COMPUTATIONAL METHODS IN PERMUTATION PATTERNS

BY BRIAN KOICHI NAKAMURA

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, 2013

ABSTRACT OF THE DISSERTATION

Computational Methods in Permutation Patterns

by Brian Koichi Nakamura

Dissertation Director: Doron Zeilberger

Given two permutations $\sigma \in \mathcal{S}_k$ and $\pi = \pi_1 \dots \pi_n \in \mathcal{S}_n$, the permutation π is said to contain the pattern σ if there exists $1 \leq i_1 < \dots < i_k \leq n$ such that $\pi_{i_1} \dots \pi_{i_k}$ is order-isomorphic to σ. Each such subsequence is called an occurrence of σ in π. Over the past few decades, the study of pattern-avoiding permutations has been a very active area of research.

This thesis will consider two types of problems in this area. The first is a variation known as consecutive patterns, where the pattern σ must occur in consecutive terms of the permutation π to count as an occurrence. The second is a generalization of the classical pattern avoiding problem, where we wish to study permutations with exactly r occurrences of a pattern (for some fixed $r \geq 0$).

Acknowledgements

First and foremost, I need to thank my advisor Doron Zeilberger for his guidance and support. He has been crucial to my success as a graduate student, and I am thankful to have had such a great academic mentor.

I also need to thank my dissertation committee members Vladimir Retakh, Michael Saks, and Neil Sloane for their patience and guidance as well as their suggestions to improve the thesis. I have also learned a lot from them throughout my time as a graduate student (sometimes as a student in their class and in other cases, through general mathematical discussions).

I also want to offer my appreciation to Gene Fiorini at DIMACS for giving me a chance to work with the DIMACS REU program and for his continued support (even after he stopped being my “boss”).

Chapter 2 and some of Chapter 3 have been through the peer-review process, and I thank the anonymous referees for their suggestions to improve the readability of the material.

I also owe thanks to many fellow (and former) graduate students for numerous helpful mathematical discussions. In particular, I would like to thank Andrew Baxter, Lara Pudwell, and Vince Vatter for the many insightful discussions regarding the field of permutation patterns.

Finally, I am deeply grateful to my family and many friends for their encouragement and support throughout my time in graduate school.

Dedication

In memory of my grandmother Sumie Negishi, who will always serve as an inspiration throughout my life.

Table of Contents

Chapter 1

Introduction and Background

1.1 Classical pattern avoidance in permutations

The focus of this thesis is on studying permutations that satisfy certain properties (namely, the avoidance or containment of certain patterns). This is an active field of research with hundreds of papers published in the past few decades. Given the enormous volume of work in this area, we will restrict this introduction chapter primarily to background that is most relevant to this thesis. There are several sources for a more broad overview of the field including recent survey articles by Kitaev and Mansour [30] and Steingrímsson [56] as well as books by Bóna [12] and Kitaev [29].

Let $\mathcal{S}_n$ be the symmetric group of permutations of $\{1, 2, \ldots, n\}$. Throughout the rest of this work, the permutations will be written in one-line notation. Given two finite sequences of distinct numbers $\sigma = \sigma_1, \sigma_2, \ldots, \sigma_k$ and $\tau = \tau_1, \tau_2, \ldots, \tau_k$, we say that σ and τ are *order-isomorphic* if for every i and j $(i \neq j)$, $\sigma_i < \sigma_j$ if and only if $\tau_i < \tau_j$. The sequence $\sigma = \sigma_1, \sigma_2, \ldots, \sigma_k$ will usually be written more compactly as $\sigma = \sigma_1\sigma_2 \ldots \sigma_k$. Also, given a finite sequence of distinct numbers $\sigma = \sigma_1\sigma_2 \ldots \sigma_k$, we define the *reduction* of this sequence, denoted by $\mathrm{red}(\sigma)$, to be the unique length k permutation that is order-isomorphic to σ. For example, $\mathrm{red}(63915) = 42513$.

1.1.1 Background

Given a (permutation) pattern $\sigma = \sigma_1 \ldots \sigma_k \in \mathcal{S}_k$ and permutation $\pi = \pi_1 \ldots \pi_n \in \mathcal{S}_n$, we say that π *contains* the pattern σ if there exists $1 \leq i_1 < \ldots < i_k \leq n$ such that $\mathrm{red}(\pi_{i_1} \ldots \pi_{i_k}) = \sigma$. Each such subsequence in π will be referred to as an *occurrence* of the pattern σ. If there are no occurrences of σ in π, then we will say that π *avoids* the pattern σ. For example, if pattern $\sigma = 123$, then the permutation 635412 avoids the pattern σ while the permutation 354621 contains two occurrences of the pattern σ (namely, the subsequences 356 and 346). In our work, the patterns discussed will always be permutations as well.

Given a pattern σ, we define the following set:

$$\mathcal{S}_n(\sigma) := \{\pi \in \mathcal{S}_n : \pi \text{ avoids } \sigma\}. \tag{1.1}$$

The number of length n permutations avoiding σ will be denoted by $s_n(\sigma) := |\mathcal{S}_n(\sigma)|$. We also consider a natural generalization. Let B be a set of patterns. We say that the permutation π avoids B if for every $\sigma \in B$, π avoids the pattern σ. The analogous set is defined as:

$$\mathcal{S}_n(B) := \bigcap_{\sigma \in B} \mathcal{S}_n(\sigma). \tag{1.2}$$

Similarly, the number of length n permutations avoiding B will be denoted by $s_n(B) := |\mathcal{S}_n(B)|$.

The study of patterns in permutations has origins in the study of sorting permutations. In computer science, a *stack* is a first-in, last-out data structure that permits two operations: *push* and *pop*. The *push* operation places the next unread entry of the input onto the stack, while the *pop* operation takes the top-most entry off the stack (whatever was last pushed onto stack) and makes it the next term of the final output. A permutation $\pi \in \mathcal{S}_n$ is defined to be *stack-sortable* if it can be "sorted" to produce the identity permutation $12 \ldots n$ with only one pass through a single stack. The study of permutation patterns began in earnest after Knuth's characterization of stack-sortable permutations [31]:

Theorem 1. *A permutation is stack-sortable if and only if it avoids the pattern* 231.

Much work has been done since then on studying permutations that are sortable with multiple stacks and other data structures. Another major area of study has been in enumerating permutations that avoid a pattern (or set of patterns). We proceed by reviewing some known results.

1.1.2 Wilf-equivalence

Two patterns σ and τ are said to be *Wilf-equivalent* if $s_n(\sigma) = s_n(\tau)$ for all $n \geq 0$. Patterns that are Wilf-equivalent are said to belong in the same *Wilf-equivalence class*. While there is no result providing necessary and sufficient conditions for Wilf-equivalence, there are several trivial Wilf-equivalences.

First, we define a few operations on permutations. Given a permutation $\pi = \pi_1 \ldots \pi_n$, we define the *reversal* of the permutation, denoted by π^r, to be the permutation $\pi_n \pi_{n-1} \ldots \pi_1$. Similarly, we define the *complement* of the permutation, denoted by π^c, to be $\pi^c = \pi'_1 \ldots \pi'_n$, where $\pi'_i = n - \pi_i + 1$. For example, if $\pi = 21435$, then $\pi^r = 53412$ and $\pi^c = 45231$. Finally, we define the *inverse* of the permutation, denoted by π^{-1}, to be $\pi^{-1} = \pi'_1 \ldots \pi'_n$, where $\pi'_i = j$ if and only if $\pi_j = i$. In essence, π is viewed as a bijection from $\{1, \ldots, n\}$ to $\{1, \ldots, n\}$, and π^{-1} is the functional inverse. For example, if $\pi = 42351$, then $\pi^{-1} = 52314$.

This leads to the following lemma:

Lemma 1. *Given a pattern $\sigma \in \mathcal{S}_k$ and a permutation $\pi \in \mathcal{S}_n$, the following are equivalent:*

- π *avoids* σ
- π^r *avoids* σ^r
- π^c *avoids* σ^c
- π^{-1} *avoids* σ^{-1}

Given any pattern σ, the lemma implies that σ, σ^r, σ^c, and σ^{-1} all belong to the same Wilf-equivalence class. It should be noted, however, that these four patterns are not necessarily distinct. For example, if $\sigma = 1234$, then $\sigma = \sigma^{-1} = 1234$ while $\sigma^r = \sigma^c = 4321$.

We also note one non-trivial Wilf-equivalence result. This requires one additional operation on permutations. Given two permutations $\tau = \tau_1 \ldots \tau_m \in \mathcal{S}_m$ and $\pi = \pi_1 \ldots \pi_n \in \mathcal{S}_n$, we define their *direct sum*, denoted by $\tau \oplus \pi$, to be the length $m + n$

permutation given by

$$\tau \oplus \pi := \tau_1\tau_2 \ldots \tau_m(\pi_1 + m)(\pi_2 + m) \ldots (\pi_n + m).$$

The direct sum can be considered a natural way of concatenating two permutations to form a larger permutation.

Using this notion, Backelin, West, and Xin proved the following result [4]:

Theorem 2. *Let $\tau = 12 \ldots m$ and let π be any permutation. Then the permutation $\tau \oplus \pi$ is Wilf-equivalent to $\tau^r \oplus \pi$.*

For example, this theorem implies that $12 \oplus 1 = 123$ and $21 \oplus 1 = 213$ are Wilf-equivalent. Using this fact and the trivial Wilf-equivalences, we have that all length 3 patterns belong to the same equivalence class. An earlier result by Stankova also shows that the patterns 1342 and 2413 are Wilf-equivalent [55]. These results, combined with the trivial Wilf-equivalences, show that there are exactly three Wilf-equivalence classes for length 4 patterns. The classes are represented by the patterns 1234, 1324, and 1342. Some known enumerative results will be discussed in the next subsection.

1.1.3 Enumerative results

A classical question in permutation patterns is to enumerate permutations that avoid a given pattern or set of patterns. In particular, given a fixed pattern σ, one would want to either find a closed-form formula (in n) for $s_n(\sigma)$ or at the least, find some efficient way to generate the sequence. It appears that finding a closed-form for $s_n(\sigma)$ is generally either very difficult or impossible. A related question is to find asymptotics for how the sequence grows, but this is also generally difficult. Many unresolved questions remain, even for certain length 4 patterns.

We will also consider the generating function for the enumerating sequence $s_n(\sigma)$. Given a pattern σ, we define the corresponding generating function as:

$$F_\sigma(x) := \sum_{n=0}^{\infty} s_n(\sigma)x^n.$$

If the pattern is clear from context, we will often write $F(x)$. Another question of interest is whether, for a given pattern, its generating function is of a certain type (for example, is it rational, algebraic, or holonomic?).

We begin by reviewing some of the known results for avoiding a single pattern. There is only one pattern of length one, namely $1 \in \mathcal{S}_1$, and we trivially have:

$$s_n(1) = \begin{cases} 1 & n = 0 \\ 0 & n \geq 1 \end{cases}.$$

For length two patterns, 12 is Wilf-equivalent to 21, so there is only one equivalence class. For all n, we trivially have:

$$s_n(12) = 1.$$

For length three patterns, the trivial Wilf-equivalences show that there are at most two Wilf-equivalence classes: $\{123, 321\}$ and $\{132, 213, 231, 312\}$. It turns out, however, that all length three patterns belong to the same equivalence class. More specifically, the exact enumeration of these permutations is known [31, 53], and we have the following:

$$s_n(123) = s_n(132) = C_n,$$

where $C_n = \frac{1}{n+1}\binom{2n}{n}$, the n-th Catalan number. Many proofs exist for this result including a number of bijective proofs [16].

The situation becomes much more complicated from length four patterns. There are three Wilf-equivalence classes, and the representative patterns are 1234, 1342, and 1324. The enumeration of the first two are known, but the pattern 1324 has been notoriously difficult to enumerate thus far.

For the pattern 1234, Gessel showed in [26]:

Theorem 3 (Gessel, 1990)**.**

$$s_n(1234) = \frac{1}{(n+1)^2(n+2)} \sum_{k=0}^{n} \binom{2k}{k}\binom{n+1}{k+1}\binom{n+2}{k+1}$$

For the pattern 1342, Bóna showed in [8]:

Theorem 4 (Bóna, 1996).

$$s_n(1342) = (-1)^{n-1} \cdot \frac{7n^2 - 3n - 2}{2} + 3 \sum_{i=2}^{n} (-1)^{n-i} 2^{i+1} \frac{(2i-4)!}{i!(i-2)!} \binom{n-i+2}{2}$$

Unfortunately, no exact enumeration or generating function is known for 1324. It is not known if the sequence $s_n(1324)$ is even polynomially recursive (or more concisely, P-recursive).[1] No non-recursive formula is known for computing this sequence. Marinov and Radoičić found a recurrence for the sequence and used it to compute the first 20 terms of the sequence [39]. Albert et al. developed another approach [1] to compute $s_n(1324)$ up to $n = 25$ and establish an asymptotic lower bound, which will be mentioned in greater detail in the next subsection. All known terms are currently listed in the On-Line Encyclopedia of Integer Sequences [54] (sequence A061552).

Finally, it should be noted that there are four general enumeration methods that tend to be applied frequently in enumerating pattern-avoiding permutations: generating trees, substitution decomposition, insertion encoding, and enumeration schemes. Vatter provides a detailed comparison of these techniques in [57]. None of these methods are guaranteed to work for every pattern. While the first three methods can sometimes produce generating functions, the fourth method (enumeration schemes) yields complicated recurrences that generally does not motivate generating functions. On the other hand, enumeration schemes is an automated approach that has been extendable to other variations of permutation pattern problems.

1.1.4 Asymptotic results

Given the difficulty of enumerating pattern-avoiding permutations exactly, a related problem is to study the asymptotic growth of the sequence. The most famous conjecture of this type in permutation patterns was the Stanley-Wilf conjecture:

Conjecture 1 (Stanley, Wilf). *Given any pattern σ, there exists a constant C (depending only on σ) such that $s_n(\sigma) < C^n$ for every n.*

---

[1] Recall that a sequence is P-recursive if it satisfies a homogeneous linear recurrence with polynomial coefficients.

Given a pattern σ, we also define the Stanley-Wilf limit of σ as

$$L(\sigma) := \lim_{n\to\infty} \sqrt[n]{s_n(\sigma)}.$$

Arratia showed in [3] that the Stanley-Wilf conjecture was equivalent to the following:

Conjecture 2. *Given any pattern σ, $L(\sigma)$ exists.*

The conjecture was finally proved by Marcus and Tardos [38], who provided a surprisingly short and elementary proof. In actuality, they proved a different conjecture, known as the Füredi-Hajnal Conjecture, which was previously shown to imply the Stanley-Wilf Conjecture.

Recall that there is only one Wilf-equivalence class of length 2 patterns. We trivially have that $L(\sigma) = 1$ for any $\sigma \in \mathcal{S}_2$. Also recall that all length 3 patterns are counted by the Catalan numbers. This gives us that $L(\sigma) = 4$ for any $\sigma \in \mathcal{S}_3$. For length 4 patterns, there are three equivalence classes that are represented by 1234, 1342, and 1324. Since we have the exact enumeration for the first two patterns, we can find that $L(1234) = 9$ and $L(1342) = 8$. However, the exact Stanley-Wilf limit for 1324 remains unknown.

A lower bound of 9.47 for $L(1324)$ was found by Albert et al. in [1]. There have also been improvements for the upper bound recently. In [15], Claesson, Jelínek, and Steingrímsson produced an upper bound of 16 for the Stanley-Wilf limit. Their approach was modified by Bóna in [7] to improve the bound to $7 + 4\sqrt{3} \approx 13.93$. Based off of computational simulations, it is believed that the actual limit is very close to 11 [56].

1.2 Classical pattern occurrences in permutations

We now consider a generalization of the pattern avoidance problem. For a pattern τ and non-negative integer $r \geq 0$, we define the set

$$\mathcal{S}_n(\tau, r) := \{\pi \in \mathcal{S}_n : \pi \text{ has exactly } r \text{ occurrences of the pattern } \tau\}.$$

We define the cardinality of the set $s_n(\tau, r) := |\mathcal{S}_n(\tau, r)|$. The corresponding generating function is defined as

$$F_\tau^r(x) := \sum_{n=0}^{\infty} s_n(\tau, r)x^n.$$

Observe that the $r = 0$ case corresponds to the classical pattern avoidance problem, which was just discussed and has been well studied.

1.2.1 Single pattern occurrences

This more general problem has also been studied, but the work has usually been restricted to small patterns (usually length 3) and small r. While the trivial Wilf-equivalences still hold for $r \geq 0$, Theorem 2 no longer applies for $r > 0$. Recall that the trivial Wilf-equvalences would show that at most two patterns need to be considered: 123 and 132. While these were Wilf-equivalent in the $r = 0$ case, they are not analogously equivalent for $r > 0$.

In [44], Noonan studied permutations containing exactly one occurrence of 123 and proved that $s_n(123, 1) = \frac{3}{n}\binom{2n}{n-3}$. Burstein recently provided a short combinatorial proof of the result [13]. In [45], Noonan and Zeilberger developed an enumeration approach using functional equations and reproved Noonan's result for $s_n(123, 1)$. In addition, they could quickly compute terms of the sequence $s_n(123, 2)$ and conjecture that it was equal to $\frac{59n^2+117n+100}{2n(2n-1)(n+5)}\binom{2n}{n-4}$. This conjecture was proved by Fulmek in [25] using "generalized" Dyck paths.[2] In addition, Callan gave a new approach [14] for enumerating permutations with exactly 3 occurrences and exactly 4 occurrences of 123.

More is known for the pattern 132 than for 123. In [45], Noonan and Zeilberger conjectured that $s_n(132, 1) = \binom{2n-3}{n-3}$. This was proved by Bóna in [10]. The $r = 2$ case was done by Fulmek [25] by again considering Dyck paths. Mansour and Vainshtein [37] presented a different method to find the corresponding generating functions $F_{132}^r(x)$ for any r and used it to explicitly find $s_n(132, r)$ for $r \leq 6$.

In [9], Bóna also shows that for any r, the sequence given by $s_n(132, r)$ is P-recursive. He actually proved a stronger statement:

[2] The paths may have "down-jumps" in addition to the traditional "up-steps" and "down-steps".

Theorem 5. *For any fixed $r \geq 0$, the generating function $F^r_{132}(x)$ is algebraic, and more specifically, is rational in the variables x and $\sqrt{1-4x}$.*

The analogous statement is believed to be true for the pattern 123 but still remains open:

Conjecture 3. *For any fixed $r \geq 0$, the generating function $F^r_{123}(x)$ is rational in the variables x and $\sqrt{1-4x}$.*

This conjecture holds for the cases $r \leq 2$, for which there are explicit generating functions.

1.2.2 Multiple patterns

This generalized problem can be further extended to consider multiple patterns simultaneously. More precisely, let $\mathbf{r} = [r_1, \ldots, r_k]$ be a list of non-negative integers, and let $L = \left[\tau^1, \ldots, \tau^k\right]$ be a list of patterns. For each n, we define the set of permutations

$$\mathcal{S}_n(L, \mathbf{r}) := \bigcap_{i=1}^{k} \mathcal{S}_n(\tau^i, r_i).$$

The cardinality of the set is denoted by $s_n(L, \mathbf{r}) := |\mathcal{S}_n(L, \mathbf{r})|$. Stated more simply, $\mathcal{S}_n(L, \mathbf{r})$ is the set of permutations that have r_i occurrences of τ^i for every i. Note that the single pattern case corresponds to $k = 1$. While some work has been done for the $k > 1$ case, the work is often restricted to length 3 patterns (and usually $k = 2$).

In [53], Simion and Schmidt enumerated $s_n([\sigma, \tau], [0, 0])$ for all $\sigma, \tau \in \mathcal{S}_3$. Robertson [50] found closed-form expressions for $s_n([123, 132], [r_1, r_2])$ for every combination of $r_1, r_2 = 0, 1$. Quite a bit of related work has been done since then (for example, in [22, 35, 36, 34, 51, 52]). However, there is still no exact enumeration of $s_n([123, 132], [r_1, r_2])$ for general r_1 and r_2.

In [45], Noonan and Zeilberger also made the general conjecture:

Conjecture 4. *For any list of occurrences $\mathbf{r} = [r_1, \ldots, r_k]$ and list of patterns $L = \left[\tau^1, \ldots, \tau^k\right]$, the sequence $s_n(L, \mathbf{r})$ is P-recursive in n.*

Although the Noonan-Zeilberger Conjecture has been shown to hold for specific patterns, it remains open even for the special case of single pattern avoidance (i.e., $k = 1$ and $r_1 = 0$).

Another extension in permutation patterns is to consider refining some set of permutations over some "permutation statistic". More precisely, a permutation statistic is a function $h : \bigcup_{n\geq 0} \mathcal{S}_n \to \mathbb{C}$, although for most permutation statistics, the range is the non-negative integers. Given a set of permutations S and a variable q, we will often consider the quantity

$$\sum_{\pi\in S} q^{h(\pi)}.$$

This is often referred to as the refinement of S over the permutation statistic h.

One of the most commonly studied permutation statistic is the number of inversions in a permutation. The *inversion number* of a permutation $\pi = \pi_1 \dots \pi_n$, denoted by $\mathrm{inv}(\pi)$, is the number of pairs (i, j) such that $1 \leq i < j \leq n$ and $\pi_i > \pi_j$. In the language of classical patterns, the inversion number is the number of occurrences of the pattern 21. For example, $\mathrm{inv}(31425) = 3$. The inversion number is the minimum number of adjacent transpositions needed to "sort" a permutation into the identity permutation $12\dots n$, and in a sense, represents how "unsorted" the permutation is.

A classical result by Netto [43] shows that for each n,

$$\sum_{\pi\in\mathcal{S}_n} q^{\mathrm{inv}(\pi)} = (1)(1+q)(1+q+q^2)\dots(1+q+\dots+q^{n-1}),$$

which is the well-known q-analog of $n!$. A more commonly studied question is to consider refining pattern-avoiding permutations (for some fixed pattern) over the inversion number. More precisely, given a fixed pattern σ, we define the polynomial

$$A_\sigma(q,n) := \sum_{\pi\in\mathcal{S}_n(\sigma)} q^{\mathrm{inv}(\pi)}.$$

One would like to know more about either this polynomial or the more general bivariate generating function (in variables q and t):

$$\sum_{n=0}^{\infty} A_\sigma(q,n)t^n.$$

Some work has been done in this area, for example in [11].

There is also an interesting conjecture by Claesson, Jelínek, and Steingrímsson [15]:

Conjecture 5. *The number of permutations in $\mathcal{S}_n(1324)$ with a fixed number k of inversions is increasing as a function of n.*

Restated using this section's notation, the conjecture is that for each fixed k, $s_n([21, 1324], [k, 0])$ is an increasing sequence (in n). It is shown in [15] that this conjecture (if true) would immediately improve the upper-bound of the Stanley-Wilf limit for 1324 to be $L(1324) < e^{\pi\sqrt{2/3}} \approx 13.001954$. The authors further believe that this conjecture holds for any non-increasing pattern:

Conjecture 6. *For each fixed k and pattern $\tau \in \mathcal{S}_m \setminus \{12\dots m\}$, $s_n([21, \tau], [k, 0])$ is an increasing sequence (in n).*

Observe that the conjecture fails when τ is an increasing pattern due to the Erdős-Szekeres Theorem, which we restate using our notation:

Theorem 6. *For any fixed j and k, $s_n([12\dots j, k\dots 21], [0, 0]) = 0$ for all $n \geq (j - 1)(k - 1) + 1$.*

1.3 Consecutive patterns in permutations

In addition to the classical notion of pattern containment, many generalizations and variations have gained attention in recent years. One such variation is the study of consecutive patterns in permutations, where the pattern is required to occur in consecutive entries. Recall that given a sequence of distinct numbers $\sigma = \sigma_1 \cdots \sigma_k$, the reduction of the sequence, denoted by $\mathrm{red}(\sigma)$ is the unique length k permutation that is order-isomorphic to σ.

Let m and n be positive integers with $m \leq n$, and let $\sigma \in \mathcal{S}_m$ and $\pi = \pi_1 \cdots \pi_n \in \mathcal{S}_n$. We say that π *contains σ consecutively* if $\mathrm{red}(\pi_i \cdots \pi_{i+m-1}) = \sigma$ for some i where $1 \leq i \leq n - m + 1$. Otherwise, we say that π *avoids σ consecutively.* For example, the permutation $123654 \in \mathcal{S}_6$ contains the permutation pattern 1243 (given by the subsequence 2365). However, the permutation $12453 \in \mathcal{S}_5$ avoids the pattern 1243 consecutively (even though 1243 is contained as a classical pattern).

The first extensive study in this area was done by Elizalde and Noy [23], where they considered length 3 and 4 patterns. There has been quite a bit of work in this area since then, for example in [2, 18, 33, 40, 17, 19, 28, 41, 49, 24]. We will utilize this section to highlight some interesting differences between the consecutive pattern problem and the classical pattern problem.

Given a fixed pattern σ, let $\alpha_n(\sigma)$ be the number of length n permutations avoiding the consecutive pattern σ. We will say that the two patterns σ and τ are *consecutively Wilf-equivalent* (or more concisely c-*Wilf-equivalent*) if $\alpha_n(\sigma) = \alpha_n(\tau)$ for all $n \geq 0$. We will also say that patterns that are c-Wilf-equivalent belong to the same *c-Wilf-equivalence class.* In this setting, a pattern is still c-Wilf-equivalent to its reversal and complement, but it is not necessarily equivalent to its inverse. For example, the inverse of the permutation 1342 is the permutation 1423, but it has been shown that these are not c-Wilf-equivalent. Additionally, the other Wilf-equivalence result, Theorem 2, no longer holds in the consecutive setting.

For length 3 patterns, the consecutive patterns 123 and 132 are not c-Wilf-equivalent. Elizalde and Noy derive exponential generating functions for these patterns in [23]. For length 4 patterns, there are 7 c-Wilf-equivalence classes (as opposed to just 3 in the classical pattern case). The classes are represented by the patterns: 1234, 2413, 2143, 1324, 1423, 1342, and 1243. The patterns 1234, 1243, and 1342 were solved in [23] while the pattern 1324 was solved in [24]. However, there are currently no closed solutions known for 1423, 2143, and 2413.

Similar to the Stanley-Wilf Conjecture for classical patterns, Warlimont made the following conjecture regarding the asymptotics [58]:

Conjecture 7. *Given any consecutive pattern σ, there exists constants $\gamma > 0$ and $w < 1$ such that $\alpha_n(\sigma)/n! \sim \gamma w^n$.*

This conjecture was proved by Ehrenborg, Kitaev, and Perry in [19] using spectral theory. In recent work [20], Elizalde showed that for $\tau \in \mathcal{S}_k$, $\alpha_n(\tau)$ is asymptotically maximal for the pattern $12\ldots k$ and asymptotically minimal for the pattern $12\ldots(k-2)(k)(k-1)$. This answered conjectures in [23] and [41]. Shortly after, Perarnau provided a probabilistic proof for the maximal pattern case [47]. It remains open whether these conjectures hold for all n (and not just asymptotically).

Chapter 2

Automated Approaches for Consecutive Patterns

Notice: *The content of this chapter is adapted from Nakamura [41].*

2.1 Overview

Let $\sigma = \sigma_1 \cdots \sigma_k$ be a sequence of k distinct real numbers. Recall that the *reduction* $\mathrm{red}(\sigma)$ is the length k permutation obtained by relabeling the elements of σ with $\{1, \ldots, k\}$ so that σ and $\mathrm{red}(\sigma)$ are order-isomorphic. For example, $\mathrm{red}(5386) = 2143$. For a permutation π, we will also write $|\pi|$ for the number of elements in the permutation.

Consider a fixed pattern $\sigma \in \mathcal{S}_m$ and permutation $\pi = \pi_1 \cdots \pi_n \in \mathcal{S}_n$. Recall that π *contains σ consecutively* if $\mathrm{red}(\pi_i \cdots \pi_{i+m-1}) = \sigma$ for some i where $1 \leq i \leq n - m + 1$. Otherwise, we say that π *avoids σ consecutively.* Similarly, if B is a set of permutations, then we say that π *avoids B consecutively* if for every $\tau \in B$, π avoids the pattern τ consecutively. For example, the permutation $123654 \in \mathcal{S}_6$ contains the permutation pattern 1243, since $\mathrm{red}(2365) = 1243$. However, the permutation $12453 \in \mathcal{S}_5$ avoids the pattern 1243 consecutively (even though 1243 is contained as a classical pattern).

In general, we are interested in counting permutations that avoid a pattern (or a set of patterns). Given a set of patterns B, let $\alpha_n(B)$ be the number of length n permutations that avoid B consecutively. If B consists of only a single pattern π, we may write $\alpha_n(\pi)$ instead, and if no ambiguity would arise, we may just write α_n. For a given set of patterns B, we would like to find the exponential generating function

$$G_B(z) := \sum_{n=0}^{\infty} \alpha_n \frac{z^n}{n!}.$$

If no ambiguity would arise, this may also be denoted by $G(z)$. In addition, we define a more general exponential generating function

$$P_B(z,t) := \sum_{k,n\geq 0} b_{k,n} \frac{z^n t^k}{n!}$$

where $b_{k,n}$ is the number of length n permutations that contain exactly k occurrences of the patterns in B. Again, we may write $P(z,t)$ if the set B is clear. We will also define $c_n(t) = \sum_{k\geq 0} b_{k,n}t^k$. Note that $P(z,0) = G(z)$ and $c_n(0) = \alpha_n$.

In addition, we will say that two sets of patterns B and B' are *consecutively Wilf-equivalent* (or more concisely *c-Wilf-equivalent*) if $G_B(z) = G_{B'}(z)$. We will also say that B and B' are *strongly c-Wilf-equivalent* if $P_B(z,t) = P_{B'}(z,t)$. Since this chapter deals solely with consecutive patterns, the word "consecutive" will be omitted in most instances. For the rest of this chapter, the reader should assume that all mentions of containment, avoidance, and Wilf-equivalence are consecutive.

In recent years, there has been an increasing amount of research done on consecutive pattern avoidance in permutations. An early paper of Elizalde and Noy [23] finds generating functions $G(z)$ and $P(z,t)$ for certain cases of single pattern avoidance. Using various techniques, additional generating functions for specific single patterns and multi-pattern sets have been found in [2, 33, 40, 17]. There is also recent work in vincular patterns (a generalization of consecutive patterns) using enumeration schemes by Baxter and Pudwell in [6]. In particular, our approach will resemble the cluster method approach in [17].

So far, generating functions have been found for specific single patterns and multi-pattern sets and for certain single pattern families where some specific structure can be exploited. In this paper, we will outline two algorithms to calculate α_n and $c_n(t)$ more efficiently, and both algorithms have been implemented in the accompanying Maple package `CAV`. The Maple package can be downloaded from the author's website. As a result of the first algorithm in Section 2.3, we get a theorem for proving when two sets of patterns are strongly c-Wilf-equivalent. During preparation of the paper containing this work [41], the author learned that this result was also independently proven by Khoroshkin and Shapiro in [28] by slightly different means. To establish the much faster second algorithm in Section 2.4, we define a new generating function which we refer to as the cluster tail generating function. We show that this generating function always satisfies a certain functional equation and give a constructive approach to finding it. This functional equation is then used to compute values for α_n much more quickly. We use our algorithm to give some asymptotic approximations in Section 2.5. We conclude with Section 2.6 by sharing some new conjectures we have based off of experimentation with our `CAV` package.

2.2 The cluster method

The results in this work utilize an extension of the cluster method of Goulden and Jackson [27, 46]. The cluster method itself is based off of the inclusion-exclusion principle. We restate some of the terminology and notation here.

Let B be a set of patterns. Without loss of generality, assume that B contains no trivial redundancies (i.e., there are no $\pi_1, \pi_2 \in B$ with $\pi_1 \neq \pi_2$ such that π_1 contains π_2). We say that an ordered pair $(\pi; [[i_1, j_1], \ldots, [i_m, j_m]])$ is a length k cluster (or more concisely, a k-cluster) if it satisfies the following:

(a) $\pi \in \mathcal{S}_k$

(b) $i_1 = 1$, $j_m = k$, and $i_n < i_{n+1} < j_n$ for $1 \leq n \leq m-1$ (i.e., each interval overlaps with the neighboring interval, and the intervals cover π)

(c) $\mathrm{red}(\pi_{i_n} \cdots \pi_{j_n}) \in B$ for all $1 \leq n \leq m$.

Given a cluster $(\pi; [[i_1, j_1], \ldots, [i_m, j_m]])$, we will refer to $\mathrm{red}(\pi_{i_n} \cdots \pi_{j_n})$ as the n-th *marked pattern* in the cluster. Note that the same underlying permutation π may appear in different k-clusters. For example, suppose that $B = \{123\}$. Then, the cluster $(12345; [[1,3],[3,5]])$, which has two marked patterns, is different from the cluster $(12345; [[1,3],[2,4],[3,5]])$, which has three marked patterns.

Let $\mathcal{C}_k$ be the set of clusters of length k, and for a cluster $w = (\pi; [[i_1, j_1], \ldots, [i_m, j_m]])$, define $weight(w) = (t-1)^m$, where t will be the variable used to track occurrences. Let $C(k) = \sum\limits_{w \in \mathcal{C}_k} weight(w)$. From an adaptation of [27] to the present context of an "infinite" alphabet and exponential generating functions, we have:

Theorem 7.

$$P(z,t) = \frac{1}{1 - z - \sum\limits_{k\geq 1} C(k)\frac{z^k}{k!}} \tag{2.1}$$

In the remainder of this chapter, we will derive a recurrence from this theorem and use this recurrence as the basis for our subsequent algorithms and results.

2.3 Automated derivation of recurrences

Let B be a set of patterns that we would like to avoid (consecutively). For the rest of this chapter, we assume that B contains no redundancies (i.e., there does not exist $p_1, p_2 \in B$ with $p_1 \neq p_2$ such that p_1 contains p_2). Again, α_n will be the number of length n permutations avoiding B, and $c_n(t)$ will be the polynomial in t where the coefficient of t^k is the number of length n permutations with exactly k occurrences of patterns in B.

From the cluster method of Goulden and Jackson (Theorem 7), we can get the equation

$$P(z,t) = 1 + zP(z,t) + P(z,t)\sum_{k\geq 1} C(k)\frac{z^k}{k!}$$

and by extracting the coefficients of z^n, we get the following recurrence:

$$c_n(t) = nc_{n-1}(t) + \sum_{k=1}^{n} \binom{n}{k} C(k) c_{n-k}(t). \tag{2.2}$$

Additionally, consider a fixed $p \in B$ and let $m = |p|$. Let $\mathcal{C}_k[p] = \{(\pi; [[i_1, j_1], \ldots, [i_r, j_r]]) \in \mathcal{C}_k : \mathrm{red}(\pi_{i_r} \cdots \pi_{j_r}) = p\}$, the set of length k clusters ending in the pattern p. Let $\mathcal{C}_k[p; [x_1, \ldots, x_m]]$ be the clusters in $\mathcal{C}_k[p]$ with the last m terms $\{x_1, \ldots, x_m\}$, where $x_1 < x_2 < \ldots < x_m$. Similarly, define

$$C(k,p) := \sum_{w \in \mathcal{C}_k[p]} weight(w) \tag{2.3}$$

$$C(k,p;[x_1,\ldots,x_m]) := \sum_{w \in \mathcal{C}_k[p;[x_1,\ldots,x_m]]} weight(w). \tag{2.4}$$

If B contains only one pattern, these may be denoted by $C(k)$ and $C(k;[x_1,\ldots,x_m])$, respectively. We will use Equation (2.2) to compute α_n and, more generally, $c_n(t)$.

2.3.1 General algorithm

Computationally, the difficulty in using Equation (2.2) lies in calculating $C(k)$ quickly. One way to do this is to create a recurrence for $C(k, p; [x_1, \ldots, x_{|p|}])$ for each $p \in B$.

We can do this as follows: for a given cluster w, let p_1 and p_2 be the last marked pattern and the second to last marked pattern in w, respectively. Let j be the length of the overlap of p_1 and p_2 in w, i.e., the tail of length j of p_2 coincides with the head of length j of p_1. We want to "chop off" the last $|p_1| - j$ terms of w and apply the reduction to get a shorter cluster, say $w' \in \mathcal{C}_{k'}[p_2; [x_1, \ldots, x_{|p_2|}]]$, which ends in the pattern p_2. Then, $weight(w) = weight(p_1) \cdot weight(w')$.

Additionally, once we "chop off" the tail of p_1 and apply the reduction to get a shorter cluster w', we actually know what the last j terms of $w' = w'_1 \cdots w'_{k'}$ will be. For each term w'_i with $|w'| - j + 1 \leq i \leq |w'|$, the reduction forces w'_i to be $w_i -$ (# of terms in w "chopped off" that were less than w_i). Thus, to compute $C(k, p_1; [x_1, \ldots, x_{|p_1|}])$, we need to sum over all possible ways to "fill out" the rest of the terms in the final p_2 pattern of w'. We also need to sum over all possible choices of $p_2 \in B$ and all possible ways that the tails of this p_2 overlap with the heads of the final p_1 pattern.

In summary, the number of length n permutations avoiding set B can be found by, first, generating a cluster recurrence for $C(k, p; [x_1, \ldots, x_{|p|}])$ for each $p \in B$. Next, use the recurrence $\alpha_n = n\alpha_{n-1} + \sum_{k=1}^{n} \binom{n}{k} C(k)\alpha_{n-k}$ using the base cases $\alpha_0 = \alpha_1 = 1$ and $\alpha_n = 0$ if $n < 0$. Use the cluster recurrences to compute $C(k)$ as needed:

$$C(k) = \sum_{p \in B} C(k, p)$$

$$C(k, p) = \sum_{1 \leq x_1 < x_2 < \ldots < x_m \leq k} C(k, p; [x_1, \ldots, x_m]).$$

Also recall that if $w = (\pi; [i_1, j_1], \ldots, [i_m, j_m])$, then $weight(w) = (t-1)^m$ will keep track of occurrences of patterns with variable t, while setting $t = 0$ and using $weight(w) = (-1)^m$ would count only the permutations that avoid the designated pattern set B.

2.3.2 Example

Let $B = \{2143\}$. Let $w = (\pi; [[i_1, j_1], \ldots, [i_m, j_m]])$ be a length k cluster and $\{x_1, \ldots, x_4\}$ be the last 4 terms of w with $x_1 < \ldots < x_4$ (i.e., $\pi_{k-3} = x_2$, $\pi_{k-2} = x_1$, $\pi_{k-1} = x_4$, and $\pi_k = x_3$). Then, the second to last pattern must also be a 2143 pattern and can have an overlap of length 1 or 2 with the last pattern.

If the overlap is of length 2, let $\pi' = \text{red}(\pi_1 \cdots \pi_{k-2})$ and $w' = (\pi'; [[i_1, j_1], \ldots, [i_{m-1}, j_{m-1}]])$, the cluster found by "chopping off" the tail of the final marked pattern in w and then canonically reducing. Now let $\{y_1, \ldots, y_4\}$ be the last 4 terms of w' with $y_1 < \ldots < y_4$ (i.e., $\pi'_{k-3} = y_2$, $\pi'_{k-2} = y_1$, $\pi'_{k-1} = y_4$, and $\pi'_k = y_3$). Notice that the terms "chopped off" from w were x_4 and x_3. Since both of these are larger than both x_2 and x_1, applying the reduction does not change their values. Thus, $y_4 = x_2$ and $y_3 = x_1$. Summing over all possible tails for w' and accounting for the last pattern that was removed from w, we get

$$\sum_{\substack{1 \leq y_1 < \ldots < y_4 \leq k-2 \\ y_3 = x_1 \\ y_4 = x_2}} weight(2143) \cdot C(k-2; [y_1, y_2, y_3, y_4]).$$

If the overlap is of length 1, let $\pi' = \text{red}(\pi_1 \cdots \pi_{k-3})$ and $w' = (\pi'; [[i_1, j_1], \ldots, [i_{m-1}, j_{m-1}]])$, since the tail that gets "chopped off" has 3 terms. Again, let $\{y_1, \ldots, y_4\}$ be the last 4 terms of w' with $y_1 < \ldots < y_4$ (i.e., $\pi'_{k-3} = y_2$, $\pi'_{k-2} = y_1$, $\pi'_{k-1} = y_4$, and $\pi'_k = y_3$). The terms "chopped off" from w are x_1, x_4, and x_3. Since exactly one term less than x_2 (only x_1) was removed, applying the reduction would reduce x_2 by 1. Thus, $y_3 = x_2 - 1$. Summing over all possible tails for w' and accounting for the last pattern that was removed from w, we get

$$\sum_{\substack{1 \leq y_1 < \ldots < y_4 \leq k-3 \\ y_3 = x_2 - 1}} weight(2143) \cdot C(k-3; [y_1, y_2, y_3, y_4]).$$

We combine the two possibilities along with the base cases to get the recurrence. For $k < 4$:

$$C(k; [x_1, x_2, x_3, x_4]) = 0$$

For $k = 4$:

$$C(k;[x_1,x_2,x_3,x_4]) = weight(2143)$$

For $k > 4$:

$$C(k;[x_1,x_2,x_3,x_4]) = \sum_{\substack{1\le y_1<\ldots<y_4\le k-3 \\ y_3=x_2-1}} weight(2143)\cdot C(k-3;[y_1,y_2,y_3,y_4]) + \sum_{\substack{1\le y_1<\ldots<y_4\le k-2 \\ y_3=x_1 \\ y_4=x_2}} weight(2143)\cdot C(k-2;[y_1,y_2,y_3,y_4]) \quad (2.5)$$

Using this recurrence, we can compute $C(k)$ for any value of k and compute α_n using Equation (2.2). To keep track of all occurrences of 2143 with the variable t, let $weight(2143) = t-1$. To only count permutations that avoid 2143, set $t = 0$ so that $weight(2143) = -1$ for the above recurrence.

2.3.3 Results for c-Wilf-equivalence

Even though Section 2.3.1 is algorithmic in nature, it yields a strong theoretical byproduct. The cluster recurrence generated by the pattern set B totally determines α_n. In fact, it also totally determines $P(z,t)$. However, the "overlapping" relations between the patterns in B totally determine the cluster recurrence.

More specifically, let B be the set of patterns we want to avoid, and let $\tau, \sigma \in B$ where $m = |\tau|$ and $n = |\sigma|$. Note that τ and σ are not necessarily distinct. Suppose that $\mathrm{red}(\sigma_{n-j+1}\cdots\sigma_n) = \mathrm{red}(\tau_1\cdots\tau_j)$ (the tail of σ and the head of τ has an overlap of length j). Then, define the following sets:

$$OverlapMap(\sigma,\tau,j) := \{(\tau_1,\sigma_{n-j+1}),(\tau_2,\sigma_{n-j+2}),\ldots,(\tau_j,\sigma_n)\}$$
$$OverlapMaps(\sigma,\tau) := \{OverlapMap(\sigma,\tau,j) : \mathrm{red}(\sigma_{n-j+1}\cdots\sigma_n) = \mathrm{red}(\tau_1\cdots\tau_j)\}$$

For example, in Section 2.3.2, the pattern 2143 has self-overlaps of length 1 and 2. For a length 1 overlap, we have $OverlapMap(2143, 2143, 1) = \{(2, 3)\}$. This combined with the length of the pattern, which is 4, and the length of the original cluster, denoted by k, completely determines the first summation in Equation (2.5). Similarly, for a length 2 overlap, we have $OverlapMap(2143, 2143, 2) = \{(2,4),(1,3)\}$. Combining this with the length of the pattern, again 4, and the length of the original cluster, again k, completely determines the second summation in Equation (2.5). Thus, $OverlapMaps(2143, 2143) = \{\{(2,3)\}, \{(2,4),(1,3)\}\}$ and $|2143| = 4$ completely determines the cluster recurrence. Therefore we have the following result based off of our algorithm:

Theorem 8. *Let B and B' be two sets of patterns with $|B| = |B'|$. Suppose there is some labeling of the elements (patterns) in sets B and B', say $B = \{p_1, \ldots, p_k\}$ and $B' = \{p'_1, \ldots, p'_k\}$, such that $|p_i| = |p'_i|$ for $1 \leq i \leq k$, and $OverlapMaps(p_i, p_j) = OverlapMaps(p'_i, p'_j)$ for all $1 \leq i, j \leq k$. Then, B and B' are strongly c-Wilf-equivalent.*

Proof. The cluster recurrence was uniquely determined by how the patterns overlapped (which terms from one pattern overlapped with which terms of another pattern) and by how they reduced after "chopping" the last pattern from the current cluster. The possible ways that two patterns can overlap are encoded by OverlapMaps and the effect of the reduction is determined by how the patterns overlapped and the length of those patterns. □

This result was also independently discovered by Khoroshkin and Shapiro [28].

Using this result, it is possible to classify c-Wilf-equivalences in some cases. For example, it is possible to classify single pattern avoidance for single patterns of length 3, 4, and 5 since all the potential equivalences that occur can be demonstrated using Theorem 8. Using the same approach, we can classify almost all single patterns of length 6. All that remains are four possible strong c-Wilf-equivalences that appear true but cannot be rigorously proven through our means. They are the following:

(1) The pattern 123546 appears to belong to the strong c-Wilf-equivalence class $\{124536, 125436\}$.

(2) The pattern 123645 appears to belong to the strong c-Wilf-equivalence class $\{124635, 126435\}$.

(3) The patterns 132465 and 142365 appear to be strongly c-Wilf-equivalent.

(4) The patterns 154263 and 165243 appear to be strongly c-Wilf-equivalent.

The above four conjectures were recently proven by Elizalde and Noy [24], which completes the classification of length 6 patterns. In total, there are 92 c-Wilf-equivalence classes for single patterns of length 6. Additionally, every pair of c-Wilf-equivalent patterns (up to length 6) is in fact strongly c-Wilf-equivalent. More detailed information (including all the equivalences mentioned above) can be found on the author's website.

2.3.4 Maple implementation

The algorithm from Section 2.3.1 has been implemented in the Maple package `CAV`. Using that algorithm and given a set of patterns B to avoid, you can find the sequence $\alpha_1, \ldots, \alpha_n$ by calling the procedure `CAV(B,n)`, where the patterns in B are represented as lists. For example, for $n = 10$ and $B = \{123, 321\}$, trying `CAV({[1,2,3],[3,2,1]},10);` returns the output:

$$[1, 2, 4, 10, 32, 122, 544, 2770, 15872, 101042]$$

To keep track of the occurrences of patterns from B, use the procedure `CAVt(B,n,t)`. For example, trying `CAVt({[1,2,3],[3,2,1]},6,t);` returns the output:

$$[1, 2, 4 + 2t, 10 + 12t + 2t^2, 58t + 28t^2 + 32 + 2t^3, 300t + 236t^2 + 122 + 60t^3 + 2t^4]$$

Also, most of the main procedures in the Maple `CAV` package have an optional verbose setting. For example, for the verbose outputs, try `CAV({[1,2,3],[3,2,1]},10,true);`.

To generate the cluster recurrence only (encoded in a data structure that we call a cluster scheme), use the procedure `SCHEME(k,B,x,y,t)`. For example, try `SCHEME(k, {[1,2,3], [3,2,1]},x,y,t);`. Note that our cluster schemes are different from enumeration schemes that currently appear in permutation patterns literature. The overlap maps between two patterns can also be found using `OverlapMaps(p1,p2)`, which checks for overlaps between tails of `p1` with heads of `p2`. For example, try `OverlapMaps([2,1,4,3], [2,1,4,3])`.

To (attempt to) classify pattern sets of m patterns with each pattern length n, we can compute α_N (for some fixed value N) for each of these pattern sets, and if the α_N values coincide, try to apply Theorem 8. This has been implemented in the procedure `WilfEqm(n,N,m)`. For example, try `WilfEqm(5,12,1)` (or for the verbose output, `WilfEqm(5,12,1,true)`) to (rigorously) classify c-Wilf-equivalence for all single patterns of length 5. An additional byproduct of Theorem 8 is that all instances of c-Wilf-equivalence in single length 5 patterns are actually strong c-Wilf-equivalence. The 25 c-Wilf-equivalence classes can be found on the paper's website.

Similarly, we can use the `WilfEqm` procedure to discover the following:

Corollary 1. *Let B_1 and B_2 both be sets containing two patterns of length 3. Then B_1 is c-Wilf-equivalent to B_2 if and only if they are trivially equivalent by reversal and/or complementation.*

Proof. Run "`WilfEqm(3,10,2,true);`" using the `CAV` Maple package. □

Corollary 2. *Let B_1 and B_2 both be sets containing two patterns of length 4. Then B_1 is c-Wilf-equivalent to B_2 if and only if they are trivially equivalent by reversal and/or complementation.*

Proof. Run "`WilfEqm(4,10,2,true);`" using the `CAV` Maple package. □

Corollary 3. *Let B_1 and B_2 both be sets containing three patterns of length 3. Then B_1 is c-Wilf-equivalent to B_2 if and only if they are trivially equivalent by reversal and/or complementation.*

Proof. Run "`WilfEqm(3,10,3,true);`" using the `CAV` Maple package. □

Similarly, nearly all c-Wilf-equivalences could be classified for sets containing three patterns of length 4. Four pairs of sets appear c-Wilf-equivalent but cannot be proven through our means. They are the following:

(1) The pattern sets $\{1234, 1243, 1342\}$ and $\{1234, 1243, 1432\}$ appear to be strongly c-Wilf-equivalent.

(2) The pattern sets $\{1234, 1243, 2341\}$ and $\{1234, 1243, 2431\}$ appear to be strongly c-Wilf-equivalent.

(3) The pattern sets $\{1324, 1342, 1423\}$ and $\{1324, 1423, 1432\}$ appear to be strongly c-Wilf-equivalent.

(4) The pattern sets $\{1324, 1423, 2341\}$ and $\{1324, 1423, 2431\}$ appear to be strongly c-Wilf-equivalent.

The four cases have been experimentally verified for up to length 14 permutations, and the rest of the classification can be found on the paper's website.

2.4 Automated derivation of functional equations

Computationally, the cluster recurrence is faster than the naive approach of checking every single permutation, but the approach is still very inefficient. For a fixed length k, not every combination of tails gives rise to a possible cluster. For example, if $B = \{123\}$, the only possible underlying permutation in a length 9 cluster is 123456789. The only possible tail is 789, but using the recurrence, we essentially try all $\binom{9}{3}$ possible tails. Each such possible tail gives its contribution of 0 only after it has recursed down to the base cases of $k \leq 3$.

We can, however, gain a substantial speed-up by considering a more complicated generating function. For a fixed pattern $p \in B$ of length m, the cluster tail generating function will be defined as:

$$F(k, p; [z_1, \ldots, z_m]) := \sum_{1 \leq x_1 < \ldots < x_m \leq k} C(k, p; [x_1, \ldots, x_m]) z_1^{x_1} \cdots z_m^{x_m} \tag{2.6}$$

If B is a single pattern set, this may also be denoted as $F(k; [z_1, \ldots, z_m])$. Otherwise, we also define:

$$F(k; [z_1, \ldots, z_m]) := \sum_{p \in B} F(k, p; [z_1, \ldots, z_m]) \tag{2.7}$$

Note that $F(k, p; [1, \ldots, 1]) = C(k, p)$ and $F(k; [1, \ldots, 1]) = C(k)$. In fact, we can always find a functional equation for $F(k, p; [z_1, \ldots, z_m])$ of a certain form. We can then combine this with Equation (2.2) to more quickly compute α_n. We begin with an illustrative example and then present the general algorithm.

2.4.1 Example

Let $B = \{132\}$ and suppose we want to only count permutations that completely avoid 132. We will set $t = 0$ which gives us $weight(132) = -1$. We then can find a functional equation for $F(k; [z_1, z_2, z_3])$ as follows. Using the procedure `SCHEME` in the Maple package `CAV`, we can get the following cluster recurrence:

$$\begin{aligned} C(k; [x_1, x_2, x_3]) &= - \sum_{\substack{1 \le y_1 < y_2 < y_3 \le k-2 \\ y_2 = x_1}} C(k-2; [y_1, y_2, y_3]) \\ &= - \sum_{\substack{1 \le y_1 < x_1 \\ x_1 < y_3 \le k-2}} C(k-2; [y_1, x_1, y_3]) \end{aligned}$$

with the base cases $C(k; [x_1, x_2, x_3]) = 0$ if $k < 3$ and $C(k; [x_1, x_2, x_3]) = -1$ if $k = 3$. Substituting into Equation (2.6) and applying the finite geometric series formula as needed, we get:

$$\begin{aligned} F(k; [z_1, z_2, z_3]) &= \sum_{1 \le x_1 < x_2 < x_3 \le k} C(k; [x_1, x_2, x_3]) z_1^{x_1} z_2^{x_2} z_3^{x_3} \\ &= - \sum_{x_1=1}^{k-2} \sum_{x_2=x_1+1}^{k-1} \sum_{x_3=x_2+1}^{k} \sum_{\substack{1 \le y_1 < x_1 \\ x_1 < y_3 \le k-2}} C(k-2; [y_1, x_1, y_3]) z_1^{x_1} z_2^{x_2} z_3^{x_3} \\ &= - \sum_{x_1=1}^{k-2} \sum_{x_2=x_1+1}^{k-1} \sum_{\substack{1 \le y_1 < x_1 \\ x_1 < y_3 \le k-2}} C(k-2; [y_1, x_1, y_3]) z_1^{x_1} z_2^{x_2} \sum_{x_3=x_2+1}^{k} z_3^{x_3} \\ &= - \frac{z_3}{1-z_3} \sum_{x_1=1}^{k-2} \sum_{x_2=x_1+1}^{k-1} \sum_{\substack{1 \le y_1 < x_1 \\ x_1 < y_3 \le k-2}} C(k-2; [y_1, x_1, y_3]) z_1^{x_1} z_2^{x_2} (z_3^{x_2} - z_3^{k}) \\ &= - \frac{z_3}{1-z_3} \sum_{x_1=1}^{k-2} \sum_{\substack{1 \le y_1 < x_1 \\ x_1 < y_3 \le k-2}} C(k-2; [y_1, x_1, y_3]) z_1^{x_1} \sum_{x_2=x_1+1}^{k-1} z_2^{x_2} (z_3^{x_2} - z_3^{k}) \end{aligned}$$

and since

$$\sum_{x_2=x_1+1}^{k-1} z_2^{x_2} (z_3^{x_2} - z_3^{k}) = \left(\frac{(z_2 z_3)^{x_1+1} - (z_2 z_3)^{k}}{1 - z_2 z_3} - z_3^{k} \frac{z_2^{x_1+1} - z_2^{k}}{1 - z_2} \right)$$

we get

$$\begin{aligned}
F(k;[z_1,z_2,z_3]) = &-\frac{z_2 z_3^2}{(1-z_3)(1-z_2z_3)} \sum_{x_1=1}^{k-2} \sum_{\substack{1\le y_1<x_1\\ x_1<y_3\le k-2}} C(k-2;[y_1,x_1,y_3])(z_1z_2z_3)^{x_1}\\
&+\frac{z_2^k z_3^{k+1}}{(1-z_3)(1-z_2z_3)} \sum_{x_1=1}^{k-2} \sum_{\substack{1\le y_1<x_1\\ x_1<y_3\le k-2}} C(k-2;[y_1,x_1,y_3])z_1^{x_1}\\
&+\frac{z_2 z_3^{k+1}}{(1-z_3)(1-z_2)} \sum_{x_1=1}^{k-2} \sum_{\substack{1\le y_1<x_1\\ x_1<y_3\le k-2}} C(k-2;[y_1,x_1,y_3])(z_1z_2)^{x_1}\\
&-\frac{z_2^k z_3^{k+1}}{(1-z_3)(1-z_2)} \sum_{x_1=1}^{k-2} \sum_{\substack{1\le y_1<x_1\\ x_1<y_3\le k-2}} C(k-2;[y_1,x_1,y_3])z_1^{x_1}\\
= &-\frac{z_2 z_3^2}{(1-z_3)(1-z_2z_3)} F(k-2;[1,z_1z_2z_3,1])\\
&+\frac{z_2^k z_3^{k+1}}{(1-z_3)(1-z_2z_3)} F(k-2;[1,z_1,1])\\
&+\frac{z_2 z_3^{k+1}}{(1-z_3)(1-z_2)} F(k-2;[1,z_1z_2,1])\\
&-\frac{z_2^k z_3^{k+1}}{(1-z_3)(1-z_2)} F(k-2;[1,z_1,1]).
\end{aligned}$$

We can then use the functional equation to compute $C(k) = F(k;[1,1,1])$ for whatever k we need and then find α_n for the desired n by Equation (2.2).

2.4.2 General algorithm

In general, if we can find a functional equation for $F(k,p;[z_1,\ldots,z_m])$ that relates it to cluster generating functions with lower order k', we can use it to compute $c_n(t)$ using Equation (2.2). One can see that most of what was done in the above example can be extended to any pattern (or pattern set by finding a functional equation for each pattern individually). The outline of the general procedure is as follows:

First, find the cluster recurrence for the initial summand $C(k,p;[x_1,\ldots,x_m])$ (as in Section 2.3.1) and substitute this into the summation in Equation (2.6). Split the summation over each summand $C(k',p';[y_1,\ldots,y_{m'}])$, and handle each one separately. Rewrite the summations over $x_1,\ldots,x_m$ and apply the finite geometric series formula as needed. Finally, express the remaining summations as cluster tail generating functions of lower order k'.

The only part that is not immediate is whether the summations for $x_1, \ldots, x_m$ can be ordered properly and whether the lower and upper bounds for each summation index can be chosen properly so that we can adequately apply the finite geometric series formula. This can in fact always be done, and the ordering and choice of bounds can be done as follows:

Let $x_{i_1}, \ldots, x_{i_j}$ be the entries from the original last pattern p in the length k cluster that coincide with entries from the new last pattern p' in the length k' cluster. In other words, $x_{i_1}, \ldots, x_{i_j}$ are the terms that occur in the y_i's of $C(k', p'; [y_1, \ldots, y_{m'}])$. Let $x_{i_{j+1}}, \ldots, x_{i_m}$ be the terms that were "chopped off" from the length k cluster. Also, assume that $x_{i_1} < \ldots < x_{i_j}$ and $x_{i_{j+1}} < \ldots < x_{i_m}$. Note that in the example in Section 2.4.1, $x_{i_1} = x_1$ (not "chopped") while $x_{i_2} = x_2$ ("chopped") and $x_{i_3} = x_3$ ("chopped").

Order of summations:

The summations will be ordered (from outermost to innermost) as x_{i_1} to x_{i_j} followed by $x_{i_{j+1}}$ to x_{i_m}. Thus, the outermost summation is indexed by x_{i_1}, the next summation inward is indexed by x_{i_2}, and so on. This places the summations over $x_{i_{j+1}}, \ldots, x_{i_m}$ to be on the "inside" so that they can be moved inward to apply the finite geometric series formula.

Lower/Upper bounds for $x_{i_1}, \ldots, x_{i_m}$:

For each l with $j + 1 \leq l \leq m$, let $b_l = k$ if $i_l > i_j$;

otherwise, let $b_l = \min(\{i_1, \ldots, i_j\} \backslash \{1, \ldots, i_l\})$, and let c_l be the index of i (so $i_{c_l} = b_l$).

For $x_{i_1}, \ldots, x_{i_j}$:

$$x_{i_1} = i_1 \text{ to } k - m + i_1$$

$$x_{i_2} = x_{i_1} + i_2 - i_1 \text{ to } k - m + i_2$$

$$\cdots$$

$$x_{i_j} = x_{i_{j-1}} + i_j - i_{j-1} \text{ to } k - m + i_j$$

For $x_{i_{j+1}}$:
Lower bound is 1 if $i_{j+1} = 1$, and $x_{i_{j+1}-1} + 1$ otherwise. Upper bound is $k - m + i_{j+1}$ if $b_{j+1} = k$, and $b_{j+1} - c_{j+1} + i_{j+1}$ otherwise.

For x_{i_l} *with* $l > j + 1$:
Lower bound is $x_{i_l - 1} + 1$. Upper bound is $k - m + i_l$ if $b_l = k$, and $b_l - c_l + i_l$ otherwise.

One can see that the indices $x_{i_1}, \ldots, x_{i_j}$ range over all necessary values and can also verify that $x_{i_{j+1}}, \ldots, x_{i_m}$ will cover all necessary values as well. Additionally, for each r, the lower and upper bounds for x_{i_r} never depends on any x_{i_s} where $s > r$. If we applied the above approach to the example in Section 2.4.1, we would get $x_{i_1} = x_1$ going from 1 to $k - 2$, $x_{i_2} = x_2$ going from $x_1 + 1$ to $k - 1$, and $x_{i_3} = x_3$ going from $x_2 + 1$ to k.

2.4.3 Additional results

We get a couple more immediate byproducts from the algorithm in Section 2.4.2. First, the method provided for finding a functional equation always works, so we get the following:

Theorem 9. *Let B be a pattern set and $p \in B$. Then, there always exists a functional equation for $F(k, p; [z_1, \ldots, z_{|p|}])$ of the form:*

$$F(k, p; [z_1, \ldots, z_{|p|}]) = (t - 1) \sum_{p' \in B} \sum_{i \in I(p')} R_i \cdot F(k_i, p'; [M_1^i, \ldots, M_{|p'|}^i])$$

where $I(p')$ is a finite index set for each $p' \in B$, $I(p')$ and $I(p'')$ are disjoint if $p' \neq p''$, each M_j^i is a specific monomial in $z_1, \ldots, z_{|p|}$, each R_i is a specific rational expression in $z_1, \ldots, z_{|p|}$, and $k_i < k$ for each i.

Additionally, we get an immediate corollary of Theorem 7.

Corollary 4. *Let B be a set of patterns we would like to avoid. Without loss of generality, assume that B contains no redundancies. Then by setting $weight(p) = t-1$ for each $p \in B$, we get:*

$$P(z,t) = \frac{1}{1 - z - \sum\limits_{k \geq 1} \sum\limits_{p \in B} F(k,p;[1,\ldots,1]) \frac{z^k}{k!}}.$$

Since we can find a functional equation given any pattern set B, in a sense, we have an expression for the exponential generating function $P(z,t)$ for any pattern set.

2.4.4 Maple implementation

The algorithm from Section 2.4.2 has also been implemented in the Maple package `CAV`. Using that algorithm, we can find the sequence $\alpha_1, \ldots, \alpha_n$ avoiding a set of patterns B by calling the procedure `CAVT(B,n)`, where the patterns in B are represented as lists. For example, for $n = 10$ and $B = \{123, 321\}$, try `CAVT({[1,2,3],[3,2,1]},10);`. To keep track of the occurrences of patterns from B, use the procedure `CAVTt(B,n,t)`. For example, try `CAVTt({[1,2,3],[3,2,1]},10,t);`. To generate the cluster tail functional equation only (encoded again in a data structure that we call a cluster scheme), use the procedure `MakeTailFE(B,k,z,t)`. For example, try `MakeTailFE({[1,3,2]},k,z,t);`.

Computationally, the algorithm in Section 2.4.2 is much more efficient than the one in Section 2.3.1, so the `CAVT` procedure is much faster than the `CAV` procedure. In general, `CAVT` should be used instead of `CAV` for computing α_n values and, similarly, `CAVTt` should be used instead of `CAVt` for $c_n(t)$.

2.5 Asymptotic approximations

Let $B = \{p\}$ be a set containing a single pattern. In [58], Warlimont gave a conjecture on the asymptotics of α_n:

$$\alpha_n \sim \gamma \cdot \rho^n \cdot n!$$

where γ and ρ are constants depending only on the single pattern p. Some initial asymptotic results for α_n were proven by Elizalde in [21]. Recently, Ehrenborg, Kitaev, and Perry prove this conjecture in [19]. With this result established, we can compute approximate values of γ and ρ for various single patterns.

Elizalde and Noy gave some approximations of γ and ρ for length 3 and a few length 4 patterns in [23]. Aldred, Atkinson, and McCaughan also gave approximations for the ρ values of the single length 4 patterns. Using the Maple package `CAV`, we can empirically verify these approximations and also quickly produce many new approximations. For example, the procedure `AsymApprox(p,N,d)` will give approximate values (up to d decimal digits) for γ and ρ for the pattern p by computing α_{N-2}, α_{N-1}, and α_N and computing their ratios. For example, try `AsymApprox([1,2,4,3],50,20)`.

To approximate γ and ρ values (up to d decimal digits) for all length n patterns and then rank them by the size of ρ, use `AsymApproxRank(n,N,d)`. For example, `AsymApproxRank(4,30,10)` gives us the approximations for the γ and ρ values for length 4 patterns, and the results are given in Table 2.1.

Pattern	γ	ρ
1 2 3 4	1.1176930011	0.9630055289
2 4 1 3	1.1375931232	0.9577180134
2 1 4 3	1.1465405299	0.9561742431
1 3 2 4	1.1510444988	0.9558503134
1 4 2 3	1.1567436851	0.9548260509
1 3 4 2 $\sim$ 1 4 3 2	1.1561985648	0.9546118344
1 2 4 3	1.1696577874	0.9528914233

Table 2.1: Approximate asymptotics for length 4 patterns

Similarly, `AsymApproxRank(5,25,20)` would give us the approximations for the γ and ρ values for length 5 patterns. The output can be found on the paper's website.

2.6 Concluding remarks

In this chapter, we outlined the key procedures in the `CAV` Maple package. The cluster tail generating function was defined, and a constructive approach was demonstrated in finding a functional equation for it. Using this functional equation, we were able to more quickly enumerate permutations avoiding a prescribed set of patterns. In addition, by applying Theorem 8, we were able to totally classify c-Wilf-equivalences in single patterns of length 3, 4, and 5 rigorously, while nearly classifying single patterns of length 6. We were also able to classify c-Wilf-equivalences in a few cases of multiple pattern sets. Finally, we were able to use the faster algorithm to compute approximate values for asymptotic constants.

Despite this, there is a lot of room for improvement algorithmically and quite a few new open problems/conjectures arise. Some of the conjectures are listed below.

Elizalde and Noy provided the following conjecture in [23]:

Conjecture 8. *For a fixed pattern length k, the increasing pattern $\sigma = 12\ldots k$ is the "maximal" pattern, in the sense that $\alpha_n(\sigma) \geq \alpha_n(\pi)$ for all $\pi \in S_k$ and all n.*

Based off of experimentation, we also have the following analogous conjecture:

Conjecture 9. *For a fixed pattern length k, the pattern $\sigma = 12\ldots(k-2)(k)(k-1)$ is the "minimal" pattern, in the sense that $\alpha_n(\pi) \geq \alpha_n(\sigma)$ for all $\pi \in S_k$ and all n.*

These two conjectures were shown to be asymptotically true by Elizalde in [20]. An additional proof was provided by Perarnau in [47]. It is believed that these conjectures hold not just asymptotically but for all n, and this more specific problem remains open.

We also have analogous conjectures for the case of multiple patterns:

Conjecture 10. *For a fixed pattern length k, the pattern set $B = \{12\ldots k, 23\ldots k1\}$ is the "maximal" pattern set among sets of 2 patterns, in the sense that $\alpha_n(B) \geq \alpha_n(B')$ for all $B' \in \binom{S_k}{2}$ and all n.*

Conjecture 11. *For a fixed pattern length k, the pattern set $B = \{12\dots(k-2)(k)(k-1), 12\dots(k-3)(k-1)(k)(k-2)\}$ is the "minimal" pattern set among sets of 2 patterns, in the sense that $\alpha_n(B') \geq \alpha_n(B)$ for all $B' \in \binom{S_k}{2}$ and all n.*

Conjecture 12. *For a fixed pattern length k, the pattern set $B = \{12\dots k, 23\dots k1, k12\dots(k-1)\}$ is the "maximal" pattern set among sets of 3 patterns, in the sense that $\alpha_n(B) \geq \alpha_n(B')$ for all $B' \in \binom{S_k}{3}$ and all n.*

In addition, based off of empirical evidence for single pattern avoidance up to length 6 patterns, we believe the following:

Conjecture 13. *For any two patterns π_1 and π_2 of the same length, either π_1 and π_2 are strongly c-Wilf-equivalent or they are not c-Wilf-equivalent at all.*

In other words, two patterns are c-Wilf-equivalent if and only if they are also strongly c-Wilf-equivalent in the consecutive setting. This certainly holds for single patterns of length 3, 4, 5, and 6. It is interesting to note that this conjecture fails when considering classical pattern avoidance (e.g., the patterns 123 and 132 would provide a counter-example).

Chapter 3

Functional Equations and Algorithms for r Occurrences of a Pattern

Notice: *The section on patterns* $12\ldots k$ *represents joint work with Doron Zeilberger.*

3.1 Overview

In this chapter, we consider the classical notion of pattern containment. Given a sequence of k distinct positive integers $\sigma = \sigma_1 \ldots \sigma_k$, recall that the *reduction* $\mathrm{red}(\sigma)$ is the length k permutation $\tau = \tau_1 \ldots \tau_k$ that is order-isomorphic to σ. Given a (permutation) pattern $\tau \in \mathcal{S}_k$, we say that a permutation $\pi = \pi_1 \ldots \pi_n$ *contains* the pattern τ if there exists $1 \leq i_i < i_2 < \ldots < i_k \leq n$ such that $\mathrm{red}(\pi_{i_1}\pi_{i_2}\ldots\pi_{i_k}) = \tau$, in which case we call $\pi_{i_1}\pi_{i_2}\ldots\pi_{i_k}$ an *occurrence* of τ.[1] We will define $N_\tau(\pi)$ to be the number of occurrences of τ in π. For example, if the pattern $\tau = 123$, the permutation 53412 avoids the pattern τ (so $N_{123}(53412) = 0$), whereas the permutation 52134 contains two occurrences of τ (so $N_{123}(52134) = 2$).

Recall that for a pattern τ and non-negative integer $r \geq 0$, we have the set

$$\mathcal{S}_n(\tau, r) := \{\pi \in \mathcal{S}_n : \pi \text{ has exactly } r \text{ occurrences of the pattern } \tau\}$$

as well as the quantity $s_n(\tau, r) := |\mathcal{S}_n(\tau, r)|$. The corresponding generating function is defined as

$$F_\tau^r(x) := \sum_{n=0}^{\infty} s_n(\tau, r) x^n.$$

Recall that the classical pattern avoidance problem corresponds to the $r = 0$ case and has been well studied. In this setting, the exact enumeration is known for all patterns

[1] This definition differs from the last chapter, where the pattern must occur as consecutive terms.

of length at most 4 with the exception of the pattern 1324. Precise asymptotics are not even known for this pattern, although there has been some recent improvements in the upper bound for the growth rate. While the more general problem (where $r \geq 0$) has also been studied, the work has usually been restricted to small patterns (usually length three) and small r.

In this chapter, we consider how to modify and extend the initial functional equations by Noonan and Zeilberger in [45]. One difficulty arising from their original approach was that it became very complicated for even the $r = 2$ case. In addition, there are many patterns that this approach does not readily extend to. One such pattern (explicitly mentioned in [45]) is 1432.

In the following sections, we first present a modified approach for handling increasing patterns (i.e., patterns of the form $12 \ldots k$). Given a fixed $r \geq 0$, the resulting enumeration algorithm for computing $s_n(12 \ldots k, r)$ is polynomial-time (in n).[2] This, in a sense, tackles the first difficulty from [45] and allows us to enumerate the sequence $s_n(12 \ldots k, r)$ for even larger fixed r. It is important to note that, in general, the cardinality of the sets $\mathcal{S}_n(\tau, r)$ grow (roughly) exponentially in size as n increases. A direct constructive approach would therefore take at least exponential-time. This specific case is joint with Zeilberger and has already appeared in publication [42].

In the sections after, we first extend the approach to the family of patterns of the form $12 \ldots (k-2)(k)(k-1)$. This extension will also yield functional equations and polynomial-time algorithms (in n) for computing the sequence $s_n(12 \ldots (k-2)(k)(k-1), r)$. We then generalize the techniques to produce similar results for the family of patterns of the form $23 \ldots k1$. Unfortunately in this case, the current implementation of the enumeration algorithm does not appear to be polynomial-time, although it is still faster than a direct/constructive approach of producing the sets $\mathcal{S}_n(\tau, r)$.

We conclude the chapter by extending the generalized approach to the notorious length 4 pattern of 1324. Unfortunately, this approach does not yield a polynomial-time enumeration algorithm either. However, we can compute terms of $s_n(1324, r)$ for

[2] According to Wilf [59], this constitutes an answer to an enumerative question.

small r and reasonably small n. In the case of $r = 0$, it is possible to extract a new recurrence (on 0–1 matrices) and compute $s_n(1324, 0)$ for $n \leq 23$.[3]

We conclude this section with one final remark. It should be noted that this general enumeration approach is different from the *enumeration schemes* approach pioneered by Zeilberger [60] and extended by Vatter [57], Pudwell [48], and Baxter [5, 6]. The enumeration schemes approach seeks to derive recurrences for $s_n(\tau, 0)$ (generally with more complicated functions that keep track of additional parameters), while the approach in this chapter is based off of deriving functional equations. Also, the enumeration schemes approach is useful for enumerating pattern-avoiding permutations (the $r = 0$ case) but does not seem readily adaptable to the generalized setting for permutations with $r > 0$ occurrences of a pattern. On the other hand, the approach in this chapter is unfortunately not yet automated to discover new functional equations for different patterns. All the functional equations, while they are rigorously derived, are done through human means for each type of pattern. However, we hope that the techniques developed in this chapter may yield insight on how a computer can be trained to discover such functional equations on its own.

3.2 Patterns of the form $12 \dots k$

In this section, we modify the approach developed by Noonan and Zeilberger in [45]. We first handle the case of 123 in full detail and then outline how to generalize this approach to the patterns 1234, 12345, and so on. The results of this section is joint work with Zeilberger.

3.2.1 Permutations containing 123

For the sake of completeness, we reconstruct the functional equation derivation from [45]. Given a (fixed) pattern τ and non-negative integer n, we define the polynomial

$$f_n(t) := \sum_{\pi \in \mathcal{S}_n} t^{N_\tau(\pi)}. \tag{3.1}$$

[3] The method in [1] can compute the first 25 terms. Given that our implementation was in Maple, a lower level programming language (such as C) should be able to produce some more terms.

Observe that the coefficient of t^r in $f_n(t)$ is exactly equal to $s_n(\tau, r)$. For a fixed pattern τ and fixed $r \geq 0$, our goal is to quickly compute $s_n(\tau, r)$. In the remainder of this subsection, we will assume that $\tau = 123$.

In addition to the variable t, we introduce the catalytic variables $x_1, \ldots, x_n$ and define the weight of a length n permutation $\pi = \pi_1 \ldots \pi_n$ to be

$$\text{weight}_{123}(\pi) := t^{N_{123}(\pi)} \prod_{i=1}^{n} x_i^{\#\{(a,b)\,:\, \pi_a = i < \pi_b,\ 1 \leq a < b \leq n\}}.$$

In general, this will be written more simply as weight(π) when the fixed pattern is clear from context (in this case 123). For example,

$$\text{weight}(12345) = t^{10} x_1^4 x_2^3 x_3^2 x_4^1 x_5^0,$$

$$\text{weight}(54321) = 1,$$

$$\text{weight}(21354) = t^4 x_1^3 x_2^3 x_3^2.$$

For each n, we define the polynomial

$$P_n(t; x_1, \ldots, x_n) := \sum_{\pi \in \mathcal{S}_n} \text{weight}(\pi).$$

Observe that P_n is essentially a generalized multi-variate polynomial for f_n and in particular, $P_n(t; 1, \ldots, 1) = f_n(t)$. We now have the following:

Lemma 2. *Let $\pi = \pi_1 \ldots \pi_n$ and suppose that $\pi_1 = i$. If $\pi' := red(\pi_2 \ldots \pi_n)$, then*

$$\textit{weight}(\pi) = x_i^{n-i}\, \textit{weight}(\pi')|_{x_i \to t x_{i+1}\ ,\ x_{i+1} \to t x_{i+2}\ ,\ \ldots\ ,\ x_{n-1} \to t x_n}.$$

Proof. We re-insert i at the beginning of π' by shifting all the terms $i, i+1, \ldots, n-1$ up by 1 (i.e., $x_j \to x_{j+1}$ for $i \leq j$). The weight of this permutation (the original π) would gain an x_i^{n-i} term while the exponents of the other shifted catalytic variables remain the same. Also, observe that $N_{123}(\pi)$ is equal to the number of occurrences of 123 in π' **plus** the number of occurrences of 12 in $\pi_2 \ldots \pi_n$, where the term corresponding to the "1" is larger than i. $\square$

This directly leads to the Noonan-Zeilberger Function Equation from [45]:

Theorem 10. *For the pattern $\tau = 123$,*

$$P_n(t; x_1, \dots x_n) = \sum_{i=1}^{n} x_i^{n-i} P_{n-1}(t; x_1, \dots, x_{i-1}, tx_{i+1}, \dots, tx_n). \qquad \text{(NZFE1)}$$

Once $P_n(t; x_1, \dots, x_n)$ is computed, the catalytic variables $x_1, \dots, x_n$ can all be set to 1 to get $f_n(t) = P_n(t; 1, \dots, 1)$. In the original approach in [45], Noonan and Zeilberger simply plugged in $t = 0$ and $x_1 = \dots x_n = 1$ to get recurrences which can be used to show that $s_n(123, 0) = \frac{1}{n+1}\binom{2n}{n}$, the Catalan numbers. For $r = 1$, they differentiated Eq. (NZFE1) with respect to t using the multi-variable calculus chain rule and then plugged in $t = 0$ and $x_1 = \dots = x_n = 1$. However, for $r = 2$, this became a computational mess, even for a computer.

For computational purposes though, it is not necessary to compute $P_n(t; x_1, \dots, x_n)$ in its entirety prior to setting the catalytic variables to 1. Observe that by Eq. (NZFE1), we have:

$$P_n(t; 1, \dots, 1) = \sum_{i=1}^{n} P_{n-1}(t; 1\ [i-1\ \ times], t\ [n-i\ \ times]).$$

We get terms of the form $P_{a_0+a_1}(t; 1\ [a_0\ times], t\ [a_1\ times])$ in the summation, which can again be plugged into Eq. (NZFE1) to get:

$$\begin{gathered} P_{a_0+a_1}(t; 1\ [a_0\ times], t\ [a_1\ times]) = \\ \sum_{i=1}^{a_0} P_{a_0+a_1-1}(t; 1\ \ [i-1\ \ times]\,, t\ \ [a_0 - i\ \ times]\,, t^2\ \ [a_1\ \ times]) \\ + \sum_{i=1}^{a_1} t^{a_1-i} P_{a_0+a_1-1}(t; 1\ \ [a_0\ \ times]\,, t\ \ [i-1\ \ times]\,, t^2\ \ [a_1 - i\ \ times]) \end{gathered}$$

We must now deal with terms of the form $P_{a_0+a_1+a_2}(t; 1\ [a_0\ times], t\ [a_1\ times], t^2\ [a_2\ times])$. We can continue this recursive process of plugging new terms into Eq. (NZFE1) to eventually compute $f_n(t) = P_n(t; 1\ [n\ \ times])$.

This is much faster than the direct weighted counting of all $n!$ permutations, although it is still unfortunately an exponential-time (and memory) algorithm. Even then, we were still able to explicitly compute $f_n(t)$ up to $n = 20$. This is implemented in the Maple procedure `fn(n,t)` in the accompanying Maple package `P123`. The procedure `L20(t);` gives the pre-computed sequence of `fn(n,t)` for $1 \le n \le 20$. Here are the first few terms:

$$f_1(t) = 1 \quad , \quad f_2(t) = 2 \quad , \quad f_3(t) = t + 5 \quad , \quad f_4(t) = t^4 + 3\,t^2 + 6\,t + 14 \quad ,$$

$$f_5(t) = t^{10} + 4\,t^7 + 6\,t^5 + 9\,t^4 + 7\,t^3 + 24\,t^2 + 27\,t + 42 \quad ,$$

$$f_6(t) = t^{20} + 5\,t^{16} + 8\,t^{13} + 6\,t^{12} + 6\,t^{11} + 16\,t^{10} + 12\,t^9 + 24\,t^8$$
$$+32\,t^7 + 37\,t^6 + 54\,t^5 + 74\,t^4 + 70\,t^3 + 133\,t^2 + 110\,t + 132 \quad ,$$

$$f_7(t) = t^{35} + 6\,t^{30} + 10\,t^{26} + 10\,t^{25} + 8\,t^{23} + 13\,t^{22} + 30\,t^{21} + 10\,t^{20} + 32\,t^{19} + 18\,t^{18}$$
$$+62\,t^{17} + 74\,t^{16} + 24\,t^{15} + 100\,t^{14} + 130\,t^{13} + 104\,t^{12} + 162\,t^{11} + 191\,t^{10} + 232\,t^9$$
$$+260\,t^8 + 320\,t^7 + 387\,t^6 + 395\,t^5 + 507\,t^4 + 461\,t^3 + 635\,t^2 + 429\,t + 429 \quad ,$$

$$f_8(t) = t^{56} + 7\,t^{50} + 12\,t^{45} + 15\,t^{44} + 10\,t^{41} + 16\,t^{40} + 40\,t^{39} + 18\,t^{38} + 47\,t^{36} + 38\,t^{35} + 68\,t^{34}$$
$$+60\,t^{33} + 58\,t^{32} + 66\,t^{31} + 154\,t^{30} + 138\,t^{29} + 115\,t^{28} + 156\,t^{27} + 252\,t^{26} + 324\,t^{25} + 228\,t^{24}$$
$$+288\,t^{23} + 537\,t^{22} + 466\,t^{21} + 546\,t^{20} + 656\,t^{19} + 682\,t^{18} + 1004\,t^{17} + 1047\,t^{16} + 886\,t^{15}$$
$$+1494\,t^{14} + 1456\,t^{13} + 1580\,t^{12} + 1818\,t^{11} + 2077\,t^{10} + 2182\,t^9 + 2389\,t^8 + 2544\,t^7 + 2864\,t^6$$
$$+2570\,t^5 + 3008\,t^4 + 2528\,t^3 + 2807\,t^2 + 1638\,t + 1430 \quad .$$

Suppose that for a small fixed $r \geq 0$, we wanted the first 20 terms of the sequence $s_n(123, r)$. By this functional equation approach, one would compute $f_n(t)$ and extract the coefficient of t^r for each n up to 20. This approach would expend quite a bit of computational effort in generating unnecessary information (namely, all the t^k terms where $k > r$). This issue can mostly be circumvented, however, by a couple of observations. First, we have the following observation:

Lemma 3. *Let $n = a_0 + a_1 + \ldots + a_s$ (where $a_i \geq 0$ for each i) and suppose $s > r + 1$. Then, the coefficients of $t^0, t^1, \ldots, t^r$ in*

$$P_n(t; 1\,[a_0 \;\; times]\,, \ldots, t^{s-1}\,[a_{s-1} \;\; times]\,, t^s\,[a_s \;\; times])$$
$$-P_n(t; 1\,[a_0 \;\; times]\,, \ldots, t^r\,[a_r \;\; times]\,, t^{r+1}\,[a_{r+1} + a_{r+2} + \ldots + a_s \;\; times])$$

all vanish.

Proof. The more general function $P_n(t; x_1, \ldots, x_n)$ is a multi-variate polynomial. $\square$

This lemma allows us to collapse all the higher powers of t into the t^{r+1} coefficient and allows us to consider objects of the form $P_n(t; 1\,[a_0 \text{ times}], \ldots, t^r\,[a_r \text{ times}], t^{r+1}\,[a_{r+1} \text{ times}])$ regardless of how large n is.

Let $n := a_0 + a_1 + \ldots + a_{r+1}$. Also, for any expression R and positive integer k, let $R\$k$ denote $R\,[k \ \ times]$. For example, $t^3\$4$ is shorthand for t^3, t^3, t^3, t^3. Now for any polynomial $p(t)$ in the variable t, let $p^{(r)}(t)$ denote the polynomial of degree (at most) r obtained by discarding all powers of t larger than r. Also, define the operator CHOP_r by $\text{CHOP}_r[p(t)] := p^{(r)}(t)$.

An application of (NZFE1) and CHOP_r to $P_n^{(r)}(t; 1\$a_0, \ldots, t^r\$a_r, t^{r+1}\$a_{r+1})$ becomes:

$$P_n^{(r)}(t; 1\$a_0\ ,\ t\$a_1\ ,\ \ldots\ ,\ t^r\$a_r\ ,\ t^{r+1}\$a_{r+1})$$

$$= \text{CHOP}_r\left[\sum_{i=1}^{a_0} P_{n-1}^{(r)}(t; 1\$(i-1),\ t\$(a_0-i),\ t^2\$a_1,\ \ldots,\ t^r\$a_{r-1},\ t^{r+1}\$(a_r+a_{r+1}))\right.$$

$$+\sum_{i=1}^{a_1} t^{(a_1-i)+a_2+\ldots+a_{r+1}} P_{n-1}^{(r)}(t; 1\$a_0,\ t\$(i-1),\ t^2\$(a_1-i),\ \ldots,\ t^{r+1}\$(a_r+a_{r+1}))$$

$$+\sum_{i=1}^{a_2} t^{2((a_2-i)+a_3+\ldots+a_{r+1})} P_{n-1}^{(r)}(t; 1\$a_0,\ \ldots\ t^3\$(a_2-i),\ \ldots,\ t^r\$a_{r-1},\ t^{r+1}\$(a_r+a_{r+1}))$$

$$+\ldots\ldots$$

$$\left.+\sum_{i=1}^{a_{r+1}} t^{(r+1)(a_{r+1}-i)} P_{n-1}^{(r)}(t; 1\$a_0,\ t\$a_1,\ \ldots,\ t^r\$a_r,\ t^{r+1}\$(a_{r+1}-1))\right].$$

Due to the CHOP_r operator, many terms automatically disappear because of the power of t in front. From a computational perspective, this observation eliminates many unnecessary terms and hence circumvents a lot of unnecessary computation. The important point is that a computer can automatically generate a "scheme" for computing the degree-r polynomials in t of the form:

$$P_{a_0+\ldots+a_{r+1}}^{(r)}(t; 1\,[a_0 \text{ times}]\,,\ t\,[a_1 \text{ times}]\,,\ \ldots,\ t^r\,[a_r \text{ times}]\,,\ t^{r+1}\,[a_{r+1} \text{ times}]),$$

with $a_0+\ldots+a_{r+1} = n$ and $a_0, \ldots, a_{r+1} \geq 0$. The number of such objects to consider is $\binom{r+n+1}{r+1}$. So each iteration involves $\mathcal{O}(n^{r+1})$ evaluations and hence $\mathcal{O}(n^{r+2})$ additions.

Doing this n times yields an $\mathcal{O}(n^{r+3})$ algorithm for finding our desired polynomial:

$$f_n^{(r)}(t) = P_n^{(r)}(t; 1\,[n \text{ times}]\,,\ t\,[0 \text{ times}]\,,\ \ldots,\ t^{r+1}\,[0 \text{ times}]).$$

Having found the “scheme”, a computer can use it to generate as many terms as desired.[4]

The accompanying Maple package `P123` implements the functional equation NZFE1 and easily generates the first 25 terms of the enumerating sequences for $0 \leq r \leq 7$. From this data, we can empirically verify the already-known results for $s_n(123, r)$ for $0 \leq r \leq 4$ [44, 25, 14] and make conjectures for $5 \leq r \leq 7$ as follows:

$$a_0(n) = 2\,\frac{(2\,n-1)!}{(n-1)!\;(n+1)!} \quad .$$

$$a_1(n) = 6\,\frac{(2\,n-1)!}{(n-3)!\;(n+3)!} \quad .$$

$$a_2(n) = \frac{(2\,n-2)!}{(n-4)!\;(n+5)!} \cdot (59\,n^2 + 117\,n + 100) \quad .$$

$$a_3(n) = \frac{(2\,n-3)!}{(n-5)!\;(n+7)!} \cdot 4\,n\left(113\,n^3 + 506\,n^2 + 937\,n + 1804\right) \quad .$$

$$a_4(n) = \frac{(2n-4)!}{(n-4)!(n+9)!} \cdot$$

$$(\,3561\,n^8 + 3126\,n^7 - 46806\,n^6 + 12384\,n^5 - 659091\,n^4$$

$$+2630634\,n^3 + 5520576\,n^2 + 26283456\,n - 39191040\,) \quad .$$

[4] As a reminder, the “scheme” mentioned here is a liberal use of the word and differs from *enumeration schemes*.

$$a_5(n) = \frac{(2n-5)!}{(n-5)!(n+11)!} \cdot$$
$$(\, 26246\, n^{10} + 136646\, n^9 - 115872\, n^8 + 22524\, n^7 - 9648450\, n^6 + 71304534\, n^5$$
$$+381205612\, n^4 + 1607633896\, n^3 + 2800103664\, n^2 + 3611692800\, n - 32891443200\,) \quad .$$

$$a_6(n) = \frac{(2n-6)!}{(n-6)!(n+13)!} \cdot$$
$$(\, 193311\, n^{12} + 2349954\, n^{11} + 13035003\, n^{10} + 95151030\, n^9 + 406430793\, n^8 + 2889552582\, n^7$$
$$+14335663329\, n^6 + 60005854890\, n^5 + 313010684796\, n^4 + 1025692693464\, n^3$$
$$+1283595375168\, n^2 - 6909513045120\, n - 28177269120000\,) \quad .$$

$$a_7(n) = \frac{(2n-7)!}{(n-5)!(n+15)!} \cdot$$
$$(\, 1386032\, n^{16} + 13111080\, n^{15} + 22526480\, n^{14} + 355187760\, n^{13} - 1654450096\, n^{12}$$
$$+10534951680\, n^{11} + 15797223760\, n^{10} - 305671694640\, n^9$$
$$+3750695521216\, n^8 - 26631101348520\, n^7 - 86395090065440\, n^6$$
$$-636425872408320\, n^5 + 3647384624274048\, n^4 + 11386434230674560\, n^3$$
$$+103032675524966400\, n^2 - 157858417817856000\, n - 763734137886720000\,) \quad .$$

3.2.2 Permutations containing 1234

We now show how to extend the previous approach to the pattern 1234. In addition to the variable t, we now introduce $2n$ catalytic variables $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$. The weight of a length n permutation $\pi = \pi_1 \ldots \pi_n$ will now be

$$\text{weight}(\pi) := t^{N_{1234}(\pi)} \prod_{i=1}^{n} x_i^{\#\{(a,b)\,:\,\pi_a = i < \pi_b,\; 1 \le a < b \le n\}} \cdot y_i^{\#\{(a,b,c)\,:\,\pi_a = i < \pi_b < \pi_c,\; 1 \le a < b < c \le n\}}.$$

For example,

$$\text{weight}(123456) = t^{15}x_1^5x_2^4x_3^3x_4^2x_5y_1^{10}y_2^6y_3^3y_4,$$

$$\text{weight}(654321) = 1,$$

$$\text{weight}(345612) = tx_1x_3^3x_4^2x_5y_3^3y_4.$$

For each n, we define the polynomial

$$P_n(t; x_1, \ldots, x_n;\ y_1, \ldots, y_n) := \sum_{\pi \in \mathcal{S}_n} \text{weight}(\pi).$$

We now observe the following:

Lemma 4. *Let $\pi = \pi_1 \ldots \pi_n$ and suppose that $\pi_1 = i$. If $\pi' := red(\pi_2 \ldots \pi_n)$, then*

$$weight(\pi) = x_i^{n-i} weight(\pi')|_{x_i \to y_i x_{i+1}\ ,\ \ldots\ ,\ x_{n-1} \to y_i x_n\ ,\ y_i \to t y_{i+1}\ ,\ \ldots\ ,\ y_{n-1} \to t y_n}.$$

Proof. We consider i to be a fixed value and re-insert i at the beginning of π' by shifting all the terms $i, i+1, \ldots, n-1$ up by 1 (i.e., $x_j \to x_{j+1}$ and $y_j \to y_{j+1}$ for $i \leq j$). The new i would create $n-i$ new 12 patterns and would require an extra x_i^{n-i} factor for the weight. Also, the re-insertion of i would recreate new 123 patterns. The number of such new patterns is exactly the number of 12 patterns in the shifted π', where the "1" is greater than i. Therefore, our x_j shift now becomes $x_j \to y_i x_{j+1}$ for $i \leq j$. Also, observe that $N_{1234}(\pi)$ is equal to the number of occurrences of 1234 in π' **plus** the number of occurrences of 123 in $\pi_2 \ldots \pi_n$, where the term corresponding to the "1" is larger than i. Therefore, our y_j shift now becomes $y_j \to t y_{j+1}$ for $i \leq j$. □

This directly leads to the Noonan-Zeilberger Function Equation for pattern 1234 from [45]:

Theorem 11. *For the pattern $\tau = 1234$,*

$$P_n(t; x_1, \ldots x_n) = \sum_{i=1}^{n} x_i^{n-i} P_{n-1}(t;\ x_1, \ldots, x_{i-1}, y_i x_{i+1}, \ldots, y_i x_n;\ y_1, \ldots, y_{i-1}, t y_{i+1}, \ldots, t y_n). \tag{NZFE2}$$

Again, our goal is to compute $f_n(t) = P_n(t; 1\,[2n \text{ times}])$. We can apply the same computational methods as before. For example, we can apply (NZFE2) directly to $P_n(t; 1\,[2n \text{ times}])$, and more generally, to objects of the form

$$P_n(t; 1\,[a_0 \text{ times}]\,, \ldots, t^{s_1}\,[a_{s_1} \text{ times}]\,; 1\,[b_0 \text{ times}]\,, \ldots, t^{s_2}\,[b_{s_2} \text{ times}])$$

to compute $f_n(t)$. This again gives us an algorithm that is faster than the direct weighted counting of $n!$ permutations but is still exponential-time (and memory). This has been implemented in the accompanying Maple package `P1234`, and the first few polynomials are:

$$f_1(q) = 1\,, \quad f_2(q) = 2\,, \quad f_3(q) = 6\,, \quad f_4(q) = q + 23\,, \; f_5(q) = q^5 + 4\,q^2 + 12\,q + 103 \quad ,$$

$$f_6(q) = q^{15} + 5\,q^9 + 8\,q^6 + 12\,q^5 + 6\,q^4 + 10\,q^3 + 63\,q^2 + 102\,q + 513 \quad ,$$

$$f_7(q) = q^{35} + 6\,q^{25} + 10\,q^{19} + 18\,q^{16} + 12\,q^{15} + 13\,q^{13} + 24\,q^{11} + 32\,q^{10} + 72\,q^9 + 10\,q^8$$
$$+46\,q^7 + 142\,q^6 + 116\,q^5 + 146\,q^4 + 196\,q^3 + 665\,q^2 + 770\,q + 2761 \quad ,$$

$$f_8(q) = q^{70} + 7\,q^{55} + 12\,q^{45} + 15\,q^{41} + 10\,q^{39} + 8\,q^{36} + 28\,q^{35} + 40\,q^{32} + 41\,q^{29} + 10\,q^{28}$$
$$+24\,q^{27} + 44\,q^{26} + 84\,q^{25} + 24\,q^{24} + 89\,q^{23} + 12\,q^{21} + 142\,q^{20} + 136\,q^{19} + 96\,q^{18} + 115\,q^{17}$$
$$+333\,q^{16} + 156\,q^{15} + 112\,q^{14} + 312\,q^{13} + 199\,q^{12} + 600\,q^{11} + 573\,q^{10} + 804\,q^9 + 503\,q^8$$
$$+885\,q^7 + 1782\,q^6 + 1204\,q^5 + 2148\,q^4 + 2477\,q^3 + 5982\,q^2 + 5545\,q + 15767 \quad .$$

Additionally, both the obvious analog of Lemma 3 as well as the computational reduction using the CHOP_r operator still apply in this setting. This has been implemented in the accompanying Maple package `F1234`. As in the previous subsection, one can get polynomial-time (in n) algorithms to compute $s_n(1234, r)$, however the $\mathcal{O}(n^{r+3})$ becomes $\mathcal{O}(n^{2r+5})$ since we have twice as many catalytic variables. Nevertheless, we were still able to compute the first 70 terms for the case $r = 1$. Here are the first 23 terms:

$$0, 0, 0, 1, 12, 102, 770, 5545, 39220, 276144, 1948212, 13817680, 98679990,$$
$$710108396, 5150076076, 37641647410, 277202062666, 2056218941678, 15358296210724,$$
$$115469557503753, 873561194459596, 6647760790457218, 50871527629923754 \quad .$$

Additionally, Manuel Kauers programmed our algorithm in C and used clever programming techniques to compute the first 200 terms. The output is available at: `http://www.math.rutgers.edu/~zeilberg/tokhniot/oF1234bManuelKauers`.

3.2.3 Extending to longer patterns

The approach for the patterns 123 and 1234 can be extended analogously to longer patterns of the form $12\ldots k$. For example, if the pattern $\tau = 12345$, we consider the variable t and $3n$ catalytic variables: $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ and $z_1, \ldots, z_n$. The weight of a length n permutation $\pi = \pi_1 \ldots \pi_n$ will now be

$$\text{weight}(\pi) := t^{N_{12345}(\pi)} \prod_{i=1}^{n} x_i^{\#\{(a,b)\,:\,\pi_a = i < \pi_b\}} \cdot y_i^{\#\{(a,b,c)\,:\,\pi_a = i < \pi_b < \pi_c\}} \cdot z_i^{\#\{(a,b,c,d)\,:\,\pi_a = i < \pi_b < \pi_c < \pi_d\}}$$

where it is always assumed that $a < b < c < d$.

An analogous functional equation is derived for the corresponding polynomial

$$P_n(t; x_1, \ldots, x_n;\ y_1, \ldots, y_n;\ z_1, \ldots, z_n) := \sum_{\pi \in \mathcal{S}_n} \text{weight}(\pi)$$

and all the analogous computational methods work in this setting as well. This has been implemented in the Maple packages `P12345` and `F12345`. For example, the first 20 terms for $r = 0$ are:

$$1, 2, 6, 24, 119, 694, 4582, 33324, 261808, 2190688, 19318688, 178108704,$$
$$1705985883, 16891621166, 172188608886, 1801013405436, 19274897768196,$$
$$210573149141896, 2343553478425816, 26525044132374656 \quad .$$

The first 20 terms for $r = 1$ are:

$$0, 0, 0, 0, 1, 20, 270, 3142, 34291, 364462, 3844051, 40632886, 432715409,$$
$$4655417038, 50667480496, 558143676522, 6223527776874, 70228214538096,$$
$$801705888742781, 9254554670121572 \quad .$$

3.3 Patterns of the form $12\ldots(k-2)(k)(k-1)$

In this section, we adapt the approach from the previous section (for increasing patterns) to the patterns $12\ldots(k-2)(k)(k-1)$. We first handle the case of 132 and then outline how to generalize this approach to patterns 1243, 12354, and so on.

3.3.1 Permutations containing 132

Given a (fixed) pattern τ and non-negative integer n, we again consider the polynomial

$$f_n(t) := \sum_{\pi \in \mathcal{S}_n} t^{N_\tau(\pi)}. \tag{3.2}$$

Recall that the coefficient of t^r in $f_n(t)$ is exactly equal to $s_n(\tau, r)$. For a fixed pattern τ and fixed $r \geq 0$, our goal is to quickly compute $s_n(\tau, r)$. In the remainder of this subsection, we will assume that $\tau = 132$.

In addition to the variable t, we introduce the catalytic variables $x_1, \ldots, x_n$ and define the weight of a length n permutation $\pi = \pi_1 \ldots \pi_n$ to be

$$\text{weight}(\pi) := t^{N_{132}(\pi)} \prod_{i=1}^{n} x_i^{\#\{(a,b)\,:\,\pi_a > \pi_b = i,\ 1 \leq a < b \leq n\}}.$$

For example, weight$(12345) = 1$, weight$(13245) = tx_2$, and weight$(25143) = t^4 x_1^2 x_3^2 x_4$.

For each n, we again define the polynomial

$$P_n(t; x_1, \ldots, x_n) := \sum_{\pi \in \mathcal{S}_n} \text{weight}(\pi).$$

Recall that $P_n(t; 1, \ldots, 1) = f_n(t)$. We can now observe the following:

Lemma 5. *Let $\pi = \pi_1 \ldots \pi_n$ and suppose that $\pi_1 = i$. If $\pi' := red(\pi_2 \ldots \pi_n)$, then*

$$weight(\pi) = x_1 x_2 \ldots x_{i-1} \cdot weight(\pi')|_{x_i \to t x_{i+1}\,,\ x_{i+1} \to t x_{i+2}\,,\ \ldots\,,\ x_{n-1} \to t x_n}.$$

Proof. We re-insert i at the beginning of π' by shifting all the terms $i, i+1, \ldots, n-1$ up by 1 (i.e., $x_j \to x_{j+1}$ for $i \le j$). The new "i" would create new 21 patterns and would require an extra factor of $x_1 x_2 \ldots x_{i-1}$ for the weight. Also, observe that $N_{132}(\pi)$ is equal to the number of occurrences of 132 in π' **plus** the number of occurrences of 21 in $\pi_2 \ldots \pi_n$, where the term corresponding to the "1" is larger than i. Therefore, our x_j shift now becomes $x_j \to t x_{j+1}$ for $i \le j$. □

This directly leads to the new functional equation:

Theorem 12. *For the pattern $\tau = 132$,*

$$P_n(t; x_1, \ldots, x_n) = \sum_{i=1}^{n} x_1 x_2 \ldots x_{i-1} \cdot P_{n-1}(t; x_1, \ldots, x_{i-1}, t x_{i+1}, \ldots, t x_n). \qquad \text{(FE132)}$$

As before, once $P_n(t; x_1, \ldots, x_n)$ is computed, the catalytic variables $x_1, \ldots, x_n$ can all be set to 1 to get $f_n(t) = P_n(t; 1, \ldots, 1)$.

We can again apply the same computational techniques as before. First, we can apply the functional equation (FE132) directly to $P_n(t; 1, \ldots, 1)$ to get:

$$P_n(t; 1, \ldots, 1) = \sum_{i=1}^{n} P_{n-1}(t; 1\ [i-1\ \ times], t\ [n-i\ \ times]).$$

We get terms of the form $P_{a_0+a_1}(t; 1\ [a_0\ times], t\ [a_1\ times])$ in the summation, which can again be plugged into (FE132) to get:

$$\begin{gathered} P_{a_0+a_1}(t; 1\ [a_0\ times], t\ [a_1\ times]) = \\ \sum_{i=1}^{a_0} P_{a_0+a_1-1}(1\ \ [i-1\ \ times]\,, t\ \ [a_0-i\ \ times]\,, t^2\ \ [a_1\ \ times]) \\ + \sum_{i=1}^{a_1} t^{i-1} P_{a_0+a_1-1}(1\ \ [a_0\ \ times]\,, t\ \ [i-1\ \ times]\,, t^2\ \ [a_1-i\ \ times]) \end{gathered}$$

Now, we must deal with terms of the form $P_{a_0+a_1+a_2}(t; 1\ [a_0\ times], t\ [a_1\ times], t^2\ [a_2\ times])$. We can continue this recursive process of plugging new terms into (FE132) to eventually compute $f_n(t) = P_n(t; 1\ [n\ \ times])$. This is much faster than the direct weighted counting of all $n!$ permutations, although it is still unfortunately an exponential-time (and memory) algorithm.

This algorithm has been implemented in the accompanying Maple package `FINCRT`. For example, here are the first few terms:

$$f_1(t) = 1 \quad , \quad f_2(t) = 2 \quad , \quad f_3(t) = t + 5 \quad , \quad f_4(t) = t^3 + 4\,t^2 + 5\,t + 14 \quad ,$$

$$f_5(t) = 3\,t^6 + 5\,t^5 + 12\,t^4 + 14\,t^3 + 23\,t^2 + 21\,t + 42 \quad ,$$

$$f_6(t) = 2\,t^{12} + 10\,t^{10} + 22\,t^9 + 29\,t^8 + 37\,t^7 + 64\,t^6$$
$$+55\,t^5 + 96\,t^4 + 82\,t^3 + 107\,t^2 + 84\,t + 132$$

$$f_7(t) = 2\,t^{20} + t^{19} + 13\,t^{18} + 14\,t^{17} + 23\,t^{16} + 55\,t^{15} + 93\,t^{14}$$
$$+126\,t^{13} + 206\,t^{12} + 175\,t^{11} + 281\,t^{10} + 298\,t^9 + 360\,t^8 + 365\,t^7$$
$$+475\,t^6 + 394\,t^5 + 526\,t^4 + 410\,t^3 + 464\,t^2 + 330\,t + 429$$

Suppose that for a small fixed $r \geq 0$, we want to compute the sequence (in n) given by $s_n(132, r)$. As in the previous sections, this current approach computes many unnecessary terms of $f_n(t)$ (namely, all the t^k terms where $k > r$). Fortunately, the same computational methods from the 123 pattern case apply in the present setting as well. In particular, the obvious analog of Lemma 3 as well as the reduction using the CHOP_r operator can be applied here. Recall that given a polynomial $p(t)$, $\mathrm{CHOP}_r[p(t)]$ discards all the powers of t larger than r.

From a computational perspective, these techniques again eliminate many unnecessary terms and hence circumvent a lot of unnecessary computation. This has been automated in the Maple package `FINCRT` so that a computer can derive a "scheme" for any fixed r (completely on its own) and use it to enumerate $s_n(132, r)$ for as many terms as the user wants.

For example, the Maple call `F132rN(5,15);` for the first 15 terms of $s_n(132, 5)$ produces the sequence:

$$0, 0, 0, 0, 5, 55, 394, 2225, 11539, 57064, 273612, 1283621, 5924924, 27005978, 121861262$$

which is the sequence A138163 in the On-Line Encyclopedia of Integer Sequences [54].

Finally, the same run-time analysis carries over from the 123 case to verify that this algorithm is polynomial-time. In particular, this approach (combined with the mentioned computational techniques) would produce an $\mathcal{O}(n^{r+3})$ algorithm for our desired computations.

3.3.2 Permutations containing 1243

We now outline how to extend the previous approach to the pattern 1243. In addition to the variable t, we now introduce $2n$ catalytic variables $x_1, \dots, x_n$ and $y_1, \dots, y_n$. The weight of a length n permutation $\pi = \pi_1 \dots \pi_n$ will now be

$$\text{weight}(\pi) := t^{N_{1243}(\pi)} \prod_{i=1}^{n} x_i^{\#\{(a,b)\,:\,\pi_a > \pi_b = i,\ 1 \le a < b \le n\}} \cdot y_i^{\#\{(a,b,c)\,:\,\pi_a = i < \pi_c < \pi_b,\ 1 \le a < b < c \le n\}}.$$

For example, $\text{weight}(123456) = 1$ and $\text{weight}(135624) = t^2 x_2^3 x_4^2 y_1^5 y_3^2$.

For each n, we define the polynomial

$$P_n(t; x_1, \dots, x_n;\ y_1, \dots, y_n) := \sum_{\pi \in \mathcal{S}_n} \text{weight}(\pi).$$

We now observe the following:

Lemma 6.

$$\textit{weight}(\pi) = x_1 x_2 \dots x_{i-1} \cdot \textit{weight}(\pi')|_{x_i \to y_i x_{i+1}\ ,\ \dots\ ,\ x_{n-1} \to y_i x_n\ ,\ y_i \to t y_{i+1}\ ,\ \dots\ ,\ y_{n-1} \to t y_n}.$$

Proof. We consider i to be a fixed value and re-insert i at the beginning of π' by shifting all the terms $i, i+1, \dots, n-1$ up by 1 (i.e., $x_j \to x_{j+1}$ and $y_j \to y_{j+1}$ for $i \le j$). The new "i" would create new 21 patterns and would require an extra factor of $x_1 x_2 \dots x_{i-1}$ for the weight. Also, the re-insertion of i would recreate new 132 patterns. The number of such new patterns is exactly the number of 21 patterns in the shifted π', where the "1" is greater than i. Therefore, our x_j shift now becomes $x_j \to y_i x_{j+1}$ for $i \le j$. Also, observe that $N_{1243}(\pi)$ is equal to the number of occurrences of 1243 in π' **plus** the number of occurrences of 132 in $\pi_2 \dots \pi_n$, where the term corresponding to the "1" is larger than i. Therefore, our y_j shift now becomes $y_j \to t y_{j+1}$ for $i \le j$. $\square$

This directly leads to the new functional equation:

Theorem 13. *For the pattern $\tau = 1243$,*

$$P_n(t; x_1, \ldots, x_n, y_1, \ldots, y_n) =$$

$$\sum_{i=1}^{n} x_1 x_2 \ldots x_{i-1} \cdot P_{n-1}(t;\ x_1, \ldots, x_{i-1}, y_i x_{i+1}, \ldots, y_i x_n;\ y_1, \ldots, y_{i-1}, t y_{i+1}, \ldots, t y_n). \tag{FE1243}$$

Again, our goal is to compute $f_n(t) = P_n(t; 1\, [2n \text{ times}])$. We can apply the same computational methods as before. For example, we can apply (FE1243) directly to $P_n(t; 1\, [2n \text{ times}])$, and more generally, to objects of the form

$$P_n(t; 1\, [a_0 \text{ times}]\,, \ldots, t^{s_1}\, [a_{s_1} \text{ times}]\,; 1\, [b_0 \text{ times}]\,, \ldots, t^{s_2}\, [b_{s_2} \text{ times}])$$

to compute $f_n(t)$. This again gives us an algorithm that is faster than the direct weighted counting of $n!$ permutations but is still exponential-time (and memory).

This algorithm has been implemented in the procedure `F1243full(n,t)` (in Maple package `FINCRT`). For example, the Maple call `F1243full(8,t);` computes $f_8(t)$ and outputs:

$$t^{36} + t^{31} + 10\,t^{30} + 3\,t^{28} + 13\,t^{27} + 9\,t^{26} + 8\,t^{25} + 37\,t^{24} + 16\,t^{23} + 16\,t^{22} + 49\,t^{21} + 60\,t^{20}$$
$$+41\,t^{19} + 130\,t^{18} + 81\,t^{17} + 157\,t^{16} + 266\,t^{15} + 184\,t^{14} + 233\,t^{13} + 542\,t^{12} + 356\,t^{11} + 771\,t^{10}$$
$$+877\,t^{9} + 975\,t^{8} + 972\,t^{7} + 2180\,t^{6} + 1710\,t^{5} + 2658\,t^{4} + 3119\,t^{3} + 4600\,t^{2} + 4478\,t + 15767$$

Additionally, both the obvious analog of Lemma 3 as well as the computational reduction using the CHOP_r operator still apply in this setting. This has also been automated in the Maple package `FINCRT`.

For example, the Maple call `F1243rN(1,15);` for the first 15 terms of $s_n(1243, 1)$ produces the sequence:

$$0, 0, 0, 1, 11, 88, 638, 4478, 31199, 218033, 1535207, 10910759, 78310579, 567588264, 4152765025$$

and the Maple call `F1243rN(2,15);` for the first 15 terms of $s_n(1243, 2)$ produces the sequence:

$$0, 0, 0, 0, 4, 56, 543, 4600, 36691, 284370, 2174352, 16533360, 125572259, 955035260, 7283925999$$

Finally, the same run-time analysis carries over from the 1234 case to verify that this algorithm is polynomial-time. In particular, this approach (combined with the mentioned computational techniques) would produce an $\mathcal{O}(n^{2r+5})$ algorithm for our desired computations.

3.3.3 Extending to longer patterns

The approach for the patterns 132 and 1243 can be extended analogously to longer patterns of the form $12\ldots(k-2)(k)(k-1)$. For example, if the pattern $\tau = 12354$, we consider the variable t and $3n$ catalytic variables: $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ and $z_1, \ldots, z_n$. The weight of a length n permutation $\pi = \pi_1 \ldots \pi_n$ will now be

$$\text{weight}(\pi) :=$$
$$t^{N_{12354}(\pi)} \prod_{i=1}^{n} x_i^{\#\{(a,b)\,:\,\pi_a>\pi_b=i\}} \cdot y_i^{\#\{(a,b,c)\,:\,\pi_a=i<\pi_c<\pi_b\}} \cdot z_i^{\#\{(a,b,c,d)\,:\,\pi_a=i<\pi_b<\pi_d<\pi_c\}}$$

where it is always assumed that $a < b < c < d$.

An analogous functional equation is derived for the corresponding polynomial

$$P_n(t; x_1, \ldots, x_n;\ y_1, \ldots, y_n;\ z_1, \ldots, z_n) := \sum_{\pi \in \mathcal{S}_n} \text{weight}(\pi)$$

and all the analogous computational methods work in this setting as well. The 12354 case has also been automated in the Maple package `FINCRT`.

For example, the Maple call `F12354rN(0,14);` for the first 14 terms of $s_n(12354, 0)$ produces the sequence:

$$1, 2, 6, 24, 119, 694, 4582, 33324, 261808, 2190688,$$
$$19318688, 178108704, 1705985883, 16891621166$$

and the Maple call `F12354rN(1,15);` for the first 15 terms of $s_n(12354, 1)$ produces the sequence:

$$0, 0, 0, 0, 1, 19, 246, 2767, 29384, 305646, 3170684,$$
$$33104118, 349462727, 3738073247, 40549242195$$

3.4 Patterns of the form $23\ldots k1$

In this section, we generalize the previous techniques to handle patterns of the form $23\ldots k1$. Although $s_n(231,r) = s_n(132,r)$ for every r and n (by reversal), we will develop an approach for handling 231 directly[5] and then show how this can be extended to longer patterns of the form $23\ldots k1$. The techniques developed in this section will be necessary for extending this general methodology to handle the pattern 1324 in the next section. This alternate approach for 231 will also be used when considering multiple patterns simultaneously in the next chapter.

3.4.1 Permutations containing 231

In this subsection, we will assume that our (fixed) pattern $\tau = 231$. We define the analogous polynomial

$$f_n(t) := \sum_{\pi\in\mathcal{S}_n} t^{N_{231}(\pi)}.$$

Recall that the coefficient of t^r in $f_n(t)$ will be exactly $s_n(231,r)$.

In addition to the variable t, we introduce $n(n+1)/2$ catalytic variables $x_{i,j}$ with $1 \le j \le i \le n$ and define the weight of a permutation $\pi = \pi_1\ldots\pi_n$ to be

$$\text{weight}(\pi) := t^{N_{231}(\pi)} \prod_{1\le j\le i\le n} x_{i,j}^{\#\{(a,b)\,:\,\pi_a>\pi_b,\ \pi_a=i,\ \pi_b<j,\ 1\le a<b\le n\}}$$

For example, $\text{weight}(12345) = 1$ and $\text{weight}(24153) = t^2x_{2,2}x_{4,2}x_{4,3}x_{4,4}^2x_{5,4}x_{5,5}$. In essence, these catalytic variables keep track of occurrences of the pattern 21 that begin with i and have a "gap" of at least $i-j$.

We will again define an analogous multi-variate polynomial P_n on all the previously defined variables. However, for notational convenience, the $x_{i,j}$ variables will be written

[5] As opposed to computing the equivalent pattern 132.

as a matrix of variables:

$$X_n := \begin{bmatrix} x_{1,1} & & \cdots & & x_{1,n} \\ & \ddots & & & \\ \vdots & & x_{i,i} & & \vdots \\ & & & \ddots & \\ x_{n,1} & & \cdots & & x_{n,n} \end{bmatrix} \tag{3.3}$$

where we will disregard the entries above the diagonal (i.e., the $x_{i,j}$ entries where $j > i$).

For each n, we now define the polynomial

$$P_n(t; X_n) := \sum_{\pi \in \mathcal{S}_n} \text{weight}(\pi).$$

Recall that $P_n(t; \mathbf{1}) = f_n(t)$, where $\mathbf{1}$ is the matrix of all 1's.

We will derive a functional equation for this P_n function, but first, we derive the following lemma:

Lemma 7. *Let $\pi = \pi_1 \ldots \pi_n$ and suppose that $\pi_1 = i$. If $\pi' := red(\pi_2 \ldots \pi_n)$, then*

$$weight(\pi) = x_{i,1}^0 x_{i,2}^1 \ldots x_{i,i}^{i-1} \cdot weight(\pi')|_A \quad ,$$

where A is the set of substitutions given by

$$A := \begin{cases} x_{b,c} \to x_{b+1,c} & b \geq i, c < i \\ x_{b,c} \to x_{b+1,c+1} & b \geq i, c > i \\ x_{b,c} \to t x_{b+1,c} \cdot x_{b+1,c+1} & b \geq i, c = i \end{cases} .$$

Proof. Observe that $N_{231}(\pi)$ is equal to the number of occurrences of 231 in $\pi_2 \ldots \pi_n$ **plus** the number of occurrences of 21 in $\pi_2 \ldots \pi_n$, where the term corresponding to the "2" is greater than i and the term corresponding to the "1" is less than i. We make the following two observations. First, in weight(π), the exponents of $x_{k,i}$ and $x_{k,i+1}$ are equal for each k (since $\pi_1 = i$). Second, the number of 231 patterns that include the first term $\pi_1 = i$ is the exponent of $x_{i+1,i}$ **plus** the exponent of $x_{i+2,i}$ **plus** $\ldots$ **plus** the exponent $x_{n,i}$.

If we re-insert i at the beginning of π', we would shift all the terms $i, i+1, \ldots, n-1$ up by 1. This (combined with the prior observations) would lead to the set of substitutions given by A. Note that there is no case for $b < i, c \geq i$ since the $x_{b,c}$ variables are only defined for $b \geq c$. Finally, the new "i" would create new 21 patterns and would require an extra factor of $x_{i,1}^0 x_{i,2}^1 \dots x_{i,i}^{i-1}$ for the weight. □

Now, define the operator R_1 on a square matrix X_n and $i < n$ to be:

$$R_1(X_n, i) := \begin{bmatrix} x_{1,1} & \cdots & x_{1,i-1} & tx_{1,i}x_{1,i+1} & x_{1,i+2} & \cdots & x_{1,n} \\ \vdots & \ddots & & \vdots & & & \vdots \\ x_{i-1,1} & & x_{i-1,i-1} & & \cdots & & x_{i-1,n} \\ x_{i+1,1} & \cdots & x_{i+1,i-1} & tx_{i+1,i}x_{i+1,i+1} & x_{i+1,i+2} & \cdots & x_{i+1,n} \\ \vdots & & \vdots & \vdots & \ddots & & \vdots \\ \vdots & & \vdots & \vdots & & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,i-1} & tx_{n,i}x_{n,i+1} & x_{n,i+2} & \cdots & x_{n,n} \end{bmatrix}. \tag{3.4}$$

In essence, the R_1 operator deletes the i-th row, merges the i-th and $(i+1)$-th columns via term-by-term multiplication, and multiplies this new column by a factor of t. If $i = n$, then $R_1(X_n, i)$ is defined to be the $(n-1) \times (n-1)$ matrix obtained by deleting the n-th row and n-th column from X_n.

The previous lemma now leads directly to the following:

Theorem 14. *For the pattern $\tau = 231$,*

$$P_n(t; X_n) = \sum_{i=1}^{n} x_{i,1}^0 x_{i,2}^1 \dots x_{i,i}^{i-1} \cdot P_{n-1}(t; R_1(X_n, i)). \tag{FE231}$$

Note that while all entries in the matrix are changed for consistency, we will continue to disregard the entries above the diagonal.

Again, our goal is to compute $P_n(t; \mathbf{1})$, and the analogous computational techniques from previous sections will also apply in this setting. For example, we can apply (FE231) directly to $P_n(t; \mathbf{1})$ as opposed to computing $P_n(t; X_n)$ symbolically and substituting $x_{i,j} = 1$ at the end. The following result, which is obvious from the definition of the operator R_1, provides a substantial simplification:

Lemma 8. *Let A be a square matrix where every row is identical (i.e., the i-th row and the j-th row are equal for every i, j). Then, $R_1(A, i)$ will also be a square matrix with identical rows.*

By Lemma 8, repeated applications of R_1 to the all ones matrix $\mathbf{1}$ will still result in a matrix with identical rows. Therefore, it is sufficient to keep track of only one row instead of the entire matrix. Also observe that repeated applications of R_1 to the matrix $\mathbf{1}$ will always result in a matrix whose entries are powers of t. Let $Q_n(t; c_1, \ldots, c_n)$ denote the polynomial $P_n(t; C)$, where C is the $n \times n$ matrix where every row is $[c_1, \ldots, c_n]$ and every c_i is a power of t. This leads to a functional equation analogous to (FE231):

$$Q_n(t; c_1, \ldots, c_n) = \sum_{i=1}^{n} c_1^0 c_2^1 \ldots c_i^{i-1} \cdot Q_{n-1}(t; c_1, \ldots, c_{i-1}, t c_i c_{i+1}, c_{i+2}, \ldots, c_n). \tag{FE231c}$$

Note that $Q_n(t; 1\,[n \text{ times}])$ is exactly our desired polynomial $P_n(t; \mathbf{1}) = f_n(t)$. However, this interpretation only forces us to deal with n catalytic variables (the c_i's) as opposed to $n(n+1)/2$ catalytic variables (the $x_{i,j}$'s). Just as in prior sections, we can repeatedly apply our functional equation (FE231c) to compute $Q_n(t; 1\,[n \text{ times}])$.

When the sequence $s_n(231, r)$ is desired for a fixed r, the obvious analog of Lemma 3 and the computational reduction using the CHOP_r operator can again be used. This has been implemented in the accompanying Maple package `F231`.[6]

3.4.2 Extending to 2341 and beyond

In this subsection, we outline how to extend the approach for 231 to an analogous (but more complicated) approach for 2341. In addition to the variable t, we now introduce $n(n+1)/2$ catalytic variables $x_{i,j}$ with $1 \le j \le i \le n$ and $n(n+1)/2$ more catalytic variables $y_{i,j}$ with $1 \le j \le i \le n$ (a total of $n(n+1)$ catalytic variables). Define the weight of a permutation $\pi = \pi_1 \ldots \pi_n$ to be

$$\mathrm{weight}(\pi) :=$$
$$t^{N_{2341}(\pi)} \prod_{1 \le j \le i \le n} x_{i,j}^{\#\{(a,b)\,:\,\pi_a > \pi_b,\ \pi_a = i,\ \pi_b < j,\ 1 \le a < b \le n\}} \cdot y_{i,j}^{\#\{(a,b,c)\,:\,\pi_c < \pi_a < \pi_b,\ \pi_a = i,\ \pi_c < j,\ 1 \le a < b < c \le n\}}$$

For example,

$$\mathrm{weight}(24351) = t^2 \cdot x_{2,2} x_{3,2} x_{3,3} x_{4,2} x_{4,3} x_{4,4}^2 x_{5,2} x_{5,3} x_{5,4} x_{5,5} \cdot y_{2,2}^3 y_{3,2} y_{3,3} y_{4,2} y_{4,3} y_{4,4}.$$

---

[6] Although all output would be equivalent to the 132 case, the approach here will be necessary when considering multiple patterns.

The $x_{i,j}$ variables and the $y_{i,j}$ variables will be written as matrices of variables:

$$X_n := \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ & \ddots & \\ \vdots & x_{i,i} & \vdots \\ & & \ddots \\ x_{n,1} & \cdots & x_{n,n} \end{bmatrix}, \quad Y_n := \begin{bmatrix} y_{1,1} & \cdots & y_{1,n} \\ & \ddots & \\ \vdots & y_{i,i} & \vdots \\ & & \ddots \\ y_{n,1} & \cdots & y_{n,n} \end{bmatrix} \tag{3.5}$$

where we will disregard the entries above the diagonal.

For each n, we define the polynomial

$$P_n(t; X_n, Y_n) := \sum_{\pi \in \mathcal{S}_n} \text{weight}(\pi)$$

and again $P_n(t; \mathbf{1}, \mathbf{1}) = f_n(t)$ is our desired polynomial. We now have the following result:

Lemma 9. *Let $\pi = \pi_1 \ldots \pi_n$ and suppose that $\pi_1 = i$. If $\pi' :=$ red$(\pi_2 \ldots \pi_n)$, then*

$$\textit{weight}(\pi) = x_{i,1}^0 x_{i,2}^1 \ldots x_{i,i}^{i-1} \cdot \textit{weight}(\pi')|_{A'} \quad ,$$

where A' is the set of substitutions given by

$$A' := \begin{cases} x_{b,c} \to y_{i,c} \cdot x_{b+1,c} & b \geq i, c < i \\ x_{b,c} \to x_{b+1,c+1} & b \geq i, c > i \\ x_{b,c} \to y_{i,i} \cdot x_{b+1,c} \cdot x_{b+1,c+1} & b \geq i, c = i \\ y_{b,c} \to y_{b+1,c} & b \geq i, c < i \\ y_{b,c} \to y_{b+1,c+1} & b \geq i, c > i \\ y_{b,c} \to t y_{b+1,c} \cdot y_{b+1,c+1} & b \geq i, c = i \end{cases} .$$

Proof. Observe that $N_{2341}(\pi)$ is equal to the number of occurrences of 2341 in $\pi_2 \ldots \pi_n$ **plus** the number of occurrences of 231 in $\pi_2 \ldots \pi_n$, where the term corresponding to the "2" is greater than i and the term corresponding to the "1" is less than i. We make the following few observations. First, in weight(π), the exponents of $x_{k,i}$ and $x_{k,i+1}$ are equal and the exponents of $y_{k,i}$ and $y_{k,i+1}$ are equal for each k (since $\pi_1 = i$). Second, the number of 2341 patterns that include the first term $\pi_1 = i$ is the exponent of $y_{i+1,i}$

plus the exponent of $y_{i+2,i}$ **plus** ... **plus** the exponent $y_{n,i}$. Third, the number of 231 patterns that include the first term $\pi_1 = i$ (i.e., the "2" is equal to i) and whose "1" term is less than k is equal to the exponent of $x_{i+1,k}$ **plus** the exponent of $x_{i+2,k}$ **plus** ... **plus** the exponent of $x_{n,k}$.

If we re-insert i at the beginning of π', we would shift all the terms $i, i+1, \ldots, n-1$ up by 1. This (combined with the prior observations) would lead to the set of substitutions given by A'. Note that there is no case for $b < i, c \geq i$ since the $x_{b,c}$ variables are only defined for $b \geq c$. Finally, the new "i" would create new 21 patterns and would require an extra factor of $x_{i,1}^0 x_{i,2}^1 \ldots x_{i,i}^{i-1}$ for the weight. $\square$

In addition to the previous R_1 operator defined in Eq. 3.4, we define another operator R_2 on two square matrices X_n and Y_n (of equal dimension) and $i < n$ to be:

$$R_2(X_n, Y_n, i) := \begin{bmatrix} x_{1,1} & \cdots & x_{1,i-1} & y_{i,i}x_{1,i}x_{1,i+1} & x_{1,i+2} & \cdots & x_{1,n} \\ \vdots & \ddots & & \vdots & & & \vdots \\ x_{i-1,1} & & x_{i-1,i-1} & & \cdots & & x_{i-1,n} \\ y_{i,1}x_{i+1,1} & \cdots & y_{i,i-1}x_{i+1,i-1} & y_{i,i}x_{i+1,i}x_{i+1,i+1} & x_{i+1,i+2} & \cdots & x_{i+1,n} \\ \vdots & & \vdots & \vdots & \ddots & & \vdots \\ \vdots & & \vdots & \vdots & & \ddots & \vdots \\ y_{i,1}x_{n,1} & \cdots & y_{i,i-1}x_{n,i-1} & y_{i,i}x_{n,i}x_{n,i+1} & x_{n,i+2} & \cdots & x_{n,n} \end{bmatrix}. \tag{3.6}$$

If $i = n$, then $R_2(X_n, Y_n, i)$ is defined to be the $(n-1) \times (n-1)$ matrix obtained by deleting the n-th row and n-th column from X_n.

The previous lemma now leads to the following:

Theorem 15. *For the pattern $\tau = 2341$,*

$$P_n(t; X_n, Y_n) = \sum_{i=1}^{n} x_{i,1}^0 x_{i,2}^1 \ldots x_{i,i}^{i-1} \cdot P_{n-1}(t; R_2(X_n, Y_n, i), R_1(Y_n, i)). \tag{FE2341}$$

As in prior sections, we recursively apply the functional equation directly to $P_n(t; \mathbf{1}, \mathbf{1})$ (and subsequent instances of P_k). Observe that in this scenario, Lemma 8 still applies for the R_1 operator and more specifically the "Y_n" matrix in P_n. While the lemma does not apply to the R_2 operator, this still allows us to reduce the number of catalytic variables. Let $Q_n(t;C;d_1,\ldots,d_n)$ denote the polynomial $P_n(t;C,D)$ where every entry of the $n\times n$ matrices C and D are powers of t and every row in D is $[d_1,\ldots,d_n]$. We derive an analogous functional equation:

$$Q_n(t;C;d_1,\ldots,d_n) =$$

$$\sum_{i=1}^{n} c_{i,1}^{0}c_{i,2}^{1}\ldots c_{i,i}^{i-1}\cdot Q_{n-1}(t;R_2(C,D,i);d_1,\ldots,d_{i-1},td_id_{i+1},d_{i+2},\ldots,d_n). \quad \text{(FE2341c)}$$

Using this recurrence to compute $Q_n(t;\mathbf{1};1\,[n \text{ times}])$ will yield the desired polynomial $f_n(t)$. This approach allows us to deal with $n(n+1)/2+n$ catalytic variables (as opposed to $n(n+1)$ such variables).

Additionally, for a fixed r, the sequence $s_n(2341,r)$ can be computed by applying Lemma 3 and the CHOP_r operator as necessary. This has been implemented in the procedure `F2341rN(r,N)` (in the Maple package `F2341`).

For example, the Maple call `F2341rN(1,15);` for the first 15 terms of $s_n(2341,1)$ produces the sequence:

$$0,0,0,1,11,87,625,4378,30671,216883,1552588,$$

$$11257405,82635707,613600423,4604595573$$

and the Maple call `F2341rN(2,15);` for the first 15 terms of $s_n(2341,2)$ produces the sequence:

$$0,0,0,0,5,68,626,5038,38541,289785,2172387,$$

$$16339840,123650958,942437531,7236542705$$

While we do not present the details here, the same methodology can be applied to longer patterns of the form $23\ldots k1$. Analogous functional equations can be derived and used for enumeration.

3.5 The pattern 1324

In this section, we extend the techniques developed for the pattern 2341 to the pattern 1324. The pattern 1324 has been notoriously difficult to study, even for the pattern avoidance case. The approach will resemble that of the previous section (for the pattern 2341). In addition, we will show how to extract a more efficient enumeration algorithm specifically for the $r = 0$ case.

3.5.1 A functional equations approach to 1324

We again consider the variable t as well as $n(n+1)/2$ catalytic variables $x_{i,j}$ with $1 \le i \le j \le n$ and $n(n+1)/2$ catalytic variables $y_{i,j}$ with $1 \le j \le i \le n$. Observe that the subscripts of the two sets of catalytic variables range over different quantities. Define the weight of a permutation $\pi = \pi_1 \dots \pi_n$ to be

$$\text{weight}(\pi) :=$$
$$t^{N_{1324}(\pi)} \prod_{1\le i\le j\le n} x_{i,j}^{\#\{(a,b)\,:\,\pi_a<\pi_b,\ \pi_a=i,\ \pi_b>j\}} \cdot \prod_{1\le j\le i\le n} y_{i,j}^{\#\{(a,b,c)\,:\,\pi_b<\pi_a<\pi_c,\ \pi_a=i,\ \pi_b\ge j\}}$$

where it is always assumed that $1 \le a < b < c \le n$. For example,

$$\text{weight}(41325) = t \cdot x_{1,1}^3 x_{1,2}^2 x_{1,3} x_{1,4} x_{2,2} x_{2,3} x_{2,4} x_{3,3} x_{3,4} x_{4,4} \cdot y_{3,1} y_{3,2} y_{4,1}^3 y_{4,2}^2 y_{4,3}.$$

The $x_{i,j}$ variables and the $y_{i,j}$ variables will be written as matrices of variables:

$$X_n := \begin{bmatrix} x_{1,1} & \cdots & & x_{1,n} \\ & \ddots & & \\ \vdots & & x_{i,i} & \vdots \\ & & \ddots & \\ x_{n,1} & \cdots & & x_{n,n} \end{bmatrix}, \quad Y_n := \begin{bmatrix} y_{1,1} & \cdots & & y_{1,n} \\ & \ddots & & \\ \vdots & & y_{i,i} & \vdots \\ & & \ddots & \\ y_{n,1} & \cdots & & y_{n,n} \end{bmatrix} \tag{3.7}$$

where we will disregard the entries below the diagonal in X_n and the entries above the diagonal in Y_n.

For each n, we define the polynomial

$$P_n(t; X_n, Y_n) := \sum_{\pi\in\mathcal{S}_n} \text{weight}(\pi)$$

and again $P_n(t; \mathbf{1}, \mathbf{1}) = f_n(t)$ is our desired polynomial. We now have the following result:

Lemma 10. *Let $\pi = \pi_1 \ldots \pi_n$ and suppose that $\pi_1 = i$. If $\pi' := red(\pi_2 \ldots \pi_n)$, then*

$$weight(\pi) = x_{i,i}^{n-i} x_{i,i+1}^{n-i-1} \ldots x_{i,n-1}^{1} \cdot weight(\pi')|_{A''} \quad ,$$

where A'' is the set of substitutions given by

$$A'' := \begin{cases} x_{b,c} \to x_{b,c+1} & b < i, c \geq i \\ x_{b,c} \to x_{b+1,c+1} & b \geq i, c \geq i \\ x_{b,c} \to y_{i,1} y_{i,2} \ldots y_{i,b} \cdot x_{b+1,c} \cdot x_{b+1,c+1} & b < i, c = i-1 \\ y_{b,c} \to y_{b+1,c} & b \geq i, c < i \\ y_{b,c} \to y_{b+1,c+1} & b \geq i, c > i \\ y_{b,c} \to t y_{b+1,c} \cdot y_{b+1,c+1} & b \geq i, c = i \quad . \end{cases}$$

Proof. Observe that $N_{1324}(\pi)$ is equal to the number of occurrences of 1324 in $\pi_2 \ldots \pi_n$ **plus** the number of occurrences of 213 in $\pi_2 \ldots \pi_n$, where the term corresponding to the "1" is greater than i. We make the following observations.

First, in weight(π), the exponents of $x_{k,i-1}$ and $x_{k,i}$ are equal and the exponents of $y_{k,i}$ and $y_{k,i+1}$ are equal for each k (since $\pi_1 = i$). Second, the number of 2341 patterns that include the first term $\pi_1 = i$ is the exponent of $y_{i+1,i+1}$ **plus** the exponent of $y_{i+2,i+1}$ **plus** ... **plus** the exponent $y_{n,i+1}$. Third, the number of 213 patterns that include the first term $\pi_1 = i$ (i.e., the "2" is equal to i) and whose "1" term is at least k is equal to the exponent of $x_{k,i}$ **plus** the exponent of $x_{k+1,i}$ **plus** ... **plus** the exponent of $x_{i-1,i}$.

If we re-insert i at the beginning of π', we would shift all the terms $i, i+1, \ldots, n-1$ up by 1. This (combined with the prior observations) would lead to the set of substitutions given by A''. Finally, the new "i" would create new 12 patterns and would require an extra factor of $x_{i,i}^{n-i} x_{i,i+1}^{n-i-1} \ldots x_{i,n-1}^{1}$ for the weight. $\square$

We will again use the R_1 operator defined in Eq. 3.4. In addition, we define another operator R_3 on two $n \times n$ square matrices X_n and Y_n and $1 < i \leq n$ to be:

$$R_3(X_n, Y_n, i) := \begin{bmatrix} x_{1,1} & \cdots & x_{1,i-2} & w_1 & x_{1,i+1} & \cdots & x_{1,n} \\ \vdots & \ddots & & \vdots & & & \vdots \\ x_{i-2,1} & \cdots & x_{i-2,i-2} & w_{i-2} & x_{i-2,i+1} & \cdots & x_{i-2,n} \\ x_{i-1,1} & \cdots & x_{i-1,i-2} & w_{i-1} & x_{i-1,i+1} & \cdots & x_{i-1,n} \\ x_{i+1,1} & \cdots & x_{i+1,i-2} & w_{i+1} & x_{i+1,i+1} & \cdots & x_{i+1,n} \\ \vdots & & \vdots & \vdots & & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,i-2} & w_n & x_{n,i+1} & \cdots & x_{n,n} \end{bmatrix} \tag{3.8}$$

where

$$w_k := \begin{cases} y_{i,1}y_{i,2}\ldots y_{i,k} \cdot x_{k,i-1} \cdot x_{k,i} & k \leq i-1 \\ 0 & k > i-1 \end{cases} .$$

If $i = 1$, then $R_3(X_n, Y_n, i)$ is defined to be the $(n-1) \times (n-1)$ matrix obtained by deleting the 1-st row and 1-st column from X_n. In essence, the R_3 operator deletes the i-th row, merges the $(i-1)$-th column with the i-th column (via term-by-term multiplication), and scales that new column by products of terms from Y_n.

The previous lemma now leads to the following:

Theorem 16. *For the pattern $\tau = 1324$,*

$$P_n(t; X_n, Y_n) = \sum_{i=1}^{n} x_{i,i}^{n-i} x_{i,i+1}^{n-i-1} \ldots x_{i,n-1}^{1} \cdot P_{n-1}(t;\ R_3(X_n, Y_n, i),\ R_1(Y_n, i)). \tag{FE1324}$$

As in prior sections, we recursively apply the functional equation directly to $P_n(t; \mathbf{1}, \mathbf{1})$ (and subsequent instances of P_k). Again, Lemma 8 still applies for the R_1 operator and more specifically the Y_n matrix in P_n. While the lemma does not apply to the R_3 operator, this still allows us to reduce the number of catalytic variables. Let $Q_n(t;\ C;\ d_1, \ldots, d_n)$ denote the polynomial $P_n(t; C, D)$ where every entry of the $n \times n$ matrices C and D are powers of t and every row in D is $[d_1, \ldots, d_n]$. We derive an analogous functional equation:

$$Q_n(t;\ C;\ d_1,\ldots,d_n) =$$
$$\sum_{i=1}^{n} c_{i,i}^{n-i} c_{i,i+1}^{n-i-1} \ldots c_{i,n-1}^{1} \cdot Q_{n-1}(t;\ R_3(C,D,i);\ d_1,\ldots,d_{i-1}, td_i d_{i+1}, d_{i+2},\ldots,d_n). \tag{FE1324c}$$

Using this recurrence to compute $Q_n(t;\mathbf{1};1\,[n \text{ times}])$ will yield the desired polynomial $f_n(t)$. This approach allows us to deal with $n(n+1)/2+n$ catalytic variables (as opposed to $n(n+1)$ such variables).

Additionally, for a fixed r, the sequence $s_n(1324,r)$ can be computed by applying Lemma 3 and the CHOP_r operator as necessary. This has been implemented in the procedure `F1324rN(r,N)` (in the Maple package `F1324`).

For example, the Maple call `F1324rN(0,17);` for the first 17 terms of $s_n(1324,0)$ produces the sequence:

$$1, 2, 6, 23, 103, 513, 2762, 15793, 94776, 591950, 3824112, 25431452,$$
$$173453058, 1209639642, 8604450011, 62300851632, 458374397312$$

and the Maple call `F1324rN(1,15);` for the first 15 terms of $s_n(1324,1)$ produces the sequence:

$$0, 0, 0, 1, 10, 75, 522, 3579, 24670, 172198, 1219974, 8776255, 64082132, 474605417, 3562460562.$$

3.5.2 Specializing the approach to $r = 0$

Unfortunately, the previous algorithm developed for the pattern 1324 is very memory intensive. In this subsection, we outline how to extract a simpler recurrence specifically for the $r = 0$ case from the previous functional equations method. The resulting algorithm is still quite memory intensive but allows us to compute the first 23 terms of the sequence (before memory runs out).

We will specialize for the $r = 0$ case beginning at functional equation (FE1324c). Recall that $Q_n(t;C;d_1,\ldots,d_n)$ is the polynomial $P_n(t;C,D)$ where every entry of the $n\times n$ matrices C and D are powers of t and every row in D is $[d_1,\ldots,d_n]$. We had the functional equation

$$Q_n(t;\ C;\ d_1,\ldots,d_n) =$$
$$\sum_{i=1}^{n} c_{i,i}^{n-i} c_{i,i+1}^{n-i-1} \ldots c_{i,n-1}^{1} \cdot Q_{n-1}(t;\ R_3(C,D,i);\ d_1,\ldots,d_{i-1}, td_i d_{i+1}, d_{i+2},\ldots,d_n)$$

and wanted to compute $Q_n(t;\ \mathbf{1};\ 1\,[n \text{ times}])$, which is exactly the desired polynomial $f_n(t)$.[7]

Observe that all the variables $c_{k,l}$ and d_k represent powers of t. Then, it is actually sufficient to keep track of powers of t through most of the algorithm. In particular, we may consider the analogous function $H_n(t;\ U;\ v_1,\ldots,v_n)$, where U is an $n \times n$ matrix of non-negative integers and each v_i is a non-negative integer. More precisely, $H_n(t; U; v_1,\ldots,v_n)$ is the polynomial $P_n(t; C, D)$, where C and D are $n \times n$ matrices, $c_{i,j} = t^{u_{i,j}}$ for every $1 \le i, j \le n$, and every row of D is $[t^{v_1},\ldots,t^{v_n}]$.

In addition, we define the analogous operator R_3' on an $n \times n$ square matrix U_n (of non-negative integers), a length n vector of non-negative integers $[v_1,\ldots,v_n]$, and $1 < i \le n$:

$$R_3'(U_n, [v_1,\ldots,v_n], i) := \begin{bmatrix} u_{1,1} & \cdots & u_{1,i-2} & w_1' & u_{1,i+1} & \cdots & u_{1,n} \\ \vdots & \ddots & & \vdots & & & \vdots \\ u_{i-2,1} & \cdots & u_{i-2,i-2} & w_{i-2}' & u_{i-2,i+1} & \cdots & u_{i-2,n} \\ u_{i-1,1} & \cdots & u_{i-1,i-2} & w_{i-1}' & u_{i-1,i+1} & \cdots & u_{i-1,n} \\ u_{i+1,1} & \cdots & u_{i+1,i-2} & w_{i+1}' & u_{i+1,i+1} & \cdots & u_{i+1,n} \\ \vdots & & \vdots & \vdots & & \ddots & \vdots \\ u_{n,1} & \cdots & u_{n,i-2} & w_n' & u_{n,i+1} & \cdots & u_{n,n} \end{bmatrix} \tag{3.9}$$

where

$$w_k' := \begin{cases} (v_1 + v_2 + \ldots + v_k) + u_{k,i-1} + u_{k,i} & k \le i-1 \\ 0 & k > i-1 \end{cases} \quad . \tag{3.10}$$

If $i = 1$, then $R_3'(U_n, [v_1,\ldots,v_n], i)$ is defined to be the $(n-1)\times(n-1)$ matrix obtained by deleting the 1-st row and 1-st column from U_n. In essence, the R_3' operator deletes

[7] Recall that $\mathbf{1}$ is the $n \times n$ matrix where every entry is 1.

the i-th row, merges the $(i-1)$-th column with the i-th column (via term-by-term addition), and adds partial sums of $[v_1, \ldots, v_n]$ into the new column.

We now have the functional equation (analogous to Eq. (FE1324c)):

$$H_n(t;\ U;\ v_1, \ldots, v_n) =$$
$$\sum_{i=1}^{n} t^{e_i} \cdot H_{n-1}(t;\ R_3'(U, [v_1, \ldots, v_n], i);\ v_1, \ldots, v_{i-1}, v_i + v_{i+1} + 1, v_{i+2}, \ldots, v_n) \tag{FE1324e}$$

where $e_i = (n-i)u_{i,i} + (n-i-1)u_{i,i+1} + \ldots + (1)u_{i,n-1}$. Observe that $H_n(t;\ \mathbf{0};\ 0\,[n \text{ times}])$ is now our desired polynomial $f_n(t)$.[8]

Since we are specifically considering the $r = 0$ case, we can make additional observations and simplifications. First, we are only interested in the constant term of $f_n(t)$. By Lemma 3, we only need to keep track of polynomials of the form $a_0 + a_1 t$ in intermediate computations. Because of this, we may consider all matrices and vectors used in H_n to be 0-1 matrices. After every addition (for example, in the w_k' term in R_3'), we can take the minimum of the resulting sum and 1.

From this, we can make additional observations on $v_1, \ldots, v_n$. These quantities are only utilized directly in the R_3' operator, and in particular, they only appear in the partial sums for w_k' in Eq. 3.10. Suppose that some of the $v_1, \ldots, v_n$ are equal to 1, and let j be the smallest number such that $v_j = 1$. Then,

$$H_n(t;\ U;\ v_1, \ldots, v_n) = H_n(t;\ U;\ 0\,[j-1 \text{ times}], 1\,[n-j+1 \text{ times}]) \quad .$$

In particular, the variables $v_1, \ldots, v_n$ are unnecessary, and it is sufficient to keep track of how many 0's there are. We can then consider this slightly simpler function

$$\widetilde{H}_n(t;\ U;\ k) := H_n(t;\ U;\ 0\,[k \text{ times}], 1\,[n-k \text{ times}])$$

where $0 \leq k \leq n$.

Finally, if we apply the CHOP_r operator to functional equation FE1324e, it would eliminate every term where the exponent $e_i > 0$. This would happen if $u_{i,i} > 0$ **or**

[8] We denote the $n \times n$ matrix consisting of all zeros by $\mathbf{0}$.

$u_{i,i+1} > 0$ **or** ... **or** $u_{i,n-1} > 0$. This observation (combined with how the R_3' operator "modifies" the matrix U_n) implies that we only need to keep track of the left-most 1 within each row of U_n. If there are multiple 1's on a row, the left-most 1 is sufficient to force $e_i > 0$ as long as it is not in the n-th column. Therefore, we can consider a function of the form

$$H_n^0(t;\ b_1, \ldots, b_n;\ k) := \widetilde{H}_n(t;\ B_n;\ k) = H_n(t;\ B_n;\ 0\,[k \text{ times}]\,, 1\,[n-k \text{ times}])$$

where $0 \le k \le n$ and $1 \le b_j \le n{+}1$ for each j and B_n is the $n \times n$ matrix where the j-th row is $[\ 0\,[n \text{ times}]\ ]$ if $b_j = n{+}1$ and otherwise is $[\ 0\,[b_j - 1 \text{ times}]\,,\ 1\,[n - b_j + 1 \text{ times}]\ ]$.

This simplified approach has been implemented in the procedure `Av1324F(n)` in the accompanying Maple package `F1324`. This approach is still quite memory intensive, but even then, we were able to compute the first 23 terms of the enumerating sequence $s_n(1324, 0)$.[9] The Maple call `Av1324N(23)` gives us the sequence:

$$1, 2, 6, 23, 103, 513, 2762, 15793, 94776, 591950, 3824112, 25431452, 173453058,$$
$$1209639642, 8604450011, 62300851632, 458374397312, 3421888118907, 25887131596018,$$
$$198244731603623, 1535346218316422, 12015325816028313, 94944352095728825$$

which is the sequence A061552 in the On-Line Encyclopedia of Integer Sequences [54].

This approach seems to differ from the existing enumeration methods for 1324-avoiding permutations. The approach by Albert et al. [1] can generate the first 25 terms of the sequence and was also used to find the best lower bound on the Stanley-Wilf limit. Recall that the Stanley-Wilf limit is

$$L(1324) := \lim_{n\to\infty} \sqrt[n]{s_n(1324, 0)}.$$

The best lower bound [1] and upper bound [7] for the limit are currently

$$9.47 < L(1324) < 13.93.$$

Our hope is that this new enumeration algorithm may be analyzed to produce an improved bound for $L(1324)$.

---

[9] The limitation came from running out of memory and not from a lengthy computational time. The first 23 terms took under 1 hour with the computing resources available to the author.

Chapter 4

Extensions for the Functional Equation Methodology

4.1 Overview

The study of permutations with r occurrences of a given pattern can be extended to consider multiple patterns simultaneously. Let $p_1, \ldots, p_k$ be k distinct (permutation) patterns, and let $r_1, \ldots, r_k$ be k non-negative integers. We define the set of length n permutations with exactly r_i occurrences of pattern p_i for every i:

$$\mathcal{S}_n([p_1, \ldots, p_k], [r_1, \ldots, r_k]) := \bigcap_{i=1}^{k} \mathcal{S}_n(p_i, r_i).$$

We also denote the cardinality of this set by

$$s_n([p_1, \ldots, p_k], [r_1, \ldots, r_k]) := |\mathcal{S}_n([p_1, \ldots, p_k], [r_1, \ldots, r_k])|.$$

Some work has already been done in the avoidance case for this problem ($r_i = 0$ for each i). In this case, most work has been restricted to studying pairs of patterns ($k = 2$) where the patterns are either of length 3 or 4. The sequences enumerating permutations avoiding pairs of patterns (of length 3 or 4) can also be found on the On-Line Encyclopedia of Integer Sequences [54] with many helpful references.

For cases where one (or more) of the r_i are greater than 0, much less is known. Most of the work has been restricted specifically to length 3 patterns. For example, the pattern class $\mathcal{S}_n([123, 132], [r_1, r_2])$ has received some attention, and there are explicit formulas for $s_n([123, 132], [r_1, r_2])$ and small r_i. It is not difficult to show that $s_n([123, 132], [0, 0]) = 2^{n-1}$. Robertson also shows in [50]:

Theorem 17.

$$s_n([123, 132], [0, 1]) = s_n([123, 132], [1, 0]) = (n-2)2^{n-3}$$

and

$$s_n([123, 132], [1, 1]) = (n-3)(n-4)2^{n-5}.$$

However, there is no concrete result for enumerating this sequence for general r_1 and r_2.

Another extension of the classical avoidance problem is to refine a permutation class over some permutation statistic. One of the most commonly studied statistic is the number of inversions in a permutation, which is one way to quantify how "unsorted" a permutation is. Recall that the inversion number of a permutation $\pi = \pi_1 \dots \pi_n$, denoted by $\text{inv}(\pi)$, is the number of pairs (i, j) such that $1 \leq i < j \leq n$ and $\pi_i > \pi_j$.

In this chapter, we show two extensions of the functional equations approach developed in the last chapter. First, we show how to extend any of the previous functional equations to also keep track of inversions. These new algorithms have been implemented in accompanying Maple packages `qINCR`, `qINCRT`, `qF231`, `qF2341`, `qF1324`, and `qAv1324`. We then show how to extend the functional equations approach to consider multiple patterns simultaneously. Any collection of the patterns in the previous chapter can be combined, however, we will only present a few constructive examples. In particular, we will show how to compute $s_n([12\dots k, 12\dots(k-2)(k)(k-1)], [r_1, r_2])$ as well as $s_n(L_3, [r_1, \dots, r_6])$, where L_3 is the list of all length 3 permutations in lexicographical order.

4.2 Refining by inversions

In this section, we show how to modify the functional equations approach to account for the inversion number and present some examples of its implementation. First, we define some additional notation. Given a pattern τ and for each n, we define the bivariate polynomial

$$g_n(t, q) := \sum_{\pi \in \mathcal{S}_n} q^{\text{inv}(\pi)} t^{N_\tau(\pi)}.$$

Observe that $g_n(t, 1)$ is exactly $f_n(t)$ from the previous chapter.

Given a permutation $\pi = \pi_1 \ldots \pi_n$, suppose that $\pi_1 = i$. Then, $\mathrm{inv}(\pi)$ is equal to the number of inversions in $\pi_2 \ldots \pi_n$ **plus** the number of elements in $\pi_2, \ldots, \pi_n$ that are less than i. For any previously defined functional equation, it is enough to insert a factor of q^{i-1} in the summation.

Given a fixed pattern τ, the polynomial P_n can now be "generalized" as

$$P_n(t, q; x_1, \ldots, x_n) := \sum_{\pi \in \mathcal{S}_n} q^{\mathrm{inv}(\pi)} \cdot \mathrm{weight}(\pi). \tag{4.1}$$

Observe that $P_n(t, 1; x_1, \ldots, x_n)$ is exactly the polynomial $P_n(t; x_1, \ldots, x_n)$ from the previous chapter.

We can now quickly derive the modified functional equations for each pattern. The functional equation (NZFE1) now extends to:

Corollary 5. *For the pattern $\tau = 123$,*

$$P_n(t, q; x_1, \ldots x_n) = \sum_{i=1}^{n} q^{i-1} x_i^{n-i} P_{n-1}(t, q; x_1, \ldots, x_{i-1}, tx_{i+1}, \ldots, tx_n). \tag{qNZFE1}$$

Similarly, the functional equation (NZFE2) now extends to:

Corollary 6. *For the pattern $\tau = 1234$,*

$$P_n(t, q; x_1, \ldots x_n) =$$

$$\sum_{i=1}^{n} q^{i-1} x_i^{n-i} P_{n-1}(t, q;\ x_1, \ldots, x_{i-1}, y_i x_{i+1}, \ldots, y_i x_n;\ y_1, \ldots, y_{i-1}, t y_{i+1}, \ldots, t y_n). \tag{qNZFE2}$$

This has been implemented in the accompanying Maple package `qFINCR`. For example, the Maple call `qF123r(8,0,t,q);` would refine the quantity $s_8(123, 0)$ by inversions and produce the output:

$$q^{28} + 7\,q^{27} + 27\,q^{26} + 70\,q^{25} + 134\,q^{24} + 196\,q^{23} + 227\,q^{22} + 215\,q^{21} + 179\,q^{20}$$
$$+139\,q^{19} + 99\,q^{18} + 64\,q^{17} + 38\,q^{16} + 20\,q^{15} + 9\,q^{14} + 4\,q^{13} + q^{12}$$

and the Maple call `coeff(qF1234r(9,2,t,q),t,2);` would refine the quantity $s_9(1234, 2)$ by inversions and produce the output:

$$20\,q^{27} + 112\,q^{26} + 408\,q^{25} + 1087\,q^{24} + 2351\,q^{23} + 4176\,q^{22} + 6258\,q^{21} + 7750\,q^{20} + 8060\,q^{19}$$
$$+7003\,q^{18} + 5444\,q^{17} + 3608\,q^{16} + 2092\,q^{15} + 915\,q^{14} + 384\,q^{13} + 68\,q^{12} + 12\,q^{11}.$$

The functional equation (FE132) extends to:

Corollary 7. *For the pattern $\tau = 132$,*

$$P_n(t,q;x_1,\ldots,x_n) = \sum_{i=1}^{n} q^{i-1} x_1 x_2 \ldots x_{i-1} \cdot P_{n-1}(t,q;x_1,\ldots,x_{i-1},tx_{i+1},\ldots,tx_n). \tag{qFE132}$$

Also, functional equation (FE1243) extends to:

Corollary 8. *For the pattern $\tau = 1243$,*

$$P_n(t,q;x_1,\ldots,x_n,y_1,\ldots,y_n) =$$
$$\sum_{i=1}^{n} q^{i-1} x_1 x_2 \ldots x_{i-1} \cdot P_{n-1}(t,q;\ x_1,\ldots,x_{i-1},y_i x_{i+1},\ldots,y_i x_n;\ y_1,\ldots,y_{i-1},ty_{i+1},\ldots,ty_n). \tag{qFE1243}$$

This has also been implemented in the accompanying Maple package `qFINCRT`. For example, the Maple call `qF132r(7,0,t,q);` would refine the quantity $s_7(132,0)$ by inversions and produce the output:

$$q^{21} + 6\,q^{20} + 15\,q^{19} + 25\,q^{18} + 35\,q^{17} + 40\,q^{16} + 43\,q^{15} + 44\,q^{14} + 40\,q^{13} + 37\,q^{12}$$
$$+32\,q^{11} + 28\,q^{10} + 22\,q^{9} + 18\,q^{8} + 13\,q^{7} + 11\,q^{6} + 7\,q^{5} + 5\,q^{4} + 3\,q^{3} + 2\,q^{2} + q + 1$$

and the Maple call `coeff(qF1243r(8,2,t,q),t,2);` would refine the quantity $s_8(1243,2)$ by inversions and produce the output:

$$16\,q^{20} + 75\,q^{19} + 216\,q^{18} + 436\,q^{17} + 649\,q^{16} + 733\,q^{15} + 683\,q^{14} + 546\,q^{13} + 412\,q^{12}$$
$$+301\,q^{11} + 209\,q^{10} + 139\,q^{9} + 86\,q^{8} + 43\,q^{7} + 25\,q^{6} + 19\,q^{5} + 8\,q^{4} + 4\,q^{2}.$$

Let X_n and Y_n be the matrices of catalytic variables and R_1, R_2, and R_3 be the matrix operators defined in the previous chapter. We can then similarly extend the functional equations for 231, 2341, and 1324. The functional equations (FE231) and (FE231c) become:

Corollary 9. *For the pattern $\tau = 231$,*

$$P_n(t,q;X_n) = \sum_{i=1}^{n} q^{i-1} x_{i,1}^{0} x_{i,2}^{1} \ldots x_{i,i}^{i-1} \cdot P_{n-1}(t,q;R_1(X_n,i)) \tag{qFE231}$$

and

$$Q_n(t,q;c_1,\dots,c_n) = \sum_{i=1}^{n} q^{i-1} c_1^0 c_2^1 \dots c_i^{i-1} \cdot Q_{n-1}(t,q;c_1,\dots,c_{i-1},tc_ic_{i+1},c_{i+2},\dots,c_n).$$

(qFE231c)

Similarly, the functional equations (FE2341) and (FE2341c) become:

Corollary 10. *For the pattern $\tau = 2341$,*

$$P_n(t,q;X_n,Y_n) = \sum_{i=1}^{n} q^{i-1} x_{i,1}^0 x_{i,2}^1 \dots x_{i,i}^{i-1} \cdot P_{n-1}(t,q;R_2(X_n,Y_n,i),R_1(Y_n,i))$$

(qFE2341)

and

$$Q_n(t,q;C,d_1,\dots,d_n) =$$

$$\sum_{i=1}^{n} q^{i-1} c_{i,1}^0 c_{i,2}^1 \dots c_{i,i}^{i-1} \cdot Q_{n-1}(t,q;R_2(C,D,i),d_1,\dots,d_{i-1},td_id_{i+1},d_{i+2},\dots,d_n).$$

(qFE2341c)

And finally, the functional equations (FE1324) and (FE1324c) become:

Corollary 11. *For the pattern $\tau = 1324$,*

$$P_n(t,q;X_n,Y_n) = \sum_{i=1}^{n} q^{i-1} x_{i,i}^{n-i} x_{i,i+1}^{n-i-1} \dots x_{i,n-1}^1 \cdot P_{n-1}(t,q;\ R_3(X_n,Y_n,i),\ R_1(Y_n,i))$$

(qFE1324)

and

$$Q_n(t,q;C;d_1,\dots,d_n) =$$

$$\sum_{i=1}^{n} q^{i-1} c_{i,i}^{n-i} c_{i,i+1}^{n-i-1} \dots c_{i,n-1}^1 \cdot Q_{n-1}(t,q;\ R_3(C,D,i);\ d_1,\dots,d_{i-1},td_id_{i+1},d_{i+2},\dots,d_n).$$

(qFE1324c)

These extensions have been implemented in the accompanying Maple packages `qF231`, `qF2341`, `qF1324` and `qAv1324`.

Finally, we observe that we can also apply the same extension to patterns that are reversals of the ones we have considered so far (e.g., $k(k-1)\dots 21$, $(k-1)(k)(k-2)\dots 21$, etc.). Given a pattern $\sigma = \sigma_1 \dots \sigma_k$, we make the simple observation that for all n,

$$s_n([21,\sigma_1\dots\sigma_k],[r_1,r_2]) = s_n([12,\sigma_k\dots\sigma_1],[r_1,r_2]) \quad .$$

Therefore, instead of considering inversions in our desired pattern, we may simply keep track of the number of *non-inversions* (occurrences of the pattern 12) in the reversal of the pattern. This merely corresponds to adding a q^{n-i} factor into the functional equation summation (as opposed to q^{i-1} as before). For example, we have the following for deceasing patterns:

Corollary 12. *For the pattern* $\tau = 321$*,*

$$P_n(t, q; x_1, \ldots x_n) = \sum_{i=1}^{n} q^{n-i} x_i^{n-i} P_{n-1}(t, q; x_1, \ldots, x_{i-1}, tx_{i+1}, \ldots, tx_n). \tag{qFE321}$$

Corollary 13. *For the pattern* $\tau = 4321$*,*

$$P_n(t, q; x_1, \ldots x_n) =$$
$$\sum_{i=1}^{n} q^{n-i} x_i^{n-i} P_{n-1}(t, q;\ x_1, \ldots, x_{i-1}, y_i x_{i+1}, \ldots, y_i x_n;\ y_1, \ldots, y_{i-1}, ty_{i+1}, \ldots, ty_n). \tag{qFE4321}$$

The reversals of the other patterns can be handled similarly.

4.3 Extending to multiple patterns

In the previous section, we considered how to extend the functional equations to account for the inversion number. That was a very special case of the more general extension of tracking multiple patterns simultaneously. It is actually possible to keep track of any subset of the patterns we have considered before. However in this section, we will only present two specific examples (other combinations of patterns can be handled similarly).

4.3.1 Permutations containing 123 and 132

We first outline how to consider the patterns $\sigma = 123$ and $\tau = 132$ simultaneously. Let s and t be the variables corresponding to 123 and 132, respectively. Our ultimate goal will be to quickly compute coefficients of the bivariate polynomial

$$f_n(s, t) := \sum_{\pi \in \mathcal{S}_n} s^{N_{123}(\pi)} t^{N_{132}(\pi)}. \tag{4.2}$$

Let $x_1, \ldots, x_n$ and $u_1, \ldots, u_n$ be two sets of catalytic variables. We recall the two weight functions for the patterns 123 and 132 (using our new variables) to be:

$$\text{weight}_{123}(\pi) := s^{N_{123}(\pi)} \prod_{i=1}^{n} x_i^{\#\{(a,b)\,:\, \pi_a = i < \pi_b,\ 1 \le a < b \le n\}}$$

and

$$\text{weight}_{132}(\pi) := t^{N_{132}(\pi)} \prod_{i=1}^{n} u_i^{\#\{(a,b)\,:\, \pi_a > \pi_b = i,\ 1 \le a < b \le n\}}.$$

We now define the weight function for considering both patterns to be:

$$\text{weight}(\pi) := \text{weight}_{123}(\pi) \cdot \text{weight}_{132}(\pi).$$

For each n, we define the analogous multivariate polynomial

$$P_n(s, t;\ x_1, \ldots, x_n;\ u_1, \ldots, u_n) := \sum_{\pi \in \mathcal{S}_n} \text{weight}(\pi).$$

Observe that $P_n(s, t;\ 1\,[n \text{ times}]\,;\ 1\,[n \text{ times}])$ is exactly our desired polynomial $f_n(s, t)$. The proofs for Lemma 2 and Lemma 5 still hold, and the variables for 123 never depend on the variables for 132 (and vice versa). We immediately have the functional equation

$$P_n(s, t;\ x_1, \ldots, x_n;\ u_1, \ldots, u_n) =$$
$$\sum_{i=1}^{n} x_i^{n-i} u_1 u_2 \ldots u_{i-1} P_{n-1}(s, t;\ x_1, \ldots, x_{i-1}, s x_{i+1}, \ldots, s x_n;\ u_1, \ldots, u_{i-1}, t u_{i+1}, \ldots, t u_n).$$

The analogous computational techniques from the previous chapter also apply here.

This has been implemented in the Maple package `F123n132`. For example, the Maple call `F123r132sN(2,2,15);` gives the first 15 terms of the sequence enumerating permutations with 2 occurrences of 123 and 2 occurrences of 132:

$$0, 0, 0, 1, 6, 26, 94, 306, 934, 2732, 7752, 21488, 58432, 156288, 411904$$

and the Maple call `F123r132sN(4,2,15);` gives the first 15 terms of the sequence enumerating permutations with 4 occurrences of 123 and 2 occurrences of 132:

$$0, 0, 0, 0, 1, 5, 23, 106, 450, 1740, 6214, 20831, 66427, 203550, 603920$$

Based off of empirical evidence, we also believe the following to be true:

Conjecture 14. *For each fixed $r \geq 0$ and $s \geq 0$, the sequence enumerating length n permutations with exactly r occurrences of 123 and s occurrences of 132 is given by $p(n)2^n$, where $p(n)$ is some polynomial of degree $r + s$.*

As mentioned before, there are a number of results considering this type of problem, but most such results limit themselves to $s = 0, 1$.[1] If this general form were shown to hold for arbitrary r and s, the `F123n132` package could quickly compute enough terms to find explicit formulas and generating functions.

4.3.2 Permutations containing 1234 and 1243

We now outline the case of considering the patterns 1234 and 1243 simultaneously. Let s and t be the variables corresponding to 1234 and 1243, respectively. Let $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ be catalytic variables corresponding to 1234. Similarly, let $u_1, \ldots, u_n$ and $v_1, \ldots, v_n$ be catalytic variables corresponding to 1243. Recall that the weight functions for these patterns (using these variables) are

$$\text{weight}_{1234}(\pi) := s^{N_{1234}(\pi)} \prod_{i=1}^{n} x_i^{\#\{(a,b)\,:\,\pi_a = i < \pi_b,\ 1 \leq a < b \leq n\}} \cdot y_i^{\#\{(a,b,c)\,:\,\pi_a = i < \pi_b < \pi_c,\ 1 \leq a < b < c \leq n\}}$$

and

$$\text{weight}_{1243}(\pi) := t^{N_{1243}(\pi)} \prod_{i=1}^{n} u_i^{\#\{(a,b)\,:\,\pi_a > \pi_b = i,\ 1 \leq a < b \leq n\}} \cdot v_i^{\#\{(a,b,c)\,:\,\pi_a = i < \pi_c < \pi_b,\ 1 \leq a < b < c \leq n\}}.$$

The weight function for considering both patterns will be:

$$\text{weight}(\pi) := \text{weight}_{1234}(\pi) \cdot \text{weight}_{1243}(\pi).$$

For each n, the polynomial P_n is now defined to be:

$$P_n(s, t;\ x_1, \ldots, x_n;\ y_1, \ldots, y_n;\ u_1, \ldots, u_n;\ v_1, \ldots, v_n) := \sum_{\pi \in \mathcal{S}_n} \text{weight}(\pi).$$

Again, our goal is to compute $P_n(s, t;\ 1\,[n \text{ times}]\,;\ 1\,[n \text{ times}]\,;\ 1\,[n \text{ times}]\,;\ 1\,[n \text{ times}]) = f_n(s, t)$. In this case as well, the proofs from the corresponding lemmas (Lemma 4 and

[1] Permutations that are 132-avoiding have nice structural properties that make them easier to study.

Lemma 6) still hold, and we immediately have the functional equation

$$P_n(s,t;\ x_1,\dots,x_n;\ y_1,\dots,y_n;\ u_1,\dots,u_n;\ v_1,\dots,v_n) =$$
$$\sum_{i=1}^{n} x_i^{n-i} u_1 u_2 \dots u_{i-1} \cdot P_{n-1}(s,t;\ L_i^x;\ L_i^y;\ L_i^u;\ L_i^v)$$

where the sequences of variables L_i^x, L_i^y, L_i^u, and L_i^v are defined to be:

$$L_i^x := x_1\ ,\ \dots\ ,\ x_{i-1}\ ,\ y_i x_{i+1}\ ,\ \dots\ ,\ y_i x_n$$
$$L_i^y := y_1\ ,\ \dots\ ,\ y_{i-1}\ ,\ s y_{i+1}\ ,\ \dots\ ,\ s y_n$$
$$L_i^u := u_1\ ,\ \dots\ ,\ u_{i-1}\ ,\ v_i u_{i+1}\ ,\ \dots\ ,\ v_i u_n$$
$$L_i^v := v_1\ ,\ \dots\ ,\ v_{i-1}\ ,\ t v_{i+1}\ ,\ \dots\ ,\ t v_n \quad .$$

The analogous computational techniques from the previous chapter also apply here.

This has been implemented in the accompanying Maple package `F1234n1243`. For example, the Maple call `F1234r1243sN(15,0,0);` for the first 15 terms of $s_n([1234, 1243]\,, [0,0])$ produces the sequence:

$$1, 2, 6, 22, 90, 394, 1806, 8558, 41586, 206098, 1037718,$$
$$5293446, 27297738, 142078746, 745387038$$

which is the sequence A006318, the large Schröeder numbers, in the On-Line Encyclopedia of Integer Sequences [54].

Similarly, the Maple call `F1234r1243sN(15,1,0);` for the first 15 terms of $s_n([1234, 1243]\,, [1,0])$ produces the sequence:

$$0, 0, 0, 1, 10, 71, 444, 2617, 14958, 84063, 467960, 2591265, 14308722, 78911943, 435066228$$

Also, the Maple call `F1234r1243sN(15,1,1);` for the first 15 terms of $s_n([1234, 1243]\,, [1,1])$ produces the sequence:

$$0, 0, 0, 0, 0, 2, 32, 322, 2634, 19216, 130662, 848284, 5334332, 32788726, 198201268$$

4.3.3 Other extensions for multiple patterns

In the previous subsections, we outlined how to extend the functional equations methodology to consider the pair of patterns 123 and 132 simultaneously as well as the pair of patterns 1234 and 1243 simultaneously. Similar extensions can be done for any collection of patterns that were handled in the previous chapter.

It is also possible to consider more than two patterns simultaneously by extending the previous approaches in the obvious ways. We consider one final extension as an example. Let $p_1, \ldots, p_6$ be the length three permutations in lexicographical order (that is, $p_1 = 123$, $p_2 = 132$, and so on). Let $t_1, \ldots, t_6$ be variables where t_i is associated with the pattern p_i. Now, we are interested in the multivariate polynomial

$$f_n(t_1, t_2, t_3, t_4, t_5, t_6) := \sum_{\pi \in \mathcal{S}_n} t_1^{N_{123}(\pi)} t_2^{N_{132}(\pi)} t_3^{N_{213}(\pi)} t_4^{N_{231}(\pi)} t_5^{N_{312}(\pi)} t_6^{N_{321}(\pi)}.$$

In the previous chapter, we defined weight functions and derived functional equations for the patterns 123, 132, and 231 directly, but analogous functional equations can also be derived for 321, 312, and 213.[2] These six functional equations can be combined to count occurrences of all the length three patterns. This has been implemented in the accompanying Maple package `FS3`. For example, the procedure `FS3full(n,Lt)` will compute the full f_n polynomial. Using this, we are able to compute the full multivariate polynomial $f_n(t_1, t_2, t_3, t_4, t_5, t_6)$ for $n \le 11$.[3]

As an example, the Maple call `FS3full(4,[t[1],t[2],t[3],t[4],t[5],t[6]]);` would produce the output:

$$\begin{gathered} {t_1}^4 + {t_1}^2{t_2}^2 + {t_1}^2 t_2 t_3 + t_1 {t_2}^2 t_4 + {t_2}^2 t_1 t_5 + {t_2}^3 t_6 + {t_3}^2 {t_1}^2 + {t_3}^2 {t_2}^2 + t_1 t_4 {t_3}^2 \\ + t_1 {t_4}^3 + 2\, t_2 t_4 t_3 t_5 + t_2 {t_4}^2 t_6 + {t_3}^2 t_5 t_1 + {t_3}^3 t_6 + t_3 t_6 {t_4}^2 + {t_4}^2 {t_5}^2 + {t_4}^2 {t_6}^2 \\ + {t_5}^3 t_1 + {t_5}^2 t_2 t_6 + {t_5}^2 t_6 t_3 + t_5 {t_6}^2 t_4 + {t_6}^2 {t_5}^2 + {t_6}^4 \end{gathered}$$

---

[2] As opposed to computing these by merely considering the complement of the pattern.

[3] The main issue is not the speed of the algorithm but the size of the output. The output for the $n = 11$ case is a multivariate polynomial that requires 450 megabytes of space as a text file.

Chapter 5

Automating Existing Techniques

5.1 Introduction

Given a pattern τ and a permutation π, recall that $N_\tau(\pi)$ will denote the number of occurrences of τ in π. For example, if the pattern $\tau = 123$, the permutation 53412 avoids the pattern τ (so $N_{123}(53412) = 0$), whereas the permutation 52134 contains two occurrences of τ (so $N_{123}(52134) = 2$).

For a pattern τ and non-negative integer $r \geq 0$, we will again consider the set

$$\mathcal{S}_n(\tau, r) := \{\pi \in \mathcal{S}_n : \pi \text{ has exactly } r \text{ occurrences of the pattern } \tau\}$$

and the quantity $s_n(\tau, r) := |\mathcal{S}_n(\tau, r)|$. When $r = 0$, we will more simply write $\mathcal{S}_n(\tau)$ and $s_n(\tau)$ instead of $\mathcal{S}_n(\tau, 0)$ and $s_n(\tau, 0)$, respectively. Recall that the corresponding generating function is defined as

$$F_\tau^r(x) := \sum_{n=0}^{\infty} s_n(\tau, r) x^n.$$

Now recall that Dyck paths are paths with endpoints on the integer lattice consisting of *up-steps* (positional changes of $(1, 1)$) and *down-steps* (positional changes of $(1, -1)$) which start at the position $(0, 0)$, end at the position $(2n, 0)$, and never go below the x-axis. For each n, the set of Dyck paths from $(0, 0)$ to $(2n, 0)$ will be denoted by D_n. Recall that the set D_n is enumerated by the Catalan numbers C_n:

$$|D_n| = C_n = \frac{1}{n+1}\binom{2n}{n}$$

For length 3 patterns in the general $r \geq 0$ setting, there are only two Wilf-equivalence classes due to the trivial Wilf-equivalences of reversal and complementation: $\{123, 321\}$ and $\{132, 213, 231, 312\}$. Recall that in the avoidance case (where $r = 0$), the patterns

123 and 132 are Wilf-equivalent. This however does not hold for $r > 0$. Therefore for general r, there are two cases to consider for length 3 patterns. For this chapter, it will be convenient to consider the patterns 312 and 321 as representatives of these two equivalence classes.

In [32], Krattenthaler gave a new combinatorial proof for $s_n(312) = C_n$ and $s_n(321) = C_n$ by providing bijections from $\mathcal{S}_n(312)$ to the set of Dyck paths D_n and from $\mathcal{S}_n(321)$ to D_n. Fulmek [25] later extended the bijection to "generalized Dyck paths", which will be defined later, to compute $F^r_{312}(x)$ and $F^r_{321}(x)$ for $r = 1, 2$.

In this chapter, we review and reformulate Fulmek's approach for the pattern 312 so that it is more conducive to being computationally automated. In particular, this more automated approach allows us to re-derive Fulmek's original results and also to (rigorously) compute $F^r_{312}(x)$ for $r = 3, 4$. This approach has been implemented in the Maple package `FULMEK`. The end result is technically not new, since Mansour and Vainshtein found these generating functions through a different approach [37], but this work provides new alternate proofs that verify Mansour and Vainshtein's results. More importantly, this chapter serves as a case study on how certain results could be computationally automated and then pushed further to derive new, yet rigorous, results using computers. As a final comment, much of the work outlined here will be presented without formal proof since doing so would merely reproduce Fulmek's work. Those interested in the formal justifications and details should see his original article [25].

5.2 Preliminary definitions

We will start off by reviewing some definitions from Fulmek's original work [25]. First, consider a generalized lattice path that satisfies the following properties:

- Consists of *up-steps* (positional change of $(1, 1)$), *down-steps* (positional change of $(1, -1)$), and *down-jumps* (positional change of $(0, -1)$).
- Every up-step must be immediately followed by a down-step or another up-step.
- Every down-jump must be immediately followed by a down-step or another down-jump.

- Starts at position $(0,0)$ and ends at position $(m,0)$, where m will be the total number of up-steps and down-steps.
- Never goes below the x-axis.

We will refer to such a path as a *generalized Dyck path.* For notational convenience, we will often encode a path by a sequence of U's (up-steps), D's (down-steps), and J's (down-jumps). The notation U^k will denote a sequence of k consecutive U's (and similarly for D^k and J^k). For example, the graphical representation of the path $UUUDJDUD$ (equivalently, U^3DJDUD) is shown in Figure 5.1.

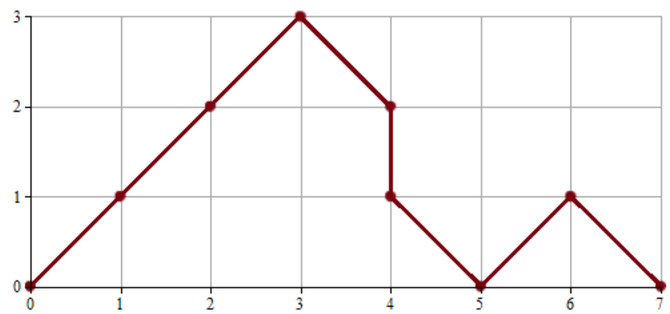


Figure 5.1: Graphical representation of $UUUDJDUD$

We will write GD_n to denote the set of generalized Dyck paths with exactly n down-steps. Note that a generalized Dyck path with no down-jumps is also a regular Dyck path. Also observe that for each $n > 1$, the set GD_n contains infinitely many paths (even though D_n is a finite set) since for each $k \geq 0$, the path $U^{n-1}U^{k+1}DJ^kD^{n-1}$ is in GD_n.

Consider an n-tuple of non-negative integers $(h_1, h_2, \ldots, h_n)$. If $h_n = 0$, we will refer to $(h_1, \ldots, h_n)$ as a *height-vector* of length n. We will define an injective function that maps each length n height-vector into a path in GD_n. The rough idea is that h_i represents the i-th down-step and is the height (or y-coordinate) of the path at the end of that down-step. Once the heights of the down-steps are established, the remaining up-steps and down-jumps are uniquely determined because of the conditions on a generalized Dyck path.

More precisely, define μ to be the function that maps the height-vector $(h_1, \ldots, h_n)$ to the path in GD_n determined by the following steps:

- Begin path at position $(0, 0)$.
- Draw $h_1 + 1$ up-steps followed by a down-step. (Note: the path is currently at height h_1).
- For each subsequent h_i, do the following:
  - If $h_i = h_{i-1} - 1$, then draw a down-step.
  - If $h_i \geq h_{i-1}$, then draw $h_i - h_{i-1} + 1$ up-steps followed by a down-step.
  - Otherwise (if $h_i \leq h_{i-1} - 2$), draw $h_{i-1} - h_i - 1$ down-jumps followed by a down-step.

Observe that this path will have exactly n down-steps, and for each i, the i-th down-step will end at height h_i. Also, since $h_n = 0$, the path must end at $(m, 0)$ for some positive integer m. It is clear (by its constructive definition) that μ is injective. For example, if $h = (0, 3, 1, 1, 0)$, then $\mu(h) = UDU^4DJDUDD$. This is graphically represented in Figure 5.2.

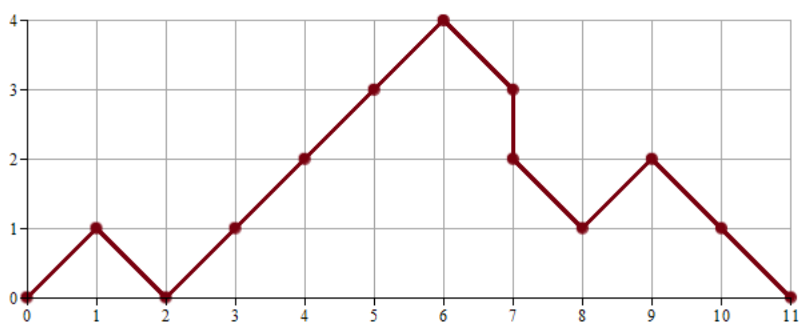


Figure 5.2: Graphical representation of $\mu(h) = UDU^4DJDUDD$

In the next section, we will define an injective mapping from permutations to height vectors and use that to create an injective function from permutations to generalized Dyck paths. We will then show what paths correspond to permutations in $\mathcal{S}_n(312, r)$ and show how to compute the generating function enumerating those paths.

5.3 Extending the approach for 312

We will start off by giving a brief overview of Fulmek's approach and then show how certain parts can be systematized and computationally automated. Given a permutation $\pi = \pi_1 \dots \pi_n$, we define the *height-vector function* (for the pattern 312) as $\mathrm{hv}(\pi) := (h_1, \dots, h_n)$ where $h_i := \#\{\pi_k : \pi_i > \pi_k,\ i < k\}$. We will refer to the quantity $(h_1, \dots, h_n)$ as the *height-vector* for π. For example, $\mathrm{hv}(32514) = (2, 1, 2, 0, 0)$. Observe that by construction, we always have $h_n = 0$, which matches our previous definition for a height-vector. Also note that this function is injective, since the permutation can be reconstructed by reading the entries of the height-vector in reverse.

Now, we define the function $\psi := \mu \circ \mathrm{hv}$, which maps a length n permutation into a path in GD_n. Since both hv and μ are injective, the function ψ is also injective. We also define the set

$$GD_n^r := \{\psi(\pi) : \pi \in \mathcal{S}_n(312, r)\}\,.$$

Observe that $GD_n^r \subseteq GD_n$ since hv maps $\pi \in \mathcal{S}_n$ into a length n height-vector, and μ maps that vector to a generalized Dyck path with n down-steps. We will also consider the larger set

$$GD^r := \bigcup_{n \geq 0} GD_n^r.$$

In [25], it is shown that down-jumps in a generalized Dyck path will correspond to one or more occurrences of a 312 pattern. More specifically, for each r, a generalized Dyck path is in GD^r if and only if it contains some collection of subpaths that correspond to ways that 312 patterns can occur within a permutation. A key point in [25] is that we can find all the ways that 312 patterns can occur in a permutation, translate those into possible subpaths, and enumerate the generalized Dyck paths containing those desired subpaths.

Given a variable x, we define a weight function w, where $w(U) = w(D) = x^{1/2}$ and $w(J) = 1$. If $p = p_1 p_2 \dots p_m$ is a path with m steps (each $p_i \in \{U, D, J\}$), we naturally extend the definition of the weight function to be $w(p) := \prod w(p_i)$. Given a path p, we also define the function $N_J(p)$ to be the number of down-jumps in the path p. As discussed in [25], our desired generating function can then be computed as

$$F_\tau^r(x) = \sum_{p \in GD^r} x^{-N_J(p)/2} w(p).$$

For notational convenience, we also define

$$c := \frac{1}{2x}\left(1 - \sqrt{1-4x}\right)$$

which is the generating function for the Catalan numbers (and in particular for ordinary Dyck-paths with no down-jumps). We begin by reviewing Fulmek's approach for $r = 1$ and $r = 2$.

5.3.1 Review of $r = 1$ case

We first begin by outlining the $r = 1$ case. Everything in this section should be assumed to be a review of Fulmek's work [25] unless otherwise indicated. Suppose that $\pi \in \mathcal{S}_n(312, 1)$. Since there is only one copy of the pattern, the terms in π that form an occurrence will be order-isomorphic to 312. We will refer to this 312 copy as the 312-base permutation (for $r = 1$). Observe that the ψ map sends 312 to the generalized Dyck path shown in Figure 5.3.

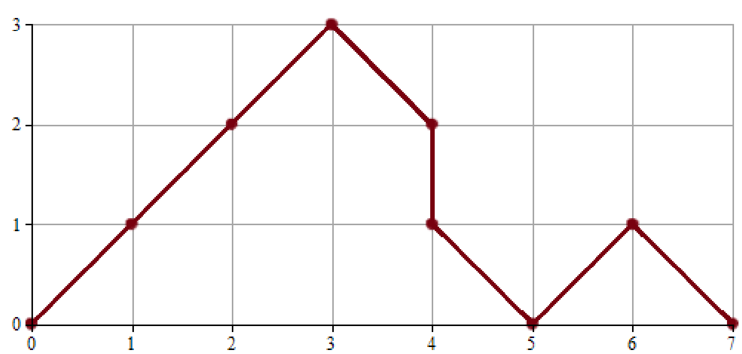


Figure 5.3: Generalized Dyck path for the permutation 312

Out of the corresponding path, the subpath that matters is $UUDJDU$, shown in Figure 5.4.

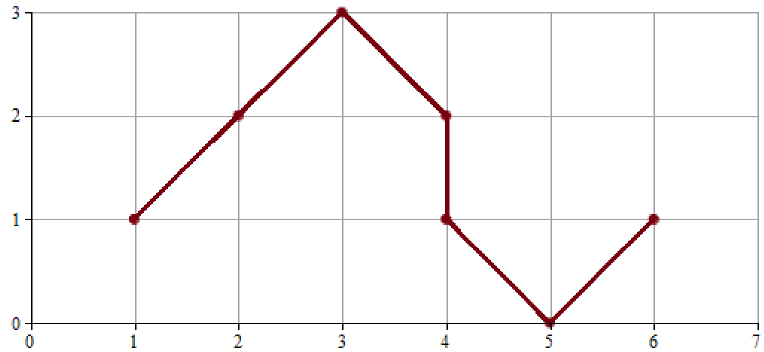


Figure 5.4: Base subpath of the permutation 312

We will refer to this as a *base subpath*.

Fulmek shows that a generalized Dyck path is in GD^1 if and only if the path contains the subpath $UUDJDU$ and contains no other down-jumps. To find the generating function $F^1_{312}(x)$, it is enough to find the generating function for paths containing $UUDJDU$ (and no other down-jumps).

Suppose that the subpath $UUDJDU$ starts at height l. First observe that $l \geq 1$, otherwise the subpath would go below the x-axis. As shown in [25], the generating function for "partial" Dyck paths that start at height 0 and end at height l is

$$c^{l+1}(w(U))^l = c^{l+1}x^{l/2}.$$

Also, there is no net change in height from the start of the subpath $UUDJDU$ to the end of the subpath. In other words, if the subpath begins at height l, the subpath will end at height l as well. Then, the generating function for the partial Dyck paths that start at height l and end at height 0 is also $c^{l+1}x^{l/2}$.

Piecing together the generating functions, we get that

$$\begin{aligned} F^1_{312} &= \sum_{l=1}^{\infty} x^{-1/2}(c^{l+1}x^{l/2})w(UUDJDU)(c^{l+1}x^{l/2}) \\ &= \sum_{l=1}^{\infty} x^{-1/2}(c^{l+1}x^{l/2})^2 x^{5/2} = \frac{c^4x^3}{1-c^2x}. \end{aligned}$$

5.3.2 Review of $r = 2$ case

We now review Fulmek's approach for $r = 2$. Again, everything in this section should be assumed to be a review of Fulmek's work [25] unless otherwise indicated. Suppose that $\pi \in \mathcal{S}_n(312, 2)$. Let $\pi' := \pi_{i_1}\pi_{i_2}\dots\pi_{i_m}$ be the subsequence of terms in π such that $N_{312}(\pi') = 2$ and every term in π' is part of a 312 occurrence. We will refer to the permutation $\pi'' := \text{red}(\pi')$ as a 312-base permutation (for $r = 2$). Fulmek shows that there are 8 possible base permutations for this case: 312645, 31524, 316452, 3412, 4132, 4213, 423615, and 4312. The map ψ sends each base permutation to a generalized Dyck path as shown in Table 5.1.

Now observe that if a generalized Dyck path contained two non-overlapping copies of the subpath $UUDJDU$ (from Figure 5.4), it would contain 2 copies of 312. Paths of this type have the form:

$$P' = P_1\ B\ P_2\ B\ P_3 \tag{5.1}$$

where P' is a generalized Dyck path, $B = UUDJDU$ (the base subpath for 312), and the subpaths P_1, P_2, and P_3 contain no down-jumps.

Observe that the base permutations 312645, 31524, 316452, and 423615 produce paths that are specific instances of this more general form. It is therefore sufficient to only consider paths containing the base subpaths derived from 3412, 4132, 4213, and 4312 in addition to paths with the general form of Eq. 5.1. The base subpaths for the necessary base permutations are shown in Table 5.2.

The generating function for each case will be computed by piecing together the generating function (as in the $r = 1$ case).

For the pattern 3412, the base subpath is $UUDUDJDU$. If this subpath begins at height l, then the subpath will also end at height l. Also observe that $l \geq 1$ (otherwise the path goes below the x-axis). Then, the generating function enumerating paths in GD^2 containing this base subpath is

$$\begin{aligned}\sum_{l=1}^{\infty} x^{-1/2}(c^{l+1}x^{l/2})w(UUDUDJDU)(c^{l+1}x^{l/2}) &= \sum_{l=1}^{\infty} x^{-1/2}(c^{l+1}x^{l/2})^2 x^{7/2} \\ &= \frac{c^4x^4}{1-c^2x}.\end{aligned} \tag{5.2}$$

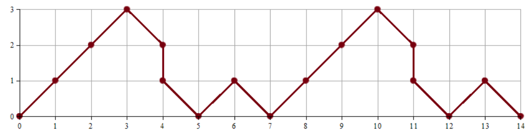
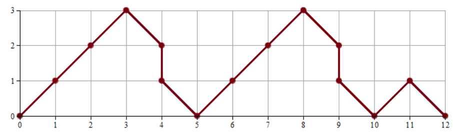
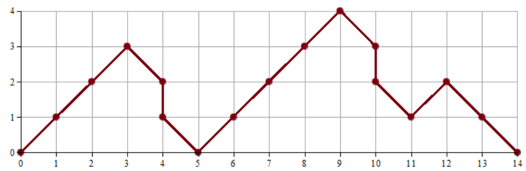
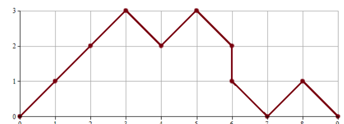
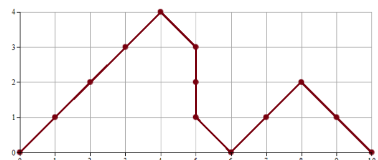
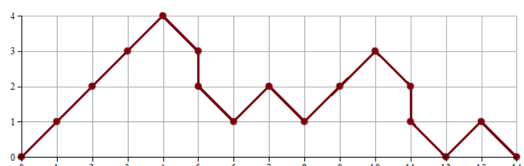
Base permutation	Generalized Dyck path
312645	
31524	
316452	
3412	
4132	
4213	
423615	
4312	

Table 5.1: Generalized Dyck paths of base permutations for $r = 2$ case

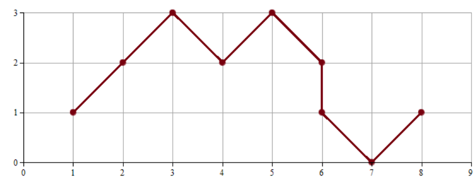
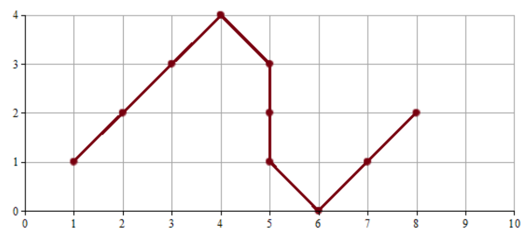
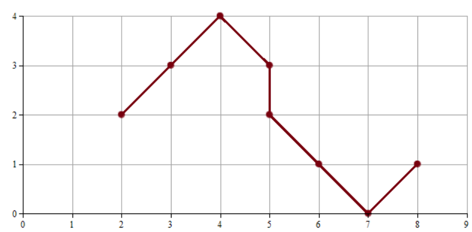
Base permutation	Generalized Dyck path
3412	
4132	
4213	
4312	

Table 5.2: Necessary base subpaths for $r = 2$ case

For the pattern 4132, the base subpath is $UUUDJJDUU$. If this subpath begins at height l, then the subpath will end at height $l+1$. Also observe that $l \geq 1$ (otherwise the path goes below the x-axis). Then, the generating function enumerating paths in GD^2 containing this base subpath is

$$\begin{aligned}\sum_{l=1}^{\infty} x^{-2/2}(c^{l+1}x^{l/2})w(UUUDJJDUU)(c^{l+2}x^{(l+1)/2}) &= \sum_{l=1}^{\infty} x^{-1}(c^{2l+3}x^{(2l+1)/2})x^{7/2} \\ &= \frac{c^5x^4}{1-c^2x}. \qquad (5.3)\end{aligned}$$

For the pattern 4213, the base subpath is $UUDJDDU$. If this subpath begins at height l, then the subpath will end at height $l-1$. Also observe that $l \geq 2$ (otherwise the path goes below the x-axis). Then, the generating function enumerating paths in GD^2 containing this base subpath is

$$\begin{aligned}\sum_{l=2}^{\infty} x^{-1/2}(c^{l+1}x^{l/2})w(UUDJDDU)(c^{l}x^{(l-1)/2}) &= \sum_{l=2}^{\infty} x^{-1/2}(c^{2l+1}x^{(2l-1)/2})x^{6/2} \\ &= \frac{c^5x^4}{1-c^2x}. \qquad (5.4)\end{aligned}$$

For the pattern 4312, the base subpath is $UUUDDJDU$. If this subpath begins at height l, then the subpath will also end at height l. Also observe that $l \geq 1$ (otherwise the path goes below the x-axis). Then, the generating function enumerating paths in GD^2 containing this base subpath is

$$\begin{aligned}\sum_{l=1}^{\infty} x^{-1/2}(c^{l+1}x^{l/2})w(UUUDDJDU)(c^{l+1}x^{l/2}) &= \sum_{l=1}^{\infty} x^{-1/2}(c^{l+1}x^{l/2})^2x^{7/2} \\ &= \frac{c^4x^4}{1-c^2x}. \qquad (5.5)\end{aligned}$$

Finally, we consider the case of two disjoint $UUDJDU$ base subpaths occurring to form the two 312 patterns in a permutation.[1] It is important to note that while the two subpaths may be disjoint in a path from GD^2, the two occurrences of 312 in the corresponding permutation may not be disjoint. For example, the permutation 316452 has one 312 occurrence (formed by the terms 645) within another 312 occurrence, but the corresponding generalized Dyck path has two non-overlapping copies of the subpath

[1] By disjoint, we mean that the two subpaths do not share a "step" together. One copy *is* allowed to start immediately at the conclusion of another copy.

$UUDJDU$. Another example is the permutation 31524, which has two 312 copies that actually have a term in common (namely the "2" in 31524), yet when converted to a generalized Dyck path, has two disjoint copies of the base subpath $UUDJDU$.

We now consider how to find the generating function enumerating paths in GD^2 with two copies of $UUDJDU$. Recall that the generating function enumerating paths (with no jumps) that climb from height 0 to height l is

$$c^{l+1}x^{l/2}.$$

We also now want to compute a more general quantity. Suppose we want the generating function for all paths that start at height k and end at height l (with $k \leq l$) and do not go below the x-axis. As shown in [25], our desired generating function is

$$C_{k,l} := \sum_{d=0}^{k} \left(\sqrt{x}\right)^{l-k+2d} c^{l-k+2d+1} = \frac{\left(\left(c^2x\right)^{k+1} - 1\right) c^{l-k+1}x^{(l-k)/2}}{c^2x - 1}.$$

(Note: the indexing variable d may be viewed as the maximum decrease in height that the path achieves relative to the starting height k. In other words, $k-d$ is the minimum height that the path achieves.)

At this point, we diverge from Fulmek's approach for computing the generating function. While the computation is essentially the same, we reformulate the set-up so that it is more conducive to generalization. We will denote the first and second occurrences of $UUDJDU$ by P_1 and P_2, respectively. Suppose that h_i denotes the starting height of subpath P_i. Also, let Δh_i denote the net change in height from the start of subpath P_i to the end of that subpath. Then, the ending height for subpath P_1 is $h_1 + \Delta h_1$ (and similarly, $h_2 + \Delta h_2$ for P_2).

From the $r = 1$ case, we know that $h_1, h_2 \geq 1$ and $\Delta h_1 = \Delta h_2 = 0$. There are two cases to consider: $h_1 + \Delta h_1 \leq h_2$ (P_2 starts at the same or a greater height than the end height of P_1) or $h_1 + \Delta h_1 > h_2$ (P_2 starts at a lower height than the end height of P_1). The corresponding generating function for the first case is:

$$\sum_{h_1=1}^{\infty} \sum_{h_2=h_1}^{\infty} x^{-2/2} \left(c^{h_1+1}x^{h_1/2}\right) w(P_1) \left(C_{h_1+\Delta h_1, h_2}\right) w(P_2) \left(c^{h_2+\Delta h_2+1}x^{(h_2+\Delta h_2)/2}\right). \tag{5.6}$$

Similarly, the corresponding generating function for the second case is:

$$\sum_{h_1=2}^{\infty} \sum_{h_2=1}^{h_1+\Delta h_1-1} x^{-2/2} \left(c^{h_1+1} x^{h_1/2}\right) w(P_1) \left(C_{h_1+\Delta h_1, h_2}\right) w(P_2) \left(c^{h_2+\Delta h_2+1} x^{(h_2+\Delta h_2)/2}\right). \tag{5.7}$$

Also, recall that $w(P_i) = x^{5/2}$. Simplifying these two expressions and adding them together, we get the generating function:

$$\frac{c^5 x^5 \left(1 + c^2 x - c^4 x^2\right)}{\left(1 - c^2 x\right)^3}. \tag{5.8}$$

Finally, we combine all the generating functions (Eq. 5.2, 5.3, 5.4, 5.5, 5.8) to get

$$F_{312}^2(x) = \frac{c^4 x^4}{1 - c^2 x} + \frac{c^5 x^4}{1 - c^2 x} + \frac{c^5 x^4}{1 - c^2 x} + \frac{c^4 x^4}{1 - c^2 x} + \frac{c^5 x^5 \left(1 + c^2 x - c^4 x^2\right)}{\left(1 - c^2 x\right)^3} \tag{5.9}$$

which matches Fulmek's result in [25].

5.3.3 Automating the approach

The approach for $r = 2$ can be systematized and computationally implemented to compute $F_{312}^r(x)$ for larger r. Consider a fixed $r > 0$. The generating function can be computed as follows.

Finding the base subpaths

The first step would be to find all the necessary base subpaths to piece together. For each $1 \leq i \leq r$, let B_i be the set of base permutations for exactly i occurrences of 312. After computing the sets $B_1, \ldots, B_r$, we will compute the sets of desired base subpaths $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_r$ iteratively.

We begin by defining a few operations on paths. Given a path $P = p_1 p_2 \ldots p_m$, let $\text{TRIMTAIL}(P) := p_1 \ldots p_{m'}$, where m' is the smallest number such that $p_{m'+1} = p_{m'+2} = \ldots = p_m = D$. In essence, $\text{TRIMTAIL}(P)$ removes the final string of consecutive down-steps from the path, if it exists. For example,

$$\text{TRIMTAIL}(UUUDJDUD) = UUUDJDU$$

$$\text{TRIMTAIL}(U^4 D J^2 D U^2 D^2) = U^4 D J^2 D U^2$$

$$\text{TRIMTAIL}(U^4 D J^2 D U^2) = U^4 D J^2 D U^2$$

Next, we define the MINFILL function. Suppose we have a subpath $P = p_1p_2 \dots p_m$. Let Δh be the net change in height from the start to the end of P. Also, let b be the maximum height decrease achieved in P relative to the starting height. Then for this path, $\text{MINFILL}(P) = U^b P D^{b+\Delta h}$. In other words, MINFILL(P) is the minimal path satisfying the conditions for a generalized Dyck path that contains P as a subpath. For example, if $P = UUDJDUU$, then $\Delta h = 1$, $b = 1$, and $\text{MINFILL}(P) = UUUDJDUUDD$.

Finally, we define the TRIMHEAD function. Again, suppose we have a subpath $P = p_1p_2 \dots p_m$. If $p_1 \neq U$, then $\text{TRIMHEAD}(P) = P$. Otherwise, let r' be the integer such that $\text{MINFILL}(P) \in GD^{r'}$, and let k be the number of consecutive U's at the beginning of P. Let j be the largest number ($0 \leq j \leq k$) such that $\text{MINFILL}(Dp_{j+1}p_{j+2} \dots p_m) \in GD^{r'}$. Then, $\text{TRIMHEAD}(p_1p_2 \dots p_m) = p_{j+1}p_{j+2} \dots p_m$. In other words, TRIMHEAD removes the maximum number of initial up-steps while ensuring that adding steps before and after the subpath will not create new 312 occurrences.

For example, $\text{TRIMHEAD}(U^3DJDU) = U^2DJDU$ since $\text{MINFILL}(DU^2DJDU)$ and $\text{MINFILL}(U^3DJDU)$ are paths with the same number of occurrences of 312, while $\text{MINFILL}(DUDJDU)$ would have more occurrences of 312 than $\text{MINFILL}(U^3DJDU)$.

We can now find the base subpaths needed to compute $F^r_{312}(x)$. First, set $\mathcal{P}_1 := \{UUDJDU\}$, which is the base subpath from the $r = 1$ case. Each subsequent $\mathcal{P}_i$ will be computed as follows. First, begin with $\mathcal{P}_i := \{\}$.

For each base permutation $\pi \in B_i$, do:

- Compute $P := \psi(\pi)$.
- IF P can be formed by piecing together base subpaths out of $\mathcal{P}_1 \bigcup \dots \bigcup \mathcal{P}_{i-1}$ (along with up-steps and down-steps), discard it.

  ELSE, add the new base subpath TRIMHEAD(TRIMTAIL(P)) into the set $\mathcal{P}_i$.

In essence, $\mathcal{P}_i$ is being constructed to prevent redundancies (the subpaths can't be created by other base subpaths) and so that it is a minimal subpath (no additional 312 patterns will be formed when piecing together subpaths). Each subpath in $\mathcal{P}_i$ corresponds to a way that i occurrences of the 312 pattern can be formed in a generalized

Dyck path.

Piecing everything together

We can now piece together the generating functions in a manner that is analogous to the $r = 2$ case. Let I_r denote the set of integer compositions of r, where the compositions will be written as a list $[d_1, d_2, \ldots, d_s]$. For example, $I_3 = \{[3], [1,2], [2,1], [1,1,1]\}$. Each integer composition will represent a way to piece together subpaths to form r occurrences of 312.

For each composition $[d_1, d_2, \ldots, d_s] \in I_r$ and every combination of subpaths $B_1 \in \mathcal{P}_{d_1}, B_2 \in \mathcal{P}_{d_2}, \ldots, B_s \in \mathcal{P}_{d_s}$, we will compute the generating function enumerating paths of the form:

$$P' = P_1\ B_1\ P_2\ B_2\ \ldots\ B_s\ P_{s+1}$$

where P' is a generalized Dyck path and the subpaths $P_1, P_2, \ldots, P_{s+1}$ contain no down-jumps. As in the $r = 2$ case, this will require considering different cases for the starting heights for each subpath B_i.

The desired generating function $F^r_{312}(x)$ will be the sum of all these generating functions (ranging over all ordered combinations of subpaths over all compositions in I_r). All of these previous steps (from deriving the base subpaths to determining the proper boundaries for all the summations and computing the generating function) can be automated and has been implemented in the Maple package `FULMEK`. Using this Maple package, we are able to extend Fulmek's approach to the cases of $r = 3$ and $r = 4$.

For example, the Maple call `GF312x3(x);` will output the generating function

$$F^3_{312}(x) = \frac{2x^3 - 5x^2 + 7x - 2}{2} + \frac{-106x^5 - 22x^6 + 292x^4 - 302x^3 + 135x^2 - 27x + 2}{2}(1-4x)^{-5/2}$$

and the Maple call `GF312x4(x);` will output the generating function

$$F^4_{312}(x) = \frac{5x^4 - 7x^3 + 2x^2 + 8x - 3}{2}$$
$$+\frac{2x^9 + 218x^8 + 1074x^7 - 1754x^6 + 388x^5 + 1087x^4 - 945x^3 + 320x^2 - 50x + 3}{2}(1-4x)^{-7/2}.$$

Unfortunately, the computations became too complicated for $r > 4$. These results verify (and provide alternate proofs) to the results of Mansour and Vainshtein [37]. Most importantly, this chapter provides an example on how certain techniques can be systematized and automated to derive results beyond what can be achieved by merely human means.

References

[1] M. H. Albert, M. Elder, A. Rechnitzer, P. Westcott, and M. Zabrocki. On the Stanley-Wilf limit of 4231-avoiding permutations and a conjecture of Arratia. *Adv. in Appl. Math.*, 36(2):96–105, 2006.

[2] R. E. L. Aldred, M. D. Atkinson, and D. J. McCaughan. Avoiding consecutive patterns in permutations. *Adv. in Appl. Math.*, 45(3):449–461, 2010.

[3] Richard Arratia. On the Stanley-Wilf conjecture for the number of permutations avoiding a given pattern. *Electron. J. Combin.*, 6:Note, N1, 4 pp. (electronic), 1999.

[4] Jörgen Backelin, Julian West, and Guoce Xin. Wilf-equivalence for singleton classes. *Adv. in Appl. Math.*, 38(2):133–148, 2007.

[5] Andrew Baxter. Refining enumeration schemes to count according to the inversion number. *Pure Math. Appl. (PU.M.A.)*, 21(2):137–160, 2010.

[6] Andrew Baxter and Lara Pudwell. Enumeration schemes for vincular patterns. *Discrete Math.*, 312(10):1699–1712, 2012.

[7] Miklós Bóna. A new upper bound for 1324-avoiding permutations. arxiv:1207.2379 [math.co], 2012.

[8] Miklós Bóna. Exact enumeration of 1342-avoiding permutations: a close link with labeled trees and planar maps. *J. Combin. Theory Ser. A*, 80(2):257–272, 1997.

[9] Miklós Bóna. The number of permutations with exactly r 132-subsequences is P-recursive in the size! *Adv. in Appl. Math.*, 18(4):510–522, 1997.

[10] Miklós Bóna. Permutations with one or two 132-subsequences. *Discrete Math.*, 181(1-3):267–274, 1998.

[11] Miklós Bóna. The absence of a pattern and the occurrences of another. *Discrete Math. Theor. Comput. Sci.*, 12(2):89–102, 2010.

[12] Miklós Bóna. *Combinatorics of permutations.* Discrete Mathematics and its Applications (Boca Raton). CRC Press, Boca Raton, FL, second edition, 2012. With a foreword by Richard Stanley.

[13] Alexander Burstein. A short proof for the number of permutations containing pattern 321 exactly once. *Electron. J. Combin.*, 18(2):Paper 21, 3, 2011.

[14] David Callan. A recursive bijective approach to counting permutations containing 3-letter patterns. arxiv:math/0211380 [math.co], 2002.

[15] Anders Claesson, Vít Jelínek, and Einar Steingrímsson. Upper bounds for the Stanley-Wilf limit of 1324 and other layered patterns. *J. Combin. Theory Ser. A*, 119(8):1680–1691, 2012.

[16] Anders Claesson and Sergey Kitaev. Classification of bijections between 321- and 132-avoiding permutations. *Sém. Lothar. Combin.*, 60:Art. B60d, 30, 2008/09.

[17] Vladimir Dotsenko and Anton Khoroshkin. Anick-type resolutions and consecutive pattern avoidance. arxiv:1002.2761 [math.co], 2010.

[18] Adrian Duane and Jeffrey Remmel. Minimal overlapping patterns in colored permutations. *Electron. J. Combin.*, 18(2):Paper 25, 38, 2011.

[19] Richard Ehrenborg, Sergey Kitaev, and Peter Perry. A spectral approach to consecutive pattern-avoiding permutations. *J. Comb.*, 2(3):305–353, 2011.

[20] Sergi Elizalde. The most and the least avoided consecutive patterns. *Proc. Lond. Math. Soc.,to appear.*

[21] Sergi Elizalde. Asymptotic enumeration of permutations avoiding generalized patterns. *Adv. in Appl. Math.*, 36(2):138–155, 2006.

[22] Sergi Elizalde and Toufik Mansour. Restricted Motzkin permutations, Motzkin paths, continued fractions and Chebyshev polynomials. *Discrete Math.*, 305(1-3):170–189, 2005.

[23] Sergi Elizalde and Marc Noy. Consecutive patterns in permutations. *Adv. in Appl. Math.*, 30(1-2):110–125, 2003. Formal power series and algebraic combinatorics (Scottsdale, AZ, 2001).

[24] Sergi Elizalde and Marc Noy. Clusters, generating functions and asymptotics for consecutive patterns in permutations. *Adv. in Appl. Math.*, 49:351–374, 2012.

[25] Markus Fulmek. Enumeration of permutations containing a prescribed number of occurrences of a pattern of length three. *Adv. in Appl. Math.*, 30(4):607–632, 2003.

[26] Ira Gessel. Symmetric functions and P-recursiveness. *J. Combin. Theory Ser. A*, 53(2):257–285, 1990.

[27] I. P. Goulden and D. M. Jackson. An inversion theorem for cluster decompositions of sequences with distinguished subsequences. *J. London Math. Soc. (2)*, 20(3):567–576, 1979.

[28] Anton Khoroshkin and Boris Shapiro. Using homological duality in consecutive pattern avoidance. *Electron. J. Combin.*, 18(2):Paper 9, 17, 2011.

[29] Sergey Kitaev. *Patterns in Permutations and Words.* EATCS monographs in Theoretical Computer Science. Springer Verlag, 2011.

[30] Sergey Kitaev and Toufik Mansour. A survey on certain pattern problems. *University of Kentucky research report 2003-09*, 2003.

[31] Donald E. Knuth. *The art of computer programming.* Vol. 1: Fundamental algorithms. Addison Wesley, Reading, Massachusetts, 1973.

[32] C. Krattenthaler. Permutations with restricted patterns and Dyck paths. *Adv. in Appl. Math.*, 27(2-3):510–530, 2001. Special issue in honor of Dominique Foata's 65th birthday (Philadelphia, PA, 2000).

[33] Jeffrey Liese and Jeffrey Remmel. Generating functions for permutations avoiding a consecutive pattern. *Ann. Comb.*, 14(1):123–141, 2010.

[34] T. Mansour and A. Vainshtein. Restricted permutations and Chebyshev polynomials. *Sém. Lothar. Combin.*, 47:Article B47c, 17, 2001/02.

[35] Toufik Mansour. Permutations containing and avoiding certain patterns. In *Formal power series and algebraic combinatorics (Moscow, 2000)*, pages 704–708. Springer, Berlin, 2000.

[36] Toufik Mansour and Alek Vainshtein. Restricted permutations, continued fractions, and Chebyshev polynomials. *Electron. J. Combin.*, 7:Research Paper 17, 9 pp. (electronic), 2000.

[37] Toufik Mansour and Alek Vainshtein. Counting occurrences of 132 in a permutation. *Adv. in Appl. Math.*, 28(2):185–195, 2002.

[38] Adam Marcus and Gábor Tardos. Excluded permutation matrices and the Stanley-Wilf conjecture. *J. Combin. Theory Ser. A*, 107(1):153–160, 2004.

[39] Darko Marinov and Radoš Radoičić. Counting 1324-avoiding permutations. *Electron. J. Combin.*, 9(2):Research paper 13, 9 pp. (electronic), 2002/03. Permutation patterns (Otago, 2003).

[40] Anthony Mendes and Jeffrey Remmel. Permutations and words counted by consecutive patterns. *Adv. in Appl. Math.*, 37(4):443–480, 2006.

[41] Brian Nakamura. Computational approaches to consecutive pattern avoidance in permutations. *Pure Math. Appl. (PU.M.A.)*, 22(2):253–268, 2011.

[42] Brian Nakamura and Doron Zeilberger. Using Noonan–Zeilberger Functional Equations to enumerate (in polynomial time!) generalized Wilf classes. *Adv. in Appl. Math.*, 50(3):356–366, 2013.

[43] Eugen Netto. *Lehrbuch der Combinatorik.* Chelsea Publishing Company, New York, 1958.

[44] John Noonan. The number of permutations containing exactly one increasing subsequence of length three. *Discrete Math.*, 152(1-3):307–313, 1996.

[45] John Noonan and Doron Zeilberger. The enumeration of permutations with a prescribed number of "forbidden" patterns. *Adv. in Appl. Math.*, 17(4):381–407, 1996.

[46] John Noonan and Doron Zeilberger. The Goulden-Jackson cluster method: extensions, applications and implementations. *J. Differ. Equations Appl.*, 5(4-5):355–377, 1999.

[47] Guillem Perarnau. A probabilistic approach to consecutive pattern avoiding in permutations. *J. Combin. Theory Ser. A*, 120(5):998–1011, 2013.

[48] Lara Pudwell. Enumeration schemes for permutations avoiding barred patterns. *Electron. J. Combin.*, 17(1):Research Paper 29, 27, 2010.

[49] Don Rawlings. The q-exponential generating function for permutations by consecutive patterns and inversions. *J. Combin. Theory Ser. A*, 114(1):184–193, 2007.

[50] Aaron Robertson. Permutations containing and avoiding 123 and 132 patterns. *Discrete Math. Theor. Comput. Sci.*, 3(4):151–154 (electronic), 1999.

[51] Aaron Robertson. Permutations restricted by two distinct patterns of length three. *Adv. in Appl. Math.*, 27(2-3):548–561, 2001. Special issue in honor of Dominique Foata's 65th birthday (Philadelphia, PA, 2000).

[52] Aaron Robertson, Herbert S. Wilf, and Doron Zeilberger. Permutation patterns and continued fractions. *Electron. J. Combin.*, 6:Research Paper 38, 6 pp. (electronic), 1999.

[53] Rodica Simion and Frank W. Schmidt. Restricted permutations. *European J. Combin.*, 6(4):383–406, 1985.

[54] Neil Sloane. The On-Line Encyclopedia of Integer Sequences,`http://oeis.org/`, 2013.

[55] Zvezdelina E. Stankova. Forbidden subsequences. *Discrete Math.*, 132(1-3):291–316, 1994.

[56] Einar Steingrímsson. Some open problems on permutation patterns. *London Mathematical Society Lecture Note Series,to appear.*

[57] Vince Vatter. Enumeration schemes for restricted permutations. *Combin. Probab. Comput.*, 17(1):137–159, 2008.

[58] R. Warlimont. Permutations avoiding consecutive patterns. *Ann. Univ. Sci. Budapest. Sect. Comput.*, 22:373–393, 2003.

[59] Herbert S. Wilf. What is an answer? *Amer. Math. Monthly*, 89(5):289–292, 1982.

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