard pattern avoidance.

Lemma 3. *Let $q \in \overline{S_m}$ such that q_i is the only barred element and either (i) $q_{i+1} = q_i \pm 1$, or (ii) $q_{i-1} = q_i \pm 1$. Then,*

$$S_n(\{q_1 \cdots q_{i-1}\overline{q_i}q_{i+1} \cdots q_m\}) = S_n(\{red(q_1 \cdots q_{i-1}q_{i+1} \cdots q_m)\})$$

Proof. Clearly, if p avoids $q_1 \cdots q_{i-1}q_{i+1} \ldots q_m$, then it avoids $q_1 \cdots q_{i-1}\overline{q_i}q_{i+1} \cdots q_m$ since there are no instances of $q_1 \cdots q_{i-1}q_{i+1} \ldots q_m$ to extend to an instance of q. Thus,

$$S_n(\{red(q_1 \cdots q_{i-1}q_{i+1} \cdots q_m)\}) \subseteq S_n(\{q_1 \cdots q_{i-1}\overline{q_i}q_{i+1} \cdots q_m\})$$

On the other hand, without loss of generality assume that $q_i = q_{i+1} \pm 1$, p avoids $q_1 \cdots q_{i-1}\overline{q_i}q_{i+1} \cdots q_m$, and there is an instance of $\underline{q} = red(q_1 \cdots q_{i-1}q_{i+1} \cdots q_m)$ that extends to an instance of $\overline{q} = q_1 \cdots q_{i-1} q_i q_{i+1} \cdots q_m$. Choose the instance of $\overline{q}$ that uses the leftmost possible element of p for q_i. Then $q_1 \cdots q_{i-1} q_i q_{i+2} \cdots q_m$ is another instance of $\underline{q}$ that does not extend to $\overline{q}$. So every element of $S_n(\{q_1 \cdots q_{i-1} \overline{q_i} q_{i+1} \cdots q_m\})$ already avoids $red(q_1 \cdots q_{i-1} q_{i+1} \cdots q_m)$. $\square$

Finally, we eliminate the case of having bars on all but one letter by the following observation.

Lemma 4. *Suppose that $q \in \overline{S_m}$ with only one unbarred letter. Then $|S_n(\{q\})| = 0$ for all $n \geq 1$.*

Proof. Notice that avoiding q means that every instance of a 1 pattern extends to an instance of $\overline{q}$. Without loss of generality, assume that q has barred entries after the lone unbarred letter. Then the final entry of any permutation is a copy of 1 that does not extend to $\overline{q}$. $\square$

We now consider permutations avoiding barred patterns of length 1, 2, 3, 4, and 5 in turn, noting that many results follow almost directly from Lemmas 2 and 3. With the exception of the work of Callan [10] for patterns of length 4 with 1 bar, this is the first comprehensive list of such results.

5.2.1 Avoiding barred patterns of length 1 or 2

We begin with avoiding patterns of length 1.

It is well known that $|S_n(\{1\})| = \begin{cases} 1 & n = 0 \\ 0 & n \geq 1 \end{cases}$.

We now see from Lemma 2 that $\left|S_n(\{\overline{1}\})\right| = \begin{cases} 0 & n = 0 \\ n! & n \geq 1 \end{cases}$.

For patterns of length 2, we observe that the Wilf equivalences

$$|S_n(\{q\})| = |S_n(\{q^r\})| = |S_n(\{q^c\})| = \left|S_n(\{q^{-1}\})\right|$$

extend to barred patterns in the obvious way, where q^r denotes q reverse, q^c denotes q complement, and q^{-1} denotes q inverse [31].

Thus, we already have $|S_n(\{12\})| = |S_n(\{21\})| = 1, n \geq 0$.

Further, by Lemma 2, we have $\left|S_n(\{\overline{12}\})\right| = \left|S_n(\{\overline{21}\})\right| = n! - 1$.

Finally, $\left|S_n(\{\bar{1}2\})\right| = \left|S_n(\{2\bar{1}\})\right| = \left|S_n(\{1\bar{2}\})\right| = \left|S_n(\{\bar{2}1\})\right| = |S_n(\{1\})|$, where the first and third equalities are by reversal, the second equality is by complement, and the final equality is by Lemma 3.

5.2.2 Avoiding barred patterns of length 3

It is well known that $|S_n(\{q\})| = \frac{\binom{2n}{n}}{n+1}$ where q is any unbarred pattern of length 3.

Thus, $|S_n(\{\overline{q}\})| = n! - \frac{\binom{2n}{n}}{n+1}$ where $\overline{q}$ is any pattern of length 3 with all bars.

By Lemma 4 it only remains to consider the case of patterns with 1 bar. The trivial Wilf equivalences and Lemma 3 give:

$$\left|S_n(\{\bar{1}23\})\right| = \left|S_n(\{32\bar{1}\})\right| = \left|S_n(\{12\bar{3}\})\right| = \left|S_n(\bar{3}21)\right| = \left|S_n(21)\right|, \text{ and}$$

$$\left|S_n(\{1\bar{2}3\})\right| = \left|S_n(3\bar{2}1)\right| = \left|S_n(21)\right|$$

It is enough to consider the pattern 132 with bars on various elements to complete the characterization. If there is a bar on 3 or 2, we may make use of Lemma 3, so the remaining interesting case is that of $S_n(\{\bar{1}32\})$.

Proposition 5. $\left|S_n(\{\bar{1}32\})\right| = (n-1)!$ *for all* $n \geq 1$.

Proof. We claim that $S_n(\{\bar{1}32\})$ is precisely the set of permutations that begin with the letter 1, thus giving the above enumeration.

First, notice that if π begins with 1, then π_1 cannot be involved in a 21 pattern since it is both the first and the smallest element of π. Thus, every 21 pattern is preceded by the smallest element, and so $\pi \in S_n(\{\bar{1}32\})$.

Now, if π does not begin with 1, then $\pi_1 1$ forms a 21 pattern that is not preceded by a smaller entry so $\pi \notin S_n(\{\bar{1}32\})$.

□

We have now finished the enumeration of permutations avoiding barred patterns of length ≤ 3.

5.2.3 Avoiding barred patterns of length 4

It is well known that for patterns with no bars, permutation patterns fall into the three classes of $|S_n(\{1234\})|$, $|S_n(\{1342\})|$, and $|S_n(\{1324\})|$. As we saw in Chapter 1, for the first of these, we have a closed form enumeration, for the second a generating function, and for the third a recurrence that allows enumeration up to $n = 20$.

As given by the Lemmas, we need only consider the case of 2 bars and 1 bar in turn.

For two bars, we have two cases: either within the pattern we have a symmetry of a pair of consecutive numbers of the form $(c-1)\bar{c}$, or not.

If we have a symmetry of $(c-1)\bar{c}$ in the pattern, it will be equivalent to avoiding a simpler pattern, similar to the argument of Lemma 3. So we need only consider the patterns where this does NOT happen. These are the patterns $\bar{1}\bar{2}43$, $\bar{1}32\bar{4}$ and their symmetries.

Proposition 6. $\left|S_n(\{\bar{1}\bar{2}43\})\right| = \left|S_n(\{\bar{1}32\bar{4}\})\right| = (n-2)!$ *for all* $n \geq 2$

Proof. It is enough to show that $S_n(\{\bar{1}\bar{2}43\})$ is precisely the set of permutations that begin with the letters 12, and that $S_n(\{\bar{1}32\bar{4}\})$ is precisely the set of permutations that begin with the letter 1 and end with the letter n.

For the first, if π begins with the letters 12, clearly neither of these letters is involved in a 21 pattern, and every 21 pattern is preceded by the smaller 12, thus $\pi \in S_n(\{\bar{1}\bar{2}43\})$. On the other hand, if π starts with the letters 21, it clearly contains the forbidden pattern 21 not preceded by a smaller 12 pattern. Also, if one of the first two letters of π is ≥ 3 then that letter is part of a forbidden 21 pattern that is not preceded by a smaller 12 pattern, so we are done.

For the case of $S_n(\{\bar{1}32\bar{4}\})$, if π begins with the letter 1 and ends with the letter n, then neither of these can be involved in a 21 pattern, so every 21 pattern is preceded by a smaller number (the 1) and succeeded by a larger number (the n). Thus, $\pi \in S_n(\{\bar{1}32\bar{4}\})$. On the other hand, if π begins with something other than 1, then $\pi_1 1$ forms a 21 pattern that is not preceded by a smaller number. Similarly if π ends with something other than n, then $n\pi_n$ forms a 21 pattern that is not succeeded by a larger number. $\square$

Finally, we consider the case of barred patterns of length 4 with precisely one bar. This was first comprehensively studied by Callan [10]. The following propositions are proved in a similar way to Callan's work, but with slightly modified notation, and are included for completeness.

Callan showed that permutations avoiding a barred pattern of length 4 with exactly one bar fall into 4 categories. By Lemma 3, 64 of these 96 ($= 4! \times 4$) patterns are equivalent to avoiding an unbarred pattern of length 3, thus yielding the Catalan numbers. The remaining 3 cases are those for which the sequence $\{|S_n(\{q\})|\}_{n\geq 0}$ gives the Bell numbers, OEIS Sequence A051295 [32], and a third case with a new sequence. The data in Table 5.1, first computed by Callan [10], lists the 32 remaining patterns, grouped by Wilf equivalence class.:

Representative	**Other Class Members**	**Sequence**
$1\bar{4}23$	$134\bar{2}, 23\bar{1}4, \bar{2}431, \bar{3}124, 32\bar{4}1, 4\bar{1}32, 421\bar{3}$	Bell
$\bar{2}413$	$2\bar{4}13, 24\bar{1}3, 241\bar{3}, \bar{3}142, 3\bar{1}42, 31\bar{4}2, 314\bar{2}$	Bell
$\bar{1}423$	$\bar{1}342, 241\bar{4}, 243\bar{1}, 312\bar{4}, 324\bar{1}, \bar{4}132, \bar{4}213$	A051295
$\bar{1}324$	$132\bar{4}, 423\bar{1}, \bar{4}231$	A051295
$\bar{1}432$	$234\bar{1}, 321\bar{4}, \bar{4}123$	new

Table 5.1: The equivalence classes of patterns of length 4 with one bar

We consider one representative from each of these classes. Other patterns that yield the same sequence can be studied using similar methods.

Proposition 7. $\left|S_n(\{1\bar{4}23\})\right|$ *satisfies the recurrence*

$$\left|S_n(\{1\bar{4}23\})\right| = \sum_{i=1}^{n} \binom{n-1}{i-1} \left|S_{n-i}(\{1\bar{4}23\})\right|$$

Proof. Let i be the position of the letter n in a $1\bar{4}23$-avoiding permutation. Then, the $i-1$ letters preceding n must be in decreasing order (otherwise $j < k < n$ forms a 123 pattern without a larger element between the j and k). The $n-k$ letters after n may be in any order, so long as they avoid $1\bar{4}23$. This gives a typical graph of a $1\bar{4}23$-avoiding permutation, considered as a function from $[n]$ to $[n]$ as in Figure 5.1.

There are $\binom{n-1}{i-1}$ ways to choose the initial k elements of the permutation, and $\left|S_{n-i}(\{1\bar{4}23\})\right|$ ways to order the last $n-i$ elements, so summing over all possible

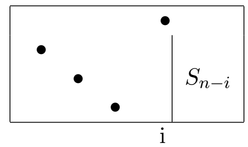


Figure 5.1: A generic $\{1\bar{4}23\}$-avoiding permutation

positions i for the entry n, we obtain the above recurrence.

This is the same recurrence satisfied by the Bell numbers. $\square$

Proposition 8. $\left|S_n(\{\bar{1}423\})\right|$ *satisfies the recurrence*

$$\left|S_n(\{\bar{1}423\})\right| = \sum_{i=1}^{n}(n-i)!\left|S_{i-1}(\{\bar{1}423\})\right|$$

Proof. Let i be the position of the letter 1. Then the $(n-i)$ entries following i may appear in any order. However, the letters before the 1 must all be smaller than the letters after the 1, otherwise $j1k$ with $j > k$ forms a 312 pattern without a smaller letter in front of it. The $i-1$ entries preceding i must merely avoid the forbidden pattern $\bar{1}423$, giving the graph of a typical $\bar{1}423$-avoiding permutation to be as in Figure 5.2.

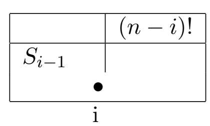


Figure 5.2: A generic $\{\bar{1}423\}$-avoiding permutation

There are $\left|S_{i-1}(\{\bar{1}423\})\right|$ ways to order the first $i-1$ elements, and $(n-i)!$ ways to order the last $n-i$ elements, so summing over all possible positions i for the letter n gives the above recurrence.

This recurrence gives sequence A051295 in the Online Encyclopedia of Integer Sequences. $\square$

Proposition 9.

$$\left|S_n(\{\bar{1}432\})\right| = (n-1)! + \sum_{j=2}^{n}\frac{(n-2)!}{(j-2)!} + \sum_{i=3}^{n}\sum_{j=2}^{n-i+2}\sum_{l=j+i-2}^{n}\frac{(n-i)!(l-j-1)!}{(l-i)!(i-3)!}$$

Proof. We break the set $S_n(\{\bar{1}432\})$ into cases depending on the location of the letter 1.

If 1 is the first letter of a permutation p, then clearly $p \in S_n(\{\bar{1}432\})$ since 1 as the first letter cannot be involved in a forbidden 321 pattern, and every 321 pattern is preceded by the 1. Thus, there are $(n-1)!$ permutations avoiding $\bar{1}432$ and beginning with 1.

If 1 is the second letter of a permutation p that begins with j, then we get the following graph:

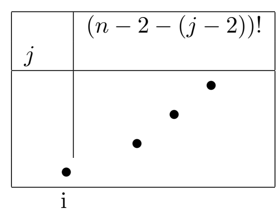


Figure 5.3: A $\{\bar{1}432\}$-avoiding permutation with 1 as the second letter

That is, all letters smaller than j must appear in increasing order (otherwise $j > a > b$ forms a 321 pattern without a smaller letter in front of it), so we may choose the positions of these letters but not their order. This can be done in $\binom{n-2}{j-2}$ ways. Further, the letters greater than j may appear in any order, but their positions are exactly the positions left over after choosing the places of the letters smaller than j. These larger letters can be ordered in $(n-2-(j-2))!$ ways. Summing over all possible values for j, we get the second term in the proposition.

Finally, we consider the case of 1 appearing in the third position or later. We obtain a permutation graph as in Figure 5.4.

That is, all letters before 1 must appear in increasing order, otherwise $a > b > 1$ is a 321 pattern not preceded by a smaller letter. If j is the smallest letter before 1 and l is the largest letter before 1, we may also conclude that

- The $n-l$ letters larger than l may appear in any order, so we may choose their positions in $\binom{n-i}{n-l}$ ways, and their order in $(n-l)!$ ways.

- The letters smaller than j must appear in increasing order, otherwise $j > a > b$

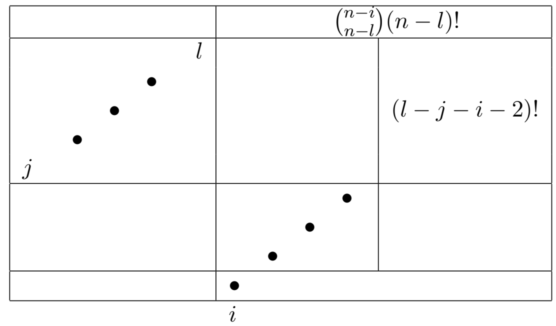


Figure 5.4: A $\{\bar{1}432\}$-avoiding permutation with 1 as the third letter or later

is a 321 pattern not preceded by a smaller letter.

- The letters smaller than j must appear strictly before the letters between j and l that are after the 1. Otherwise, let a be a letter $j < a < l$ that occurs before letter b, with $b < j$. Then lab is a 321 pattern not preceded by a smaller letter.

- Now, the positions of the remaining $(l - j - i - 2)$ letters are determined, and they can be ordered in $(l - j - i - 2)!$ ways.

Thus, summing over all possibilities for j, l, and i gives the third and final term in the proposition. □

5.2.4 Avoiding barred patterns of length 5

A comprehensive study of permutations avoiding patterns of length 5 is not yet completed, however, computational data shows that a number of new non-degenerate cases remain to be studied. We give a survey of computational data for $n \leq 7$ and patterns with 1, 2, or 3 bars.

The symmetries of reverse, complement, and inverse give 89 distinct equivalence classes for the sequence $|S_n(\{q\})|$ when q is a pattern of length 5 with one bar. Of these classes, 52 are equivalent to avoiding a pattern of length 4 by Lemma 3.

For the 37 remaining classes, computation suggests that there are at least 17 different possible sequences for $|S_n(\{q\})|$. 15 of these are new to the literature. Table 5.2 below sorts these non-degenerate results by their first 7 terms.

Pattern Class Representatives	**Sequence**	**OEIS Number**
$25\bar{3}14$, $35\bar{2}41$, $45\bar{3}12$, $51\bar{4}23$	1, 2, 6, 23, 104, 530, 2958	A117106
$352\bar{4}1$	1, 2, 6, 23, 104, 530, 2959	new
$1\bar{4}235$, $4\bar{2}513$	1, 2, 6, 23, 104, 531, 2977	new
$423\bar{1}5$, $425\bar{1}3$, $5314\bar{2}$(**)	1, 2, 6, 23, 104, 531, 2982	A110447
$\bar{4}2153$, $5\bar{1}423$	1, 2, 6, 23, 104, 532, 3002	new
$5134\bar{2}$	1, 2, 6, 23, 104, 532, 3003	new
$25\bar{1}34$(*), $25\bar{1}43$(*), $253\bar{1}4$, $\bar{3}5241$ $42\bar{5}13$, $43\bar{5}12$(*), $43\bar{5}21$(*)	1, 2, 6, 23, 104, 532, 3004	new
$1\bar{5}324$, $\bar{4}1523$	1, 2, 6, 23, 104, 532, 3005	new
$\bar{4}1253$	1, 2, 6, 23, 104, 533, 3026	new
$1\bar{5}234$, $412\bar{5}3$	1, 2, 6, 23, 104, 533, 3027	new
$\bar{1}3425$, $3524\bar{1}$	1, 2, 6, 23, 104, 533, 3038	new
$\bar{1}3245$, $3241\bar{5}$, $\bar{5}1432$, $\bar{5}3412$	1, 2, 6, 23, 104, 534, 3060	new
$\bar{5}1342$	1, 2, 6, 23, 104, 534, 3064	new
$\bar{5}2143$	1, 2, 6, 23, 104, 535, 3081	new
$5234\bar{1}$	1, 2, 6, 23, 104, 535, 3082	new
$\bar{5}1243$	1, 2, 6, 23, 104, 535, 3085	new
$\bar{5}1234$, $\bar{5}1324$	1, 2, 6, 23, 104, 535, 3088	new

(*) This sequence will be proven by the method of prefix enumeration schemes.
(**) This sequence has been proven by Callan in [11].

Table 5.2: Number of permutations avoiding a pattern of length 5 with one bar

Similarly, for patterns of length 5 with 2 bars, there are 172 equivalence classes and 150 of these reduce to avoiding an unbarred pattern of length 3. Of the 22 non-degenerate cases, we get at least 13 distinct sequences, 9 of these new to the literature. These sequences are given in Table 5.3.

Pattern Class Representatives	**Sequence**	**OEIS Number**
$2\bar{5}3\bar{1}4$, $\bar{3}5\bar{1}42$	1, 2, 5, 14, 43, 143, 509	A006789
$\bar{4}25\bar{1}3$, $5\bar{1}32\bar{4}$	1, 2, 5, 14, 43, 143, 510	A098569
$5\bar{1}\bar{2}43$(*)	1, 2, 5, 14, 43, 143, 511	new
$31\bar{5}\bar{4}2$(*), $415\bar{3}\bar{2}$	1, 2, 5, 14, 43, 144, 522	new
$\bar{3}\bar{1}542$	1, 2, 5, 14, 43, 144, 523	A047970
$\bar{2}413\bar{5}$, $425\bar{3}\bar{1}$, $42\bar{5}3\bar{1}$	1, 2, 5, 14, 43, 144, 525	new
$\bar{1}43\bar{5}2$	1, 2, 5, 14, 43, 145, 538	A122993
$\bar{1}5\bar{2}43$	1, 2, 5, 14, 43, 146, 550	new
$\bar{2}\bar{1}453$, $243\bar{1}\bar{5}$, $423\bar{1}\bar{5}$, $\bar{5}\bar{4}231$(*)	1, 2, 5, 14, 43, 146, 561	new
$\bar{5}324\bar{1}$, $\bar{5}\bar{4}132$(*)	1, 2, 5, 14, 43, 147, 575	new
$\bar{4}\bar{5}123$	1, 2, 5, 14, 43, 147, 578	new
$\bar{1}432\bar{5}$	1, 2, 5, 14, 43, 148, 592	new
$345\bar{2}\bar{1}$	1, 2, 5, 14, 43, 150, 617	new

(*) This sequence will be proven by the method of prefix enumeration schemes later in this chapter.

Table 5.3: Number of permutations avoiding a pattern of length 5 with two bars

Finally, for patterns of length 5 with 3 bars, all cases are degenerate to either $S_n(\{q\}) = 1$ or $S_n(\{q\}) = (n-3)!$.

Now that we have exhausted comprehensive case by case analysis of permutations avoiding a single barred permutation, we consider a method to compute recurrences for $S_n(Q)$ where Q is an arbitrary set of barred permutation patterns.

5.3 Enumeration Schemes

Our goal in this section is to introduce a single method that works to enumerate the elements of many classes $S_n(Q)$ where Q is a set of barred permutation patterns. As for pattern-avoiding words, we derive an algorithm whose input is a set of permutation patterns Q, and whose output can be read as a recurrence counting the elements of $S_n(Q)$. Notation from the Zeilberger's and Vatter's original work with unbarred permutation patterns will be adapted as necessary.

In the following sections, we discuss in turn the notions of *refinement*, *reversibly deletable elements*, *gap vectors*, and *stop points*. These four concepts are combined to form an *enumeration scheme*, or recurrence computing $|S_n(Q)|$.

5.3.1 Refinement

Since the set $S_n(Q)$ may be complicated, we first partition $S_n(Q)$ into disjoint subsets and look for recurrences between these subsets.

As we have seen, one natural and useful way to partition the permutations of $S_n(Q)$ is by the patterns formed by the first i letters of a permutation. The following notation will be useful to discuss this partitioning of $S_n(Q)$:

$$S_n(Q; p_1 \cdots p_i) = \{\pi \in S_n \mid \pi \text{ avoids } q \text{ for all } q \in Q, \pi_1 \cdots \pi_i \text{ reduces to } p_1 \dots p_i\}$$

$$S_n\left(Q; \begin{matrix} p_1 \cdots p_l \\ l_1 \cdots l_i \end{matrix}\right) = \left\{ \pi \in S_n \;\middle|\; \begin{array}{l} \pi \text{ avoids } Q, \\ \pi_1 \cdots \pi_i \text{ reduces to } p_1 \dots p_i, \text{ and} \\ l_1, \dots, l_i \text{ are the first } i \text{ letters of } \pi \end{array} \right\}.$$

For example, $S_3(\{132\}; 12) = \{123, 231\}$, i.e. of the 5 permutations of length 3 that avoid the pattern 132, only 123 and 231 begin with an increasing pair of letters.

Similarly, $S_3\left(\{132\}; \begin{matrix} 12 \\ 23 \end{matrix}\right) = \{231\}$.

Given $p = p_1 \cdots p_i$, a *refinement* of p is a permutation $q = q_1 \cdots q_{i+1}$ such that $red(q_1 \cdots q_i) = p$. For example, the refinements of $\emptyset$ are $\{1\}$. The refinements of 1 are $\{12, 21\}$. The refinements of 12 are $\{123, 132, 231\}$.

As before, we have the following useful observation:

$$S_n(Q; p) = \bigcup_{q \in \{\text{refinements of } p\}} S_n(Q; q),$$

and thus

$$|S_n(Q; p)| = \sum_{q \in \{\text{refinements of } p\}} |S_n(Q; q)|.$$

Thus, for any set of patterns Q, we have $S_n(Q) = S_n(Q; 1) = S_n(Q; 12) \cup S_n(Q; 21)$, etc.

This partitioning of $S_n(Q)$ into disjoint sets depending on the initial few letters is drawn from the work of Zeilberger [42].

Graphically, we may represent refinement using a graph, where the vertices correspond to the sets $S_n(Q;p)$, and there is a directed edge from a prefix pattern to each of its refinements. To count the elements of $S_n(Q)$ it is enough to count the elements of the subsets $S_n(Q;p)$ represented by the leaves of the graph.

An example of such a graph of refinements is given in Figure 5.5.

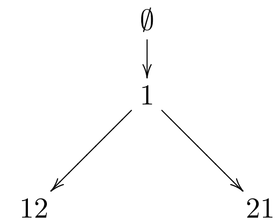


Figure 5.5: The graph of refinements for an arbitrary pattern set

Now that we have a way to partition $S_n(Q)$ into disjoint subsets, we consider ways to find recurrences between these subsets.

5.3.2 Reversibly Deletable Elements

The key tool for finding recurrences between $S_n(Q;p)$ for various prefixes p is the following:

Definition 13. *Given Q, a set of barred permutation patterns, and p, a prefix of length l, $l > 0$, we say that position r $(1 \leq r \leq l)$ is* reversibly deletable *if and only if the action of removing p_r from a Q-avoiding permutation of length n and inserting p_r into a Q-avoiding permutation of length $n-1$ is a bijection between $S_n\left(Q; \begin{matrix} p_1 \cdots p_l \\ i_1 \cdots i_l \end{matrix}\right)$ and $S_{n-1}\left(Q; \begin{matrix} p_1 \cdots p_{r-1}p_{r+1} \cdots p_l \\ i_1 \cdots i_{r-1}i_{r+1} \cdots i_l \end{matrix}\right)$.*

In the case of unbarred pattern avoidance, it is enough to check that the insertion of p_r into a Q-avoiding permutation of length $n-1$ gives a Q-avoiding permutation of length n, since the deletion of a letter from a permutation cannot cause a bad pattern. More specifically, for unbarred pattern avoidance, p_r is reversibly deletable if and only if every forbidden pattern involving p_r implies the existence of a forbidden pattern without p_r.

For example, if $Q = \{123\}$, and $p = 21$, we have that $p_1 =$"2" is reversibly deletable, since the only way for p_1 to be involved in a 123 pattern is for there to be $21 \cdots p_s \cdots p_t$ with "2"$< p_s < p_t$. But this means that "1"$< p_s < p_t$, and $1p_sp_t$ forms a forbidden 123 pattern without $p_1 =$"2". That is, every 123 pattern involving p_1 implies the existence of a 123 pattern without p_1. Thus, it is impossible to create a $\{123\}$-containing permutation by inserting p_1 into a $\{123\}$-avoiding permutation. So inserting and deleting p_1 is indeed a bijection between $\{123\}$-avoiding permutations of length $n-1$ and $\{123\}$-avoiding permutations of length n.

For the case of barred patterns, the definition of reversibly deletable elements is equivalent to the old definition with the added caveat that p_r cannot be the only letter to play the role of a barred letter in extending a forbidden $\underline{q}$ pattern to $\overline{q}$. That is, deleting p_r from a Q-avoiding permutation can only fail to produce another Q-avoiding permutation if p_r plays the role of a barred letter and its removal makes an instance of a forbidden pattern $\underline{q}$ no longer extendable to the larger barred pattern $\overline{q}$.

In summary, to check algorithmically that p_r is reversibly deletable, we must check two things.

1. Check that *inserting* p_r into a Q-avoiding permutation always produces a Q-avoiding permutation.

2. Check that *deleting* p_r from a Q-avoiding permutation always produces a Q-avoiding permutation.

We discuss how to rigorously check each of these in turn.

1. **Insertion**

   For insertion, as in the unbarred case, we check that every possible occurrence of a forbidden pattern with p_r implies the existence of a forbidden pattern without p_r. This is easily seen to happen in a finite number of scenarios. First, choose the letters of the prefix p (including p_r) that will be involved in the forbidden pattern.

Then, choose all the ways that the remaining letters of the forbidden pattern can be spaced between the letters of p.

For example, for $Q = \{\bar{1}423\}$, a forbidden pattern is a 312 pattern without a smaller letter before it. Consider $p = 123$, and reversibly deletable candidate $p_2 =$"2". Recall that p is a prefix, denoting that the first three letters of our permutation are increasing, not that they are specifically the letters 1, 2, and 3. So the only way for p_2 to be involved in a bad pattern is for p_2 to be followed by a smaller increasing pair. These letters may be (a) both less than p_1, (b) one less than p_1 and one greater than p_1, or (c) both greater than p_1 and less than p_2, as in the permutation graphs in Figure 5.6 below. As before, we use $\circledast$ to mark p_2 to be deleted, $\blacksquare$ to denote the letters of $123ab$ that, together with p_2, form a forbidden pattern, and $\boxdot$ to mark another letter that, together with the letters marked $\blacksquare$, forms a forbidden pattern without p_2.

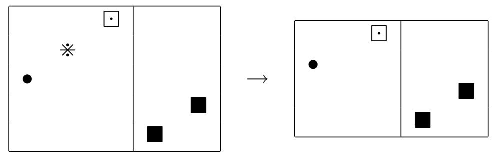
We quickly eliminate case (c) since $123ab$ where $2ab$ is a 312 pattern, and "1"$<$ $a < b <$"2", actually extends to the 1423 pattern $12ab$. Now, it is easy to check that deleting p_2 in each of cases (a) and (b) gives another $\{\bar{1}423\}$-containing permutation.

Additionally, to check that p_r is reversibly deletable for prefix p and forbidden pattern q where q has b bars, we must also check scenarios with b additional letters.

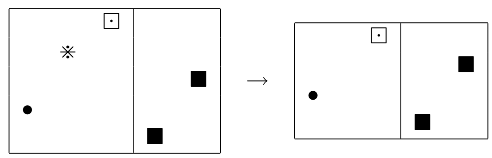
These additional scenarios are indeed necessary. For example, let $Q = \{134\bar{2}\}$, and $p = 21$, and consider p_1. p_1 being involved in a bad pattern with two letters after the prefix may happen in one way, namely:

So, $21ab$ containing the forbidden pattern "2"$< a < b$ *does* imply that "1"$< a < b$ is a forbidden pattern without p_1. It seems from this that p_1 is reversibly deletable. However, $21abc$ containing the forbidden pattern $2ab$ does not imply that $1abc$ is bad (if "1"$< c <$"2"), since c may act as a barred letter extending the $1ab$ pattern, but not the $2ab$ pattern.

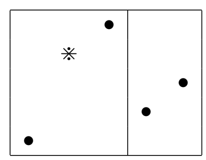
To show that this is always enough, note that if π contains a forbidden pattern $\underline{q}$



Case (a): both post-prefix letters less than p_1



Case (b): one post-prefix letter less than p_1 and one greater than p_1



Case (c): both post-prefix letters greater than p_1 and less than p_2

Figure 5.6: An example of checking that insertion is bijective

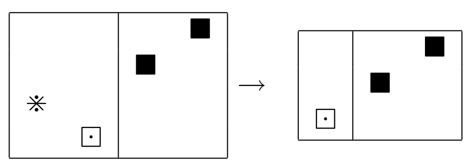


Figure 5.7: A $\{134\overline{2}\}$-avoiding permutation with prefix 21

where q has b bars, then only b letters need be inserted to form a copy of the $\overline{q}$, so adding even more letters is redundant.

2. **Deletion**

   Now that we have rigorously shown that insertion of p_r is a map from Q-avoiding permutations to Q-avoiding permutations, we check that deletion also has this property.

   To do this, we check the scenarios for a forbidden pattern involving p_r as above. Namely, we want to show that if π^* begins with prefix $p^* = p_1 \cdots p_{r-1} p_{r+1} \cdots p_l$

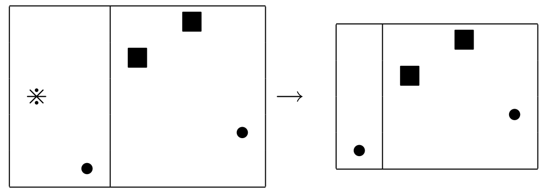


Figure 5.8: A $\{134\bar{2}\}$-containing permutation with prefix 21

and has a forbidden pattern, then π, beginning with $p = p_1 \cdots p_r \cdots p_l$ has a forbidden pattern. Thus, if we compute all the scenarios beginning with p^*, insert p_r, and check that each one contains a forbidden pattern, then we are done.

For example, if $Q = \{\bar{1}423\}$ and $p = 123$, we again consider p_2. There are a number of ways for $p_1 p_3$ to be involved in a forbidden pattern that does *not* extend to 1423, namely:

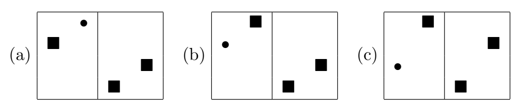


Figure 5.9: $\{\bar{1}423\}$-containing patterns with prefix 12.

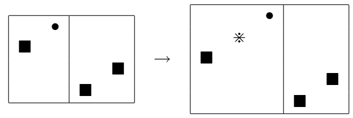
Now, p_2 can be inserted into each of these scenarios in possibly multiple ways as in Figure 5.10.

We may inspect that each of these resulting permutations contains a 312 pattern that does *not* extend to a 1423 pattern, and thus, p_2 is reversibly deletable.

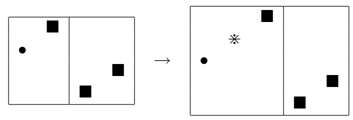
To show that p_r is reversibly deletable graphically, we draw a dotted arrow from p to p^* labelled with d_r, which denotes the deletion map of deleting the rth letter of π and reducing. For example, if $p = 21$ had p_1 reversibly deletable, we would encode this as in Figure 5.11

5.3.3 Gap Vectors

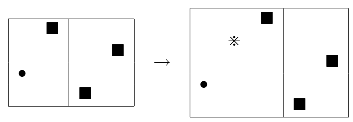
Unfortunately, as for words, the reversibly deletable elements are usually not sufficient to find recurrences counting the elements of $S_n(Q)$, so following Vatter [36], we introduce



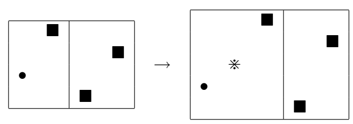
Case (a): inserting p_2 ($p_1 < p_2 < p_3$) into a $\{\bar{1}423\}$-avoiding permutation



Case (b): inserting p_2 ($p_1 < p_2 < p_3$) into a $\{\bar{1}423\}$-avoiding permutation



Case (c1): inserting p_2 ($p_1 < p_2 < p_3$) into a $\{\bar{1}423\}$-avoiding permutation



Case (c2): inserting p_2 ($p_1 < p_2 < p_3$) into a $\{\bar{1}423\}$-avoiding permutation

Figure 5.10: An example of checking that deletion is bijective

the notion of gap vectors. Given a set of forbidden patterns Q and prefix $p = p_1 \cdots p_i$, a *spacing vector* v is a vector in $\mathbb{N}^{i+1}$. We write $||v||$ to denote the sum of the entries of v. Spacing vectors help further narrow down the set $S_n(Q; p)$ into smaller subsets in the following way.

Definition 14. *Given Q, a set of forbidden patterns, p a prefix of length i, and v, a spacing vector of length $l + 1$, let $s_1 \cdots s_l$ be the permutation obtained by sorting p. $S_n(Q; p; v)$ denotes the set of permutations of length n, avoiding Q, beginning with prefix p, and with exactly v_1 letters smaller than s_1, exactly v_j letters that are greater than s_{j-1} and smaller than s_j, and exactly v_{i+1} letters that are greater then s_l.*

For example, $S_n(\{123\}; 12; \langle 0, 1, 2 \rangle)$ denotes the set of permutations avoiding 123, beginning with an increasing pair of letters $p_1 p_2$, with one letter a such that $p_1 < a < p_2$ and two letters bigger then p_2.

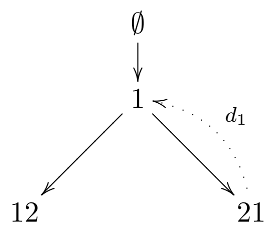


Figure 5.11: Representation of reversibly deletable elements

Definition 15. *A spacing vector v is a* gap vector *for $[Q, p]$ if there are no permutations avoiding Q with prefix p and spacing vector $\geq v$ (componentwise).*

For example, if $Q = \{123\}$ and $p = 12$, then $v = \langle 0, 0, 1\rangle$ is a gap vector since if we have at least one letter a larger than p_1 and p_2, then $p_1 p_2 a$ forms a 123 pattern.

We check this similarly to the unbarred case, with yet another constraint.

To check that v is a gap vector in the unbarred case, we consider permutations starting with p, and name v_1 letters smaller than $s_1 = 1$ (say $\frac{1}{v_1+1}, \ldots, \frac{v_1}{v_1+1}$), v_2 letters between $s_1 = 1$ and $s_2 = 2$ (say $1 + \frac{1}{v_2+1}, \ldots 1 + \frac{v_2}{v_2+1}$), $\ldots$, and v_{l+1} letters bigger than $s_l = l$ (say $l + \frac{1}{v_{l+1}+1}, \ldots, l + \frac{v_{l+1}}{v_{l+1}+1}$). Now, consider all permutations that begin with p and end with any of the $(v_1 + \cdots + v_{l+1})!$ permutations of these fractional letters. If *each* of these permutations contains a forbidden pattern from Q, then v is a gap vector for $[Q, p]$.

In the barred case, while the algorithm in the previous paragraph is necessary to show that v is a gap vector, it is no longer sufficient. For example, when avoiding $Q = \{23\bar{1}\}$, with prefix $p = 1$, $v = \langle 0, 1\rangle$ appears to be a gap vector when considering permutations of length 2. However, there are permutations of length 3 with spacing vector $v^* = \langle 1, 1\rangle$ that avoid Q. That is, we may have a vector such that $|S_n(Q; p; v)| = 0$, but there is some $w > v$ (componentwise) such that $|S_n(Q; p; w)| > 0$. However, we want to find a basis for the set of vectors v such that $|S_n(Q; p; w)| = 0$ for all n and for all $w \geq v$.

In light of this complication, to show that v is a gap vector for $[Q, p]$, not only do we need to confirm that there are no Q-avoiding permutations with prefix p and spacing v, but also that there are no Q-avoiding permutations with spacing w, $w > v$

componentwise. More precisely, if we are concerned with finding the basis of gap vectors for $[Q, p]$, most of the time we proceed as in the unbarred case, with one important exception. More work needs to be done to check for gap vectors when avoiding a pattern of the form $q = q_1 \cdots q_{m-i-1}\overline{q}_{m-i} \cdots \overline{q}_m$. We begin with the case when $i = 0$, i.e. q ends with exactly one barred letter.

Theorem 22. *Let $q \in \overline{S}_m$ such that $\underline{q} = q_1 \cdots q_{m-1}$. Then there are* no *basis gap vectors for $[\{q\}, p]$ for any prefix p.*

Proof. Assume that q is as in the proposition, and that v is a basis gap vector for $[\{q\}, p]$. Now, let $\pi = \pi_1 \cdots \pi_l \in S_{|p|+\|v\|}$ that has prefix p. Since v is a basis gap vector, π contains q, but if the last letter of π is deleted, then it avoids q. That is, by definition of basis gap vector, the last letter of π is involved in a forbidden $\underline{q}$ pattern that does not extend to a $\overline{q}$ pattern.

For each instance of $\underline{q}$ in π, choose a letter to append to π which will extend the instance to $\overline{q}$; write L for the set of such letters to be appended to π. Without loss of generality, assume that $q_{m-1} < q_m$. Then append the letters of L to π in increasing order, and call the resulting permutations π^*. We claim that either π^* is a $\{q\}$-avoiding permutation or can be further extended to be $\{q\}$-avoiding, with prefix p and spacing w, $w > v$, so v is not a gap vector, and by contradiction we are done.

To see that $\pi^* = \pi_1^* \cdots \pi_{l_2}^*$ is $\{q\}$-avoiding, we consider several cases.

First, by Lemma 3, we may assume that $\overline{q}$ is not a monotone pattern and that there exists q_c with $q_{m-1} < q_c < q_m$.

Construct π^* by appending each letter of L individually. Suppose that $\pi_1 \cdots \pi_{l+i}$ contains a forbidden $\underline{q}$ pattern that uses π_{l+i}. Then either:

- The rest of the forbidden pattern consists of letters π_j with $j \leq l$. In this case, the letter of L that was meant to extend the bad pattern formed by replacing π_{l+i} with π_l has yet to be appended to π, and will extend this instance of $\underline{q}$ to an instance of $\overline{q}$ as well.

- If the rest of the forbidden pattern consists of both letters from $\pi_1 \cdots \pi_l$ and letters from L, then we note that by Lemma 3, there exists π_c in the instance of the forbidden pattern with $\pi_{l+j} < \pi_c < \pi_{l+i}$, $j < i$, $c < l$, so we may append another letter π_{l+i+1} to π^* extending this to a copy of $\overline{q}$. Again, there must be π_{c_2} with $c_2 < l$ so that $\pi_{l+i} < \pi_{c_2} < \pi_{l+i+1}$. If this letter π_{l+i+1} is involved in another instance of $\underline{q}$, repeat. We know this process terminates because there are a finite number of letters in π, so there is a maximum letter of $\{\pi_1, \ldots, \pi_l\}$ to play the role of π_{c_i} in this construction.

In both cases, we have shown that it is possible to append enough letters to π to make it $\{q\}$-avoiding, and thus there are no gap vectors for $[\{q\}, p]$. $\square$

As an example of this construction, consider permutations that avoid the pattern $q = 241\overline{3}$. For the prefix 123, there are no permutations avoiding q with spacing $\langle 1, 0, 0, 0\rangle$, and there are no permutations avoiding q with a spacing vector of weight 2; however, we can construct a permutation with prefix $p = 123$ and a spacing vector of weight 3 that avoids q. Notice that in the language of the proposition, $\pi = 2341$, which contains $\overline{q} = 231$ in several places, namely, 231, 241, and 341. Thus, we require the addition of a letter greater than "2" and less than "3", a letter greater than "2" and less than "4", and a letter greater than "3" and less than "4", to extend each of these copies of $\underline{q}$ to a copy of $\overline{q}$. Choosing two letters, a and b with "2" $< a <$ "3" $< b <$ "4" suffices. We append a and b to the end of π to obtain $246135 \in S_n(\{241\overline{3}\}; 123; \langle 1, 1, 1, 0\rangle)$. Note that 246135 is $\{241\overline{3}\}$-avoiding, with prefix 123 and a spacing vector which is greater than $\langle 1, 0, 0, 0\rangle$.

We note that Theorem 22 is necessary. In general, we can find p and q so that the smallest weight of a spacing vector v where $S_n(\{q\}; p; v) \neq \emptyset$ is arbitrarily large, and checking all the appropriate scenarios would be time-consuming. With this Theorem, we need not consider all of these scenarios, but rather return that the set of gap vectors for $[\{q\}, p]$ is empty. Since gap vectors were included in the scheme algorithm to help find recurrences, eliminating gap vectors in this case may seem to limit the success of our algorithm. However, we note that via the symmetries of the square, if we cannot find an enumeration scheme for $S_n(\{q\})$ where q ends in a barred letter, we may still find an enumeration scheme for $S_n(\{q^r\})$ where q^r does not end in a barred letter.

Further, if q is part of a set of forbidden patterns Q, the other patterns not ending in a barred letter may still help find gap vectors for the enumeration scheme for $S_n(Q)$.

We also note that this construction does not necessarily generalize to patterns of the form $q = q_1 \cdots q_{m-i-1}\overline{q}_{m-i} \cdots \overline{q}_m$ where $i > 0$. For example, if $q = 351\overline{4}\overline{2}$ and $p = 231$, we note that $v = \langle 0,0,0,0 \rangle$ is a gap vector because two letters a and b, with $p_3 < b < p_1 < a < p_2$ must be appended to p to extend p to an instance of 35142. But now, p_1ab is a new forbidden 351 pattern that requires two letters c and d to be appended with $b < d < p_1 < c < a$ to extend p_1ab to an instance of 35142. This process continues indefinitely.

For the case of patterns which consist of a block of unbarred letters followed by a block of more than one barred letter, we must check extra scenarios to determine if v is a gap vector. Namely, if $S_n(Q;p;v) = \emptyset$, we must also check that $S_n(Q;p;w) = \emptyset$ for all $w > v$ with $\|w\| = \|v\| +$ (total number of bars in all patterns in Q)$\cdot$ (total number of occurrences of $q_1 \cdots q_{m-i-1}$ in p}) before concluding that v is in fact a gap vector.

With this more specific definition of gap vector, we have an ideal in $\mathbb{N}^{l+1}$ which necessarily has a finite basis. We have also exhibited a method to find basis vectors for a scheme. These serve to narrow down the cases we must consider to decide if an element is reversibly deletable.

Graphically, we write a basis for the gap vectors corresponding to p below p. For example, if $\langle 0,0,1 \rangle$ is a gap vector for the prefix 12, and this causes p_2 to be reversibly deletable, we would represent this situation as in Figure 5.12

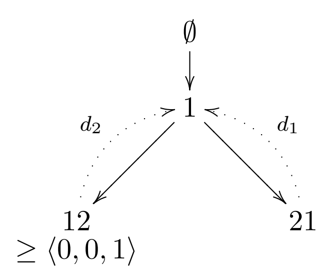


Figure 5.12: Representation of gap vectors

A final remark for gap vectors concerns the vector $v_0 = \langle 0,\ldots,0 \rangle$. Note that if v_0 is a gap vector for $[Q,p]$, then there are no permutations of any length avoiding Q and beginning with prefix p. Since the gap vector v_0 already indicates that $|S_n(Q;p)| = 0$, it is unnecessary to write $S_n(Q;p)$ in terms of smaller sets. However, for completeness of the definition of enumeration scheme (below), if v_0 is a gap vector, we allow any one of the p_i to be "exceptionally" reversibly deletable. We will return to this remark later.

5.3.4 Stop Points

As we have observed with reversibly deletable elements and with gap vectors, barred patterns require added considerations to find a rigorous enumeration schemes. While we have introduced enough notation to find recurrences between the subsets $S_n(Q;p)$, we require one extra tool to find the base cases for these recurrences, i.e. *stop points.*

The key observation is that there may be *no* permutations of length n that avoid Q and begin with prefix p, but there may be such permutations of length $n+k$ for some $k > 0$. For example, if $Q = \{23\bar{1}\}$, there are no permutations of length 2 avoiding Q, but 231 is a permutation of length 3, beginning with a 12 pattern. Thus, in the notation of the enumeration scheme, we require a mechanism to indicate at what length we may begin to consider permutations beginning with that prefix.

Definition 16. *Given a set of forbidden patterns Q, and a prefix p* without *reversibly deletable elements, we say $s \geq |p|$ is a stop point for $[Q,p]$ if there are* no *permutations of length $\leq s$ that avoid Q and begin with prefix p*

For example, the set of stop points for $(\{23\bar{1}\}, 12)$ is $\{2\}$.

Proposition 10. *Given Q and p, the set S of stop points is finite.*

Proof. Notice that since S is a set of positive integers, it is enough to show that S has a well defined maximum element.

It is enough to note that stop points are only defined for prefixes with no reversibly deletable elements. If there were *no* permutations beginning with prefix p, we would obtain the gap vector $\langle 0, \ldots, 0 \rangle$, and by convention, position 1 is reversibly deletable, so p by definition has an empty set of stop points.

Since p has no reversibly deletable elements, then, we know that there is a permutation π of minimal length that begins with p and avoids Q. The set of stop points has maximum $|\pi| - 1$. $\square$

The simplest example of a scheme that requires stop points is the scheme for permutations avoiding $\{123, 321, 23\bar{1}\}$. Graphically, we represent stop points as a set after an asterisk, listed next to the permutation prefix p, as by $p = 12$ in the scheme for $S_n(\{123, 321, 23\bar{1}\})$ in Figure 5.13

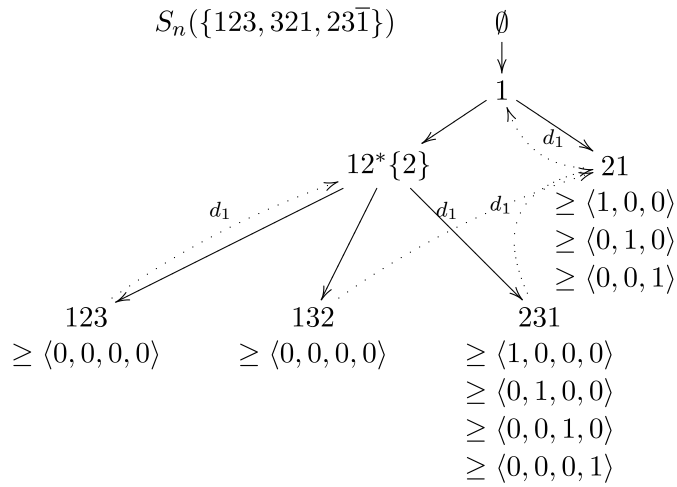


Figure 5.13: A scheme involving stop points

From this scheme, we have

- $|S_0(Q)| = 1$
- $|S_1(Q)| = 1$
- $|S_2(Q)| = |S_2(Q; 12)| + |S_2(Q; 21)| = 0 + |S_1(Q; 1)| = 0 + 1 = 1$
- $|S_3(Q)| = |S_3(Q; 123)| + |S_3(Q; 132)| + |S_3(Q; 231)| + |S_3(Q; 21)|$

$$= 0 + 0 + |S_2(Q; 21)| + 0$$
$$= 0 + 0 + |S_1(Q; 1)| + 0$$
$$= 0 + 0 + 1 + 0 = 1$$

- $|S_n(Q)| = 0$ for all $n \geq 4$

Without stop points, we would have computed $|S_2(Q)| = 2$.

5.3.5 Enumeration Schemes

Finally, we have all the necessary tools to algorithmically find recurrences counting the elements of $S_n(Q)$ where Q contains barred permutation patterns. More specifically:

Definition 17. *An* enumeration scheme $\mathbb{S}$ *is a set of* 4*-tuples* $t = [p_j, R_j, G_j, S_j]$ *such that for each* t*:*

- p_j *is a reduced prefix of length* i*.*
- R_j *a (possibly empty) subset of* $\{1, \ldots, i\}$*.*
- G_j *is a (possibly empty) set of vectors of length* $i + 1$*.*
- S_j *is a (possibly empty) finite set of positive integers whose minimum element is* $\geq |p_j|$
  and
- *either* R_j *is non-empty, or all refinements of* p_j *are also in the scheme.*

We have detailed how to find each of the elements of such a 4-tuple, namely if p_j is a prefix, denoting the set $S_n(Q; p_j)$, then R_j, G_j, and S_j are the corresponding reversibly deletable elements, set of gap vectors, and set of stop points.

The last condition ensures that the enumeration scheme can be read as a recurrence counting the elements of $S_n(Q)$. Recall that if R_j is non-empty, then we have a bijection between $S_n(Q; p)$ and $S_{n-1}(Q; p^*)$ for some p^*. If R_j is empty, then we require all refinements of p_j to be in the scheme for completeness.

Given an enumeration scheme $\mathbb{S}$ corresponding to pattern set Q, we can compute $|S_n(Q)|$ in the following way:

1. Let P be the set of p_j such that either (i) p_j is a prefix of length $\leq n$ with reversibly deletable elements or (ii) p_j is a prefix of length n without reversibly deletable elements. We have $|S_n(Q)| = \sum_{p_j \in P} |S_n(Q; p_j)|$.

2. For each $p_j \in P$, if $n \in S_j$, then we have $|S_n(Q; p_j)| = 0$.

3. For each remaining $p_j \in P$, associate the set of spacing vectors v_j^* of all vectors of length $|p_j| + 1$ and weight $n - |p_j|$ minus the set of gap vectors G_j. We have $|S_n(Q; p_j)| = \sum_{v \in v_j^*} |S_n(Q; p_j; v)|$.

4. For each $p_j \in P$, and $v \in v_j^*$, if R_j is non-empty, we have $|S_n(Q; p_j; v)| = \left|S_{n-1}(Q; p_j^*, v^*)\right|$ for prefix p_j^* (p_j with letter r deleted) and vector $v^* = \langle v_1, \ldots, v_{r-1} + v_{r+1}, \ldots, v_{n+1}\rangle$. If R_j is empty, then $\left|S_n(Q; p_j^*)\right| = 1$.

5.4 The Maple Package `bVATTER`

The algorithms both (i) to find a scheme and (ii) to read a scheme into a sequence have been programmed in the `Maple` package `bVATTER`, available from the author's website: `http://www.math.rutgers.edu/~lpudwell/maple.html`. The main functions are `SchemeImage`, `SeqS`, `Sipur`.

`SchemeImage` inputs a set of patterns Q, a maximum depth scheme to search for, and a maximum weight of gap vectors to search for, and outputs a concrete enumeration scheme for words avoiding Q of the specified maximum depth. If it cannot find a scheme for Q, it searches for a scheme for a symmetry-equivalent pattern set and returns that scheme instead.

`SeqS` inputs a scheme and an integer K, and uses the scheme to compute $|S_i(Q)|$ for $1 \leq i \leq K$.

`Sipur` inputs a list $[L]$ of pairs of integers, a maximum scheme depth, a maximum weight of gap vectors, and an integer K. It outputs all information about schemes for permutations avoiding one pattern of each length in L where each pair is of the form [length,number of bars]. For example, `Sipur([[4,1]],4,2,30)` outputs all information about permutations avoiding one pattern of length 4 with 1 bar. It will search for schemes of depth 4 with maximum gap vector weight 2 and will output the first 30 terms of the sequence $|S_i(Q)|$ given by each scheme it finds.

`Sipur` has been run on $[L]$ for various lists of the form $[[3, x_i]^a, [4, y_i]^b, [5, z_i]^c]$, and the output is available from the author's website.

5.5 Success Rate

In this section, we consider the success rate of prefix enumeration schemes for $S_n(Q)$ where Q is a set of barred permutation patterns.

Recall that sets of permutation patterns can be put into equivalence classes based on the permutation involutions of reverse, complement and inverse. We measure success in terms of the number of such equivalence classes for which there is an enumeration scheme. In the Table 5.4, pattern lengths denotes the lengths of patterns, as well as the number of bars. For example pattern lengths [4,0],[4,1] denotes two patterns of length 4, one without bars, and one with precisely one bar. Specific schemes for the data in the table can be found at the author's website. As for words, it should be noted that pattern sets that are counted as unsuccessful do not necessarily lack an enumeration scheme; they may have enumerations schemes of greater depth than the computer has searched.

Pattern Lengths	**Success Rate**	**Pattern Lengths**	**Success Rate**
[2,0]	1/1 (100%)	[3,0],[3,0],[3,0]	5/5 (100%)
[2,1]	1/1 (100%)	[3,0],[3,0],[3,1]	43/45 (95.6%)
[2,0],[2,0]	1/1 (100%)	[3,0],[3,0],[3,2]	45/45 (100%)
[2,1],[2,0]	2/2 (100%)	[3,0],[3,1],[3,1]	135/138 (97.8%)
[2,1],[2,1]	2/2 (100%)	[3,0],[3,1],[3,2]	280/280 (100%)
[3,0]	2/2 (100%)	[3,0],[3,2],[3,2]	138/138 (100%)
[3,1]	4/4 (100%)	[3,1],[3,1],[3,1]	115/118 (97.5%)
[3,2]	4/4 (100%)	[3,1],[3,1],[3,2]	378/378 (100%)
[3,0],[3,0]	5/5 (100%)	[3,1],[3,2],[3,2]	378/378 (100%)
[3,0],[3,1]	18/20 (90%)	[3,2],[3,2],[3,2]	118/118 (100%)
[3,0],[3,2]	20/20 (100%)	[4,0]	2/7 (28.6%)
[3,1],[3,1]	27/28 (96.4%)	[4,1]	12/16 (75%)
[3,1],[3,2]	50/50 (100%)	[4,2]	25/26 (96.2%)
[3,2],[3,2]	28/28 (100%)	[4,3]	16/16 (100%)
[3,1],[4,0]	59/71 (83.1%)	[5,1]	15/89 (16.9%)
[3,1],[4,1]	229/240 (95.4%)	[5,2]	(in progress)
[3,1],[4,2]	355/364 (97.5%)		
[3,0],[4,1]	84/88 (95.5%)		
[3,0],[4,2]	133/136 (97.8%)		
[4,0],[5,1]	(in progress)		

Table 5.4: Success rate of schemes for various sets of barred patterns

5.5.1 Examples

We now examine the enumeration schemes for permutations avoiding some specific barred patterns of length 5, thus exhibiting recurrence relations for five of the new sequences given in the tables of Section 5.2.

First, we consider the four classes of patterns of length 5 with one bar that give the sequence 1, 2, 6, 23, 104, 532, 3004. These are the classes with representatives $25\bar{1}43$, $25\bar{1}34$, $43\bar{5}21$, and $43\bar{5}12$.

For the class represented by $25\bar{1}43$ we have the scheme in Figure 5.14:

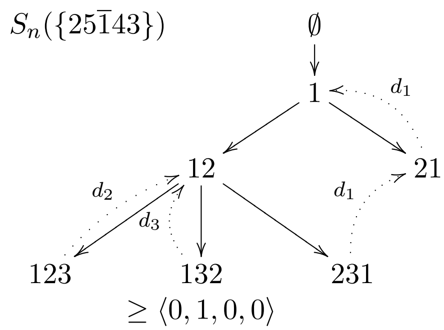


Figure 5.14: The scheme for $S_n(\{25\bar{1}43\})$

This scheme gives the sequence 1, 2, 6, 23, 104, 532, 3004, 18426, 121393, 851810, 6325151, 49448313, 405298482, 3470885747, 30965656442 for $S_n(\{25\bar{1}43\})$ with $1 \leq n \leq 30$.

Next, the equivalence class represented by the pattern $25\bar{1}34$ has the scheme in Figure 5.15.

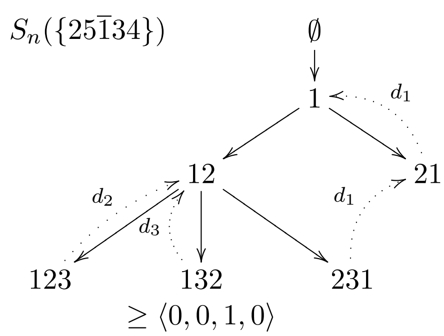


Figure 5.15: The scheme for $S_n(\{25\bar{1}34\})$

This differs from the scheme for $S_n(\{25\bar{1}43\})$ only by the gap vector associated to 132, and yields the same sequence.

The equivalence class with representative $43\bar{5}21$ has the scheme in Figure 5.16.

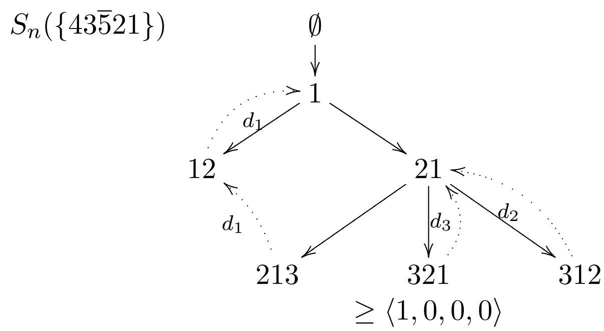


Figure 5.16: The scheme for $S_n(\{43\bar{5}21\})$

This is also symmetric to the previous schemes and yields the same sequence.

Finally, the equivalence class with representative $43\bar{5}12$ has the scheme in Figure 5.17.

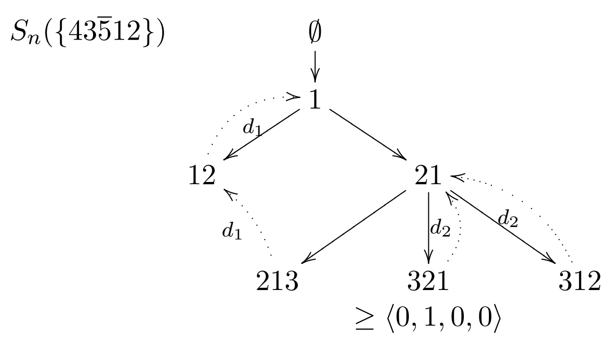


Figure 5.17: The scheme for $S_n(\{43\bar{5}12\})$

Again, since the only difference from the scheme for $S_n(\{43\bar{5}21\})$ is the gap vector associated with prefix 321, so we get the same sequence yet again.

Now, we consider schemes for the 4 patterns of length 5 with two bars that yield new sequences.

The pattern $5\bar{1}\bar{2}43$ has the scheme in Figure 5.18. This yields the new sequence 1, 2, 5, 14, 43, 143, 511, 1950, 7903, 33848, 152529, 720466, 3555715, 18285538, 97752779 for $S_n(\{5\bar{1}\bar{2}43\})$, $1 \leq n \leq 30$.

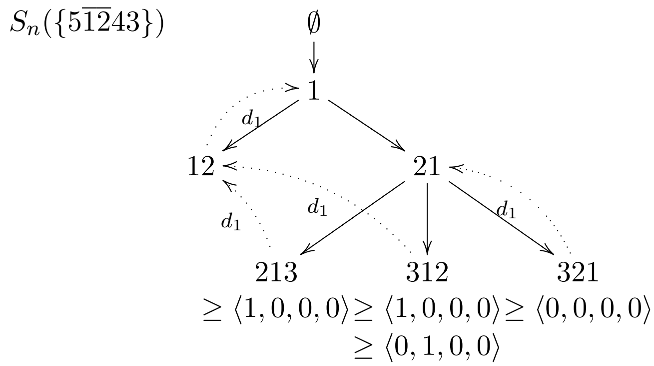


Figure 5.18: The scheme for $S_n(\{5\overline{12}43\})$

The equivalence class with representative $31\overline{54}2$ has the scheme in Figure 5.19, which yields the new sequence 1, 1, 2, 5, 14, 43, 144, 522, 2030, 8398, 36714, 168793, 813112, 4091735, 21451972, 116891160 for $S_n(\{31\overline{54}2\})$, $1 \leq n \leq 30$.

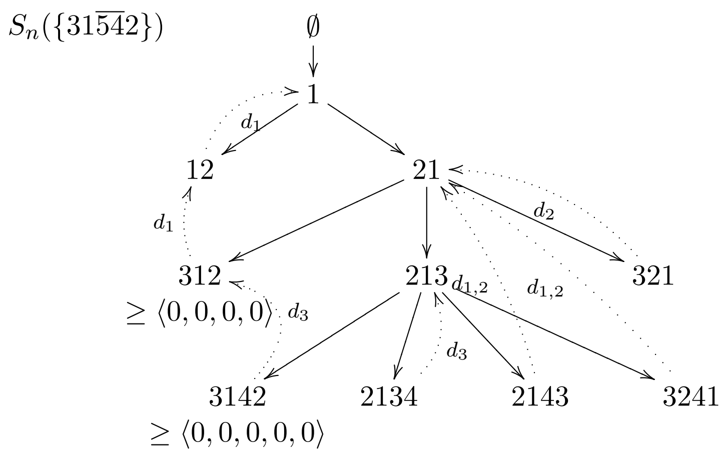


Figure 5.19: The scheme for $S_n(\{31\overline{54}2\})$

The equivalence class with representative $\overline{54}231$ has the scheme in Figure 5.20, which gives the new sequence 1, 2, 5, 14, 43, 146, 561, 2518, 13563, 88354, 686137, 6191526, 63330147, 720314930, 8985750097 for $S_n(\{\overline{54}231\})$, $1 \leq n \leq 30$.

Finally, the pattern $\overline{54}132$ has the scheme in Figure 5.21, which gives the new sequence 1, 1, 2, 5, 14, 43, 147, 575, 2648, 14617, 96696, 754585, 6794015, 69116493, 781266266, 9688636317 for $S_n(\{\overline{54}132\})$, $1 \leq n \leq 30$.

In light of the preceding discussion, each of these schemes can be considered as a rigorously proven recurrence counting pattern-avoiding permutations, each sequence

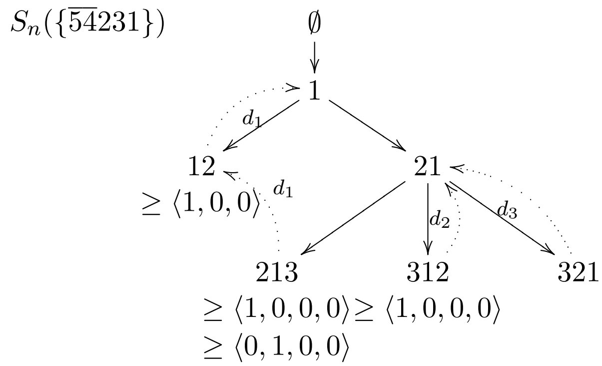


Figure 5.20: The scheme for $S_n(\{\overline{54}231\})$

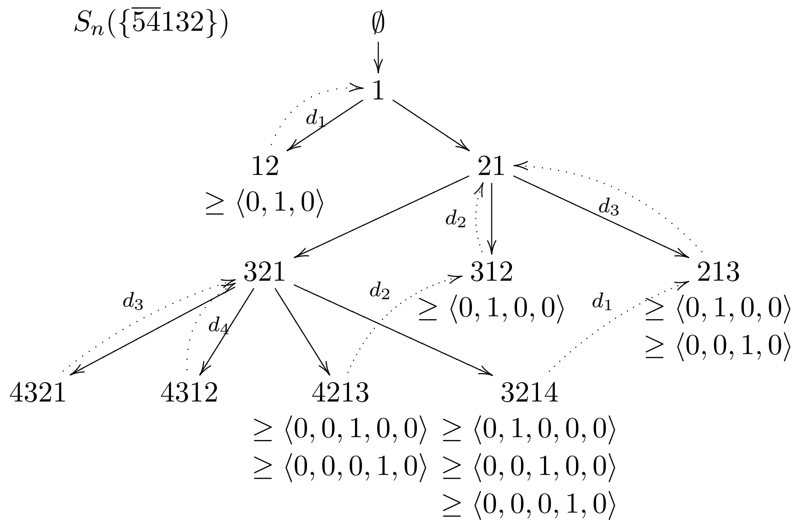


Figure 5.21: The scheme for $S_n(\{\overline{54}132\})$

completely new to the literature.

Chapter 6

Conclusions and Future Work

We have now demonstrated rigorous methods to find recurrences counting both pattern-avoiding words and permutations avoiding barred patterns. These methods have been completely automated in `Maple` allowing the discovery of many new sequences which enumerate these pattern-avoiding objects. The results of these computer algorithms can be found on the author's webpage `http://www.math.rutgers.edu/~lpudwell/maple.html`. However, there is more to do.

6.1 Schemes for Words

The modification of Zeilberger and Vatter's enumeration schemes detailed in Chapter 3 provides the beginning of a universal method for counting pattern-avoiding words. The success rate is quite good for avoiding sets of patterns, and has reasonable success for avoiding single patterns. The schemes of Chapter 4 begin to set the stage for the enumeration of words avoiding other words, and present a method for counting words avoiding monotone patterns. The following questions remain open:

- Find other general techniques for enumerating classes of permutation-avoiding words not counted by prefix schemes.
- Find ways to simplify schemes to compute even more values in the sequence of the number of pattern-avoiding words for fixed k and Q.
- Find ways to convert concrete enumeration schemes to closed form formulas or generating functions when possible.
- Modify the method of schemes for monotone patterns given in Chapter 4 to count words avoiding non-monotone patterns.

At present, the method of enumeration schemes gives a recurrence that makes it easy to compute $\left|A_{[a_1,\ldots a_k]}(Q)\right|$ for a variety of pattern sets Q, and various alphabet vectors. However, these values only allow for conjecture at an appropriate generating function or closed form formula. Finding a way to automate generating function solutions to count pattern-avoiding words would be the next ideal improvement in this area.

6.2 Schemes for Barred Patterns

For the case of permutations avoiding barred patterns, one should note that this method of enumeration schemes, already very successful for counting pattern-avoiding permutations and pattern-avoiding words in the standard sense extends nicely to enumerate permutations avoiding barred patterns as well. Moreover, this is the *first* such method for computing the size of many classes of permutations avoiding barred patterns, and ushers in the study of barred pattern avoidance in its own right

As for words, it would be interesting to automate generating function solutions to count permutations avoiding barred patterns as well.

6.3 Future Directions

The method of enumeration schemes, already shown to be widely successful for pattern-avoiding permutations, pattern-avoiding words, and permutations avoiding barred patterns, can be extended even further still. One clear generalization of both of these areas would be to compute schemes for words that avoid barred patterns. Also, motivated by other applications, it is our hope to extend it to distanced pattern avoidance, in the sense of Firro [17], and to pattern avoidance in other Coxeter groups than S_n, in both cases counting even more general objects than at present.

References

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

[2] M. Albert, R. Aldred, M.D. Atkinson, C. Handley, D. Holton, Permutations of a multiset avoiding permutations of length 3, *Europ. J. Combin.* **22** (2001), 1021–1031.

[3] M. H. Albert, S. Linton, and N. Ruškuc, The insertion encoding of permutations, *Electon. J. Combin.* **12(1)** (2005), R47.

[4] A. Bernini, L. Ferrari, and R. Pinzani, Enumeration of some classes of words avoiding two generalized patterns of length three, arXiv:math.CO/0711.3387v1.

[5] M. Bona, Combinatorics of Permutations, Chapman & Hall, 2004, Chapter 4, 129–173.

[6] P. Brändén, T. Mansour, Finite automata and pattern avoidance in words, *J. Combin. Theory, Series A* **110(1)** (2005), 127–145.

[7] A. Burstein, *Enumeration of Words with Forbidden Patterns*, Ph.D. Thesis, University of Pennsylvania, 1998.

[8] Alexander Burstein, Toufik Mansour, Words restricted by patterns with at most 2 distinct letters, *Electron. J. Combin.* **9(2)** (2002–2003), R3.

[9] S. Butler and M. Bousquet-Melou, Forest-like permutations, To appear in *Ann. Comb.*, arXiv:math.CO/0603617v3.

[10] D. Callan, A combinatorial interpretation for the eigensequence for composition, *Journal of Integer Sequences* **9** (2006), A06.1.4.

[11] D. Callan, A Wilf equivalence related to two stack sortable permutations, arXiv:math.CO/0510211v1.

[12] F. R. K. Chung, R. L. Graham, V. E. Hoggatt Jr., and M. Kleiman, The number of Baxter permutations, *J. Combin. Theory Ser. A* **24** (1978), 382–394.

[13] R. Cori, B. Jacquard, and G. Schaeffer, Description trees for some families of planar maps, *Proceedings of the 9th Conference on Formal Power Series and Algebraic Combinatorics*, Vienna, Austria, 1997, 196–208.

[14] S. B. Ekhad, L. Pudwell, and V. Vatter, *Systematic Studies in Pattern Avoidance*, published electronically at `http://www.math.rutgers.edu/~lpudwell/webbook/bookmain.html`, 2006.

[15] S. Elizalde, Asymptotic enumeration of permutations avoiding generalized patterns, *Advances in Applied Mathematics*, **36(2)** (2006), 138–155.

[16] P. Erdős, and G. Szekeres, A combinatorial problem in geometry. *Compositio Mathematica* **2** (1935), 463–470.

[17] G. Firro, *Distanced patterns*, Ph.D. Thesis, University of Haifa, 2007.

[18] Z. Füredi and P. Hajnal, Davenport-Schinzel theory of matrices, *Discrete Math.* **103** (1992), 233–251.

[19] I. Gessel, Symmetric functions and P-recursiveness, *J. Combin. Theory Ser. A* **53(2)** (1990), 257–285.

[20] S. Heubach, T. Mansour, Avoiding patterns of length three in compositions and multiset permutations, *Advances in Applied Mathematics* **36(2)** (2006), 156–174.

[21] M. Klazar, The Füredi–Hajnal conjecture implies the Stanley–Wilf conjecture. *Formal Power Series and Algebraic Combinatorics*, Springer Verlag, Berlin, Germany, 2000, 250–255.

[22] S. L. Kleiman and D. Laskov, "Problem 15. Rigorous foundation of Schubert's enumerative calculus" F.E. Browder (ed.), Mathematical developments arising from Hilbert problems , *Proc. Symp. Pure Math., Amer. Math. Soc.* **28** (1976), 445-482

[23] D. E. Knuth, *The Art of Computer Programming, 2nd ed.*, Addison-Wesley, Reading, MA, 1973.

[24] V. Lakshmibai and B. Sandhya, Criterion for smoothness of Schubert varieties in $SL(n)/B$, *Proc. Indian Acad. Sci. Math. Sci.* **100** (1990), 45–52.

[25] T. Mansour, Restricted 132-avoiding k-ary words, Chebyshev polynomials, and continued fractions, The Third International Conference on Permutation Patterns, University of Florida, Gainesville, Florida, March 7–11, 2005.

[26] A. Marcus and G. Tardos, Excluded permutation matrices and the Stanley–Wilf Conjecture, *J. Combin. Theory Ser. A* **107 (1)** (2004), 153–160.

[27] D. Marinov and R. Radoičić, Counting 1324-avoiding permutations, *Electon. J. Combin., Special Volume on Permutation Patterns* **9(2)** (2002–2003), R13.

[28] L. Pudwell, "Enumeration Schemes for Words Avoiding Permutations", submitted.

[29] L. Pudwell, "Enumeration Schemes for Words Avoiding Patterns with Repeated Letters", submitted.

[30] A. Regev, Asymptotic values for degrees associated with strips of Young diagrams, *Advances in Mathematics* **41** (1981), 115–136.

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

[32] N. J. A. Sloane, The On-Line Encyclopedia of Integer Sequences, published electronically at `http://www.research.att.com/~njas/sequences/`, 2008.

[33] R. Stanley, *Enumerative Combinatorics, Volume 1*, Cambridge University Press, Cambridge UK, second edition, 1997.

[34] R. E. Tarjan, Sorting using networks of queues and stacks, *J. of Association for Computing Machinery* **19** (1972), 341–346.

[35] V. Vatter, Finitely labeled generating trees and restricted permutations, *Journal of Symbolic Computation* **41** (2006), 559–572.

[36] V. Vatter, Enumeration schemes for restricted permutations, *Combinatorics, Probability, and Computing* **17** (2008), 137–159

[37] J. West, *Permutations with forbidden subsequences and stack-sortable permutations*, Ph.D. thesis, MIT (1990).

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

[39] J. West, Generating trees and forbidden sequences, *Discrete Math* **157** (1996), 363–374.

[40] A. Woo and A. Yong, "When is a Schubert variety Gorenstein?", *Adv. Math.* **207 (1)**, (2006), 205–220.

[41] D. Zeilberger, A proof of Julian West's conjecture that the number of 2-stack-sortable permutations of length n is $2(3n)!/((2n+1)!(n+1)!)$, *Discrete Math.* **102** (1992), 85–93.

[42] D. Zeilberger, Enumeration schemes, and more importantly, their automatic generation, *Annals of Combinatorics* **2** (1998), 185–195.

[43] D. Zeilberger, On Vince Vatter's brilliant extension of Doron Zeilberger's enumeration schemes for Herb Wilf's classes, *The Personal Journal of Ekhad and Zeilberger*, 2006. `http://www.math.rutgers.edu/~zeilberg/pj.html`.

Vita

Lara Kristin Pudwell

2003-2008	Ph.D. in Mathematics, Rutgers University
1999-2003	B.S. in Mathematics, Valparaiso University
1999-2003	B.A. in Computer Science, Valparaiso University
1999	Graduated from White Station High School

2007-2008	Rutgers Carnegie Academy for the Scholarship of Teaching and Learning Fellow
2007-2008	TA Project Assistant, Graduate School - New Brunswick, Rutgers University
2006-2007	PEO Graduate Fellow
2005-2008	Teaching Assistant, Department of Mathematics, Rutgers University
2005-2007	REU Graduate Coordinator, DIMACS, Rutgers University