# AN EXPERIMENTAL WALK IN PATTERNS, PARTITIONS AND WORDS

**By**

**MINGJIA YANG**

**A dissertation submitted to the**

**School of Graduate Studies**

**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, 2018**

## ABSTRACT OF THE DISSERTATION


# An Experimental Walk in Patterns, Partitions and Words

**By MINGJIA YANG**

**Dissertation Director: Doron Zeilberger**

Experimental mathematics, broadly speaking, is the philosophy that computers are a valuable tool that should be used extensively in mathematical research. This thesis explores topics related to partitions, patterns and words, incoporating the spirit of experimental math. There are four main projects in this thesis, and we will take a walk from the least experimental to the most experimental.

In the first project, we extend Shar and Zeilberger's work [SZ] on generating functions enumerating 123-avoiding words (with $r$-occurrences of each letter) to words (with $r$-occurrences of each letter) having exactly one 123 pattern. After a system of equations has been established (by human means), we use computer to find the defining algebraic equation for generating functions for words with $r$ occurrences of each letter and with exactly one 123 pattern, and derive relevant recurrences.

Next, we move on to explore consecutive pattern avoidance, in particular, words that avoid the increasing consecutive pattern $12 \cdots r$ for any $r \geq 2$. We use computer to conjecture the corresponding generating function and then tweak the Goulden-Jackson cluster method to prove the result by human means. We also treat the more general case of counting words with a specified number of the pattern of interest.

After these, we dive into the world of partitions. More precisely, we introduce the combinatorial object which we call "relaxed partitions". A relaxed partition of a positive integer $n$ is a finite sequence of positive integers $\lambda_1, \lambda_2, ..., \lambda_k$ $(\lambda_i - \lambda_{i+1} \geq r)$ whose sum is equal to $n$, where $r$ is allowed to be negative (note that if we only allow $r$ to be non-negative, then we get traditional partitions). We use computer to conjecture and prove the formula for the number of $r$-partitions $(r < 0)$ with fixed first part and number of parts. We also use computer to explore corresponding generating functions.

Last but not least, we go back to traditional partitions and design an efficient algorithm to count restricted partitions. We start out with a more basic algorithm and then generalize it to account for more complicated partitions, like in the Kanade-Russell conjectures/theorems. We then make use of Frank Garvan's $q$-series Maple package and Amarel cluster computing to search for new partition identities. Many new identities have been discovered and (at least) one of them generalizes to an infinitely family.

## Acknowledgements

All the acknowledgments.

# Dedication

To...

# Table of Contents

**Abstract** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ii

**Acknowledgements** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . iv

**Dedication** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . v

**1. Introduction** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

1.1. Words that contain the pattern 123 exactly once . . . . . . . . . . . . . 2

1.2. Increasing consecutive patterns in words . . . . . . . . . . . . . . . . . 3

1.3. Relaxed partitions . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

1.4. Systematic counting of restricted partitions and searching for new partition identities . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

**2. Words that contain the pattern 123 exactly once** . . . . . . . . . . . 6

2.1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

2.2. Enumeration of words that contain the pattern 123 exactly once . . . . 8

2.3. Generating functions . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

2.4. Using Maple packages . . . . . . . . . . . . . . . . . . . . . . . . . . . 16

**3. Increasing consecutive patterns in words** . . . . . . . . . . . . . . . 18

3.1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

3.2. Method, experimentation, and results . . . . . . . . . . . . . . . . . . 20

3.3. Proofs by tweaking the Goulden-Jackson cluster method . . . . . . . . . 28

**4. Relaxed partitions** . . . . . . . . . . . . . . . . . . . . . . . . . . . 33

4.1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33

4.2. The recurrence relation for $F(M,N,r,q)$ . . . . . . . . . . . . . . . . . 35

4.3. Using Maple to discover (and prove!) patterns for $F(M,N,r,1)$ . . . . . 35

4.4. Can we find a pattern for $F(M, N, r, q)$? . . . . . . 38

4.5. Future work and connection to other combinatorial objects . . . . . . 41

**5. Systematic counting of restricted partitions and searching for new partition identities** . . . . . . 43

5.1. Introduction . . . . . . 43

5.2. Adapting the Goulden-Jackson cluster method . . . . . . 47

5.3. A more straight-forward approach . . . . . . 50

5.4. Generalization . . . . . . 53

5.5. Searching for new partition identities . . . . . . 57

5.6. Future work . . . . . . 63

**References** . . . . . . 65

# Chapter 1

# Introduction

Francis Su said in his book *Mathematics for Human Flourishing*: Mathematics is the science of patterns and the art of engaging the meaning of patterns.

To me, experimental mathematics, is an art of using computers in the discovery and study of patterns. With the ever-increasing power of modern computers and the flexibility to program in a desired mathematical software (in our case, Maple), the way we do mathematical research is changing. To see an interesting philosophical discussion of experimental mathematics, please read the first chapter of *Experimental Mathematics in Action*.

This thesis, in particular Chapter 3-5, highlights the usefulness of computers in discovering patterns. While a large chunk of Chapter 2 is done by human analysis, computer is indispensable in helping us solve a systems of equations and find defining algebraic equations, as well as deriving recurrences. This chapter is least experimental in the sense that computer is mainly used in "solving" things, not "discovering" things. Also, we use existing Maple packages to a large extent. In Chapter 3, we use an existing Maple package implementing the Goulden-Jackson cluster method, along with a new package written by us to discover patterns and then prove them by human means. In Chapter 4, the spirit of "experiment" becomes even more apparent. First of all, the combinatorial object itself is an experiment–we introduce what we call "relaxed partitions", something that can be seen as a traditional integer partitions with some conditions relaxed. We program in Maple using an easy recurrence relation, and conjecture (and prove!) using the program for $q = 1$ case. For the $q$ case, we also program and experiment in Maple to find patterns. Intriguing patterns have appeared in our experiments, although an explicit form is yet to be discovered. Chapter 5 is probably the most experimental in the sense that after devising an efficient algorithm for enumeration of restricted partitions, all the searching for new partition identities are done by computer, actually, by a cluster of computers! Without Amarel cluster computing available at Rutgers, the searching process would have gone on for years instead of days.

Below I will summarize the material in the four chapters and how we use experimental math on this journey.

## 1.1 Words that contain the pattern 123 exactly once

Recall that a word $w = w_1 ... w_k$ is an ordered list of letters on some alphabet. To say a word contains a pattern (a certain permutation of $\{1, ..., m\}$) $\sigma$ is to say there exist $1 \leq i_1 < i_2 < ... < i_m \leq k$ such that the subword $w_{i_1} ... w_{i_m}$ is *order isomorphic* to $\sigma$ (for example, 246 is order isomorphic to 123). A word avoids the pattern $\sigma$ if it does not contain it.

Enumeration problems related to words avoiding patterns as well as permutations that contain the pattern 123 exactly once have been studied in great detail. However, the problem of enumerating words that contain the pattern 123 exactly once is new and is the focus of this chapter. Previously, Doron Zeilberger provided a shortened version of Alexander Burstein's combinatorial proof of John Noonan's theorem that the number of permutations with exactly one 321 pattern is equal to $\frac{3}{n}\binom{2n}{n+3}$. Surprisingly, a similar method can be directly adapted to words. We use this method to find a formula enumerating the words with exactly one 123 pattern.

Further inspired by Nathaniel Shar and Zeilberger's paper on generating functions enumerating 123-avoiding words with $r$ occurrences of each letter, we are able to set up a system of equations and use an existing algorithm on Maple to find the defining algebraic equation for generating functions for words with $r$ occurrences of each letter and with exactly one 123 pattern, for $r = 2$ and 3. We then use the **SCHUTZENBERGER** Maple package written by Doron Zeilberger to derive recurrences for those algebraic equations.

## 1.2 Increasing consecutive patterns in words

Next, we go from classical patterns to consecutive patterns. Recall a word $\pi = \pi_1 \cdots \pi_n$ avoids a consecutive pattern $\sigma = \sigma_1 \cdots \sigma_k$ if none of the $n-k+1$ length-$k$ consecutive subwords, $\pi_i \pi_{i+1} \cdots \pi_{i+k-1}$ of $\pi$, reduces to $\sigma$.

In particular, we use an existing Maple package implementing the Goulden-Jackson cluster method, along with a newly written package to explore how to enumerate words in $1^{m_1} \cdots n^{m_n}$ that avoid the increasing consecutive pattern $12 \cdots r$ for any $r \geq 2$. With not a huge amount of effort, patterns emerge and a closed form is discovered. We then tweak the Goulden-Jackson cluster method to prove the result by hand.

By simple manipulation of the result and symmetry, we get an $O(n^{s+1})$ algorithm to enumerate words in $1^s \cdots n^s$, avoiding the consecutive pattern $1 \cdots r$, for *any* $s$, and *any* $r$.

we also treat the more general case of counting words with a specified number of the pattern of interest (the avoiding case corresponding to zero appearances). Although the proof idea of the avoidance case extend to the general case, and we are able to come up with an explicit form for the general case by hand, we use computer to confirm our result.

## 1.3 Relaxed partitions

One of the cornerstones of enumerative combinatorics (and number theory!) are integer partitions. Recall that a partition of a non-negative integer $n$ is a list of integers $(\lambda_1, \ldots, \lambda_k)$ such that $\lambda_1 \geq \cdots \geq \lambda_k \geq 1$ and $\lambda_1 + \cdots + \lambda_k = n$. In this chapter, we relax the condition $\lambda_i - \lambda_{i+1} \geq 0$ to $\lambda_i - \lambda_{i+1} \geq r$ where $r$ can be negative. We also call these $r$-partitions. For example, $(2, 3, 1, 1)$ is a $(-1)$-partition of 7.

With the help with OEIS, we are able to find the generating function for the number of $(-1)$-partitions of the integer $n$. We present a short bijective proof of this and generalize the result to general $r$-partitions where $r$ is negative.

We then study "restricted" $r$-partitions with the first part and the number of parts fixed. This is the part where the fun of computer experimentation begins.

Let $a_r(M, N, n)$ be the number of $r$-partitions of $n$ with the first part equal to $M$ and exactly $N$ parts. Let $F(M, N, r, q)$ be the generating function for $a_r(M, N, n)$. Using a simple recurrence relation $F(M, N, r, q)$ satisfies, we program in Maple (using dynamical programming) to generate a specific $F(M, N, r, q)$ with input $M$, $N$, $r$ and $q$. Then we set $q = 1$ to explore the total number of $r$-partitions with the first part equal to $M$ and exactly $N$ parts.

Basically, we fix $N$ and range $M$, and use our program to conjecture a polynomial in $M$ for a fixed $N$, and then generalize. This will probably take notoriously long to do by hand, if possible–first we need to find the number sequence, which can easily result in error, then we need to guess the polynomial from a sequence of numbers, which is not at all easy to do by hand.

However, using Maple, a clean and nice pattern quickly emerges and before long, a "meta-pattern" emerges and we have a conjecture. We also prove the result using Maple by verifying that the conjectured formula satisfies the recurrence relation.

For the general $q$ case, we have not yet found an explicit formula for $F(M, N, r, q)$. However, by Maple experimentation, an intriguing relationship between the coefficients of a linear transformation of the Gaussian polynomials and $F(M, N, r, q)$ comes to light.

By further experimentation and the help of OEIS, we also discover connection between $F(M, N, -1, 1)$ with Catalan numbers and Young Tableau.

### 1.4 Systematic counting of restricted partitions and searching for new partition identities

As mentioned before, this chapter is the most "experimental", in the sense that computer is not just used as a valuable tool for conducting mathematical research, but is an indispensable copilot. First, we give a definition for "difference condition".

**Definition 1.1.** *A difference condition is a list* $a = [a_1, \ldots, a_r]$ *of length* $r \geq 1$ *of non-negative integers. A partition* $\lambda = (\lambda_1, \ldots, \lambda_k)$ **contains** *the difference condition* $a = [a_1, \ldots, a_r]$ *if there exists* $1 \leq i \leq k - r$ *such that*

$$\lambda_i - \lambda_{i+1} = a_1 \quad , \quad \lambda_{i+1} - \lambda_{i+2} = a_2 \quad , \quad \ldots \lambda_{i+r-1} - \lambda_{i+r} = a_r \quad .$$

A partition **avoids** the difference condition if it does not contain the difference condition. A partition avoids the set of difference conditions A, if it **avoids** every difference condition in A. For example, partitions whose adjacent parts differ by at least 2 is equivalent to partitions that avoid $\{[0],[1]\}$.

Our first goal of this chapter is to devise an efficient algorithm, that inputs an **arbitrary** set of difference conditions, $P$, and an arbitrary positive integer $N$, and outputs the first $N$ terms of the sequence enumerating partitions of $n$ avoiding the set of difference conditions $P$.

We use two approaches to attack this problem. One approach is to adapt the celebrated Goulden-Jackson ([8], [**?**]) method to this new context. The Goulden-Jackson method traditionally only deals with words. We extend this method to partitions. This approach (described in Section 5.2) is of considerable *theoretical* interest, but turns out to be less efficient than a more straightforward approach (described in Section 5.3), where the basic idea is to cut off the largest part in the partition and get a system of recurrences and use dynamical programming. This algorithm can be made quadratic in time and memory.

Next, in Section 5.4, we generalize this algorithm in order to deal with partitions with restrictions depending also on congruence conditions, for example, in Schur's celebrated 1926 theorem (see [An], p. 116), or the more complicated restrictions featuring in Shashank Kanade and Matthew C. Russell's intriguing conjectures ([KS], see also [S], pp. 149-152).

Finally, with this algorithm and help of Frank Garvan's q-series Maple package, we search over various parameter restrictions, using **Amarel cluster computing** `https://oarc.rutgers.edu/amarel/`. We have already found many seemingly new Rogers-Ramanujan type identities, and has generalized one of them to an infinite family. This is an ongoing research project, and while it is a long shot, we hope that the new identities that we discover will help us in uncovering the elusive big picture of where and why a certain type of partition identities exist.

# Chapter 2

# Words that contain the pattern 123 exactly once

We start with classical patterns. This chapter is adapted from the article:

## 2.1 Introduction

Recall that word $w = w_1...w_k$ is an ordered list of letters on some alphabet. To say a word contains a pattern (a certain permutation of $\{1, ..., m\}$) $\sigma$ is to say there exist $1 \leq i_1 < i_2 < ... < i_m \leq k$ such that the subword $w_{i_1}...w_{i_m}$ is *order isomorphic* to $\sigma$ (for example, 246 is order isomorphic to 123). A word avoids the pattern $\sigma$ if it does not contain it.

For a lucid history on the study of *forbidden patterns*, readers are welcome to refer to the introduction of Shar and Zeilberger's paper [SZ].

We say that a word $w$ in the alphabet $\{a_1, a_2, ..., a_n\}$ $(a_1 < a_2 < ... < a_n)$ is *associated* with the list $[l_1, ..., l_n]$ if $w$ has $l_i$ many $a_i$'s in it, for $i$ from 1 to $n$. For example, 231113233 is a word associated with the list $[3, 2, 4]$, and 223344 is a word associated with the list $[2, 2, 2]$. When not specified, our default alphabet will be $\{1, ..., n\}$ for some $n \geq 1$.

In the second section, we will generalize Zeilberger's bijective proof [Z1] (a shortened version of Alexander Burstein's elegant combinatorial proof [Bu]) that the number of permutations of $\{1, ..., n\}$ that contain the pattern 321 exactly once equals $\frac{3}{n}\binom{2n}{n+3}$ and apply it to words. Although no closed form formula was found, we have a summation whose summands are expressions involving enumeration of 123-avoiding words (for details, see Theorem 1).

In the third section, we will study, using ideas from the second section, how to extend Shar and Zeilberger's work [SZ] on generating functions enumerating 123-avoiding words (with $r$ occurrences of each letter) to words (with $r$ occurrences of each letter) having exactly one pattern 123. More precisely, for every positive integer $r$, Shar and Zeilberger found an algorithm for finding the defining algebraic equation for the ordinary generating function enumerating 123-avoiding words of length $rn$ where each of the $n$ letters of $\{1, 2, ..., n\}$ occurs exactly $r$ times.

We will present an algorithm for finding an analogue of that, that is, a defining algebraic equation for the ordinary generating function enumerating words of length $rn$ where each of the $n$ letters of $\{1, 2, ..., n\}$ occurs exactly $r$ times, now with exactly one pattern 123. We used the same (as in Shar and Zeilberger's paper [SZ]) memory-intensive, and exponential time, Buchberger's algorithm for finding Gröbner bases, and our computer (running Maple) found the defining algebraic equation for $r = 2$:

$$x^4\,(x+4)^2\,F^4 + 2\,x^3\,(x+4)\,(11\,x+23)\,F^3 - 4\,x\left(3\,x^4 - 10\,x^3 - 97\,x^2 - 146\,x + 1\right)F^2$$

$$+\left(-168\,x^4 - 840\,x^3 - 744\,x^2 + 336\,x - 24\right)F + 144\,x^3\,(x+2) = 0.$$

This took about a second. The minimal algebraic equation for $r = 3$ has 12 as the highest power for $F$ and the computation took about 20 seconds. Interested readers can find it on the website accompanying this chapter: `http://sites.math.rutgers.edu/~my237/One123`. The case when $r = 4$ already took too long to compute (more than a month).

Now, let $a_r(n)$ be the number of words of length $rn$ where each of the $n$ letters of $\{1, 2, ..., n\}$ occurs exactly $r$ times, with exactly one pattern 123. In the last section, we will use the Maple package **SCHUTZENBERGER** to derive recurrence relations for our sequences. Having obtained the defining algebraic equations of the generating functions for $a_r(n)$ in the cases $r = 2$ and $r = 3$, Manuel Kauers kindly helped us in finding the asymptotics for our sequences $a_2(n)$ and $a_3(n)$ (thanks to Kauers, the constants in front are fully rigorous and were computed via a step by step procedure; for details, please refer to [KP]):

$$a_2(n) = \frac{3(13-\sqrt{21})}{49} \cdot \frac{1}{\sqrt{\pi}} \cdot 12^n \cdot n^{-3/2} \cdot (1 + O(n^{-1})),$$

$$a_3(n) = \frac{-7+6\sqrt{7}}{56} \cdot \frac{1}{\sqrt{\pi}} \cdot 32^n \cdot n^{-3/2} \cdot (1+O(n^{-1})).$$

We noticed how similar these are to the asymptotics of the sequences enumerating 123-avoiding words with $r$ occcurences of each letter, given on page 8 of [SZ], and we have a similar conjecture as on page 3 of [SZ] (the Shar-Zeilberger conjecture was proved by Guillaume Chapuy [C]) that $a_r(n)$ was asymptotically $C_r \cdot ((r+1)2^r)^n \cdot n^{-3/2}$, where $C_r$ is a constant depending on $r$ (possibly $\frac{1}{\sqrt{\pi}}$ times a fraction of expressions involving square roots).

## 2.2 Enumeration of words that contain the pattern 123 exactly once

### 2.2.1 Zeilberger's shortened version of Bursteain's proof on permutations containing 321 exactly once

In a paper published in 2011 [Bu], Burstein gave an elegant combinatorial proof of John Noonan's theorem [N] that the number of permutations of $\{1, ..., n\}$ that contain the pattern 321 exactly once equals $\frac{3}{n}\binom{2n}{n+3}$. Zeilberger [Z1] was able to shorten Burstein's proof by using a bijection between a permutation with exactly one pattern 321, denoted as $\pi_1 c \pi_2 b \pi_3 a \pi_4$ $(a < b < c)$, with the pair $(\pi_1 b \pi_2 a, c \pi_3 b \pi_4)$ where $\pi_1 b \pi_2 a$ is a 321-avoiding permutation of $\{1, ..., b\}$ and $c\pi_3 b \pi_4$ is a 321-avoiding permutation of $\{b, ..., n\}$. Readers are encouraged to read Zeilberger's proof as a motivation and warm-up. Below we will see how we can use the same logic and apply it to words.

### 2.2.2 Extension to words

**Theorem 2.1.** *Let $A(l_1, ..., l_n)$ be the number of* 123*-avoiding words associated with the list $[l_1, ..., l_n]$. And let $B(l_1, ..., l_n)$ be the number of words associated with list $[l_1, ..., l_n]$ that contain the pattern 123 exactly once. Then we have*

$$B(l_1, ..., l_n) = \sum_{b=2}^{n-1} \sum_{j=0}^{l_b - 1} (A(l_1, ..., l_{b-1}, j+1) - A(l_1, ..., l_{b-1}, j)) \cdot (A(l_b - j, l_{b+1}, ..., l_n) - A(l_b - j - 1, l_{b+1}, ..., l_n)).$$

Before we start the proof, let us first define what a *good pair* of words is. Fix any $2 \le b \le n-1$ and a list $[l_1, ..., l_n]$. For any $0 \le j \le l_b - 1$ , the pair of words $(\sigma_1, \sigma_2)$ is *good* if $\sigma_1$ is a 123-avoiding word in $\{1, ..., b\}$ associated with the list $[l_1, ..., l_{b-1}, j+1]$ that does not start with $b$ and $\sigma_2$ is a 123-avoiding word in $\{b, ..., n\}$ associated with the list $[l_b - j, l_{b+1}, ..., l_n]$ that does not end with $b$. For example, if $[l_1, ..., l_n] = [2, 2, 2, 2]$, and $b = 2$, then $(112, 422433)$ is a good pair. We will also say $\sigma_i$ $(i = 1, 2)$ is good if it belongs to a good pair $(\sigma_1, \sigma_2)$ (Note the definitions for $\sigma_1$ and $\sigma_2$ to be good are different, but things will be made clear in context).

*Proof.* Any word $w$ associated with the list $[l_1, l_2, ..., l_n]$ with exactly one pattern 123 can be written as $\pi_1 a \pi_2 b \pi_3 c \pi_4$ $(a < b < c)$, where $abc$ is the unique 123 pattern. All entries to the left of $b$, except $a$, must be greater than or equal to $b$, and all the entries to the right of $b$, except for $c$, must be smaller than or equal to $b$. Also, $\pi_2$ and $\pi_3$ must not contain any $b$'s, otherwise there will be another 123 pattern. Observe that $a\pi_3 b\pi_4$ is a word in $\{1, 2, ..., b\}$ avoiding pattern 123 and does not start with $b$ and $\pi_1 b \pi_2 c$ is a word in $\{b, b+1, ..., n\}$ avoiding pattern 123 and does not end with $b$. Therefore $(a\pi_3 b\pi_4, \pi_1 b\pi_2 c)$ is a good pair.

We now verify that there is indeed a bijection from the set of words having exactly one pattern 123 to the set of good pairs $(\sigma_1, \sigma_2)$ (for $2 \le b \le n-1$ and $0 \le j \le l_b - 1$).

Fix $b$ and $j$ $(2 \le b \le n-1,\ 0 \le j \le l_b - 1)$. Given a word $\pi_1 a \pi_2 b \pi_3 c \pi_4$ (that has exactly one 123 pattern: $abc$), we can easily map it to a unique good pair $(a\pi_3 b\pi_4, \pi_1 b \pi_2 c)$ by first finding out what $a$, $c$ are. This is easy since we have only one 123 pattern. Conversely, given a good pair $(\sigma_1, \sigma_2)$ , we take the first letter of $\sigma_1$ as "$a$" and the leftmost occurrence of $b$ as "$b$" and get $\pi_3$ and $\pi_4$ $(\sigma_1 = a\pi_3 b\pi_4)$. Similarly, we take the last letter of $\sigma_2$ as "$c$" and the rightmost occurrence of $b$ as "$b$" and get $\pi_1$ and $\pi_2$ $(\sigma_2 = \pi_1 b \pi_2 c)$. Putting everything together we get a unique $\pi_1 a \pi_2 b \pi_3 c \pi_4$.

Now, for any $b$ and $j$, the number of good $\sigma_1$ is $A(l_1, ..., l_{b-1}, j+1) - A(l_1, ..., l_{b-1}, j)$ and the number of good $\sigma_2$ is $A(l_b - j, l_{b+1}, ..., l_n) - A(l_b - j - 1, l_{b+1}, ..., l_n)$. Therefore the number of words $\pi_1 a \pi_2 b \pi_3 c \pi_4$ with exactly one pattern 123 is: $(A(l_1, ..., l_{b-1}, j+1) - A(l_1, ..., l_{b-1}, j)) \cdot (A(l_b - j, l_{b+1}, ..., l_n) - A(l_b - j - 1, l_{b+1}, ..., l_n))$. Summing over all $b$ and $j$, we get the desired result.

□

**Corollary 2.2.** $B(l_1, ..., l_n) = B(l_n, ..., l_1)$.

*Proof.* By Theorem 1, we have

$$B(l_1,...,l_n) = \sum_{b=2}^{n-1}\sum_{j=0}^{l_b-1}(A(l_1,...,l_{b-1},j+1) - A(l_1,...,l_{b-1},j)) \cdot(A(l_b-j,l_{b+1},...,l_n)-A(l_b-j-1,l_{b+1},...,l_n)) \quad (1)$$

and

$$B(l_n,...,l_1) = \sum_{b=2}^{n-1}\sum_{j=0}^{l_{n-b+1}-1}(A(l_n,...,l_{n-b+2},j+1) - A(l_n,...,l_{n-b+2},j)) \cdot(A(l_{n-b+1}-j,l_{n-b},...,l_1)-A(l_{n-b+1}-j-1,l_{n-b},...,l_1)). \quad (2)$$

When $b = k$ $(2 \leq k \leq n-1)$, the inner sum of (1) becomes

$$\sum_{j=0}^{l_k-1}(A(l_1,...,l_{k-1},j+1) - A(l_1,...,l_{k-1},j)) \cdot(A(l_k-j,l_{k+1},...,l_n)-A(l_k-j-1,l_{k+1},...,l_n)) \quad (3)$$

while when $b = n-k+1$ $(2 \leq k \leq n-1$, notice this is the "symmetric counterpart" of $b = k)$, the inner sum of (2) becomes

$$\sum_{j=0}^{l_k-1}(A(l_n,...,l_{k+1},j+1) - A(l_n,...,l_{k+1},j)) \cdot(A(l_k-j,l_{k-1},...,l_1)-A(l_k-j-1,l_{k-1},...,l_1)). \quad (4)$$

We only need to show (3) = (4) in order to show (1) = (2). Now notice that when $j = t$ $(0 \leq t \leq l_k - 1)$, the summand of (3) is

$$(A(l_1,...,l_{k-1},t+1)-A(l_1,...,l_{k-1},t))\cdot(A(l_k-t,l_{k+1},...,l_n)-A(l_k-t-1,l_{k+1},...,l_n)) \quad (5)$$

and when $j = l_k - 1 - t$ (the "symmetric counterpart" of $j = t$), the summand of (4) is

$$(A(l_n,...,l_{k+1},l_k-t)-A(l_n,...,l_{k+1},l_k-t-1))\cdot(A(t+1,l_{k-1},...,l_1)-A(t,l_{k-1},...,l_1)). \quad (6)$$

After a small rearrangement, we can see (5) = (6) because of an important result that $A(l_1, ..., l_n)$ is symmetric in its arguments (this is not true in general for $B(l_1, ..., l_n)$; for details of this result, see [SZ], page 4). Therefore as $j$ ranges from 0 to $l_k - 1$, we have (3) = (4). And as $b$ ranges from 0 to $n-1$, we have (1) = (2).

□

**Corollary 2.3.** *Fix a list* $L := [l_1, ..., l_n]$*. The number of words associated with* $L$ *that contain exactly one pattern 123 (i.e.,* $B(l_1, ..., l_n)$*) is equal to the number of words associated with* $L$ *that contain exactly one pattern 321.*

*Proof.* Let $S_1$ be the set of words associated with $[l_1, ..., l_n]$ that contain exactly one pattern 123 and $S_2$ be the set of words associated with $[l_n, ..., l_1]$ that contain exactly one pattern 321. Take any $w_1 \in S_1$, we can map it to a word $w_2$ associated with $[l_n, ..., l_1]$ by mapping letter $i$ to letter $n-i+1$ (for all $i$ from 1 to $n$). For example, the word 121322 is mapped to 323122. Observe that $w_2$ must contain exactly one pattern 321, which occurs at the same location in $w_2$ as the location of the 123 pattern in $w_1$. Therefore $w_2 \in S_2$. Clearly this is a bijection from $S_1$ to $S_2$. So $|S_1| = |S_2|$. This along with 2.2 gives 2.3. □

**Remark 1.** *One may wonder if the number of words associated with* $[l_1, ..., l_n]$ *that contain exactly one pattern 123 is equal to the number of words associated with* $[l_1, ..., l_n]$ *that contain exactly one pattern 132. This is not the case (if this were the case, we would have an analogue of the result that the 123-avoiding words associated with* $[l_1, ..., l_n]$ *are equinumerous with the 132-avoiding words associated with* $[l_1, ..., l_n]$*, see [Z2]). For example, the number of permutations of* $\{1, 2, ..., n\}$ $(n \geq 1)$ *that contain exactly one pattern 123 is* $\frac{3}{n}\binom{2n}{n+3}$ *[Z1] while the number of permutations of* $\{1, 2, ..., n\}$ $(n \geq 1)$ *that contain exactly one pattern 132 is* $\binom{2n-3}{n-3}$ *[Bó].*

## 2.3 Generating functions

### 2.3.1 Some crucial background and generating functions for words avoiding pattern 123

In the beautiful paper by Shar and Zeilberger [SZ], methods for finding the algebraic equation for the ordinary generating function enumerating 123-avoiding words of length $rn$, where each of the $n$ letters of $\{1, 2, ..., n\}$ occurs exactly $r$ times were given. First we present some important definitions and results of that paper here.

For $0 \leq i \leq j \leq r-1$ and $n \geq 0$, let $W_r^{(i,j)}(n)$ be the set of 123-avoiding words of length $rn+i+j$, in the alphabet $\{1, 2, ..n, n+1, n+2\}$, with $i$ occurrences of the letter 1, $j$ occurrences of the $n+2$, and exactly $r$ occurrences of the other $n$ letters. And let $W_r^{(i,j)}$ be the union of $W_r^{(i,j)}(n)$ over all $n \geq 0$. Let $g_r^{(i,j)}(x)$ be the *weight enumerator* for $W_r^{(i,j)}$, with respect to the weight $w \to x^{length(w)}$. (Note that the $W_r^{(i,j)}$'s have the same weight enumerator if any two letters have $i$ and $j$ occurrences respectively, and the remaining letters each occurs exactly $r$ times. For a detailed explanation, see [SZ], page 3-4.)

Shar and Zeilberger were able to find a system of $\binom{r+1}{2}$ equations for $g_r^{(i,j)}(x)$ ($0 \leq i \leq j \leq r-1$), with the convention that if $s > k$ then $g_r^{(s,k)} = g_r^{(k,s)}$:

$$g_r^{(i,j)}(x) = \delta_{i,0}\delta_{j,0} + x \sum_{t=0}^{r-1} g_r^{(i,t)}(x) g_r^{((r-t) \bmod r,\ (j-1) \bmod r)}(x) + \sum_{m=0}^{i-1} x^{m+1} g_r^{(i-m,j-1)}(x)$$

where

$$\delta_{i,j} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} .$$

For example, in the case when $r = 2$, we would get the following system of equations:

$$g_2^{(0,0)}(x) = 1 + x g_2^{(0,0)}(x) g_2^{(0,1)}(x) + x g_2^{(0,1)}(x) g_2^{(1,1)}(x)$$
$$g_2^{(0,1)}(x) = x g_2^{(0,0)}(x)^2 + x g_2^{(0,1)}(x)^2$$
$$g_2^{(1,1)}(x) = x g_2^{(0,0)}(x) g_2^{(0,1)}(x) + x g_2^{(0,1)}(x)(1 + g_2^{(1,1)}(x))$$

Solving this system of equations in the three unknowns $g_2^{(0,0)}(x), g_2^{(0,1)}(x), g_2^{(1,1)}(x)$, we get the weight enumerators for $W_2^{(0,0)}$, $W_2^{(0,1)}$ and $W_2^{(1,1)}$.

Once we have the weight enumerators, we can easily get the corresponding generating functions by doing a little operation. For example, because we have an explicit expression for $g_2^{(0,0)}(x)$ ($g_2^{(0,0)}(x) = 1 + x^2 + 6x^4 + 43x^6 + 352x^8 + 3114x^{10} + ...$), the corresponding generating function is $f_2^{(0,0)}(x) = 1+x+6x^2+43x^3+352x^4+3114x^5+...$ (that is, $f_2^{(0,0)}(x) = g_2^{(0,0)}(x^{1/2})$).

### 2.3.2 Extension to generating functions for words with exactly $r$ occurrences of each letter, and with exactly one pattern 123

**Definition 2.4.** *Let $V_r(n)$ be the set of words in the alphabet $\{1, ..., n\}$ with exactly $r$ occurrences of each letter, and with exactly one pattern 123. Let $V_r$ be $\bigcup_{n=0}^{\infty} V_r(n)$.*

*Let $h_r(x)$ be the weight enumerator for $V_r$ (as always, with weight $w \to x^{length(w)}$) and let $f_r(x)$ be the corresponding generating function.*

**First warm-up:** $r = 1$

Claim: $h_1(x) = (g_1^{(0,0)}(x) - xg_1^{(0,0)}(x) - 1)^2/x$.

*Proof.* Recall that $g_1^{(0,0)}(x)$ ($= 1 + x + 2x^2 + 5x^3 + 14x^4 + 42x^5 + ...$) is the weight enumerator for 123-avoiding permutations on $\{1, ..., n, ...\}$. We prove this claim by showing that the coefficient of $x^n$ ($n \geq 0$) on the right hand side is exactly the number of good pairs $(a\pi_3 b\pi_4, \pi_1 b\pi_2 c)$ $(2 \leq b \leq n-1)$, which equals to $B(1, 1, ..., 1)$ (with $n$ 1's) (by Zeilberger's proof [Z1]).

For any fixed $b$ $(2 \leq b \leq n-1)$, a good $a\pi_3 b\pi_4$ would be a 123-avoiding permutation on $\{1, ..., b\}$ that does not start with $b$. Similarly, a good $\pi_1 b\pi_2 c$ would be a 123-avoiding permutation on $\{b, ..., n\}$ that does not end with $b$.

Note that the coefficient of $x^b$ in $g_1^{(0,0)}(x) - xg_1^{(0,0)}(x) - 1$ is exactly the number of good $a\pi_3 b\pi_4$. (The $x$ in front of $g_1^{(0,0)}(x)$ corresponds to having $b$ in front of a permutation, and the $-1$ corresponds to an empty permutation. We don't want either of these.)

Similarly, the coefficient of $x^{n-b+1}$ in $g_1^{(0,0)}(x) - xg_1^{(0,0)}(x) - 1$ is the number of good $\pi_1 b\pi_2 c$. Multiplying the two, we have that the coefficient of $x^{n+1}$ $(= x^b \cdot x^{n-b+1})$ in

$$(g_1^{(0,0)}(x) - xg_1^{(0,0)}(x) - 1)^2$$

is the number of good pairs $(a\pi_3 b\pi_4, \pi_1 b\pi_2 c)$ ($b$ ranges from 2 to $n-1$). Dividing by $x$, we get the coefficient of $x^n$ in

$$(g_1^{(0,0)}(x) - xg_1^{(0,0)}(x) - 1)^2/x$$

is the number of good pairs $(a\pi_3 b\pi_4, \pi_1 b\pi_2 c)$.

□

**Second warm-up:** $r = 2$

Claim: $h_2(x) = 2 \cdot (g_2^{(0,0)}(x) - xg_2^{(0,1)}(x) - 1)(g_2^{(0,1)}(x) - xg_2^{(0,0)}(x))/x$.

*Proof.* Recall that $g_2^{(0,0)}(x)$ $(= 1 + x^2 + 6x^4 + 43x^6 + ...)$ is the weight enumerator for 123-avoiding words on $\{1, 1, ..., n, n, ...\}$ (or equivalently, 123-avoiding words associated with $[2, 2, ...]$) and $g_2^{(0,1)}(x)$ $(= x + 3x^3 + 19x^5 + 145x^7 ...)$ is the weight enumerator for 123-avoiding words on $\{1, 2, 2, 3, 3, ..., n, n, ...\}$. As in the first warm-up, we prove this claim by showing that the coefficient of $x^{2n}$ $(n \geq 0)$ on the right hand side is exactly the number of good pairs $(a\pi_3 b\pi_4, \pi_1 b\pi_2 c)$ $(2 \leq b \leq n-1)$, which equals to $B(2, 2, ..., 2)$ (with n 2's)(by the proof of Theorem 1).

For any $b$ $(2leqb \leq n-1)$, we have the following two cases: either $\pi_4$ contains one $b$ and $\pi_1$ contains no $b$ or the other way around.

**Case 1**: $\pi_4$ contains one $b$ and $\pi_1$ contains no $b$.

Then a good $a\pi_3 b\pi_4$ would be a 123-avoiding word on $\{1, 1, ..., b, b\}$ that does not start with $b$. Similarly, a good $\pi_1 b\pi_2 c$ would be a 123-avoiding word on $\{b, b+1, b+1, ..., n, n\}$ that does not end with $b$.

Note that the coefficient of $x^{2b}$ in $g_2^{(0,0)}(x) - xg_2^{(0,1)}(x) - 1$ is exactly the number of good $a\pi_3 b\pi_4$. (The $x$ in front of $g_2^{(0,1)}(x)$ corresponds to having $b$ in front of a word, and the $-1$ corresponds to an empty word. We don't want either of these.)

Similarly, the coefficient of $x^{2(n-b)+1}$ in $g_2^{(0,1)}(x) - xg_2^{(0,0)}(x)$ is the number of good $\pi_1 b\pi_2 c$.

Multiplying the two, and let $b$ range from 2 to $n-1$, we have that the coefficient of $x^{2n+1}$ ($= x^{2b} \cdot x^{2(n-b)+1}$) in

$$(g_2^{(0,0)}(x) - xg_2^{(0,1)}(x) - 1)(g_2^{(0,1)}(x) - xg_2^{(0,0)}(x))$$

is the number of good pairs $(a\pi_3 b\pi_4, \pi_1 b\pi_2 c)$ if the additional $b$ is in $\pi_4$. Dividing by $x$, we get the coefficient of $x^{2n}$ in

$$(g_2^{(0,0)}(x) - xg_2^{(0,1)}(x) - 1)(g_2^{(0,1)}(x) - xg_2^{(0,0)}(x))/x$$

is the number of good pairs $(a\pi_3 b\pi_4, \pi_1 b\pi_2 c)$ if the additional $b$ is in $\pi_4$.

**Case 2**: $\pi_1$ contains one $b$ and $\pi_4$ contains no $b$.

Then a good $a\pi_3 b\pi_4$ would be a 123-avoiding word on $\{1, 1, ..., b-1, b-1, b\}$ that does not start with $b$. A good $\pi_1 b\pi_2 c$ would be a 123-avoiding word on $\{b, b, ..., n, n\}$ that does not end with $b$.

Now, the coefficient of $x^{2b-1}$ in $g_2^{(0,1)}(x) - xg_2^{(0,0)}(x)$ is exactly the number of good $a\pi_3 b\pi_4$. Similarly, the coefficient of $x^{2(n-b)+2}$ in $g_2^{(0,0)}(x) - xg_2^{(0,1)}(x) - 1$ is the number of good $\pi_1 b\pi_2 c$.

Multiplying the two, we have that the coefficient of $x^{2n+1}$ ($= x^{2b-1} \cdot x^{2(n-b)+2}$) in

$$(g_2^{(0,1)}(x) - xg_2^{(0,0)}(x))(g_2^{(0,0)}(x) - xg_2^{(0,1)}(x) - 1)$$

is the number of good pairs $(a\pi_3 b\pi_4, \pi_1 b\pi_2 c)$ ($b$ ranges from 2 to $n-1$) if the additional $b$ is in $\pi_1$. Dividing by $x$, we get the coefficient of $x^{2n}$ in

$$(g_2^{(0,1)}(x) - xg_2^{(0,0)}(x))(g_2^{(0,0)}(x) - xg_2^{(0,1)}(x) - 1)/x$$

is the number of good pairs $(a\pi_3 b\pi_4, \pi_1 b\pi_2 c)$ if the additional $b$ is in $\pi_1$.

Therefore the coefficient of $x^{2n}$ in $2 \cdot (g_2^{(0,0)}(x) - xg_2^{(0,1)}(x) - 1)(g_2^{(0,1)}(x) - xg_2^{(0,0)}(x))/x$ is the number of good pairs $(a\pi_3 b\pi_4, \pi_1 b\pi_2 c)$, which is equal to $B(2, 2, ..., 2)$ (with n 2's). Note that the coefficient of $x^{2n+1}$ in $2 \cdot (g_2^{(0,0)}(x) - xg_2^{(0,1)}(x) - 1)(g_2^{(0,1)}(x) - xg_2^{(0,0)}(x))/x$ is 0 because $g_2^{(0,1)}(x)$ has only odd powers of $x$ and $g_2^{(0,0)}(x)$ has only even powers of

$x$. So we have shown the weight enumerator for $V_2$ is as claimed to be. To get the generating function $f_2(x)$ for $V_2$ we simply let $f_2(x) = h_2(x^{1/2})$. □

Readers are welcome to compare $h_2(x)$ with the earlier formula in the case when $l_i = 2(1 \le i \le n)$:

$$\sum_{b=2}^{n-1}\sum_{j=0}^{1}(\underbrace{A(j+1,2,2,...,2)}_{b-1 \text{ many 2's}} - \underbrace{A(j,2,2,...,2)}_{b-1 \text{ many 2's}}) \cdot (\underbrace{A(2-j,2,2,...,2)}_{n-b \text{ many 2's}} - \underbrace{A(1-j,2,2,...,2)}_{n-b \text{ many 2's}}).$$

**The general case**

**Theorem 2.5.**

$$h_r(x) = \frac{1}{x}\sum_{i=1}^{r}(g_r^{(0,\ i\ mod\ r)} - x g_r^{(0,\ i-1)} - \delta_{(i\ mod\ r,\ 0)})(g_r^{(0,\ (r+1-i)\ mod\ r)} - x g_r^{(0,\ r-i)} - \delta_{((r+1-i)\ mod\ r,\ 0)}).$$

The general case is derived using the exact same logic as for the warm-up cases. Instead of having two cases as in the second warm-up, here we have $r$ cases. Interested readers are welcome to verify the formula for $r = 3$ by themselves, and the general case should be apparent after this verification. As before, to get the generating function for $V_r$ we simply let $f_r(x) = h_r(x^{1/r})$.

## 2.4 Using Maple packages

As noted in Shar and Zeilberger's paper ([SZ], page 7), now that we know $f_r(x)$ has the property of being algebraic, the sequence $a_r(n)$ satisfies some homogeneous linear recurrence equation with polynomial coefficients.

Using the *algtorec* procedure in the **SCHUTZENBERGER** package written by Doron Zeilberger, available from:

`http://www.math.rutgers.edu/~zeilberg/tokhniot/SCHUTZENBERGER.txt`

we are able to find (rigorously) recurrences (in operator notation) for our sequences when $r = 1$ and $r = 2$ ($r = 3$ took too long to compute):

For $r = 1$ we get: $(2\,n\,(2\,n+1) - (n+4)\,(n-2)\,N)a_1(n) = 0$ (which agrees with the already known formula $a_1(n) = \frac{3}{n}\binom{2n}{n+3}$).

In the case when $r = 2$, *algtorec* returned a operator of degree 8, but it can be reduced to a minimal operator of degree 4 (thanks to Manuel Kauers for pointing it out), that is:

$$(36(1+n)(2+n)(1+2n)(3+2n)(18154800+23101940n+10635771n^2+2093616n^3+147833n^4)$$
$$+12(2+n)(3+2n)(1283329440+3700267618n+4200957553n^2+2408049238n^3+735936616n^4$$
$$+113774584n^5+6948151n^6)N+(282564806400+1066356868608n+1704365727480n^2$$
$$+1511140337906n^3+814587362081n^4+273775889012n^5+56080140110n^6+6405068474n^7$$
$$+312371129n^8)N^2-2(4+n)(11939685120+40890299130n+56943840213n^2+41794221496n^3$$
$$+17488032270n^4+4183030930n^5+531527997n^6+27792604n^7)N^3+8(1+n)(4+n)(5+n)(11$$
$$+2n)(3742848+7519914n+5241921n^2+1502284n^3+147833n^4)N^4)a_2(n)=0$$

Here are some initial terms of $f_2(n)$ (i.e., the generating function for $a_2(n)$): $f_2(n) = 12\,x^3+174\,x^4+2064\,x^5+23082\,x^6+252966\,x^7+2755332\,x^8+30001026\,x^9+327381492\,x^{10}+\cdots$.

(We can easily get many more terms.)

Everything in this chapter is implemented (with explanation) in the Maple packages **Words123New** and **PW123** and available from: `http://sites.math.rutgers.edu/~my237/One123`, which also includes some sample input and output files.

# Chapter 3

# Increasing consecutive patterns in words

In this chapter, we move from classical patterns to consecutive patterns. This chapter is adapted from the article:

## 3.1 Introduction

Simion and Wilf initiated the study of enumerating *classical* pattern-avoidance. This is a very dynamic area with its own annual conference.

Recall that a permutation $\pi = \pi_1 \cdots \pi_n$ avoids a pattern $\sigma = \sigma_1 \cdots \sigma_k$ if none of the $\binom{n}{k}$ length-$k$ subsequences of $\pi$, reduces to $\sigma$.

Burstein [2], in a 1998 PhD thesis, under the direction of Wilf, pioneered the enumeration of *words* avoiding a set of patterns. This field is also fairly active today, with notable contributions by, *inter alia*, Mansour [3] and Pudwell [12].

The enumeration of permutations avoiding a given (classical) pattern, or a set of patterns, is notoriously difficult, and it is widely believed to be intractable for most patterns, hence it would be nice to have other notions for which the enumeration is more feasible. Such an analog was given, in 2003, by Elizalde and Noy, in a seminal paper [5], that introduced the study of the enumeration of permutations avoiding *consecutive* patterns. A permutation $\pi = \pi_1 \cdots \pi_n$ avoids a consecutive pattern $\sigma = \sigma_1 \cdots \sigma_k$ if none of the $n - k + 1$ length-$k$ consecutive subwords, $\pi_i \pi_{i+1} \cdots \pi_{i+k-1}$ of $\pi$, reduces to $\sigma$.

Algorithmic approaches to the enumeration of *permutations* avoiding sets of consecutive patterns were given by Nakamura, Baxter, and Zeilberger [10, 1]. Our present approach may be viewed as an extension, from permutations to words, of Nakamura's paper, who was also inspired by the Goulden-Jackson cluster method, but in a sense, is more straightforward, and closer in spirit to the original Goulden-Jackson cluster method ([8], that is beautifully exposited (and extended!) in [11]).

In this chapter we will consider consecutive patterns of the form $1 \cdots r$, i.e. *increasing consecutive patterns*, and show how to count words in $1^{m_1} \cdots n^{m_n}$ avoiding the pattern $1 \cdots r$ (Theorem 3.1, that is due to Ira Gessel [6]). Throughout this chapter we will only consider consecutive patterns, so the word "consecutive" may be omitted. In particular, we will look at how to efficiently count words in $1^s \cdots n^s$ avoiding the pattern $1 \cdots r$. All the sequences for $s = 1$ and $3 \leq r \leq 9$ are in the On-Line Encyclopedia of Integer Sequences, with many terms. Also, quite a few of theses sequences for $s > 1$ are already there, but with very few terms. Our implied algorithms are $O(n^{s+1})$ and hence yield many more terms, and, of course, new sequences.

In the last part of the paper, we will provide a new proof of Theorem 3.1 by tweaking the Goulden-Jackson cluster method. Using this proof, along with a little more effort, we will generalize Theorem 3.1 to counting words with a specified number of the pattern $12 \cdots r$ (Theorem 3.3), instead of just *avoiding*, that is, having *zero* occurrence of the pattern of interest.

We close this introduction by mentioning the pioneering work of Mendes and Remmel [9], in combining the two keywords "consecutive patterns" and "words". We were greatly inspired by their article, but our focus is algorithmic.

**Maple Packages**: This chapter is accompanied by three Maple packages available from the webpage:

`http://sites.math.rutgers.edu/~zeilberg/mamarim/mamarimhtml/icpw.html` .

These are

• `ICPW.txt`: For fast enumeration of sequences enumerating words avoiding increasing consecutive patterns.

• `ICPWt.txt`: For fast computation of sequences of weight-enumerators for words according to the number of increasing consecutive patterns ($t = 0$ reduces to the former case).

• `GJpats.txt`: For conjecturing generating functions (that still have to be proved by humans).

This page also has links to numerous input and output files. The input files can be modified to generate more data, if desired.

## 3.2 Method, experimentation, and results

### 3.2.1 The Goulden-Jackson cluster method

Recall that the original Goulden-Jackson method [8, 11] inputs a *finite* alphabet, $A$, that may be taken to be $\{1, \ldots, n\}$, and a finite set of "bad words", $B$.

It outputs a certain **rational function**, let us call it $F(x_1, \ldots, x_n)$, that is the multi-variable generating function, in $x_1, \ldots, x_n$, for the discrete $n$-variable function

$$f(m_1, \ldots, m_n) \quad ,$$

that counts the words in $1^{m_1} \cdots n^{m_n}$ (there are altogether $(m_1 + \cdots + m_n)!/(m_1! \cdots m_n!)$ of them) that **never** contain as *consecutive* subwords (aka *factors* in linguistics) any member of $B$. In other words:

$$F(x_1, \ldots, x_n) = \sum_{(m_1, \ldots, m_n) \in N^n} f(m_1, \ldots, m_n)\, x_1^{m_1} \cdots x_n^{m_n} \quad .$$

This is nicely implemented in the Maple package `DavidIan.txt`, that accompanies [11], and is freely available from

`http://sites.math.rutgers.edu/~zeilberg/tokhniot/DavidIan.txt` .

For example, let $n = 4$, so the alphabet is $\{1, 2, 3, 4\}$, and let the set of "bad words" to avoid be $\{1234, 1432\}$. Starting a Maple session and typing

```
read `DavidIan.txt`:  lprint(subs(t=0,GJgf(1,2,3,4,[1,2,3,4],[1,4,3,2],x,t)));
```

immediately returns

```
1/(1-x[1]-x[2]-x[3]-x[4]+ 2*x[1]*x[2]*x[3]*x[4])
```
,

that in Human language reads

$$\frac{1}{1 - x_1 - x_2 - x_3 - x_4 + 2\, x_1 x_2 x_3 x_4} \quad .$$

### 3.2.2 Enuerating words avoiding consecutive patterns: let the computer do the guessing

*Now* we are interested in words in an *arbitrarily large* alphabet $\{1, \ldots, n\}$ avoiding a set of consecutive patterns, but each pattern, e.g., 123, entails an *arbitrarily large* set of forbidden consecutive subwords. For example, in this case, the set of forbidden consecutive subwords is

$$\{i_1\, i_2\, i_3 \,|\, 1 \leq i_1 < i_2 < i_3 \leq n\} \quad .$$

We can ask `DavidIan.txt` to find the generating function for each specific $n$, and then hope to conjecture a **general** formula in terms of $x_1, \ldots, x_n$, for *general* (i.e., **symbolic**) $n$.

This is accomplished by the Maple package `GJpats.txt`, available from the webpage of this chapter. It uses the original `DavidIan.txt` to produce the corresponding generating functions for increasing values for $n$, and then attempts to conjecture a *meta-pattern*. For example for words avoiding the consecutive pattern 123 (alias the word 123), for $n = 3$,

`GFpats(`$\{[1,2,3]\}, x, 3, 0$`);` (the 0 stands for having zero occurrences of (i.e., avoiding) the pattern of interest) yields

$$1/(1 - x_1 - x_2 - x_3 + x_1x_2x_3) \quad .$$

This is simple enough. Moving right along,

`GFpats(`$\{[1,2,3]\}, x, 4, 0$`);` yields

$$1/(1 - x_1 - x_2 - x_3 - x_4 + x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4 - x_1x_2x_3x_4) \quad ,$$

while `GFpats(`$\{[1,2,3]\}, x, 5, 0$`);` yields

$$1/(1 - x_1 - x_2 - x_3 - x_4 - x_5 + x_1x_2x_3 + x_1x_2x_4 + x_1x_2x_5 + x_1x_3x_4 + x_1x_3x_5 + x_1x_4x_5 + x_2x_3x_4 + x_2x_3x_5 + x_2x_4x_5 + x_3x_4x_5 - x_1x_2x_3x_4 - x_1x_2x_3x_5 - x_1x_2x_4x_5 - x_1x_3x_4x_5 - x_2x_3x_4x_5) \quad .$$

These look like *symmetric* functions. Procedure `SPtoM(P,x,n,m)` expresses a polynomial, P, in the indexed variables $x[1], \ldots, x[n]$ in terms of the *monomial symmetric polynomials* $m_\lambda$. Applying this procedure we have

`SPtoM(denom(GFpats(`$\{[1,2,3]\},x,5,0)),x,5,m$`);` yields

`-m[1, 1, 1, 1] + m[1, 1, 1] - m[1] + m[]` .

`SPtoM(denom(GFpats(`$\{[1,2,3]\},x,6,0)),x,6,m$`);` yields

`m[1,1,1,1,1,1]-m[1,1,1,1]+m[1,1,1]-m[1]+m[]` .

`SPtoM(denom(GFpats(`$\{[1,2,3]\},x,7,0)),x,7,m$`);` yields

`-m[1,1,1,1,1,1,1]+m[1,1,1,1,1,1]-m[1,1,1,1]+m[1,1,1]-m[1]+m[]` .

You do not have to be a Ramanujan to conjecture the following result.

**Fact:** The generating function for words in $\{1, 2, \ldots, n\}$ avoiding the consecutive pattern 123, let us call it $F_3(x_1, \ldots, x_n)$ is

$$F_3(x_1, \ldots, x_n) = \frac{1}{1 - e_1 + e_3 - e_4 + e_6 - e_7 + e_9 - e_{10} + \cdots} \quad ,$$

where $e_i$ stands for the *elementary symmetric function* of degree $i$ in $x_1, \ldots, x_n$, i.e., the coefficient of $z^i$ in $(1 + x_1 z) \cdots (1 + x_n z)$. (Note that $e_i = m_{1^i}$.)

Doing the analogous guessing for the consecutive patterns 1234 and 12345, a *meta-pattern* emerges, and we were safe in formulating the following theorem that we discovered using the present experimental mathematics approach. After the first version of this chapter was posted, we found out, thanks to Justin Troyka, that this theorem is due to Ira Gessel [6, p. 51].

**Theorem 3.1.** *(Gessel [6]) For $n \geq 1, r \geq 2$, the generating function for words in $\{1, 2, \ldots, n\}$ avoiding the consecutive pattern $12 \cdots r$, let us call it $F_r(x_1, \ldots, x_n)$ is*

$$F_r(x_1, \ldots, x_n) = \frac{1}{1 - e_1 + e_r - e_{r+1} + e_{2r} - e_{2r+1} + e_{3r} - e_{3r+1} + \cdots} \quad .$$

Of course, if Gessel did not prove it before us, these would have been "only" guesses, but once known, humans can prove them. We did it by tweaking the cluster method to apply to an *arbitrarily large* alphabet, i.e. where even the size of the alphabet, $n$, is *symbolic*. Our proof of Gessel's theorem will be given at the end of this chapter.

### 3.2.3 Efficient computations

Theorem 3.1 immediately implies the following partial recurrence equation for the actual coefficients.

**Fundamental Recurrence**: Let $f_r(\mathbf{m})$ be the number of words in the alphabet $\{1,\ldots,n\}$ with $m_1$ 1's, $m_2$ 2's, …, $m_n$ $n$'s (where we abbreviate $\mathbf{m}=(m_1,\ldots,m_n)$) that avoid the consecutive pattern $1\cdots r$. Also let $V_i$ be the set of $0-1$ vectors of length $n$ with $i$ ones, then

$$
\begin{aligned}
f_r(\mathbf{m}) = \sum_{\mathbf{v}\in V_1} f_r(\mathbf{m}-\mathbf{v}) \; &- \sum_{\mathbf{v}\in V_r} f_r(\mathbf{m}-\mathbf{v}) \\
+ \sum_{\mathbf{v}\in V_{r+1}} f_r(\mathbf{m}-\mathbf{v}) \; &- \sum_{\mathbf{v}\in V_{2r}} f_r(\mathbf{m}-\mathbf{v}) \\
+ \sum_{\mathbf{v}\in V_{2r+1}} f_r(\mathbf{m}-\mathbf{v}) \; &- \sum_{\mathbf{v}\in V_{3r}} f_r(\mathbf{m}-\mathbf{v}) \\
+ \sum_{\mathbf{v}\in V_{3r+1}} f_r(\mathbf{m}-\mathbf{v}) \; &- \sum_{\mathbf{v}\in V_{4r}} f_r(\mathbf{m}-\mathbf{v}) + \cdots \quad .
\end{aligned}
$$

(Readers can check this derivation by multiplying each side of the equation in Theorem 3.1 by the denominator of the right hand side and then using the fact that $F_r(x_1,\ldots,x_n)=\sum_{(m_1,\ldots,m_n)\in N^n} f_r(m_1,\ldots,m_n)\, x_1^{m_1}\cdots x_n^{m_n}$.)

Suppose that we want to compute $f_3(1^{100})$, i.e., the number of permutations of length 100 that avoid the consecutive pattern 123. If we use the above recurrence literally, we would need about $2^{100}$ computations, but there is a shortcut!

It follows from the symmetry of the generating function $F_r(x_1,\ldots,x_n)$, that $f_r(m_1,\ldots,m_n)$ is symmetric, hence the above Fundamental Recurrence immediately implies the following recurrence, that enables a very fast computation of the sequences, let us call them $a_r(n)$, for the number of *permutations* of length $n$ that avoid the consecutive pattern $1\cdots r$.

#### 3.2.3.1 Fast recurrence for enumerating permutations avoiding the consecutive pattern $1\cdots r$

$$
\begin{aligned}
a_r(n) = na_r(n-1) - \binom{n}{r}a_r(n-r) + \binom{n}{r+1}a_r(n-r-1) - \binom{n}{2r}a_r(n-2r) + \binom{n}{2r+1}a_r(n-2r-1) \\
- \binom{n}{3r}a_r(n-3r) + \binom{n}{3r+1}a_r(n-3r-1) - \cdots \quad .
\end{aligned}
$$

This recurrence goes back to F. N. David and D. Barton [4, p. 157], whose proof used a probabilistic language and an inclusion-exclusion argument that may be viewed as a precursor of the cluster method, applied to the special case of increasing patterns. Note that it takes $O(n^2)$ steps to compute $a_r(n)$ using the recurrence above.

Equivalently, we have the following exponential generating function:

$$\sum_{n=0}^{\infty} a_r(n)\frac{x^n}{n!} = \frac{1}{1 - x + \frac{x^r}{r!} - \frac{x^{r+1}}{(r+1)!} + \frac{x^{2r}}{(2r)!} - \frac{x^{2r+1}}{(2r+1)!} + \frac{x^{3r}}{(3r)!} - \frac{x^{3r+1}}{(3r+1)!} + \cdots} \quad .$$

While this 'explicit' (exponential) generating function is 'nice', it is more efficient to use the fast recurrence. And indeed, the OEIS has these sequences for $3 \leq r \leq 9$, with many terms. These are (in order): A001212, A117158, A177523, A177533, A177553, A230051, A230231.

#### 3.2.3.2 Efficient computations of permutations of words with two occurrences of each letter

Let $b_r(n)$ be the number of words with 2 occurrences of each of $1, 2, \ldots, n$ avoiding the pattern $1 \cdots r$. Quite a few of them are currently (April 17, 2018) in the OEIS, but with relatively few terms

- $b_3(n)$: A177555 (15 terms)
- $b_4(n)$: A177558 (15 terms)
- $b_5(n)$: A177564 (14 terms)
- $b_6(n)$: A177574 (14 terms)
- $b_7(n)$: A177594 (14 terms)

$b_r(n)$ for $r > 7$ are not yet (April 17, 2018) in the OEIS.

We can compute $b_r(n)$ in cubic time as follows. If you plug-in $f_r(2^n)$ into the Fundamental Recurrence, you are forced to consider the more general quantities of the form $f_r(2^\alpha 1^\beta)$. Defining

$$B_r(\alpha, \beta) = f_r(2^\alpha 1^\beta) \quad ,$$

and using symmetry, we get the following recurrence for $B_r(\alpha, \beta)$.

$$B_r(\alpha, \beta) = \alpha B_r(\alpha-1, \beta+1) + \beta B_r(\alpha, \beta-1)$$

$$- \sum_{i_1+i_2=r} \binom{\alpha}{i_1}\binom{\beta}{i_2} B_r(\alpha-i_1, \beta-i_2+i_1) + \sum_{i_1+i_2=r+1} \binom{\alpha}{i_1}\binom{\beta}{i_2} B_r(\alpha-i_1, \beta-i_2+i_1)$$

$$- \sum_{i_1+i_2=2r} \binom{\alpha}{i_1}\binom{\beta}{i_2} B_r(\alpha-i_1,\beta-i_2+i_1) + \sum_{i_1+i_2=2r+1} \binom{\alpha}{i_1}\binom{\beta}{i_2} B_r(\alpha-i_1,\beta-i_2+i_1) - \cdots \quad .$$

In particular $b_r(n) = B_r(n,0)$. Using this recurrence we (easily!) obtained 80 terms of each of the sequences $b_r(n)$ for $3 \le r \le 9$, and could get many more. See the output file

`http://sites.math.rutgers.edu/~zeilberg/tokhniot/oICPW1.txt` .

#### 3.2.3.3 Efficient computations of permutations of words with three occurrences of each letter

Let $c_r(n)$ be the number of words with 3 occurrences of each of $1,2,\ldots,n$ avoiding the pattern $1\cdots r$. Quite a few of them are currently (April 17, 2018) in the OEIS, but with relatively few terms

- $c_3(n)$: A177596 (Only 10 terms)
- $c_4(n)$: A177599 (Only 10 terms)
- $c_5(n)$: A177605 (Only 10 terms)
- $c_6(n)$: A177615 (Only 9 terms)
- $c_7(n)$: A177635 (Only 9 terms)

$c_r(n)$ for $r > 7$ are not yet in the OEIS.

We can compute $c_r(n)$ in quartic time as follows. If you plug-in $f_r(3^n)$ into the Fundamental Recurrence, you are forced to consider the more general quantities of the form $f_r(3^\alpha 2^\beta 1^\gamma)$. Defining

$$C_r(\alpha,\beta,\gamma) = f_r(3^\alpha 2^\beta 1^\gamma) \quad ,$$

and using symmetry, we get the following recurrence for $C_r(\alpha,\beta,\gamma)$.

$$C_r(\alpha,\beta,\gamma) = \alpha C_r(\alpha-1,\beta+1,\gamma) + \beta C_r(\alpha,\beta-1,\gamma+1) + \gamma C_r(\alpha,\beta,\gamma-1)$$
$$- \sum_{i_1+i_2+i_3=r} \binom{\alpha}{i_1}\binom{\beta}{i_2}\binom{\gamma}{i_3} C_r(\alpha-i_1,\beta-i_2+i_1,\gamma-i_3+i_2)$$
$$+ \sum_{i_1+i_2+i_3=r+1} \binom{\alpha}{i_1}\binom{\beta}{i_2}\binom{\gamma}{i_3} C_r(\alpha-i_1,\beta-i_2+i_1,\gamma-i_3+i_2)$$
$$- \sum_{i_1+i_2+i_3=2r} \binom{\alpha}{i_1}\binom{\beta}{i_2}\binom{\gamma}{i_3} C_r(\alpha-i_1,\beta-i_2+i_1,\gamma-i_3+i_2)$$

$$+ \sum_{i_1+i_2+i_3=2r+1} \binom{\alpha}{i_1}\binom{\beta}{i_2}\binom{\gamma}{i_3} C_r(\alpha - i_1, \beta - i_2 + i_1, \gamma - i_3 + i_2) - \cdots$$

In particular, $c_r(n) = C_r(n, 0, 0)$. Using this recurrence we (easily!) obtained 40 terms of each of the sequences $c_r(n)$ for $3 \le r \le 9$, and could get many more. See the output file

`http://sites.math.rutgers.edu/~zeilberg/tokhniot/oICPW1.txt` .

#### 3.2.3.4 Efficient computations of permutations of words with four occurrences of each letter

Let $d_r(n)$ be the number of words with 4 occurrences of each of $1, 2, \ldots, n$ avoiding the pattern $1 \cdots r$. Quite a few of them are currently (April 17, 2018) in the OEIS, but with relatively few terms.

- $d_3(n)$: A177637 (8 terms)
- $d_4(n)$: A177640 (8 terms)
- $d_5(n)$: A177646 (8 terms)
- $d_6(n)$: A177656 (8 terms)
- $d_7(n)$: A177676 (8 terms)

$d_r(n)$ for $r > 7$ are not yet in the OEIS.

We can compute $d_r(n)$ in quintic time as follows. If you plug-in $f_r(4^n)$ into the Fundamental Recurrence, you are forced to consider the more general quantities of the form $f_r(4^\alpha 3^\beta 2^\gamma 1^\delta)$. Defining

$$D_r(\alpha, \beta, \gamma, \delta) = f_r(4^\alpha 3^\beta 2^\gamma 1^\delta) \quad ,$$

and using symmetry, we get the following recurrence for $D_r(\alpha, \beta, \gamma, \delta)$.

$$D_r(\alpha,\beta,\gamma,\delta) = \alpha D_r(\alpha-1,\beta+1,\gamma,\delta) + \beta D_r(\alpha,\beta-1,\gamma+1,\delta) + \gamma D_r(\alpha,\beta,\gamma-1,\delta+1) + \delta D_r(\alpha,\beta,\gamma,\delta-1)$$

$$- \sum_{i_1+i_2+i_3+i_4=r} \binom{\alpha}{i_1}\binom{\beta}{i_2}\binom{\gamma}{i_3}\binom{\delta}{i_4} D_r(\alpha - i_1, \beta - i_2 + i_1, \gamma - i_3 + i_2, \delta - i_4 + i_3)$$

$$+ \sum_{i_1+i_2+i_3+i_4=r+1} \binom{\alpha}{i_1}\binom{\beta}{i_2}\binom{\gamma}{i_3}\binom{\delta}{i_4} D_r(\alpha - i_1, \beta - i_2 + i_1, \gamma - i_3 + i_2, \delta - i_4 + i_3)$$

$$- \sum_{i_1+i_2+i_3+i_4=2r} \binom{\alpha}{i_1}\binom{\beta}{i_2}\binom{\gamma}{i_3}\binom{\delta}{i_4} D_r(\alpha - i_1, \beta - i_2 + i_1, \gamma - i_3 + i_2, \delta - i_4 + i_3)$$

$$+\sum_{i_1+i_2+i_3+i_4=2r+1} \binom{\alpha}{i_1}\binom{\beta}{i_2}\binom{\gamma}{i_3}\binom{\delta}{i_4} D_r(\alpha-i_1, \beta-i_2+i_1, \gamma-i_3+i_2, \delta-i_4+i_3) - \cdots$$

In particular $d_r(n) = D_r(n, 0, 0, 0)$. Using this recurrence we (easily!) obtained 20 terms of each of the sequences $c_d(n)$ for $3 \le r \le 9$, and could get many more. See the output file

`http://sites.math.rutgers.edu/~zeilberg/tokhniot/oICPW1.txt` .

**Comment**: Ira Gessel kindly informed us that an alternative approach to extracting coefficients from the generating function in Theorem 3.1 is to use the elegant method described in section 3 of his paper on symmetric functions and P-recursiveness [7].

### 3.2.4 Keeping track of the number of occurrences

Above we showed how to enumerate words *avoiding* the consecutive pattern $1 \cdots r$, in other words, the number of words, with a specified number of each letters, with **zero** such patterns. With a little more effort we can answer the more general question about the number of such words with exactly $j$ consecutive patterns $1 \cdots r$ for *any* $j$, not just $j = 0$. Let $W(\mathbf{m}) = W(m_1, \ldots, m_n)$ be the set of words in the alphabet $1, \ldots, n$ with $m_1$ 1's, ... , $m_n$ $n$'s (note that the number of elements of $W(\mathbf{m})$ is $(m_1 + \cdots + m_n)!/(m_1! \cdots m_n!)$).

We are interested in the polynomials in $t$

$$g_r(\mathbf{m}; t) = \sum_{w \in W(\mathbf{m})} t^{\alpha(w)} \quad ,$$

where $\alpha(w)$ is the number of occurrences of the consecutive pattern $1 \cdots r$ in the word $w$. (For example $\alpha(831456178) = 3$ if $r = 3$. Note that $\alpha(w) = 0$ if $w$ avoids the pattern.)

[Also note that $g_r(\mathbf{m}; 0) = f_r(\mathbf{m})$ and $g_r(\mathbf{m}; 1) = (m_1 + \cdots + m_n)!/(m_1! \cdots m_n!)$.]

Using GJpats.txt we were able to conjecture the following theorem, whose proof will be presented later.

We first need to define certain families of polynomial sequences.

**Definition 3.2.** *For any integer $k \ge 1$ and $r \ge 2$, $P_k^{(r)}(t)$ is defined as follows.*

If $k < r$, then it is 0. If $k = r$ then it is $t-1$, and if $k > r$ then

$$P_k^{(r)}(t) \;=\; (t-1)\sum_{i=1}^{r-1} P_{k-i}^{(r)}(t) \quad .$$

**Theorem 3.3.** *For $k \geq 1, r \geq 2$, the generating function of $g_r(\mathbf{m};t)$, let us call it $G_r(x_1,\ldots,x_n;t)$, is*

$$G_r(x_1,\ldots,x_n;t) = \frac{1}{1 - e_1 - \sum_{k=r}^{n} P_k^{(r)}(t) e_k} \quad .$$

This implies the

**Fundamental Recurrence for** $g_r$: Let $g_r(\mathbf{m};t)$ be the weight-enumerator of words in the alphabet $\{1,\ldots,n\}$ with $m_1$ 1's, $m_2$ 2's, $\ldots m_n$ $n$'s (where we abbreviate $\mathbf{m} = (m_1,\ldots,m_n)$), using the weight "$t$ raised to the power of the number of occurrences of the consecutive pattern $1\cdots r$".

Also, let $V_k$ be the set of $0-1$ vectors of length $n$ with $k$ ones. Then we have

$$g_r(\mathbf{m}) = \sum_{\mathbf{v}\in V_1} g_r(\mathbf{m}-\mathbf{v}) \;+\; \sum_{k=r}^{n}\sum_{\mathbf{v}\in V_k} P_k^{(r)}(t)\, g_r(\mathbf{m}-\mathbf{v}) \quad .$$

Analogously to the avoidance case we can get efficient recurrences for the permutations, and words in $1^s\cdots n^s$, for each $s > 1$. For each $s$ it is still polynomial time, but things are slower because of the variable $t$. This is implemented in the Maple package `ICPWt.txt` .

## 3.3 Proofs by tweaking the Goulden-Jackson cluster method

### 3.3.1 Proof of Theorem 1

We will use the general set-up of the Goulden-Jackson cluster method as described in Noonan and Zeilberger's paper [11], but will be able to make things simpler by taking advantage of the specific structure of our forbidden patterns, which are the increasing patterns $1\cdots r$. That will enable us to use an elegant combinatorial argument, without solving a system of linear equations.

First let us quickly review some basic definitions. (We will not go into the details of the cluster method but readers who wish to see an excellent and concise summary of the cluster method are welcome to refer to the first section of Wen's paper [14].) A **marked word** is a word with some of its factors (consecutive subwords) marked. We are assuming that all the marks are in the set of bad words $B$. For example (13212; [1,3]) is a marked word with 132 marked, with [1,3] denoting the location of the mark. A **cluster** is a marked word where the adjacent marks overlap with each other and all the letters in the underlying word belong to at least one mark of the cluster. For example (145632; [1,3],[2,4],[4,6]) is a cluster whereas (145632; [1,3],[4,6]) is not. We let the weight of a marked word $w = w_1 w_2 \cdots w_k$ be $weight(w) := (-1)^{|S|} \cdot \prod_{i=1}^{k} x[w_i]$ where $S$ is the set of marks in $w$. For example, the weight of the cluster (135632; [1,3],[2,4],[4,6]) is $(-1)^3 x_1 x_2 x_3^2 x_5 x_6$.

Let $M$ be the set of all marked words in the alphabet $\{1, \ldots, n\}$. Recall from [11] that $weight(M) = 1 + weight(M) \cdot (x_1 + x_2 + \cdots + x_n) + weight(M) \cdot weight(C)$ where $C$ is the set of all possible clusters. We also know from page 4 and 5 of [11] that $weight(M)$ is equal to the generating function for words avoiding the set of bad words. This implies that the multivariate generating function for words avoiding the increasing pattern $1 \cdots r$ (i.e., our target generating function) is equal to $weight(M) = \frac{1}{1 - e_1 - weight(C)}$. So we only need to figure out $weight(C)$. However, to use the classical Goulden-Jackson cluster method, we would have to solve a system of $\binom{n}{r}$ (the number of bad words) equations (recall that we write C as a summation of $C[v]$'s where $v$ is a word in B, and for each $C[v]$ there is an equation) and no obvious symmetry argument seems to help. So we will use a slick combinatorial approach.

Notice that since the pattern to be avoided is $12 \cdots r$, the clusters can only be of the form

$$(a_1 \cdots a_j; [1, r], \ldots)$$

where

$$1 \leq a_1 < a_2 < \cdots < a_j \leq n \quad .$$

Therefore $weight(C)$ is a summations of multivariate monomials on $x_1, x_2, \ldots, x_n$ where the exponent of each variable $x_i$ is zero or one.

Any fixed monomial in $weight(C)$ can come from many different clusters. The number of clusters it comes from and the coefficient of the monomial are uniquely determined by the number of variables in the monomial. For example, for $r = 3$, the monomial $x_1x_3x_5x_6x_7$ can come from the cluster $(13567; [1, 3], [2, 4], [3, 5])$ or $(13567; [1, 3], [3, 5])$. The first cluster contributes weight $(-1)^3x_1x_3x_5x_6x_7$ whereas the second cluster contributes weight $(-1)^2x_1x_3x_5x_6x_7$. So when summing up, they cancel each other out and there is no monomial $x_1x_3x_5x_6x_7$ in $weight(C)$. So is the case with any other monomial of degree 5. Therefore, let us focus on the monomial $x_1x_2x_3 \cdots x_k$ and figure out its coefficient.

**Definition 3.4.** *Let coeff*$(x_1x_2 \cdots x_k)$ $(k \geq 1)$ *be the coefficient of* $x_1x_2 \cdots x_k$ *in* $weight(C)$.

It is clear that for $k < r$, coeff$(x_1x_2x_3 \cdots x_k) = 0$, because $12 \cdots k$ cannot be a cluster (it does not have enough letters to be marked). And when $k = r$, we have coeff$(x_1x_2 \cdots x_k) = -1$, since there can be only one mark. So let us move on to the case when $k > r$. We have the following Lemma.

**Lemma 3.5.** *For* $k > r$, *coeff*$(x_1x_2 \cdots x_k) = -$ *coeff*$(x_2x_3 \cdots x_k) -$ *coeff*$(x_3x_4 \cdots x_k) - \cdots -$ *coeff*$(x_rx_{r+1} \cdots x_k)$. *(Equivalently, coeff*$(x_1x_2 \cdots x_k) = -$ *coeff*$(x_1x_2 \cdots x_{k-1}) -$ *coeff*$(x_1x_2 \cdots x_{k-2}) - \cdots -$ *coeff*$(x_1x_2 \cdots x_{k-r+1})$.*)*

This is because there are $(r - 1)$ ways in which the left-most marked word can "interface" with the one to its immediate right. For example, if the clusters are of the form $(1 \cdots k; [1, r], [3, r + 2], \ldots)$ (that is, the second mark starts at 3), then the contribution will be $(-1)\cdot$ coeff$(x_3x_4 \cdots x_k)$. This is simply because of the bijection between the set of clusters in the form of $(1 \cdots k; [1, r], [3, r + 2], \ldots)$ with set of the clusters in the form $(3 \cdots k; [3, r + 2], \ldots)$. By peeling off the first mark $[1, r]$, we just lose a factor of $(-1)$ in the coefficient of our monomial.

Similarly, if the clusters are of the form $(1 \cdots k; [1, r], [u, u + r - 1], \ldots)$ $(1 < u \leq r)$, then the contribution from this case will be $(-1)\cdot$ coeff$(x_ux_{u+1} \cdots x_k)$. Note that if $k < 2r - 1$, there cannot be as many as $(r - 1)$ cases. However, in this case, we can make the convention that there are $(r - 1)$ places for the second mark because for $k < r$ the coefficient of $x_1x_2x_3 \cdots x_k$ is 0. So the above formula still holds.

For example, for the clusters associated with the word 123456, and $r = 4$, the first mark has to be 1234, the second mark can only be 2345 or 3456. But, according to the natural convention, the second mark can also start with 4 and be 456, and so, $\text{coeff}(x_1x_2x_3x_4x_5x_6) = -\text{coeff}(x_2x_3x_4x_5x_6) - \text{coeff}(x_3x_4x_5x_6) - \text{coeff}(x_4x_5x_6) = -\text{coeff}(x_2x_3x_4x_5x_6) - \text{coeff}(x_3x_4x_5x_6)$.

So we have: $\text{coeff}(x_1x_2\cdots x_r) = -1$; $\text{coeff}(x_1x_2\cdots x_{r+1}) = (-1)\cdot(-1) = 1$; $\text{coeff}(x_1x_2\cdots x_{r+2}) = -\text{coeff}(x_2x_3\cdots x_{r+2}) - \text{coeff}(x_3x_4\cdots x_{r+2}) = -\text{coeff}(x_1x_2\cdots x_{r+1}) - \text{coeff}(x_1x_2\cdots x_r) = 0$. Continuing this process, it is easy to see that $x_1x_2\cdots x_{mr}$ ($m \geq 1$) has coefficient $-1$ (so is any other monomial of degree $mr$ ) and $x_1x_2\cdots x_{mr+1}$ has coefficient 1 (so is any other monomial of degree $mr+1$). The monomials with other number of variables all have coefficient 0. From this argument and summing over all clusters, we conclude $weight(C) = -e_r + e_{r+1} - e_{2r} + e_{2r+1} + \cdots$ and therefore $weight(M) = \frac{1}{1-e_1+e_r-e_{r+1}+e_{2r}-e_{2r+1}+\cdots}$.

### 3.3.2 Proof of Theorem 3

This proof can be directly generalized from the proof of Theorem 3.1 based on the '$t$-generalization' described in Noonan and Zeilberger's paper [11]. Again, let the set of marked words on $\{1, 2, \ldots, n\}$ be $M$. However, this time we let the weight of a marked word $w$ of length $k$ be $weight(w) := (t-1)^{|S|} \cdot \prod_{i=1}^{k} x[w_i]$ where $S$ is the set of marks in $w$. We still have $weight(M) = 1 + weight(M)\cdot(x_1+x_2+\cdots+x_n) + weight(M)\cdot weight(C)$ and $G_r(x_1, \ldots, x_n; t)$ is equal to $weight(M)$, which is $\frac{1}{1-e_1-weight(C)}$ (for details, see page 11 and 12 of [11]).

The procedure to calculate $weight(C)$ directly follows from the proof of Theorem 3.1. We simply replace $(-1)$ with $(t-1)$ in various places, because the only difference is that now we assign a different weight to a marked word. For example, we have $\text{coeff}(x_1x_2\cdots x_r) = t-1$; $\text{coeff}(x_1x_2\cdots x_{r+1}) = (t-1)(t-1) = (t-1)^2$; $\text{coeff}(x_1x_2\cdots x_{r+2}) = (t-1)(\text{coeff}(x_2x_3\cdots x_{r+2}) + \text{coeff}(x_3x_4\cdots x_{r+2})) = (t-1)((t-1)+(t-1)^2)$. Again it is clear that for $k < r$, $\text{coeff}(x_1x_2x_3\cdots x_k) = 0$ and when $k = r$, $\text{coeff}(x_1x_2\cdots x_k) = t-1$. For the case when $k > r$, we generalize Lemma 3.5 to the following:

**Lemma 3.6.** *For $k > r$, coeff$(x_1 x_2 \cdots x_k) = (t-1)$ (coeff$(x_2 x_3 \cdots x_k)+$ coeff$(x_3 x_4 \cdots x_k)+ \cdots +$coeff$(x_r x_{r+1} \cdots x_k)$). (Equivalently, coeff$(x_1 x_2 \cdots x_k) = (t-1)$ (coeff$(x_1 x_2 \cdots x_{k-1})+$ coeff$(x_1 x_2 \cdots x_{k-2}) + \cdots +$ coeff$(x_1 x_2 \cdots x_{k-r+1})$).)*

The proof of Lemma 3.6 directly generalizes from the proof of Lemma 3.5. Now one mark contributes a factor of $(t-1)$ instead of $(-1)$ to the weight of a marked word. For example, for the clusters associated with the word 123456, and $r = 3$, the first mark has to be 123, the second mark can be 234 or 345. So coeff$(x_1 x_2 x_3 x_4 x_5 x_6) = (t-1)$(coeff$(x_2 x_3 x_4 x_5 x_6)$+coeff$(x_3 x_4 x_5 x_6)$). In general, like in the proof of Theorem 3.1, if we are interested in keeping track of the number of appearances of the consecutive pattern $12 \cdots r$, then there are $(r-1)$ scenarios of clusters that can give rise to the monomial $x_1 x_2 \cdots x_k$, depending on where the second mark is. By peeling off the first mark, now we lose a factor of $(t-1)$ instead of $(-1)$ in the coefficient of our monomial.

As the coefficients of the monomials of the same length are the same, Lemma 6 immediately implies that $weight(C) = \sum_{k=r}^{n} P_k^{(r)}(t) e_k$ where $P_k^{(r)}(t)$ satisfies the recurrence

$$P_k^{(r)}(t) \;=\; (t-1) \sum_{i=1}^{r-1} P_{k-i}^{(r)}(t) \quad .$$

(In fact $P_k^{(r)}(t)$ is just a concise way of writing coeff$(x_1 x_2 \cdots x_k)$, where the consecutive pattern of interest is $12 \cdots r$.) From this Theorem 3.3 follows directly.

# Chapter 4

# Relaxed partitions

Starting this chapter, we venture into the fascinating territory of integer partitions. More strictly speaking, in this chapter, we introduce a new combinatorial object that can be seen as a variation of traditional integer partitions. We experiment on this combinatorial object and find intruiging results. Next chapter we will go back to traditional integer partitions.

## 4.1 Introduction

Recall that a partition of a positive integer $n$ is a finite nonincreasing sequence of positive integers $\lambda_1$, $\lambda_1 \ldots \lambda_k$ whose sum is equal to $n$. A lot of beautiful theories and conjectures have been developed in this area and this field is blooming.

In this chapter, we are going to take a road less traveled and study an object which we call "relaxed partitions", or more specifically, $r$-partitions with $r$ to be specified. Unlike the traditional partitions where we require $\lambda_i - \lambda_{i+1} \geq 0$, for $r$-partitions we require $\lambda_i - \lambda_{i+1} \geq r$ where $r$ can be negative. For example, $(2, 3, 1, 1)$ is a $(-1)$-partition of 7.

Just as with traditional partitions, there are many questions one could ask about $r$-partitions. Perhaps one of the first questions to ask is: "for a fixed $r$, how many $r$-partitions of the integer $n$ do we have?" Using an easy recurrence relation, we used Maple to program a procedure which we called $\mathbf{NPr(n, r)}$, and it returns the number of $r$-partitions of $n$ for any given $n$ and $r$. But can we find a generating function for a given $r$? The answer turned out to be yes! And there is a nice single-sum formula for it. By typing in a sequence of entries produced by $\mathbf{NPr(n, -1)}$ into the **OEIS**, we found that its generating function seemed to be the reciprocal of

$$\sum_{k=0}^{\infty} \frac{(-1)^k q^{k^2}}{\prod_{i=1}^{k}(1-q^i)} .$$

Recognizing this is the generating function for the "weighted" (with weight $(-1)^k$) number of (traditional) partitions of integer $n$ into parts with difference at least 2, Doron Zeilberger provided an short and elegant bijective proof for it (interested reader can look up the proof in the comments of the sequence **A003116** in **OEIS**). This proof can be directly generalized to general $r$, and the generating function for the number of $r$-partitions of $n$ is the reciprocal of

$$\sum_{k=0}^{\infty} \frac{(-1)^k q^{k(2+(1-r)(k-1))/2}}{\prod_{i=1}^{k}(1-q^i)} ,$$

which is the "weighted" number of partitions of integer $n$ into parts with difference at least $(-r+1)$.

In this chapter, we will focus on restricted $r$-partitions such that the first part and the number of the parts are fixed. Let $a_r(M,N,n)$ be the number of $r$-partitions of $n$ with the first part equal to $M$ and exactly $N$ parts. To go with the notation in our Maple package **rPar** (the Maple package can be found at `https://sites.math.rutgers.edu/~my237/RP`), let $F(M,N,r,q)$ be the generating function for $a_r(M,N,n)$. Using a simple recurrence relation $F(M,N,r,q)$ satisfies, we were able to program it in Maple and happily used Maple to conjecture (**and prove!**) the closed form for the case $q=1$ (i.e., the total number of $r$-partitions with the first part equal to $M$ and exactly $N$ parts). It is:

$$F(M,N,r,1) = \frac{(M-r)(M+(1-r)N-2)!}{(N-1)!(M-rN)!} = \binom{M+(1-r)N-2}{N-1} + r\binom{M+(1-r)N-2}{N-2}$$

Although we were not yet able to find a closed form formula for the generating function $F(M,N,r,q)$, we found out (using Maple) some initial terms (according to $N$) of it. In Section 3 of this chapter, we will present the first few terms corresponding to $N \le 5$.

In the last section, we will present some possible future work to be done as well as connections with other combinatorial objects.

## 4.2 The recurrence relation for $F(M, N, r, q)$

It is not hard to come up with a recurrence relation for $F(M, N, r, q)$. Given an $r$-partition of $n$ with the first part equal to $M$ and exactly $N$ parts, we can knock off the first row (we know by doing this we take away a factor of $q^M$) and what is left is an $r$-partition of $n - M$ with the first part equal to $M_1$ and exactly $N - 1$ parts, where $1 \leq M1 \leq M - r$. Therefore we have the following recurrence relation:

$$F(M, N, r, q) = q^M \sum_{M_1=1}^{M-r} F(M_1, N - 1, r, q) \quad .$$

It is clear that $F(M, 1, r, q) = q^M$. With this information, we could program the procedure $\mathbf{F(M, N, r, q)}$ in Maple, which allows us to input specific $M$,$N$,$r$ ($q$ can be symbolic), and it will output the corresponding generating function.

## 4.3 Using Maple to discover (and prove!) patterns for $F(M, N, r, 1)$

We start our experiment with $F(M, N, -1, 1)$, which represents the total number of $(-1)$-partitions with the first part equal to $M$ and exactly $N$ parts. By fixing N and varying $M$, we generate the first 20 terms of $F(M, N, -1, 1)$, which allows us to then use the procedure **GuessPol** in our package **rPar** to try guessing a polynomial for this sequence.

For example, typing $[\mathbf{seq(F(M1, 1, -1, 1), M1 = 1..20)}]$; yields

$$[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$$

To guess a polynomial for this sequence, type:

$\mathbf{GuessPol([seq(F(M1, 1, -1, 1), M1 = 1..20)], M, 1)}$;

Not surprisingly, it yields the constant polynomial 1.

Now try $N = 2$. Typing $[\mathbf{seq(F(M1, 2, -1, 1), M1 = 1..20)}]$; yields

$$[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]$$

**GuessPol**([**seq**(**F**(**M1**, **2**, −**1**, **1**), **M1** = **1**..**20**)], **M**, **1**); yields

$$M + 1$$

Also not a surpise. Moving right along,

**GuessPol**([**seq**(**F**(**M1**, **3**, −**1**, **1**), **M1** = **1**..**20**)], **M**, **1**); yields

$$\frac{(M+4)(M+1)}{2}$$

**GuessPol**([**seq**(**F**(**M1**, **4**, −**1**, **1**), **M1** = **1**..**20**)], **M**, **1**); yields

$$\frac{(M+6)(M+5)(M+1)}{6}$$

**GuessPol**([**seq**(**F**(**M1**, **5**, −**1**, **1**), **M1** = **1**..**20**)], **M**, **1**); yields

$$\frac{(M+8)(M+7)(M+6)(M+1)}{24}$$

**GuessPol**([**seq**(**F**(**M1**, **6**, −**1**, **1**), **M1** = **1**..**20**)], **M**, **1**); yields

$$\frac{(10+M)(9+M)(8+M)(7+M)(M+1)}{120}$$

Without much effort, one can conjecture the following:

$$F(M, N, -1, 1) = \frac{(M+1)(M+2N-2)!}{(N-1)!(M+N)!} \quad .$$

Now let us experiment with $r = -2$.

**GuessPol**([**seq**(**F**(**M1**, **1**, −**2**, **1**), **M1** = **1**..**20**)], **M**, **1**); yields

$$1$$

**GuessPol**([**seq**(**F**(**M1**, **2**, −**2**, **1**), **M1** = **1**..**20**)], **M**, **1**); yields

$$M + 2$$

**GuessPol**([**seq**(**F**(**M1**, **3**, −**2**, **1**), **M1** = **1**..**20**)], **M**, **1**); yields

$$\frac{(M+7)(M+2)}{2}$$

**GuessPol**([**seq**(**F**(**M1**, **4**, −**2**, **1**), **M1** = **1**..**20**)], **M**, **1**); yields

$$\frac{(M+10)(M+9)(M+2)}{6}$$

**GuessPol**([**seq**(**F**(**M1**, **5**, −**2**, **1**), **M1** = **1**..**20**)], **M**, **1**); yields

$$\frac{(M+13)(M+12)(M+11)(M+2)}{24}$$

**GuessPol**([**seq**(**F**(**M1**, **6**, −**2**, **1**), **M1** = **1**..**20**)], **M**, **1**); yields

$$\frac{(M+16)(M+15)(M+14)(M+13)(M+2)}{120}$$

Again without much effort, one can conjecture that

$$F(M,N,-2,1) = \frac{(M+2)(M+3N-2)!}{(N-1)!(M+2N)!} \quad .$$

Comparing these two guesses, one can easily conjecture the formula for a general r:

$$F(M,N,r,1) = \frac{(M-r)(M+(1-r)N-2)!}{(N-1)!(M-rN)!} \quad .$$

Now, how do we prove this conjecture?

Recall that we programmed **F**(**M**, **N**, **r**, **q**) using the recurrence relation

$$F(M,N,r,q) = q^M \sum_{M_1=1}^{M-r} F(M_1, N-1, r, q)$$

and the initial condition $F(M,1,r,q) = q^M$. Note that $F(M,N,r,q)$ can be fully defind by this information. In other words, if we have found a formula that satisfies this recurrence relation and initial condition, then it *is* the formula for $F(M,N,r,q)$. This also applies to our current case when $q = 1$.

Therefore we went ahead and programmed the procedure **checkF**(**M**, **N**, **r**) and hooray! Maple was able to show that, by using symbolic computation, our conjectured formula for $F(M,N,r,1)$ indeed satisfies the recurrence relation (for $q = 1$). It is also easy to verify that the initial condition is satisfied. Therefore we have arrived, with a lot of help from Maple, at the following theorem:

**Theorem 4.1.**

$$F(M,N,r,1) = \frac{(M-r)(M+(1-r)N-2)!}{(N-1)!(M-rN)!} \quad .$$

Now the next step is to try to conjecture a formula for $F(M,N,r,q)$.

## 4.4 Can we find a pattern for $F(M,N,r,q)$?

Just like with the case $q = 1$, before we get so bold to go figuring out a pattern for $F(M,N,r,q)$, let us start by trying to figure out a pattern for $F(M,N,-1,q)$. Below are the guesses (using the **qGuessPol** procedure) for $N \leq 5$:

$\mathbf{qGuessPol([seq(F(M1,1,-1,q),M1=1..20)],M,q,1)}$; yields

$$q^M$$

$\mathbf{qGuessPol([seq(F(M1,2,-1,q),M1=1..20)],M,q,1)}$; yields

$$\frac{q^{M+1}(q^{M+1}-1)}{(q-1)}$$

$\mathbf{qGuessPol([seq(F(M1,3,-1,q),M1=1..20)],M,q,1)}$; yields

$$\frac{q^{M+2}(q^{M+3}+q^2-q-1)(q^{M+1}-1)}{(q-1)^2(q+1)}$$

$\mathbf{qGuessPol([seq(F(M1,4,-1,q),M1=1..20)],M,q,1)}$; yields

$$\frac{q^{M+3}(q^{2M+8}+q^{M+7}-q^{M+5}-q^{M+4}-q^{M+3}+q^6-2q^4-q^3+2q+1)(q^{M+1}-1)}{(q-1)^3(q+1)(q^2+q+1)}$$

$\mathbf{qGuessPol([seq(F(M1,5,-1,q),M1=1..20)],M,q,1)}$; yields

$$\frac{q^{M+4}(q^{M+5}+q^4-q-1)(q^{2M+10}-q^{M+4}-q^{M+3}+q^8-q^5-2q^4+2q+1)(q^{M+1}-1)}{(q-1)^4(q+1)^2(q^2+1)(q^2+q+1))}$$

We can again **prove** that they are true by using the recurrence relation

$$F(M,N,r,q) = q^M \sum_{M_1=1}^{M-r} F(M_1,N-1,r,q)$$

and the initial condition $F(M,1,r,q) = q^M$, setting $r = -1$.

Note that, although the formulas above look like rational functions, they are in fact polynomials. It is easy to see why the first two formulas are polynomials. The rest are also polynomials simply because of the recurrence relation (a summation of polynomials is also a polynomial). So far we have not yet been able to find a closed form for $F(M, N, -1, q)$. However, Drew Sills made an interesting observation.

Drew Sills's Observation: $F(M, N, -1, q)$ has denominator $(q; q)_N$ and a numerator of degree $N(M+N-1)$. Thus it is plausible that the numerator is a (possibly alternating) sum of polynomials that are a power of q times a Gaussian polynomial of the form $G(M, N) := GP(2N + M - 1, N)$, where GP stands for the usual gaussian polynomial:

$$GP(m, r) := \frac{(q^{m-r+1}; q)_r}{(q; q)_r}$$

So far we have not made much progress in this, but we noticed an interesting pattern:

$$\begin{cases} G(2,1) = \underline{q^2 + q + 1} \\ F(2,2,-1,q) = \underline{q^5 + q^4 + q^3} \end{cases}$$

$$\begin{cases} G(2,2) = \underline{q^6 + q^5 + 2q^4 + 2q^3 + 2q^2 + q} + 1 \\ F(2,3,-1,q) = \underline{q^9 + q^8 + 2q^7 + 2q^6 + 2q^5 + q^4} \end{cases}$$

$$\begin{cases} G(2,3) = \underline{q^{12} + q^{11} + 2q^{10} + 3q^9 + 4q^8 + 4q^7 + 5q^6 + 4q^5} + 4q^4 \\ \qquad +3q^3 + 2q^2 + q + 1 \\ F(2,4,-1,q) = \underline{q^{14} + q^{13} + 2q^{12} + 3q^{11} + 4q^{10} + 4q^9 + 5q^8} \\ \qquad \underline{+4q^7} + 3q^6 + q^5 \end{cases}$$

In each case, the underlined parts for $G$ and $F$ have the same coefficients. This pattern continues for larger values of $N$(we tested until N=9). In addition, it is straightforward to deduce from definition that the highest power of $F(2, N, -1, q)$ is $N(N+3)/2$. Therefore, we can conjecture $2N$ many terms of $F(2, N, -1, q)$ using the first $2N$ terms of $G(2, N-1)$ $(N \geq 3)$. Similarly, we have the following patterns for $F(3, N, -1, q)$:

$$\begin{cases} G(3,1) = \underline{q^3+q^2+q+1} \\ F(3,2,-1,q) = \underline{q^7+q^6+q^5+q^4} \end{cases}$$

$$\begin{cases} G(3,2) = \underline{q^8+q^7+2q^6+2q^5+3q^4+2q^3+2q^2+q}+1 \\ F(3,3,-1,q) = \underline{q^{12}+q^{11}+2q^{10}+2q^9+3q^8+2q^7+2q^6+q^5} \end{cases}$$

$$\begin{cases} G(3,3) = \underline{q^{15}+q^{14}+2q^{13}+3q^{12}+4q^{11}+5q^{10}+6q^9+6q^8+6q^7+6q^6} \\ \qquad\qquad +5q^5+4q^4+3q^3+2q^2+q+1 \\ F(3,4,-1,q) = \underline{q^{18}+q^{17}+2q^{16}+3q^{15}+4q^{14}+5q^{13}+6q^{12}+6q^{11}+6q^{10}+6q^9} \\ \qquad\qquad\qquad +4q^8+3q^7+q^6 \end{cases}$$

$$\begin{cases} G(3,4) = \underline{q^{24}+q^{23}+2q^{22}+3q^{21}+5q^{20}+6q^{19}+9q^{18}+10q^{17}+13q^{16}} \\ \qquad\qquad \underline{+14q^{15}+16q^{14}+16q^{13}}+18q^{12}+16q^{11}+16q^{10} \\ \qquad\qquad +14q^9+13q^8+10q^7+9q^6+6q^5+5q^4+3q^3+2q^2+q+1 \\ F(3,5,-1,q) = \underline{q^{25}+q^{24}+2q^{23}+3q^{22}+5q^{21}+6q^{20}+9q^{19}+10q^{18}} \\ \qquad\qquad\qquad \underline{+13q^{17}+14q^{16}+16q^{15}+16q^{14}} \\ \qquad\qquad\qquad +17q^{13}+15q^{12}+14q^{11}+11q^{10}+7q^9+4q^8+q^7 \end{cases}$$

Here the conjecture is that we can predict the first $(2N+2)$ terms of $F(3,N,-1,q)$ using the first $(2N+2)$ terms of $G(3,N-1)$ $(N \geq 3)$.

For $M = 4$, the conjecture is that we can predict the first $(2N+4)$ terms of $F(4,N,-1,q)$ using the first $(2N+4)$ terms of $G(4,N-1)$ $(N \geq 3)$.

Now, a "meta-pattern" appears, and the conjecture is that we can predict the first $(2N+2M-4)$ terms of $F(M,N,-1,q)$ using the first $(2N+2M-4)$ terms of $G(M,N-1)(N \geq 3)$.

## 4.5 Future work and connection to other combinatorial objects

Interestingly, there is a direct connection between $F(M, N, -1, 1)$ and Catalan's triangle (thanks again to **OEIS** to help us find such a connection). Catalan's triangle is a number triangle whose entries $C(n, k)$ is the number of strings consisting of $n$ $X$'s and $k$ $Y$'s such that no initial segment of the string has more $Y$'s than $X$'s. It satisfies the following:

$$C(n,k) = \binom{n+k}{k} - \binom{n+k}{k-1}$$
$$C(n,k) = \frac{(n+k)!(n-k+1)}{k!(n+1)!}$$
$$C(n+1,k) = C(n+1,k-1) + C(n,k)$$

Since we have shown $F(M, N, -1, 1) = \frac{(M-r)(M+(1-r)N-2)!}{(N-1)!(M-rN)!}$, it is clear that:

$$F(M, N, -1, 1) = C(M + N - 1, N - 1)$$

.

This turns out to be more or less obvious by a geometric interpretation. If one thinks in terms of lattice path, $C(M+N-1, N-1)$ is the number of lattice paths from the origin to the point $(M + N - 1, N - 1)$ that do not go above the line $y = x$ in the $xy$-plane and with $N-1$ steps up. (Note in each step we are only allowed to move right or up one step.) Each such path corresponds uniquely to a $(-1)$-partition with the first part exactly equal to $M$ and exactly $N$ parts. The part of the path that is above the line $y = N - 2$ can take $(M + 1)$ shapes. The number of horizontal steps in the shape determines the second part of the corresponding $(-1)$-partition (note there are $(M+1)$ possibilities for the second part). If there are $k$ horizontal steps, then the second part of the partition is $M - k + 1$. Similarly, the third part of the partition is determined by how many horizontal steps we take in the lattice path above line $y = N - 3$ and below (including) the line $y = N - 2$.

There also seems to be a bit of connection between the standard Young Tableau and $F(M, N, -1, 1)$. For example, typing the sequence $[seq(F(5, N, -1, 1), N = 1..20]$ into

**OEIS**, we will find it can also represent the number of standard Young Tableau of shape $(N+3, N-2)$. (A003517) If we change the value of M, then we can find correspondence with other standard Young Tableau. We conjecture that $F(M, N, -1, 1)$ is equal to the number of standard Young Tableau of shape $(N + \lceil M/2 \rceil, N - \lfloor M/2 \rfloor)$.

For general $r$, we haven't found nice connections yet.

# Chapter 5

# Systematic counting of restricted partitions and searching for new partition identities

In this chapter we go back to traditional integer partitions. We will first devise an efficient algorithm to count for certain restricted partitions, and then use this algorithm to search for new partition identities.

## 5.1 Introduction

Recall that a partition of a non-negative integer $n$ is a list of integers $(\lambda_1, \ldots, \lambda_k)$ such that $\lambda_1 \geq \cdots \geq \lambda_k \geq 1$ and $\lambda_1 + \cdots + \lambda_k = n$.

As usuall, we will denote the number of integer partitions of $n$ by $p(n)$. This is a very famous sequence, OEIS sequence A41.

While there is no 'explicit' formula for $p(n)$, there is a nice generating function, that goes back to Leonhard Euler. Denoting the number of integer partitions of $n$ by $p(n)$, Euler discovered that

$$\sum_{n=0}^{\infty} p(n)\, q^n = \prod_{i=1}^{\infty} \frac{1}{1-q^i} \quad .$$

The **bible** of the theory of partitions is George Andrews' classic [An]. We also strongly recommend Drew Sills' fascinating monograph [S].

Suppose that you did not know about Euler's generating function, and you were given the task of computing the first, say, 1000 terms of the sequence $p(n)$, how would you proceed? The most straightforward way would be to try and use *dynamical programming.* Note that partitions have the *hereditary* property. If you chop-off the largest entry of the partition of $n$, $(\lambda_1, \ldots, \lambda_k)$, you would get a shorter partition, $(\lambda_2, \ldots, \lambda_k)$, of $n - \lambda_1$. Alas, because of the condition $\lambda_1 \geq \lambda_2$, we have to 'remember' what $\lambda_1$ was, after kicking it out. So we are **forced** to consider a more general quantity, let's call it $P(n,m)$, enumerating the set of partitions of $n$ whose largest part is **exactly** $m$. Once we can compute this more general quantity, the original object of interest, $p(n)$, is given by

$$p(n) \,=\, \sum_{m=1}^{n} P(n,m) \quad .$$

In order to compute $P(n,m)$ we have the obvious recurrence (alias partial **difference equation**)

$$P(n,m) \,=\, \sum_{m'=1}^{m} P(n-m,m') \quad , \quad n \geq m \geq 1, \qquad (FundamentalRecurrence)$$

subject to the boundary conditions $P(m,m)=1$ and $P(n,m)=0$ if $n<m$. Replacing $n$ by $n-1$ and $m$ by $m-1$ in the above recurrence, and subtracting, one gets the even simpler recurrence

$$P(n,m) = P(n-1,m-1) + P(n-m,m) \quad . \, (SimplifiedFundamentalRecurrence)$$

This gives a **quadratic time** (and quadratic memory) algorithm, $O(N^2)$ for compiling a table of $p(n)$ for $1 \leq n \leq N$.

This is not the most efficient way to compile such a table. An even better way is via Euler's recurrence (e.g. [An], p. 12)

$$p(n) = \sum_{j=1}^{\infty} (-1)^{j-1} \left( p(n-j(3j-1)/2) + p(n-j(3j+1)/2) \right) \quad ,$$

that was famously used by Major Percy MacMahon to compile such a table, that lead to Ramanujan's discovery of his famous congruences (see [An]).

Already Euler considered the enumeration of sets of partitions obeying some *restrictions*. For example the set of partitions into **distinct** parts, let's call it $d(n)$, is given by the generating function ([An], p. 5)

$$\sum_{n=0}^{\infty} d(n) q^n \,=\, \prod_{i=1}^{\infty} (1+q^i) \,=\, \prod_{i=0}^{\infty} \frac{1}{1-q^{2i+1}} \quad .$$

More recently, Rogers and Ramanujan (with the help of MacMahon, see [An] and [S]) considered the problem of enumerating partitions with the property that the difference between consecutive parts is at least 2, i.e. for which

$$\lambda_i - \lambda_{i+1} \geq 2 \quad .$$

The first Rogers-Ramanujan identity states that these numbers, let's call them $d_2(n)$, also have a nice product generating function

$$\sum_{n=0}^{\infty} d_2(n)q^n = \prod_{i=0}^{\infty} \frac{1}{(1-q^{5i+1})(1-q^{5i+4})} \quad .$$

We can say that distinct partitions **avoid** the 'pattern' $[a, a]$ and Rogers-Ramanujan partitions avoid both the pattern $[a, a]$ and the pattern $[a, a-1]$.

This naturally leads to the question of enumerating partitions avoiding an *arbitrary* (finite) set of patterns, but first let's formally define the notion of a 'pattern' in the context of partitions. Note that a commonly used term for our 'pattern' is 'difference condition'. They are equivalent in our setting, but we choose 'pattern' for simplicity.

**Definition 5.1.** *A partition-pattern (pattern for short) is a list* $a = [a_1, \ldots, a_r]$ *of length* $r \geq 1$ *of non-negative integers.*

**Definition 5.2.** *A partition* $\lambda = (\lambda_1, \ldots, \lambda_k)$ **contains** *the pattern* $a = [a_1, \ldots, a_r]$ *if there exists* $1 \leq i \leq k - r$ *such that*

$$\lambda_i - \lambda_{i+1} = a_1 \quad , \quad \lambda_{i+1} - \lambda_{i+2} = a_2 \quad , \quad \ldots \lambda_{i+r-1} - \lambda_{i+r} = a_r \quad .$$

For example, the partition $(7, 6, 5, 4, 4)$ contains the pattern [1] (several times), the pattern [0] (since $4-4 = 0$), the pattern $[1, 1]$ (because of 765 and 654), the pattern $[1, 0]$ (because of 544), the pattern $[1, 1, 1]$ (because of 7654), the pattern $[1, 1, 0]$ (because of 6544), and the pattern $[1, 1, 1, 0]$.

**Definition 5.3.** *A partition* $\lambda$ **avoids** *the pattern* $a$ *if it does* **not** *contain the pattern* $a$.

**Definition 5.4.** *A partition* $\lambda$ **avoids** *the set of patterns* $A$, *if it avoids every pattern in* $A$.

With this language, the class of distinct partitions are those that avoid the pattern [0], while the class of partitions whose differences are at least 2 avoids the set of patterns $\{[0], [1]\}$.

Our goal in 5.2 and 5.3 is to devise an efficient algorithm, that inputs an **arbitrary** set of patterns, $P$, and an arbitrary positive integer $N$, and outputs the first $N$ terms of the sequence enumerating partitions of $n$ avoiding the set of patterns $P$.

A natural approach is to adapt the celebrated Goulden-Jackson [GJ] method to this new context. Since it is based on *sieving* (i.e. 'signed-counting' using the deep identity $1 + (-1) = 0$) we call it a *negative* approach.

The Goulden-Jackson method is lucidly explained (and significantly extended) in the article [NZ]. Recently it has been adapted [EZ] to counting **compositions** avoiding (a different kind of) patterns.

As it turned out, while this 'negative' approach is of considerable *theoretical* interest, it is less efficient than a more straightforward, 'positive', approach, to be described in 5.3.. Readers can feel free to jump to 5.3. without reading 5.2., if the extension of Goulden-Jackson cluster method is not of interest. The rest of the chapter does not depend on 5.2..

In 5.4., we will generalize the efficient, 'positive' approach to account for partitions with more general restrictions (depending on congruence conditions), for example, in Schur's celebrated 1926 theorem (see [An], p. 116), or the more complicated restrictions featuring in Shashank Kanade and Matthew C. Russell's intriguing conjectures ([KS], see also [S], pp. 149-152).

With this efficient, generalized algorithm, we search for new Rogers-Ramanujan type identities (Section 5.5). We should note that computer searches for Rogers-Ramanujan type identities have been around for a while (for a lucid history, see [S] and [MSZ] and [KR]), but our approach is interesting because our notation provides a new way of looking at the restrictions, in addition to the efficiency of the algorithm. Because of the size of our search is quite large, we also make use of the Amarel cluster computing available at Rutgers to make our search even more efficient. For each search, we look at a batch of approximately $2^{27}$ (or less) 'sum-sides' and see which ones 'factor' into nice 'product sides'.

Many new Rogers Ramanujan type have been found using this approach, and at least one of them generalize to an infinite family of identities. This is an ongoing research project, and currently we are working on varying our algorithm and searching for Nandi-type partition identities as well as other types of partition identities.

## 5.2 Adapting the Goulden-Jackson cluster method

Recall that in the Goulden-Jackson Cluster method [GJ][NZ], one finds the weight enumerator for 'marked words' and that turns out to be exactly the same as the target weight enumerator, that is, the weight enumerator for words avoiding a given set of subwords. Since the cluster method involves the *signed* counting of a larger set, and often involves negative numbers, we call it the "negative" approach here. However, in the setting of partitions, we cannot directly use the Goulden-Jackson Cluster method for the following reason:

In the Goulden-Jackson cluster method, one uses the important fact that if one peels off the first letter, or cluster, of a (non-empty) marked word, then the result can be ANY marked word. So we have the following:

$M = \{empty\_word\} \cup V\,M \cup C\,M$

(Note: $M$ is the set of all marked words, $V$ is the alphabet, $C$ is the set of all clusters)

Our basic idea is the same as in the Goulden-Jackson cluster method (we may call it the cluster method for simplicity from now on), however, since we are working with partitions, not words, we need the parts of the partition to be in non-increasing order. Therefore, when we peel off the first letter or cluster of a (non-empty) marked partition, the result is not any marked partition, but a marked partition with possibly a smaller first part such that after adding the cluster or the letter (that we peeled off) in front, it would still be a partition.

We also define weight a little differently than in the cluster method. Recall that in the cluster method, $weight(w, S) = (-1)^{|S|} s^{length(w)}$ ($S$ is the set of marks this word has). Here we define $weight(p, S) = (-1)^{|S|} s^{sum(p)}$ (where $sum(p)$ denotes the sum of the parts of $p$, that is, the integer that $p$ is partitioning.)

In order to use dynamical programming, we define the following:

- $P(A, k, m)$: the set of marked partitions that start with $k$ and having m parts, $A$ being the set of patterns to avoid.
- $C(A, k, l, w)$: the set of clusters starting with $k$, ending with $l$ and of width $w$, $A$ being the set of patterns to avoid.
- $w(P(A, k, m))$: the weight enumerator of $P(A, k, m)$
- $w(C(A, k, l, w))$: the weight enumerator of $C(A, k, l, w)$

Let us start with a marked partition of largest part $k$ and $m$ parts. If the partition is empty ($m = 0$), then the weight enumerator is 1. If $m = 1$, then the weight enumerator is $q^k$. If $m \geq 2$, the first part of the marked partition can be either part of a cluster or not, so for a fixed set of forbidden patterns $A$, we have the following decomposition:

$$P = kP \cup CP'$$

($P$ is the set of all marked partitions that start with $k$, having $m$ parts; $C$ is the set of all clusters starting with $k$, with width no greater than $m$; $P'$ is the set of marked partitions whose first part is no greater than the last part of clusters in $C$). More precisely, for $m \geq 2$, we have:

$$w(P(A, k, m)) = q^k \sum_{r=1}^{k} w(P(A, r, m-1)) + \sum_{l=1}^{k} \sum_{w=1}^{m} (w(C(A, k, l, w)) \sum_{r=1}^{l} w(P(A, r, m-w)))$$

It remains to find $w(C(A, k, l, w))$. In order to do this, we introduce $w(C(A, v, k, l, w))$: the weight enumerator for clusters starting with $k$, with $v$ ($v \in A$) being the first pattern, ending with $l$ and of width $w$, $A$ being the set of patterns to avoid. For example, if $A = \{[2, 1], [1, 1]\}$, consider the cluster $\{8, 6, 5, 3, 2, 1, \{[8, 6, 5], [5, 3, 2], [3, 2, 1]\}\}$. $v$ in this case would be $[2, 1]$ (corresponding to the first mark $[8, 6, 5]$). It is apparent that $w(C(A, k, l, w)) = \sum_{v \in A} w(C(A, v, k, l, w))$. So how do we find $w(C(A, v, k, l, w))$?

For a given cluster, we have two scenarios:

($S1$) if the cluster has only one mark, then the weight for the cluster will just be $(-1) \cdot q^{sum(s)}$ ($s$ being the underlying partition). For example, the cluster $\{3, 2, 1, \{[3, 2, 1]\}\}$ has weight $-q^6$;

$(S2)$ if the cluster has more than one mark, we can peel off the first mark (leaving the overlapping part), and we get a smaller cluster. For example, for the cluster $\{8,6,5,3,2,1,\{[8,6,5],[5,3,2],[3,2,1]\}\}$, after peeling off the first mark, we are left with the cluster $\{5,3,2,1,\{[5,3,2],[3,2,1]\}$. So, $weight(\{8,6,5,3,2,1,\{[8,6,5],[5,3,2],[3,2,1]\}\})$ $= -q^{14}weight(\{5,3,2,1,\{[5,3,2],[3,2,1]\}).$

This is done in similar fashion as in the Goulden-Jackson cluster method. However, because of the nature of our extension, the details are more complicated. The first scenario occurs only if our input has width exactly 1 greater than the length of $v$, and the smallest part to "match" $k$ (the largest part) and the forbidden pattern, that is, $k = l + sum(v)$. To compute the weight for clusters in the second scenario, we first define $OVERLAP$, which takes two partitions $u$ and $v$ and outputs a set of lists. Each list is in the form $[q^i, j]$, where $j$ denotes the number of parts that $u$ and $v$ are overlapping, and $i$ denotes the sum of the parts of $u$ that is not overlapping with $v$. For example, $OVERLAP([4,3,2,2],[2,2,2,1])$ would return $\{[q^7,2],[q^9,1]\}$ because there are two possible ways of overlapping here. (Note: "overlapping" is defined in the usual sense, as in the cluster method, here the two possible overlaps are $[2,2]$ and $[2]$, the power 7 comes from $4+3$, the power 9 comes from $4+3+2$.)

Now, since we are really working with patterns (the $v$ in the input for $C(A,v,k,l,w)$ is a pattern, not a partition), we define $OVERLAP1$ which takes two patterns $u$ and $v$ and two integers $k1$ and $k2$ and let $u1$ and $u2$ be the corresponding partitions that start with $k1$ and $k2$ and with underlying pattern $u$ and $v$ respectively, and use $u1$ and $u2$ as input for $OVERLAP$.

For example, $OVERLAP1([1,1,0],[0,0,1],4,2)$ corresponds to $OVERLAP([4,3,2,2],[2,2,2,1])$ and also outputs $\{[q^7,2],[q^9,1]\}$.

Now we are ready to compute $w(C(A,v,k,l,w))$:

$$w(C(A,v,k,l,w)) = (-1)q^{sum\{v,k\}}(\text{if } k = l + sum(v) \text{ and } w = |v|+1)$$
$$-\sum_{k1=1}^{k}\sum_{u\in A}\sum_{p\in OVERLAP1(v,u,k,k1)} p[1]\cdot w(C(A,u,k1,l,w-|v|-1+p[2]))$$

(Note: $\{v,k\}$ denotes the partition that start with $k$ and has underlying pattern $v$, for example, $\{[0,1],4\} = [4,4,3]$. $|v|$ is the length of the pattern $v$. $p[1]$ denotes the first part of the list $p$, $p[2]$ denotes the second part of $p$.)

In this formula, the part before the minus sign correspond to the first scenario, where the cluster have only one mark, and we will leave it to the reader to verify. If we are in the second scenario (computing the weight of the clusters that have more than one mark), we choose a pattern $u$ from $A$, and a largest part $k1$ ($1 \le k1 \le k$), and $\{u, k1\}$ is chosen to be the second mark of the cluster. We need to find all the ways $\{u, k1\}$ can overlap with $\{v, k\}$ (that is, compute $OVERLAP1(v, u, k, k1)$). Let us use the previous example $\{v, k\} = \{[1,1,0],4\} = [4,3,2,2]$, $\{u, k1\} = \{[0,0,1],2\} = [2,2,2,1]$. There are two ways they can overlap, if the overlap is $[2,2]$, then $p[1]$ would be $q^7$, and $p[2]$ would be 2. After chopping off the $[4,3]$ (that is, chopping off the first mark, leaving the overlapping part [2,2]) we would get a smaller cluster that starts with $k1 = 2$, still ends with $l$, and with width $(w - |v| - 1 + p[2])$, thus the formula above.

**Remark 2.** *One may wonder why we have to include the width as a variable. If we do not, and if [0] or [0,0], or [0,0,0] etc. is in A, then we would have infinitely many clusters (suppose there exist at least one cluster, we can then insert as many marks as we wanted in the middle) and we would have an infitely loop in our program.*

## 5.3 A more straight-forward approach

While, for enumerating words (in a fixed alphabet) avoiding a given set of ‘patterns’ (occurrences of consecutive subwords), the negative approach, pioneered by Goulden and Jackson [GJ] is (usually) more efficient, it turns out that this is not the case for the present problem of counting partitions avoiding the kind of patterns discussed here.

The “positive” approach, to be described in this section, turns out to be much more efficient than the negative approach described in the previous section. Nevertheless, we believe that this partition analog of the Goulden-Jackson method is very elegant and has theoretical interest. It is also possible that it may lead to more efficient approaches.

We use an extension of the dynamical programming approach described in the introduction that gave a quadratic-time and quadratic memory algorithm to compute the original partition sequence $\{p(n)\}$, the iconic OEIS sequence $A41$.

It relied on the obvious fact that removing the largest part, $\lambda_1$, from a partition $\lambda = (\lambda_1, \dots, \lambda_k)$, results in a smaller partition, $\lambda = (\lambda_2, \dots, \lambda_k)$, without extra conditions, except that $\lambda_2 \leq \lambda_1$. That's why in the dynamical programming approach described in the introduction, we were **forced** to compute the more *refined* quantity, with **two** arguments, $P(n, m)$, and that set-up the *recurrence scheme* rolling.

If the set of forbidden patterns, $A$, consists only of patterns of length 1,

$$A = \{[a_1], [a_2], \dots, [a_k]\} \quad ,$$

then the analog of $(FundamentalRecurrence)$ is easy. Let $p_A(n)$ be the number of partitions of $n$ that avoid the patterns in the set $A$, and let $P_A(n, m)$ be the number of such partitions whose largest part is $m$. Then

$$P_A(n, m) \ = \sum_{\substack{1 \leq m' \leq m \\ m - m' \notin \{a_1, \dots, a_k\}}} P_A(n - m, m') \quad , \quad n \geq m \geq 1 \, ,$$

and $p_A(n) = \sum_{m=1}^{n} P_A(n, m)$.

In order to motivate the general case, let's first do a simple special case, where we want to avoid the single pattern $[1, 1, 1]$. In other words, the set of patterns that we want to avoid is the singleton set $A = \{[1, 1, 1]\}$. Consider a typical such partition $\lambda = (\lambda_1, \dots, \lambda_k)$, whose largest part, $\lambda_1$, is $m$. If $\lambda_2 \neq \lambda_1 - 1$, then removing $\lambda_1$ results with the same type of partition, hence the number of partitions of $n$, that we are interested in, with $\lambda_1 = m$ and $\lambda_2 = m'$ is exactly the same as number of such partitions of $n - m$ with largest part $\lambda_2$, since there is a one-to-one correspondence. If you have a good partition of $n - m$ with largest part $m'$, then sticking $m$ in the front can't cause trouble, since $m - m' \neq 1$, so the forbidden pattern $[1, 1, 1]$ can't emerge.

On the other hand if $m' = m - 1$ then we *can* create new trouble. If you have a partition of, $n$, the form

$$(m, m - 1, \lambda_3, \dots, \lambda_k) \quad ,$$

then the 'be-headed' partition,of $n - m$

$$(m - 1, \lambda_3, \dots, \lambda_k) \quad ,$$

must, **in addition** to avoiding the pattern $[1,1,1]$ also avoid the pattern $[1,1]$ at the start. This **forces** us to introduce a new quantity, let's call it $P'_{[1,1,1]}(n,m)$ the number of partitions of $n$ with largest part $m$, avoiding the pattern $[1,1,1]$ *everywhere*, and in addition, avoiding the pattern $[1,1]$ at the very beginning

$$P_{[1,1,1]}(n,m) \;=\; \sum_{\substack{1\le m'\le m\\ m'\ne m-1}} P_{[1,1,1]}(n-m,m') \;+\; P'_{[1,1,1]}(n-m,m-1) \quad .$$

We now need to set-up a scheme for $P'_{[1,1,1]}(n,m)$. If you have a partition of $n$ whose largest part is $m$, avoiding $[1,1,1]$, and in addition avoiding $[1,1]$ at the beginning, and the second largest part is $m'$ with $m-m'\ne 1$, then removing the largest part, $m$, results in a partition of $n-m$ avoiding the pattern $[1,1,1]$, and **no conditions** at the beginning. On the other hand, if $m'=m-1$, then we have a partition of $n-m$ with largest part $m-1$, avoiding $[1,1,1]$, and *in addition*, avoiding the pattern $[1]$ at the beginning. Let $P''_{[1,1,1]}(n,m)$ be the number of such partitions. We have

$$P'_{[1,1,1]}(n,m) \;=\; \sum_{\substack{1\le m'\le m\\ m'\ne m-1}} P_{[1,1,1]}(n-m,m') \;+\; P''_{[1,1,1]}(n-m,m-1) \quad .$$

Similarly

$$P''_{[1,1,1]}(n,m) \;=\; \sum_{\substack{1\le m'\le m\\ m'\ne m-1}} P_{[1,1,1]}(n-m,m') \;+\; P'''_{[1,1,1]}(n-m,m-1) \quad ,$$

where $P'''_{[1,1,1]}(n,m)$ is the number of partitions of $n$ with largest part $m$ avoiding the pattern $[1,1,1]$ and in addition avoiding the empty list, [], at the beginning. But this can never happen so $P'''[1,1,1](n,m)$ is always zero.

Note that we were forced to introduce two auxiliary quantities, $P'(n,m)$, and $P''(n,m)$ that arose naturally.

In general, for any given set of patterns $A$, the computer automatically sets-up a scheme, introducing more general quantities, parameterized, in addition to the set of **global** conditions $A$, by a set of *local* conditions that should be avoided at the very beginning. Then, for each such set of beginning restrictions, $A'$, depending on $m'$, either we are back to only the global conditions, $A$, i.e. the new $A'$ is the empty set, or if

$m - m'$ happens to be one of the starting entries of $A$ or $A'$, the chopped partition, of $n - m$, in addition to obeying the global restrictions of $A$, must obey a brand-new kind of restrictions $A''$. So each 'state' $(m, m', A')$ gives rise to a state $(m', m'', A'')$ for some (possibly empty) set $A''$. Finding these "children" state is automatically done by the computer, setting up a quadratic-time scheme. At the end of the day, we are only interested in the case where $A' = \emptyset$, but we are forced to consider these auxiliary quantitities. Since there are only finitely many of them, and there are still only two arguments (namely $n$ and $m$, where $1 \leq m \leq n$), the algorithm remains quadratic time and quadratic memory.

## 5.4 Generalization

What if we want to count partitions with more specific restrictions, for example, to avoid patterns in the beginning of the partition, not just "globally", or based on congruence conditions?

Recall Shur's celebrated 1926 theorem: the number of partitions of $m$ into parts with minimal difference 3 and with no consecutive multiples of 3 is equal to the number of partitions of $m$ into distinct parts $\equiv 1, 2 \pmod 3$. We are interested in the "sum-side" of this identity, that is, the number of partitions of $m$ into parts with minimal difference 3 and with no consecutive multiples of 3.

And how about more complicated restrictions feathuring in Shashank Kanade and Matthew C. Russell's intriguing conjectures, for example, the sum-side in 3.1.1. of [KR2]:

(a) No consecutive parts allowed.

(b) Odd parts do not repeat.

(c) Even parts appear at most twice. (d) If a part $2j$ appears twice then $2j \pm 3, 2j \pm 2$ are forbidden to appear at all.

(e) 2+2 is not allowed as a sub-partition.

We are motivated to refine and generalize our "positive approach" to account for these fascinating, more generalized creatures, hoping that our efficient algorithm will lead to the discovery of more of them.

In order to generalize our "positive" approach, we introduce some new notions and formulate these sum-sides in a different way, which turns out to be very flexible and also easy to feed to a computer.

Now the input of our algorithm will not just be $A$, $m$ and $n$, but will be the following:

$m$: the largest part of the partition (same as before).

$n$: the integer to be partitioned (same as before).

$A$: the set of patterns to avoid "globally" (same as before).

$Mod$: the list of patterns to avoid according to mod conditions of the largest part of a sub-partition. If $Mod = [A_0, A_1, ..., A_{k-1}]$ (each $A_i$ is a set of patterns), then we are considering modulo $k$ and avoiding the sub-partitions $\{a_0, kj\}$, $\{a_1, kj+1\}$,..., $\{a_k, kj+(k-1)\}$ where $a_i$ can be any pattern in $A_i$ and $j$ can be any positive integer. (Note we are using the notation $\{v, k\}$ to denote the partition that start with $k$ and has underlying pattern $v$, same as the notation at the bottom of page 48.) Equivalently, we are saying: pick any part of the partition (let us call it $m_1$), if $m_1 \equiv i \mod k$, then we want to avoid patterns in $A_i$, starting from $m_1$ going to the right. For example, if $Mod = [\{[0]\}, \{[0,0]\}]$, it means even parts are not allowed to repeat, and odd parts can appear at most twice.

$B$: the set of patterns to avoid at the beginning of the partition.

$IC$: the list of sub-partitions to avoid (we call this "initial conditions"). Let $p_1$ and $p_2$ be partitions. We say $p_1$ is a sub-partition of $p_2$ if $p_1$ (seen as a sequence) is a consecutive subsequence of $p_2$. For example, $[1,1]$ is a sub-partitions of $[2,1,1]$.

Let us look at these notations in light of the two examples above:

Schur: parts with minimal difference 3 translates to $A = \{[0],[1],[2]\}$. No consecutive multiples of 3 translates to $Mod = [\{[3]\},\{\},\{\}]$.

3.1.1. in [KR2]: No consecutive parts allowed translates to the global condition: $A = \{[1]\}$. Odd parts do not repeat, even parts appear at most twice, along with part (d) translate to $Mod = [\{[0,0],[0,3],[0,2],[2,0]\},\{[0],[3,0]\}]$. Part (e), that is, 2+2 is not allowed as a sub-partition translates to IC=[[2,2]]. We do not have any beginning restrictions, so $B = \{\}$.

Having introduced our new language, we now look at the inner-workings of this generalized algorithm. Let $GP(m,n,A,Mod,B,IC)$ be the number of partitions of $n$, with largest part m, and the restrictions $A$, $Mod$, $B$, $IC$ described above. In general, we follow the following steps:

(1) If $m > n$, return 0.

(2) Check if $m$ is equal to the largest part of any of the forbidden sub-partitions in IC: if so (if not, we do nothing), and if the forbidden sub-partition is just [m] then return 0, otherwise we add the underlying partition pattern to $B$ and we have a set of new beginning restrictions $B'$.

(3) If $m = n$, return 1.

(4) If $Mod = \{\}$, then by chopping off the largest part we get the recurrence:

$$GP(m,n,A,Mod,B,IC) = \sum_{\substack{1 \le m' \le m \\ [m-m'] \notin A \cup B'}} GP(m', n-m, A, Mod, B'', IC) \quad .$$

Note the "contributing" $m'$ will be those such that the singleton $[m-m']$ is not in the forbidden patterns (either globally or at the beginning). $B''$ is the set of new beginning restrictions, obtained from $A \cup B'$ by chopping off the difference $m - m'$ from the patterns in $A \cup B'$ (if a pattern in $A \cup B'$ does not contain $m - m'$ as its first part, then we can "forget" about this pattern because there is no contribution there to $B''$).

(5) If $Mod \neq \{\}$, let the length of $Mod$ be $k$. If $m \equiv i \pmod{k}$ then we get the recurrence:

$$GP(m,n,A,Mod,B,IC) = \sum_{\substack{1 \leq m' \leq m \\ [m-m'] \notin A \cup B' \cup Mod[i+1]}} GP(m', n-m, A, Mod, B'', IC) \quad .$$

Note the "illegible" $m'$ will be those such that the singleton $[m - m']$ is not in the forbidden patterns (either globally or at the beginning or according to the mod condition). $B''$ is the set of new beginning restrictions, obtained from $A \cup B' \cup Mod[i+1]$ by chopping off the difference $m - m'$ from the patterns in $A \cup B' \cup Mod[i+1]$ (if a pattern in $A \cup B' \cup Mod[i+1]$ does not contain $m - m'$ as its first part, then we can "forget" about this pattern because there is no contribution there to $B''$).

Let us apply this algorithm to

$$GP(8, 24, \{[1]\}, [\{[0,0],[2,0],[0,2],[0,3]\}, \{[0],[3,0]\}], \{\}, [[2,2]]).$$

(Note the restrictions correspond to the sum-side of 3.1.1. of [KR2].)

$k = 2$, $m = 8 \equiv 0 \pmod{k}$, $m \neq 2$, so we do not need to worry about the first 3 steps, and we are in situation (5): $A = \{[1]\}$; $B' = B = \{\}$; $Mod[i+1] = Mod[1] = \{[0,0],[2,0],[0,2],[0,3]\}$, so $A \cup B' \cup Mod[i+1] = \{[1],[0,0],[2,0],[0,2],[0,3]\}$. $[m - m'] = [8 - m'] \notin A \cup B' \cup Mod[i+1]$ implies $m'$ can be 1,2,3,4,5,6,8. It will be tedious to work out the different cases for 7 different $m's$, and we will just choose $m' = 8$ here for the purpose of illustration. Since $m - m' = 0$, we have $B'' = \{[0],[2],[3]\}$ and $GP(8, 16, \{[1]\}, [\{[0,0],[2,0],[0,2],[0,3]\}, \{[0],[3,0]\}], \{[0],[2],[3]\}, [[2,2]])$.

In theory, this generalized algorithm can also be made quadratic in time and memory, in order to compute a table for the number of partitions of $n$ obeying restrictions

$A$, $Mod$, $B$ and $IC$, using similar ideas as in deriving the *SimplifiedFundamentalRecurrence* on page 43. This is because $A$, $Mod$, $B$ and $IC$ are all finite quantities. We will not go into the details here, but can at least say that this algorithm experimentally run very fast on our computer.

## 5.5 Searching for new partition identities

### 5.5.1 Background and preliminaries

Rogers Ramanujan identities are the following pair of (fascinating!) identities:

$$\sum_{n=0}^{\infty} \frac{q^{n^2}}{(q;q)_n} = \prod_{i=0}^{\infty} \frac{1}{(1-q^{5i+1})(1-q^{5i+4})} \tag{5.1}$$

$$\sum_{n=0}^{\infty} \frac{q^{n^2+n}}{(q;q)_n} = \prod_{i=0}^{\infty} \frac{1}{(1-q^{5i+2})(1-q^{5i+3})} \tag{5.2}$$

Where the "$q$-Pochhammer symbol" $(a;q)_n$ is defined as: $(a;q)_n := \prod_{i=0}^{n-1}(1-aq^i)$.

We refer to the left-hand-side of these two identities as "sum sides" and the right-hand-side as "product sides", for obvious reasons.

These identities also have very nice combinatorial meaning:

For the first identity, $\frac{q^{n^2}}{(q;q)_n}$ is the generating function for partitions with exactly $n$ parts such that adjacent parts have difference at least 2; $\frac{1}{(q;q^5)_\infty(q^4;q^5)_\infty}$ is the generating function for partitions such that each part is congruent to either 1 or 4 modulo 5. Therefore, the first identity is "saying" that the number of partitions of $n$ such that the adjacent parts differ by at least 2 is the same as the number of partitions of n such that each part is congruent to either 1 or 4 modulo 5.

Similarly, the second identity is "saying" that the number of partitions of $n$ such that the adjacent parts differ by at least 2 and such that the smallest part is at least 2 is the same as the number of partitions of $n$ such that each part is congruent to either 2 or 3 modulo 5.

There are many partition identities of this "type", where we have a "sum side" that deals with how parts interact with each other and a "product side" that deals with partitions with parts that fall into certain congruence classes. However, the "type" in "Rogers-Ramanujan type identities" has a deeper meaning (see the introduction of [MSZ]). Also, the terms "sum side" and "product side" may refer to a broader context. In this chapter, we refer to these terms as the following:

"Sum side": a generating function that counts the pattern-avoiding partitions that we are currently interested in (according to $A$, $Mod$ and $I$). May or may not have an analytic (multi-)sum.

"Product side": the side with infinite products.

We use a "list notation" to denote a "product side", for example $[-2,-1,0,1,0]$ denotes $\frac{(q^4;q^5)_\infty}{(q;q^5)^2_\infty (q^2;q^5)_\infty}$. That is, if there are $m$ entries in the list, and the $i$-th entry is $k$, that means we have a factor of $(q^i;q^m)^k_\infty$ on the product side.

As another example, if the list has only $-1$ and $0$ in it, that means the "product side" satisfies certain congruence conditions. For example, $[-1,0,0,-1,0]$ denotes $\frac{1}{(q;q^5)_\infty (q^4;q^5)_\infty}$, that is, the parts are 1 or 4 modulo 5, which is the "product side" of (5.1).

Our goal is to use our efficient algorithm, along with Amarel cluster computing and the use of Frank Gravan's **qseries** package (used to factor the "sum sides") to search for "sum sides" that can be factored into nice "product sides", where "nice" means we only allow $-2$, $-1$, $0$ and $1$ in our list. We include $-2$ because we are also interested in "two-colored" partitions, and we allow 1 because it can be used to cancel out terms in the denominator.

### 5.5.2 Search strategy

Using the efficient generalized algorithm for $GP(m, n, A, Mod, B, IC)$, we programmed $\mathbf{GxnSeq}(\mathbf{N}, \mathbf{A}, \mathbf{Mod}, \mathbf{B}, \mathbf{I})$, which returns the first $N$ terms of the sequence enumerating partitions obeying restrictions according to $A$, $Mod$, $B$ and $I$. And then we programmed the search procedure $\mathbf{Search}(\mathbf{N}, \overline{\mathbf{A}}, \mathbf{Mod}, \mathbf{B}, \mathbf{I}, \mathbf{S})$ that searches partition identities that have "product side" up to modulo $\lfloor N/2 \rfloor$. Here, $\overline{\mathbf{A}}$ is a *set of sets* of forbidden patterns. For example: $\overline{\mathbf{A}} = \{\{[0]\}, \{[1], [2, 2]\}\}$.

For inputs $\overline{\mathbf{A}}, \mathbf{Mod}, \mathbf{B}, \mathbf{I}, \mathbf{S}$, We will search through all sets in $\overline{\mathbf{A}}$, the Cartesian product of the powersets of the sets in $\mathbf{Mod}$, as well as powerset of $\mathbf{I}$. $\mathbf{S}$ is the set of elements allowed to appear in the "list notation" of the "product side". As discussed on the last page, we take $\mathbf{S} = \{-2, -1, 0, 1\}$. And we ususally just assume $\mathbf{B}$ to be $\{\}$ since we are not interested in restrictions at the beginning at this moment.

Let's look at some examples to see how $\mathbf{Search}(\mathbf{N}, \overline{\mathbf{A}}, \mathbf{Mod}, \mathbf{B}, \mathbf{I}, \mathbf{S})$ works.

- $\overline{A} = \{\{[0]\}, \{[1], [2]\}\}$, $Mod = [\ ]$, $I = \{\}$

  –This means we search for partitions that avoid either $A = \{[0]\}$ globally, or avoid $A = \{[1], [2]\}$ globally.

- $\overline{A} = \{\}$, $Mod = [\{[0]\}, \{[0, 0]\}]$, $I = \{[1]\}$

  –This means we search for partitions that avoid nothing globally, but has either $Mod = [\{\}, \{[0, 0]\}]$, $Mod = [\{[0]\}, \{\}]$, $Mod = [\{[0]\}, \{[0, 0]\}]$ or $Mod = [\{\}, \{\}]$ (which is no restriction at all) for $Mod$ restrictions. In addition, we are either avoiding 1 as a part, or nothing at all for "initial conditions".

Thanks to Amarel cluster computing (`https://oarc.rutgers.edu/amarel/`), we are able to split our job into smaller tasks and feed the tasks to 500 nodes and "theoretically" increase our speed by 500 times.

### 5.5.3 Search Space

Let $Comp(m, k)$ be the set of patterns of length at most $m$ (i.e., at most $m$ parts) and largest part at most $k$. For example, $Comp(2, 1) = \{[0, 0], [0, 1], [1, 0], [1, 1]\}$. We use the notation $\{Comp(2, 1), 2\}$ to denote the set of subsets of $Comp(2, 1)$ such that each of the subsets contain 2 patterns or less. That is, $\{Comp(2, 1), 2\} = \{\{[0, 0]\}, \{[0, 1]\}, \{[1, 0]\},$ {[1,1]}, {[0,0],[0,1]},{[0,0],[1,0]},{[0,0],[1,1]},{[0,1],[1,0]},{[0,1],[1,1]}},{[1,0],[1,1]}}.

Below are the a list of 5 essential batches of searches that we conducted.

| | $\overline{\mathbf{A}}$ | **Mod** | **I** |
|---|---|---|---|
| (1) | powerset of $Comp(2, 2)$ | $[Comp(1, 3), Comp(1, 3), Comp(1, 3)]$ | $\{[1], [2]\}$ |
| (2) | $\{Comp(5, 1), 5\}$ | [ ] | $\{[1]\}$ |
| (3) | $\{Comp(4, 2), 5\}$ | [ ] | $\{[1]\}$ |
| (4) | $\{Comp(4, 1), 3\}$ | $[Comp(1, 3), Comp(1, 3), Comp(1, 3)]$ | $\{[1]\}$ |
| (5) | $\{\}$ | $[\{Comp(2, 3), 3\}, \{Comp(2, 3), 3\}]$ | $\{[1]\}$ |

Basically, we have to ensure that we are checking (approximately) at most $2^{27}$ "sum-sides" for Amarel to be able to handle in one day. **Disclaimer**: even for the list above, we did not exhaust everything in them because a few of the tasks ran into problems on Amarel. For most cases we take $N$ to be 30. That is, we search for identities with "product side" with modulo at most 15.

In next section, we will show what new identities we discovered within this search space. Note that it is an incomplete list, as the process involves manually (and it is a tedious process!) look for and collect identities that seem interesting and checking that they are new, often by checking on the OEIS, looking at papers, and asking people. But hopefully the list below will give the readers some ideas of the characteristics of the identities that we are finding.

For one of the identities we are able to extend it to an infinite family. The discovery of the infinite family also involves computer experimentation. We will put this infinite family into a bigger context with some already known identities. Of course, all these identities are still conjectures, but we have checked 100 terms of the generating function to make sure the "sum side" matches with the "product side"–to us this is already very good evidence that these conjectures are true.

### 5.5.4 Discoveries

Needless to say, we discovered many old identities, like Gordon, Andrews-Bressoud, Capparelli, among many others. But many of them are new. Below is an incomplete list. The notation is: A, Mod, I → Product Side.

(1) {}, [{[1], [2]}, {[0], [2], [3]}, {}], {} → 0, 1, 3, 6, 7, 8, 9, 11 *mod* 12

(2) {}, [{}, {[1], [2]}, {[0], [2], [3]}], {[1], [2]} → 0, 3, 4, 5, 6, 9, 11 *mod* 12

(3) {[1]}, [{[0], [3]}, {}, {}], {} → 1, 2, 4, 6, 8, 10, 11 *mod* 12

(4) {[1]}, [{},{[0], [3]}, {}], {[1]} → 0, 2, 3, 4, 6, 9, 10 *mod* 12

(5) {[1]}, [{}, {},{[0], [3]}], {[1]} → 0, 2, 3, 6, 8, 9, 10 *mod* 12

(6) {[1]}, [{}, {[0], [3]},{[0], [3]}], {[1]} → 0, 2, 3, 6, 9, 10 *mod* 12

(7) {}, [{[0, 1]}, {[2], [1, 1]}], {} → 0, 1, 2, 3, 6, 7, 8, 9, 10 *mod* 12

(8) {}, [{[0, 1], [1, 2]}, {[0], [1, 1], [2, 2]}], {} → 0, 1, 3, 4, 7, 8, 9, 10 *mod* 12

(9) {[1, 0], [1, 1, 1]}, [ ], {[1]} → 0, 2, 3, 4, 5, 6, 8, 9, 11 *mod* 12

(10) {[1, 1], [0, 0, 0], [1, 0, 1], [1, 0, 0, 1]}, [ ], {} → 1, 2, 3, 5, 7, 9, 10, 11 *mod* 12

From (9), we obtain its "companion identity" by hand (by "flipping" the product side and **A**, and guessing the initial condition **I**):

{[0, 1], [1, 1, 1]}, [ ], {[1,1], [3,2,1]} $\rightarrow$ 0, 1, 3, 4, 6, 7, 8, 9, 10 *mod* 12

(10) is so far the most exciting one, since we are able to generalize it to an infinite family of identities.

Let us revisit (10): {[1, 1], [0, 0, 0], [1, 0, 1], [1, 0, 0, 1]}, [ ], {} $\rightarrow$ 1, 2, 3, 5, 7, 9, 10, 11 *mod* 12

Observe that the "sum side" of (10) is equivalent to:

–At most 3 occurrences of every part

–For all $i$, not allowed to have $i$, $i+1$, $i+2$ in the partition (they do not have to be consecutive)

This seems to generalize to an infinite family:

– At most $k$ occurrences of any given part

– For all $i$, not allowed to have $i$, $i+1, \ldots, i+k-1$ all as parts in the partition

We have experimentally verified this for many different values of $k$.

The "product sides" of (1) – (10) all correspond to partitions whose parts satisfy certain congruence conditions, or equivalently, only 0 and $-1$ are present in the "list notation". Here are some identities we found that also allow 1 (again, a very incomplete list):

(11) {[0,1,0]}, [{[0]}, {}, {}], {} $\rightarrow$ [-1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, 0] (mod 12)

(12) {}, [{[1], [2]}, {[2]}, {[0], [3]}], {} $\rightarrow$ [-1, -1, -1, 1, 0, -1, -1, -1, -1, 0, 0, -1] (mod 12)

(13) {}, [{[1], [2]}, {[0], [2], [3]}, {[0], [3]}], {} → [-1, 0, -1, 1, 0, -1, -1, -1, -1, 0, 0, -1] (mod 12)

(14) {[1, 2], [2, 1]}, [{}, {[0], [1], [2], [3]}, {[2]}], {[1], [2]} → [-1, 0, -1, 1, 0, -1, -1, -1, -1, 0, 0, -1] (mod 12)

## 5.6 Future work

There are lots of future work to be done for this project. We will list a few below:

(1) Search for larger modulo identities by increasing the **N** in $\mathbf{Search}(\mathbf{N}, \overline{\mathbf{A}}, \mathbf{Mod}, \mathbf{B}, \mathbf{I}, \mathbf{S})$. We have already tried this out for a small batch of inputs, and one identity we found is the following: {[0]}, [{[2], [1, 1]}, {[1, 2], [3, 2]}], {} → [-1, 0, -1, 0, -1, 1, -1, -1, -1, 1, -1, -1, -1, 1, -1, 0, -1, 0, -1, 0] (mod 20). We are hopeful that we will find many more such identities.

(2) Put more variations on the initial conditions.

(3) Incorporate Nandi's $*$ operator (the asterisk in the pattern $[3, 2*, 3, 0]$; for details, please see [N]) into our program to search for more Nandi-type partition identities. It is not hard to adapt our algorithm to look for Nandi-type partitions, in fact, we have already done that. But we would need more insight on "where to look", as some initial searches did not help us find new identities.

(4) Some identities have "wierd" "sum side", for example, a big Göllnitz companion identity (see Theorem A in [AA]) "sum side" requires difference of at least 6 between parts EXCEPT that it is OK if the smallest two parts are 1 and 6. Maybe many such "wierd" partition identites are out there, we would like to modify our algorithm to search for them.

(5) Currently our approach only deals with conditions on contiguous sub-partitions. It will be nice to develop a general frame work/an efficient way to search for identities that avoid sub-partitions that are not necessarily contiguous (like in the infinite family we presented).

# References


[1] A. Baxter, B. Nakamura, and D. Zeilberger, Automatic generation of theorems and proofs on enumerating consecutive-Wilf classes, in *Advances in Combinatorics: Waterloo Workshop in Computer Algebra, W80, May 26-29, 2011* [Volume in honor of Herbert S. Wilf], edited by Ilias Kotsireas and Eugene Zima, Springer, 2013, pp. 121–138.

[2] A. Burstein, Enumeration of words with forbidden patterns, Ph.D. thesis, University of Pennsylvania, 1998.

[3] A. Burstein and T. Mansour, Words restricted by patterns with at most 2 distinct letters, *Electron. J. Combin.* **9** (2002), #R3.

[4] F. N. Davis and D. E. Barton, *Combinatorial Chance*, Hafner, New York, 1962.

[5] S. Elizalde and M. Noy, Consecutive patterns in permutations, *Adv. Appl. Math.* **30** (2003), 110–125 .

[6] I. M. Gessel, Generating functions and enumeration of sequences, PhD thesis, Massachusetts Institute of Technology, 1977.

[7] I. M. Gessel, Symmetric functions and P-recursiveness, *J. Combin. Theory Ser. A* **53** (1990), 257–285.

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

[9] A. Mendes and J.B. Remmel, Permutations and words counted by consecutive patterns, *Adv. Appl. Math*, **37** (2006), 443–480.

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

[11] J. Noonan and D. Zeilberger, The Goulden-Jackson cluster method: extensions, applications, and implementations, *J. Difference Eq. Appl.* **5** (1999), 355–377.

[12] L. Pudwell, Enumeration schemes for pattern-avoiding words and permutations, Ph.D. thesis, Rutgers University, May 2008, `http://faculty.valpo.edu/lpudwell/papers/pudwell_thesis.pdf`

[13] N. J. A. Sloane, *The On-Line Encyclopedia of Integer Sequences*, `http://oeis.org`.

[14] X. Wen, The symbolic Goulden-Jackson cluster method, *J. Difference Eq. Appl.* **11** (2005), 173–179.