2023

Long Increasing Subsequences

Hannah Friedman

Long Increasing Subsequences

Hannah Friedman

Prof. Michael E. Orrison, Advisor

Prof. Peter O. Kagey, Reader

HARVEY MUDD COLLEGE

Department of Mathematics

May, 2023

Abstract

In my thesis, I investigate long increasing subsequences of permutations from two angles. Motivated by studying interpretations of the longest increasing subsequence statistic across different representations of permutations, we investigate the relationship between reduced words for permutations and their RSK tableaux in Chapter 3. In Chapter 4, we use permutations with long increasing subsequences to construct a basis for the space of k-local functions.

Contents

Abstract iii

Acknowledgments xi

Preface 1

1 Introduction 3
1.1 Permutation Statistics 4
1.2 The Symmetric Group 7
1.3 Pattern Avoidance 10

2 Background 13
2.1 RSK Tableaux 13
2.2 Reduced Words and Runs 19
2.3 k-Local Functions: Combinatorially 22
2.4 k-Local Functions: Algebraically 25
2.5 A Change in Perspective 27

3 Reduced Words and RSK Tableaux 33
3.1 Canonical Reduced Words 33
3.2 Some New Terminology 38
3.3 Visualizing Canonical Run Decompositions 39
3.4 From Reduced Words to RSK Tableaux 43
3.5 Future Work 48

4 A Basis for the k-Local Space 51
4.1 Utilizing Full Local Modules 51
4.2 Dimension 55
4.3 1-Local Case 56

4.4 2-Local Case 61
4.5 Future Work 69

Conclusion **71**

Bibliography **73**

List of Figures

1.1 Support of $\mathbf{1}_{(I,J)}$ functions contributing to the excedance function in $\mathfrak{S}_4$. Each box contains the support of an $\mathbf{1}_{(I,J)}$ function which contributes to the excedance function, so the number of excedances of a permutation is the same as the number of boxes it is in. The colors correspond to the position of the excedance, so no permutation can be in two boxes of the same color. 6
1.2 Summing over all possible images or indices gives the all ones function. This figure shows the 1-local case for $\mathfrak{S}_3$. The green bubbles represent mappings of the form $1 \mapsto a$ and the magenta bubbles represent mappings of the form $b \mapsto 1$. Every permutation is in exactly one green bubble and exactly one magenta bubble because the supports of the functions in $\{\mathbf{1}_{(1,a)}\}$ are disjoint and the supports of the functions in $\{\mathbf{1}_{(b,1)}\}$ are disjoint. 8
1.3 2-local representation of $\sigma = 4132$. There is 1 in the $14-21$ entry of the matrix, because $\sigma \cdot (2,1) = (\sigma(2), \sigma(1)) = (1,4)$. The 0s are omitted for clarity. 10

2.1 A visualization of the partial permutation (45, 24) embedded in the permutation 35124. The positions or preimages are in the first row, and their images under the permutation and partial permutation are in the second row. The partial permutation is emphasized with the darker magenta color. 23
2.2 Representation of 4132 acting on ordered pairs, i.e., $\rho^2(4132)$. 32

3.1 Diagram for the reduced word 4321 · 765 · 3456, which is a reduced word for the permutation 51384627. The lines representing runs are drawn to intersect every line corresponding to a number which the run does not fix. The permutation can be read off of Fig. 3.1b by putting j in position i if and only if j is the same color on the left as i is on the right. 40
3.2 Diagrams with and without paths for nested reduced words. 41
3.3 Visualization of how to read insertion tableaux directly from the diagram for a word with decreasing runs. 44

4.1 Visualization of the proof techniques for showing $\mathcal{B}_1, \mathcal{B}_2$ are linearly independent. In the 1-local case, we eliminate matrices with nonzero entries ij with $|i - j| \geq 2$, then those with $|i - j| = 1$, and finally the identity. The 2-local case is almost the same, but the $|i - j| \geq 2$ is split into sub-cases: $|i - j| \geq 3$, which is similar to the $|i - j| \geq 2$ case in the 1-local proof, and $|i - j| = 2$, which is more difficult. This technique is difficult to generalize to larger k because the number of cases, represented by lines, increases with k. 62
4.2 Visualization of the order in which permutations containing mappings with excedance 2 are zeroed out in the proof of Proposition 4.17. 68

List of Tables

2.1 Insertion tableaux (P) and recording tableaux (Q) for permutations in S_3 found using the Robinson-Schensted algorithm. 18
2.2 One-line and cycle notation for permutations in $\mathfrak{S}_3$ with factorizations into adjacent transpositions and corresponding reduced words. 20

Acknowledgments

First, I would like to thank my parents for always supporting my education and for their enthusiasm for my mathematical career. Thank you to my friends at Harvey Mudd for sharing their knowledge with me and for helping me grow as a person. Their kindness and support has been invaluable to me, particularly in finishing this thesis. A special thank you to Aldrin Feliciano, Jasper Bown, and Ian Shors for speaking with me about the ideas in my thesis and for inspiring me mathematically.

Thank you to the Harvey Mudd mathematics department for the many opportunities I've had during my time here. I want to thank in particular Professors Heather Zinn-Brooks, Haydee Lindo, and Jamie Haddock for the time and conversations we've shared. Their mentorship has greatly improved my time at Mudd and has made me more confident heading into the next phase of my life. Thank you to Professor Peter Kagey for his dedication as a second reader, all of the beautiful and playful math he's shared with me, and all the advice he's given me. Finally, I want to thank Professor Michael Orrison for his continued support and mentorship over the past three and a half years. He made me feel connected to Harvey Mudd and excited about mathematics during the pandemic, helped me find numerous opportunities, provided insightful feedback on many applications and various pieces of mathematical writing, cheered me on, and inspired me. I could not have asked for a better mentor.

Preface

I began studying permutation statistics, harmonic analysis, and the symmetric group in the spring of 2020 with my advisor, Professor Michael Orrison, and my friend, Aldrin Feliciano. We spent a lot of our time learning the mathematical background for the project and building a code base we could use to test conjectures and run experiments. We also spent a lot of time thinking about different lenses through which one could view these ideas, and I hope to introduce some of those perspectives in this document as they have shaped my experience with these ideas.

I continued the project, advised by Professor Orrison, in the summer of 2022, and ultimately ended up focusing on class functions and connections to symmetric functions. A lot of ideas we discussed that summer did not tie nicely into the paper we are working on, so some of them appear in this document.

The work I have actually completed during my thesis is in two parts. The part I focused on in the fall, Chapter 4, is work on a conjecture I made last summer. Without getting too technical, I conjectured a basis for a set of nested spaces depending on a parameter k. During my thesis, I have refined the conjecture and proved it for the $k = 1$ and $k = 2$ cases. In the spring, I wanted to work on something totally different and decided to study connections between reduced words and RSK tableaux for permutations; this work is in Chapter 3.

Most of the technical ideas in this document are combinatorial, but there are parts that involve a bit of representation theory. To the reader wishing to better understand these parts, I recommend Sagan (2001). If you're reading this document because you're curious about what I've been working on, I recommend browsing Chapter 3 as the ideas in that chapter require less technical background and, of course, checking out all of the pictures.

With that, I hope you enjoy reading my thesis and I encourage you to reach out if you'd like to talk :)

Chapter 1

Introduction

If I had to rank the colors of the rainbow from favorite to least favorite, I would probably put them in the order blue, green, purple, red, yellow, orange. However, my friend might rank them as yellow, green, blue, purple, red, orange. Now we might ask how similar our rankings are, and there are a number of ways to answer this question. One measure of how similar these are is to look at the biggest subset of colors that we ranked in the same order. In this case, we both ranked blue before purple before red before orange. Since we thought four of the colors should be in the same relative order, maybe our rankings were not so different after all. How different is your ranking of the colors from mine or my friend's?

We have really just asked a question about longest increasing subsequences. The only that is missing is to give the colors an order. Since my friend is imaginary and their feelings won't be hurt if I say they're wrong, we'll say my ordering is the correct one and assign the colors numbers according to my ordering.

Color	Ranking
Red	4
Orange	6
Yellow	5
Green	2
Blue	1
Purple	3

This means that I could encode my ranking as 123456 and my friend's ranking as 521346. These orderings are examples of permutations and the answer to how similar our rankings are is simply the length of the longest increasing subsequence in my friend's ranking, which is 1346.

Long increasing subsequences have a surprising number of connections to other ideas related to permutations. In this thesis, we explore long increasing subsequences from two perspectives. First, how do long increasing subsequences show up in other representations of permutations? Second, how can long increasing subsequences help us better understand other functions on permutations?

1.1 Permutation Statistics

Permutations are ubiquitous, not only in mathematics, but in computer science, biology, physics, and everyday life. One way to study permutations is to ask questions about them. For example, we might ask, "How close is a permutation to being sorted?" Intuition suggests that the permutation 1234567 should be the most sorted list of the numbers from 1 to 7, but beyond that, how do we compare two permutations? Is 2134567 more sorted than 3214567? We can formulate an answer to these questions using a permutation statistic. In general, we define a *permutation statistic* as a function from the set of permutations on $[n] = \{1, 2, \ldots, n\}$ to $\mathbb{C}$. One measure of "sortedness" has already been introduced: the length of the longest increasing subsequence. Another useful statistic for evaluating the "sortedness" of a permutation is the number of inversions, which measures disorder by counting how often a larger number appears before a smaller number. For example, 1234567 has zero inversions: every number is followed only by larger numbers. On the other hand, 2134567 has one inversion, 21, and 3214567 has 3 inversions $32, 31$, and 21. Thus, using inversions as our measure of sortedness, we would conclude that the permutation 2134567 is more sorted than 3214567.

In the last few decades, permutation statistics have become a popular topic in the combinatorics community. If you are interested in connecting with the community, check out the International Conference on Permutation Patterns: https://permutationpatterns.com. There are a lot of beautiful connections between permutations statistics and other branches of mathematics, as well as applications to statistical mechanics and computational biology. See Kitaev (2011) for both a detailed account of these connections and a large collection of further resources. Among classically studied statistics are inversions, excedances, descents, major index, longest increasing subsequence, and number of cycles; see Claesson and Kitaev (2008).

Almost all of these well-known statistics exhibit a property called localness, which was recently introduced in Hamaker and Rhoades (2022). In particular, a function is *k-local* if it can be evaluated at a permutation by looking only at where that permutation maps k-tuples. More concretely, we can define an action of a permutation a k-tuple of letters in $[n]$ by

$$\sigma \cdot (\ell_1, \ell_2, \ldots, \ell_k) = (\sigma(\ell_1), \sigma(\ell_2), \ldots, \sigma(\ell_k))$$

where $\sigma(\ell_i)$ is the number in the ℓ_i position when we write σ as a list. If we can determine the value a function takes on a permutation given only information about how it acts on k-tuples, then the function is k-local. For example, suppose we know that a permutation in $\mathfrak{S}_4$ maps 2-tuples in the following way:

$$\begin{aligned}
(1,2) &\mapsto (2,3)\\
(1,3) &\mapsto (2,4)\\
(1,4) &\mapsto (2,1)\\
(2,3) &\mapsto (3,4)\\
(2,4) &\mapsto (3,1)\\
(3,4) &\mapsto (4,1).
\end{aligned}$$

We can count the number of times a smaller number occurs after a bigger number from this information by simply counting the pairs on the right with a larger number first. Those pairs are $(2,1), (3,1)$, and $(4,1)$, so the permutation has 3 inversions. Since we are able to evaluate the inversion statistic by only looking at this information about ordered pairs, the inversion function is 2-local.

We can formalize this using indicator functions. A function is k-local if it can be written as a linear combination of indicator functions $\mathbf{1}_{(I,J)}$ where I, J are lists of length k without repetition and $\mathbf{1}_{(I,J)}$ is 1 on permutations which map every entry of I to the corresponding entry in J and 0 on all other permutations. For example, $\mathbf{1}_{(1,3)}(3421) = 1$ because 3 is in the first position, but $\mathbf{1}_{(12,32)}(3412) = 0$ because, while 3 is still in the first position, 2 is not in the second position.

Example 1.1. The excedance function is defined as the number of numbers which are greater than their positions in the permutation. For example, 3412 has two excedances, because $3 > 1$ and $4 > 2$, but $1 < 3$ and $2 < 4$. The excedance function is 1-local, as we can evaluate it by counting specific excedances it contains. For example, 3412 contains the excedances $1 \mapsto 3$ and $2 \mapsto 4$. We can write the excedance function as the sum of indicator functions of specific occurrences of excedances:

$$\mathrm{Exc} = \sum_{i=1}^{n} \sum_{j>i} \mathbf{1}_{(i,j)}.$$

In $\mathfrak{S}_4$, the possible excedances are $1 \mapsto 2$, $1 \mapsto 3$, $1 \mapsto 4$, $2 \mapsto 3$, $2 \mapsto 4$, and $3 \mapsto 4$, so

$$\mathrm{Exc} = \mathbf{1}_{(1,2)} + \mathbf{1}_{(1,3)} + \mathbf{1}_{(1,4)} + \mathbf{1}_{(2,3)} + \mathbf{1}_{(2,4)} + \mathbf{1}_{(3,4)}.$$

See Figure 1.1 for an illustration of this idea.

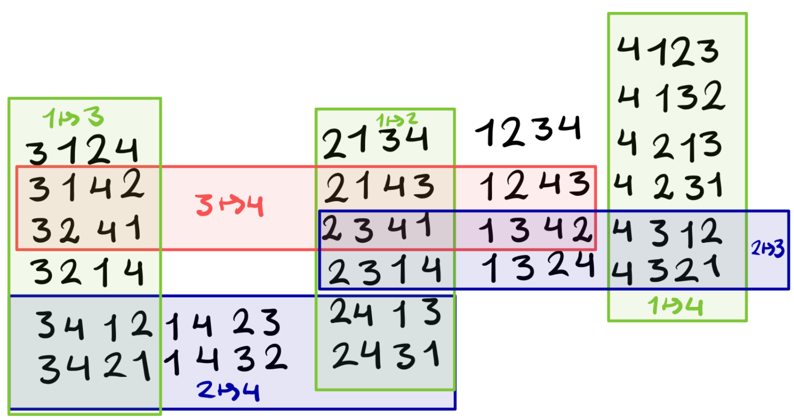


Figure 1.1 Support of $\mathbf{1}_{(I,J)}$ functions contributing to the excedance function in $\mathfrak{S}_4$. Each box contains the support of an $\mathbf{1}_{(I,J)}$ function which contributes to the excedance function, so the number of excedances of a permutation is the same as the number of boxes it is in. The colors correspond to the position of the excedance, so no permutation can be in two boxes of the same color.

Descents, inversions, and major index are all 2-local because they rely on information about pairs. The length of the longest increasing subsequence and the number of cycles in the cycle decomposition of a permutation are not local statistics because they rely on information that cannot be found just from looking at k-tuples. However, these functions do have interesting local analogs such as the number of increasing subsequences of length k or the number of k-cycles in the cycle decomposition. It would be interesting to further explore the relationships between global functions and their k-local analogs.

Permutation statistics have, for the most part, been studied from a combinatorial perspective. However, algebraic techniques are useful for studying permutation statistics because the permutations on n objects form a group called the *symmetric group*, which we denote by $\mathfrak{S}_n$. For example, recent work such as Gaetz and Ryba (2021) and Gaetz and Pierson (2022) has made use of the representation theory of the symmetric group to study the distributions of these functions.

1.2 The Symmetric Group

Because a permutation statistic is a function $f : \mathfrak{S}_n \to \mathbb{C}$, it can be viewed as an element of the group algebra $\mathbb{C}\mathfrak{S}_n$ as

$$\sum_{\sigma \in \mathfrak{S}_n} f(\sigma)\sigma.$$

This connection allows us to use a large variety of well-studied tools associated with the symmetric group. We focus in particular on the representation theory of the symmetric group, an introduction to which can be found in Sagan (2001).

Viewing permutation statistics, and particularly local statistics, through an algebraic lens has a number of advantages. The building blocks for the k-local space, the $\mathbf{1}_{(I,J)}$ functions, are simply indicator functions on two-sided cosets of $\mathfrak{S}_{n-k}$. These indicator functions are quite benign and easy to understand, so they can help us understand more complicated k-local functions as well.

Since a function is k-local if and only if it can be written as a linear combination of $\mathbf{1}_{(I,J)}$-functions, the $\mathbf{1}_{(I,J)}$ functions form a spanning set for the *k-local subring* of $\mathbb{C}\mathfrak{S}_n$. However, it turns out that they are not linearly independent, so they do not form a basis. To see that the indicator functions are linearly dependent, observe that by summing over all possible values or positions, we get the all-ones function on the symmetric group. This is because, in every permutation, a given position has exactly one value and a given value appears in exactly one position. Thus, fixing $a, b \in [n]$,

$$\sum_{i=1}^{n} \mathbf{1}_{(i,b)} - \sum_{j=1}^{n} \mathbf{1}_{(a,j)} = 0,$$

so there exists a nontrivial linear combination of our indicator functions which is zero; see Figure 1.2. While these functions do not form a basis, they are still useful because of how easy they are to describe and understand.

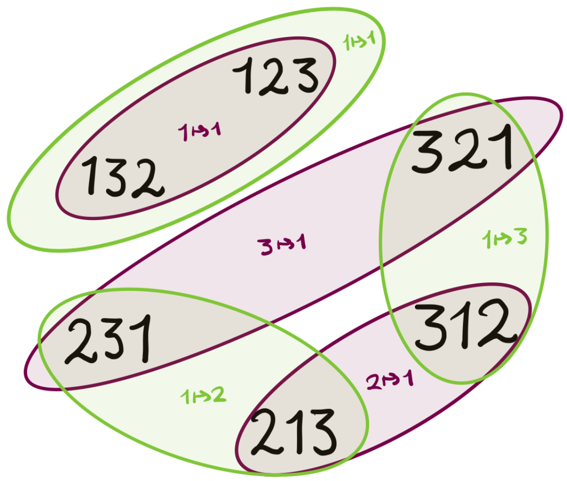


Figure 1.2 Summing over all possible images or indices gives the all ones function. This figure shows the 1-local case for $\mathfrak{S}_3$. The green bubbles represent mappings of the form $1 \mapsto a$ and the magenta bubbles represent mappings of the form $b \mapsto 1$. Every permutation is in exactly one green bubble and exactly one magenta bubble because the supports of the functions in $\{\mathbf{1}_{(1,a)}\}$ are disjoint and the supports of the functions in $\{\mathbf{1}_{(b,1)}\}$ are disjoint.

1.2.1 Permutation Representations

So far, we have considered the properties of permutations using their one-line notations. We can also view permutations as functions. That is, if σ is a permutation, then it is a function $\sigma : [n] \to [n]$ such that $\sigma(i)$ is simply the number in the ith spot of σ when σ is written in one-line notation. With this perspective, a natural representation $\rho^{(n-1,1)} : \mathfrak{S}_n \to M^{n \times n}$ of permutations as $n \times n$ matrices arises such that $\rho^{(n-1,1)}(\sigma)$ has a 1 in the ji-entry whenever

$\sigma(i) = j$ and all other entries are zero. For example,

$$\rho^{(n-1,1)}(4132) = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} & 1 & & \\ & & & 1 \\ & & 1 & \\ 1 & & & \end{pmatrix} \end{array}$$

where the zeros are omitted for clarity. We can extend this map linearly to a map $D^{(n-1,1)} : \mathbb{C}\mathfrak{S}_n \to M^{n\times n}$ so that

$$D^{(n-1,1)}(f) = \sum_{\sigma\in\mathfrak{S}_n} f(\sigma)\rho^{(n-1,1)}(\sigma).$$

We call this the *1-local discrete Fourier transform* (DFT) of a function. While $\rho^{(n-1,1)}$ is a group isomorphism, $D^{(n-1,1)}$ is not a ring isomorphism. However, this map becomes a ring isomorphism if we restrict the domain to the 1-local subring of $\mathbb{C}\mathfrak{S}_n$. We will show this in Chapter 4. Consider, for instance, the 1-local DFT of the excedance function in $\mathfrak{S}_4$:

$$D^{(n-1,1)}(\mathrm{Exc}) = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 6 & 8 & 10 & 12 \\ 12 & 6 & 8 & 10 \\ 10 & 12 & 6 & 8 \\ 8 & 10 & 12 & 6 \end{pmatrix} \end{array}.$$

The excedance function is the unique 1-local preimage of this matrix, but there are other non-local preimages of the same matrix. For instance, the function $f : \mathfrak{S}_n \to \mathbb{C}$ is defined by

$$f(\sigma) = \begin{cases} (n-2)!\left(\frac{n(n-1)}{2} - p\right) & \sigma = (123\cdots n)^p \\ 0 & \text{else} \end{cases}$$

has the same 1-local DFT, but is not 1-local.

These matrix representations are useful because we only need a polynomial number of coefficients to describe local functions, and the entries of the matrices often have nice patterns. For instance, the permutation representation of the excedance function is always a circulant. This can be seen by examining the structure of $D^{(n-1,1)}(\mathbf{1}_{(I,J)})$.

	12	13	14	21	23	24	31	32	34	41	42	43
12						1						
13					1							
14				1								
21											1	
23												1
24										1		
31								1				
32									1			
34							1					
41	1											
42			1									
43		1										

Figure 1.3 2-local representation of $\sigma = 4132$. There is 1 in the $14 - 21$ entry of the matrix, because $\sigma \cdot (2,1) = (\sigma(2), \sigma(1)) = (1,4)$. The 0s are omitted for clarity.

Viewing σ as a function from $[n]$ to $[n]$ gives rise to an action of σ on lists of length 2 with entries from $[n]$. For the most part, we will not allow repetition in our lists. We can define this map so that $\sigma((a,b)) = (\sigma(a), \sigma(b))$. Generating a matrix in the same way as we did previously, but indexing using ordered pairs instead of single numbers gives rise to the *2-local representation* and *DFT*. For example, the 2-local representation of 4132 is shown in Fig. 1.3.

1.3 Pattern Avoidance

Pattern avoidance is a field complementary to the study of permutation statistics. The question is, given a permutation in $\mathfrak{S}_n$ and a permutation in $\mathfrak{S}_k$, is it possible to find a subsequence of the permutation in $\mathfrak{S}_n$ which has the same relative order as the permutation in $\mathfrak{S}_k$? For example, the permutation 13425 contains two occurrences of the pattern 132, given by the subsequences 132 and 142, whereas the permutation 54123 *avoids* the pattern 132 because there is no subsequence abc with the property that $a < c < b$.

The most famous instance of a pattern avoidance problem is Exercise 5 in Knuth (1968), which shows that a list is stack sortable if and only if it avoids the pattern 231. Since this problem appeared, enumerating permutations which avoid a certain pattern has become a popular problem in this field. For example, the Stanley-Wilf conjecture, proposed in the 1980s and proved in 2004, shows that for every pattern σ there exists a number C such that for all n, no more than C^n permutations in $\mathfrak{S}_n$ avoid σ; see Marcus and Tardos (2004).

Searching for avoidance has also recently been generalized to searching for containment, that is, a permutation statistic which counts the number of instances of a pattern in a permutation. The function counting instances of a pattern is k-local if the pattern has length k, which has been used to study the distributions of these functions in Gaetz and Ryba (2021), Gaetz and Pierson (2022), and Hamaker and Rhoades (2022).

1.3.1 Longest Increasing Subsequences

Long increasing subsequences are really just a special permutation pattern. The length of the longest increasing subsequence of a permutation is the largest k such that $123\ldots k$ is contained in the permutation. Longest increasing subsequences were studied even before Knuth popularized the study of pattern avoidance. Perhaps the best known result is the Erdős-Szekeres Theorem, which says that if a permutation is in $\mathfrak{S}_{(r-1)(s-1)+1}$, then it contains either an increasing subsequence of length r or a decreasing subsequence of length s. The study of the distribution of the longest increasing subsequence function has an interesting history and has connections to a number of other ideas in physics and mathematics; see Romik (2015).

There is a remarkable connection between the longest increasing subsequence of a permutation and the Robinson-Schensted correspondence. The Robinson-Schensted correspondence, described in more detail in Section 2.1, is a bijection between permutations in $\mathfrak{S}_n$ and pairs of standard tableaux of the same shape which has a number of convenient properties. A notable property of this correspondence is that the length of the longest increasing subsequence in a permutation σ is the same as the length of the first row of the tableaux associated with σ under the Robinson-Schensted correspondence; see Schensted (1961). An important consequence is that the dimension of the k-local subring is equal to the number of permutations with an increasing subsequence of length $n-k$, which is useful for constructing a basis for the k-local space.

Long increasing subsequences and their connection to RSK tableaux are the threads connecting all of the ideas in my thesis. To better understand long increasing subsequences, we study how they show up across different representations of permutations. Furthermore, we can use permutations with long increasing subsequences to form a basis for the k-local space.

Chapter 2

Background

In this chapter, we provide some background which will be helpful for understanding the remainder of the thesis. Sections 2.1 and 2.2 comprise the background for Chapter 3. Sections 2.1, 2.3, 2.4, and 2.5 comprise the background for Chapter 4.

2.1 RSK Tableaux

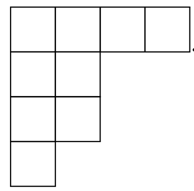
Robinson-Schensted-Knuth correspondence is a relationship between pairs of standard Young tableaux and permutations which plays nicely with longest increasing subsequences. Before understanding this relationship, we first need to understand the combinatorial objects known as standard tableaux.

Given an integer n we say λ is a *partition* of n, denoted $\lambda \vdash n$, if λ is a weakly decreasing list of positive integers which sum to n.

Example 2.1. $(4, 2, 1, 1)$ is a partition of 8. We sometimes write partitions based on how many times each number appears. For example, we could write $(4, 2, 1, 1)$ as $(4^1, 2^1, 1^2)$.

We can use *Young diagrams* to visualize partitions. To construct a Young diagram for a partition $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$, arrange n boxes in k left-justified rows so that row i has λ_i boxes.

Example 2.2. The Young diagram for $(4, 2, 2, 1)$ is



Remark. We draw our diagrams using the English convention. The French convention has the longer rows on the bottom and shorter rows on the top. The Russian convention places the squares on a diagonal by rotating the French version forty-five degrees counterclockwise.

Once we have Young diagrams, we can define Young tableaux. Young tableaux of shape $\lambda \vdash n$ are Young diagrams of shape λ with the integers in $[n] = \{1, \ldots, n\}$ filled into the diagram.

Example 2.3. Here is an example of a tableau of shape $(4, 2, 1)$:

1	7	4	2
3	5		
6			

Usually, we are interested in a specific kind of Young tableau called standard Young tableaux.

Definition 2.1. A Young tableau is *standard* if its rows and columns are strictly increasing, sreading left to right and top to bottom.

Example 2.4. The Young tableau

1	7	4	2
3	5		
6			

is not standard because the first row and second column are not strictly increasing, but the Young tableau

1	2	4	7
3	5		
6			

is standard.

We can now introduce the Robinson-Schensted algorithm, which takes in a permutation in $\mathfrak{S}_n$ and outputs two standard Young tableaux of the same shape. The algorithm works by inserting the images of the permutation into an *insertion tableau* while simultaneously inserting the corresponding indices into a *recording tableau* so that the tableaux are always standard and always have the same shape. The insertion procedure follows the steps in Algorithm 1 and is applied first to $1, \sigma(1)$ and two empty tableaux and then iteratively applied to 2 and $\sigma(2)$, 3 and $\sigma(3)$, all the way up to n and $\sigma(n)$, always using the updated tableaux from the previous run.

Notation. We typically denote the RSK tableaux of a permutation by (P, Q), where P is the insertion tableau and Q is the recording tableau. Let RSK(σ) be the partition of n that is the shape of both P and Q.

Example 2.5. Given the permutation 51342, we will find two standard Young tableaux P and Q using the Robinson-Schensted algorithm. We can find P and Q by inserting the permutation into P. We first insert 5, which leads to the following two partial tableaux for P and Q:

$$P = \begin{array}{|c|}\hline 5 \\ \hline\end{array} \qquad Q = \begin{array}{|c|}\hline 1 \\ \hline\end{array}.$$

The 5 comes from inserting the first letter into P. The 1 comes from the fact that 5 is in the first position. Next, we insert 1. Because $1 < 5$, it will "bump" 5 into the next row. We then insert 5 into the next row, which happens to be empty, and we insert 2 in Q into the same position in which we inserted 5 in P:

$$P = \begin{array}{|c|}\hline 1 \\ \hline 5 \\ \hline\end{array} \qquad Q = \begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline\end{array}.$$

Next, we insert 3. Because $1 < 3$, we do not insert 3 in the for loop, and so we simply insert 3 into the same row as 1:

$$P = \begin{array}{|c|c|}\hline 1 & 3 \\ \hline 5 \\ \cline{1-1}\end{array} \qquad Q = \begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 \\ \cline{1-1}\end{array}.$$

Because $4 > 3$, we insert 4 in the same way:

$$P = \begin{array}{|c|c|c|}\hline 1 & 3 & 4 \\ \hline 5 \\ \cline{1-1}\end{array} \qquad Q = \begin{array}{|c|c|c|}\hline 1 & 3 & 4 \\ \hline 2 \\ \cline{1-1}\end{array}.$$

Because 3 is the first number in the first row which is greater than 2, 2 will bump 3 into the second row when we insert it. Because 3 is less than 5, 3 will bump 5 into the next row. This results in the final RSK tableaux for 51342:

$$P = \begin{array}{|c|c|c|}\hline 1 & 2 & 4 \\ \hline 3 \\ \cline{1-1} 5 \\ \cline{1-1}\end{array} \qquad Q = \begin{array}{|c|c|c|}\hline 1 & 3 & 4 \\ \hline 2 \\ \cline{1-1} 5 \\ \cline{1-1}\end{array}.$$

Algorithm 1 Insertion Procedure for Robinson-Schensted Algorithm

Require: $i \in [n]$
Require: $\sigma(i) \in [n]$
Require: Insertion tableau P
Require: Recording tableau Q
  if P is empty **then**
    $P = \boxed{\sigma(i)}$
    $Q = \boxed{i}$
    return (P, Q)
  end if
  for j in the first row of P **do**
    if $\sigma(i) < j$ **then**
      Replace j with $\sigma(i)$
      Let $\tilde{P}$ be P with the first row removed
      Let $\tilde{Q}$ be Q with the first row removed
      Set $\tilde{P}'$ and $\tilde{Q}'$ to be the output of recursing on $i, j, \tilde{P}, \tilde{Q}$
      Replace all but the first row of P with $\tilde{P}'$
      Replace all but the first row of Q with $\tilde{Q}'$
      return (P, Q)
    end if
  end for
  Append $\sigma(i)$ to the end of the first row of P
  Append i to the end of the first row of Q
  return (P, Q)

We can now state the result that the algorithm described above gives a bijection between permutations and pairs of standard Young tableaux.

Theorem 2.1 (Schensted (1961))**.** *The Robinson-Schensted Algorithm gives a* *bijection between permutations and pairs of standard Young tableau of the same shape.*

Example 2.6. There are 6 permutations in $\mathfrak{S}_3$. The standard young tableaux of size 3 are

$$\begin{array}{|c|c|c|}\hline 1&2&3\\\hline\end{array} \qquad \begin{array}{|c|c|}\hline 1&2\\\hline 3\\\cline{1-1}\end{array} \qquad \begin{array}{|c|c|}\hline 1&3\\\hline 2\\\cline{1-1}\end{array} \qquad \begin{array}{|c|}\hline 1\\\hline 2\\\hline 3\\\hline\end{array}.$$

There is one pair of standard tableaux of shape (3),

$$\left(\begin{array}{|c|c|c|}\hline 1&2&3\\\hline\end{array}, \begin{array}{|c|c|c|}\hline 1&2&3\\\hline\end{array}\right),$$

there are four pairs of shape $(2,1)$:

$$\left(\begin{array}{|c|c|}\hline 1&2\\\hline 3\\\cline{1-1}\end{array}, \begin{array}{|c|c|}\hline 1&2\\\hline 3\\\cline{1-1}\end{array}\right), \left(\begin{array}{|c|c|}\hline 1&2\\\hline 3\\\cline{1-1}\end{array}, \begin{array}{|c|c|}\hline 1&3\\\hline 2\\\cline{1-1}\end{array}\right),$$

$$\left(\begin{array}{|c|c|}\hline 1&3\\\hline 2\\\cline{1-1}\end{array}, \begin{array}{|c|c|}\hline 1&2\\\hline 3\\\cline{1-1}\end{array}\right), \left(\begin{array}{|c|c|}\hline 1&3\\\hline 2\\\cline{1-1}\end{array}, \begin{array}{|c|c|}\hline 1&3\\\hline 2\\\cline{1-1}\end{array}\right)$$

and one pair of shape (1^3):

$$\left(\begin{array}{|c|}\hline 1\\\hline 2\\\hline 3\\\hline\end{array}, \begin{array}{|c|}\hline 1\\\hline 2\\\hline 3\\\hline\end{array}\right).$$

The pairs of standard Young tableau are listed again below with their corresponding permutations in Table 2.1.

RSK tableaux have some useful properties which will rely on later.

Theorem 2.2 (Schützenberger (1963))**.** *If σ has RSK tableaux P and Q, then σ^{-1} has RSK tableaux Q and P.*

This result is useful because it often allows us to prove something in only one case and automatically deduce the other case.

Now you might be wondering, what does any of this have to do with long increasing subsequences? The connection between longest increasing subsequences and RSK tableaux was made when this algorithm was first introduced.

σ	RSK(σ)	P	Q
123	(3)	$\begin{matrix}1&2&3\end{matrix}$	$\begin{matrix}1&2&3\end{matrix}$
132	$(2,1)$	$\begin{matrix}1&2\\3&\end{matrix}$	$\begin{matrix}1&2\\3&\end{matrix}$
312	$(2,1)$	$\begin{matrix}1&2\\3&\end{matrix}$	$\begin{matrix}1&3\\2&\end{matrix}$
231	$(2,1)$	$\begin{matrix}1&3\\2&\end{matrix}$	$\begin{matrix}1&2\\3&\end{matrix}$
213	$(2,1)$	$\begin{matrix}1&3\\2&\end{matrix}$	$\begin{matrix}1&3\\2&\end{matrix}$
321	(1^3)	$\begin{matrix}1\\2\\3\end{matrix}$	$\begin{matrix}1\\2\\3\end{matrix}$

Table 2.1 Insertion tableaux (P) and recording tableaux (Q) for permutations in S_3 found using the Robinson-Schensted algorithm.

Theorem 2.3 (Schensted (1961))**.** *The length of the longest increasing subsequence of σ is equal to the first component of* RSK(σ)*.*

Example 2.7. See Table 2.1. 123 has a longest increasing subsequence of length 3. 321 has a longest increasing subsequence of length 1. All other permutations in $\mathfrak{S}_3$ have a longest increasing subsequence of length 2.

Example 2.8. In Example 2.5, we showed that RSK(51342) = $(3, 1, 1)$. The permutation 51342 has an increasing subsequence of length 3, for example 134, but no increasing subsequence of length 4. So 3 is the length of the longest increasing subsequence and the first part of RSK(51342).

If the length of the first row is the same as the length of the longest increasing subsequence in a permutation, can else can the shape of the RSK tableaux tell us? More work has been done to investigate this relationship; see, for example, work by the authors of Gunawan et al. (2022).

Since the RSK algorithm gives a bijection, we can think of RSK tableaux as another representation of permutations. In the next section, we introduce another way to represent permutations called reduced words. In Chapter 3, we study the relationship between RSK tableaux and reduced words.

2.2 Reduced Words and Runs

In this section, we introduce a new way of representing permutations called a reduced word. As we will see, the length of the longest increasing subsequence of a permutation is intimately connected with a statistic on the reduced words of the permutation.

The set of permutations, $\mathfrak{S}_n$, has a group structure, which means that there are rules for composing permutations in $\mathfrak{S}_n$ to get new permutations. A set of generators for $\mathfrak{S}_n$ is a set of permutations from which we can form the whole group by taking compositions of the elements of the set. Adjacent transpositions are a particularly natural generating set for $\mathfrak{S}_n$.

Definition 2.2. The *adjacent transposition* at i, denoted s_i, is the permutation satisfying

$$s_i(j) = \begin{cases} i+1 & \text{if } j = i \\ i & \text{if } j = i+1 \\ j & \text{else.} \end{cases}$$

In cycle notation, s_i is written as $(i\ i+1)$.

Theorem 2.4 (Sagan (2001))**.** *The adjacent transpositions $s_1, s_2, \ldots, s_{n-1}$ generate $\mathfrak{S}_n$.*

In other words, any permutation in $\mathfrak{S}_n$ can be written as a product of adjacent transpositions. See Table 2.2.

When we write the factorization using the s_i, or simply i, notation rather than the adjacent transposition cycle notation, we obtain a word for the permutation. If the factorization is minimal, i.e. there does not exist a factorization with fewer letters than the given factorization, then the corresponding word is called a reduced word.

Definition 2.3. A word $w = w_1 w_2 \ldots w_\ell$ made from letters in $[n-1]$ is a *reduced word* for a permutation σ if and only if

1. $\sigma = s_{w_1} \circ s_{w_2} \circ \cdots \circ s_{w_\ell}$.

One-line notation	Cycle notation	Factorizations	Reduced Words
123	1	$\emptyset$	[]
132	(2 3)	s_2	2
213	(1 2)	s_1	1
231	(1 2 3)	$s_1 s_2$	12
312	(1 3 2)	$s_2 s_1$	21
321	(1 3)	$s_1 s_2 s_1, s_2 s_1 s_2$	121, 212

Table 2.2 One-line and cycle notation for permutations in $\mathfrak{S}_3$ with factorizations into adjacent transpositions and corresponding reduced words.

2. if another word $w' = w'_1 w'_2 \dots w'_k$ satisfies the above condition, then $k \geq \ell$.

See Table 2.2 for a list of reduced words for all permutations in $\mathfrak{S}_3$. As we can see in the table, reduced words are not unique, but they are unique up to the following relations:

1. (Commutativity Relation) $s_i s_j = s_j s_i$ if $|i - j| > 1$.

2. (Braid Relation) $s_i s_{i+1} s_i = s_{i+1} s_i s_{i+1}$.

Notice that the second condition only becomes relevant if some s_i appears more than once. Following the convention in Gunawan et al. (2022), we call permutations *boolean* when every letter i appears at most once in every reduced word for the permutation. The fact that reduced words are not unique can make them difficult to work with. Gunawan et al. define a canonical reduced word for boolean permutations in Gunawan et al. (2022). In Chapter 3, we recall this definition and attempt to extend it to all permutations.

In working with reduced words, it is helpful to note that the letters we are working with will always be the same. In other words, if supp(σ) denotes the letters in a reduced word for σ, then supp(σ) is well-defined, i.e. it is independent of the reduced word chosen. This is because the braid and commutativity relations do not change the letters in the permutation, only their order and how many times they appear. We also use supp(w) to denote the letters which appear in a reduced word w.

There is one more idea we need to introduce, which relates reduced words back to longest increasing subsequences in permutations.

Definition 2.4. A *run* in a reduced word is a consecutive subword which is monotone and has all adjacent letters. An *increasing run* is a consecutive subword of the form $a(a+1)(a+2)\cdots(a+k)$ and a *decreasing run* is a consecutive subword of the form $(a+k)\cdots(a+2)(a+1)a$. A *run decomposition* of a word partitions the word into the longest possible runs.

Example 2.9. The reduced word 12365437 has three runs: 123, 6543, and 7 and run decomposition $123 \cdot 6543 \cdot 7$.

Definition 2.5. A reduced word for a permutation is called a *minimal run word* if there are no reduced words for the same permutation which can be decomposed into fewer disjoint runs. Let $\text{Run}(\sigma)$ denote the number of runs in a minimal run word for σ.

Example 2.10. The reduced word 615432 has three runs: 6, 1, and 5432. We can use the commutativity relation to rewrite it as 654312 which has two runs: 6543 and 12. Since it is impossible to find an equivalent reduced word with only one run, 615432 is not a minimal run word, but 654312 is.

We are now ready to state the result which relates to longest increasing subsequences of $\mathfrak{S}_n$.

Theorem 2.5 (Gunawan et al. (2022))**.** *If $\sigma \in \mathfrak{S}_n$ and λ_1 is the first part of* RSK(σ)*, then*

$$\text{Run}(\sigma) + \lambda_1 = n.$$

Example 2.11. The word 654312 is a minimal run word for the permutation 2713456. One longest increasing subsequence in 2713456 is 13456, which has length 5, which is what we expect since $\text{Run}(2713456) = 2$ and $2713456 \in \mathfrak{S}_7$.

This result is somewhat remarkable. The conclusion is that the minimum number of runs needed to write a reduced word for a permutation added to the length of the longest increasing subsequence is always n, so one way to better understand the longest increasing subsequence function is to study Run. Our focus will be to extend the canonical reduced word defined in Gunawan et al. (2022) to general permutations and make some progress towards showing that it is a minimal run word. In doing so, we introduce a new kind of diagram for canonical reduced words and give a method of reading RSK tableaux directly off of the reduced words in certain cases.

2.3 k-Local Functions: Combinatorially

Long increasing subsequences also have connections to special kinds of complex-valued functions on the symmetric group. One motivation for studying functions on the symmetric group is to study voting, as permutations can be thought of as total rankings of a set of candidates. In this case, the function of interest might be, for a given permutation, how many people ranked the options in that relative order. However, given the significance of the symmetric group in combinatorics and algebra, functions which depend on properties of the permutations are interesting to study in their own right as well. Generally, we call complex-valued functions on $\mathfrak{S}_n$ which depend on some structural property of the permutation *permutation statistics*. We can think of permutations as orderings of any objects, but we typically think of ordering elements of the set $[n] = \{1, \dots, n\}$. One of the nice properties of this set is that it has a total ordering, which allows us to ask questions about the relative sizes of the objects being ordered. Some examples of functions like this are the number of inversions (Inv), descents (Des), and excedances (Exc). Inv and Exc will serve as useful examples through this work.

Definition 2.6. A permutation σ has an *excedance* at $i \in [n]$ if $i < \sigma(i)$. The statistic Exc is the number of excedances in a permutation, i.e.,

$$\text{Exc}(\sigma) = \#\{i \in [n] \mid i < \sigma(i)\}.$$

Definition 2.7. A permutation σ has an *inversion* at $i, j \in [n]$ if $i < j$ and $\sigma(i) > \sigma(j)$. The statistic Inv is the number of inversions in a permutation, i.e.,

$$\text{Inv}(\sigma) = \#\{(i, j) \in [n] \times [n] \mid i < j, \sigma(i) > \sigma(j)\}.$$

We observe from both of these definitions that the functions are simply counts of occurrences of a phenomenon in the permutation. Looking a little more closely, we can see there is a difference in the amount of information needed to determine whether there is an occurrence of the phenomenon. For example, if I hand you a number $i \in [n]$, you could easily tell me whether a permutation σ has an excedance at i by comparing i and $\sigma(i)$. However, if I just hand you $i \in [n]$, you would have to check all the numbers in $[n]$ to see if i contributed to the inversion count at all. On the other hand, if I hand you $i, j \in [n]$, you can easily tell me whether σ has an inversion at (i, j) by evaluating the permutation at i and j and comparing the values. In other words, while the excedance function encodes information about single entries, the inversion function somehow needs information about ordered pairs.

One challenge that arises when studying functions on permutations is that the symmetric group grows very quickly with n, since its size is $n!$. The idea that the function depends on information given by single numbers or pairs can be used to represent the function in a new way which is much more efficient space-wise. To do so, we define some notation which is used in Hamaker and Rhoades (2022).

Definition 2.8. Let $I = (i_1, \ldots, i_k)$ and $J = (j_1, \ldots, j_k)$ be length-k lists without repetition whose entries are in $[n]$. Then we call the pair (I, J) a *partial permutation* of length k. Let $\mathfrak{S}_{n,k}$ denote the set of all partial permutations of length k with entries in $[n]$.

Example 2.12. $((4,5),(2,4)) \in \mathfrak{S}_{5,2}$ is a length 2 partial permutation with entries drawn from $[5]$. $((5,5),(2,4))$ is not a partial permutation because the first tuple has repetition.

Notation. The ordered-pair-of-lists notation is a bit clunky, so we omit the inner parentheses and commas, writing $(45, 24)$ instead of $((4,5),(2,4))$.

The name partial permutation is suggestive. It is helpful to think of i_ℓ mapping to j_ℓ for all $\ell \in [k]$, so the partial permutation defines a bijection between two size k subsets of $[n]$; see Figure 2.1.

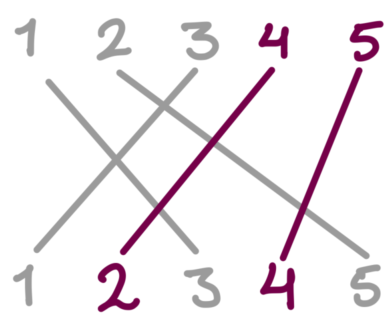


Figure 2.1 A visualization of the partial permutation $(45, 24)$ embedded in the permutation 35124. The positions or preimages are in the first row, and their images under the permutation and partial permutation are in the second row. The partial permutation is emphasized with the darker magenta color.

Partial permutations can help us formalize the idea of looking at single elements or pairs of elements to evaluate functions on the symmetric group.

Consider, for example, the partial permutation $(1, 2)$. If a permutation contains this partial permutation, then we know it has an excedance at 1. Similarly, if a permutation contains the partial permutation $(14, 32)$, then the permutation has an inversion at $(1, 4)$. On the other hand, if a permutation contains $(45, 24)$, then it does not have an inversion at $(4, 5)$. We can evaluate Exc and Inv by asking questions about which elements of $\mathfrak{S}_{n,1}$ and $\mathfrak{S}_{n,2}$, respectively, are contained in a given permutation. We do so using indicator functions.

Definition 2.9. Given a partial permutation $(I, J) \in \mathfrak{S}_{n,k}$, define the indicator function $\mathbf{1}_{(I,J)} : \mathfrak{S}_n \to \mathbb{C}$ to be 1 on all permutations containing (I, J) and zero on all other permutations, i.e.,

$$\mathbf{1}_{(I,J)}(\sigma) := \begin{cases} 1 & \sigma(i_\ell) = j_\ell \text{ for all } \ell \in [k] \\ 0 & \text{else.} \end{cases}$$

Example 2.13. From Figure 2.1, we can see that $\mathbf{1}_{(45,24)}(35124) = 1$. On the other hand, $\mathbf{1}_{(35,34)}(35124) = 0$ because 35124 does not map 3 to 3.

Many functions of interest are linear combinations of these indicator functions. For example,

$$\textsc{Exc} = \sum_{i<j} \mathbf{1}_{(i,j)} \qquad \text{and} \qquad \textsc{Inv} = \sum_{\substack{i<j \\ \ell>m}} \mathbf{1}_{(ij,\ell m)}.$$

To get the excedance function, we sum over all partial permutations (i, j) that represent an excedance, i.e. $i < j$. Similarly, to get the inversion function, we sum over all partial permutations $(ij, \ell m)$ that represent inversions, i.e. $i < j$ and $\ell > m$.

When a permutation statistic can be written as a linear combination of indicator functions on partial permutations in $\mathfrak{S}_{n,k}$, we call the permutation statistic k-local.

Definition 2.10. A complex-valued function on $\mathfrak{S}_n$ is *k-local* if it is in $\operatorname{span}\{\mathbf{1}_{(I,J)} \mid (I, J) \in \mathfrak{S}_{n,k}\}$.

Example 2.14. Exc is 1-local and Inv is 2-local.

This definition can help us prove that functions are k-local without too much difficulty, and can help us build intuition for the k-local space. What makes the idea of a local function so powerful is that it has a second definition which leans heavily on the representation theory of the symmetric group.

Remark. Why do so many well-studied statistics happen to be local? Are they fundamentally easier to study or understand intuitively because they're local? The `FindStat` database is a great computing resource for studying functions on the symmetric group; see Rubey et al. (2022). How many permutation statistics in the `FindStat` database are k-local and for which k?

2.4 k-Local Functions: Algebraically

The representation theory of the symmetric group is both well-studied and currently an active area of research. I recommend as a good starting point Sagan (2001). We introduce some notation here, but assume some familiarity with the representation theory of the symmetric group.

Definition 2.11 (Sagan (2001))**.** The irreducible modules of $\mathfrak{S}_n$, called *Specht modules*, are indexed by partitions of n and denoted by S^λ where $\lambda \vdash n$.

The symmetric group is very playful and fun to work with. The representation theory of the symmetric group has deep relationships to combinatorial objects; see, for example, Stanley and Fomin (1999) and Sagan (2001). The following result is an example of this playful interaction.

Theorem 2.6 (Sagan (2001))**.** *The dimension of S^λ is the number of standard tableaux of shape λ.*

To make use of representation theory of the symmetric group, we observe that there is a natural identification of complex-valued functions on $\mathfrak{S}_n$ with elements of the group ring $\mathbb{C}\mathfrak{S}_n$ via

$$f : \mathfrak{S}_n \to \mathbb{C} \longleftrightarrow \sum_{\sigma \in \mathfrak{S}_n} f(\sigma)\sigma.$$

Now recall that the group ring $\mathbb{C}\mathfrak{S}_n$ is a $\mathbb{C}$-algebra and by the Artin-Wedderburn Theorem there is a $\mathbb{C}$-algebra isomorphism

$$\Psi : \mathbb{C}\mathfrak{S}_n \to \bigoplus_{\lambda \vdash n} M^{f^\lambda \times f^\lambda} \qquad \text{defined by} \qquad f \mapsto \bigoplus_{\lambda \vdash n} \sum_{\sigma \in \mathfrak{S}_n} f(\sigma)\rho_\lambda(\sigma)$$

where ρ_λ is a representation for the Specht module associated with λ and f^λ is the dimension of ρ_λ. The following theorem has been proved in a number of different ways, and we omit the proof here for brevity. The proof that uses the language closest to that of this document can be found in Hamaker and Rhoades (2022), along with references to other papers containing proofs of the statement.

Theorem 2.7 (Hamaker and Rhoades (2022))**.** *A function f with*

$$\Psi(f) = \bigoplus_{\lambda \vdash n} \sum_{\sigma \in \mathfrak{S}_n} f(\sigma)\rho_\lambda(\sigma)$$

is k-local if and only if the components with $\lambda_1 < n - k$ are zero.

Example 2.15. We give a detailed example of how this plays out when $n = 3$. In this case,

$$\mathbb{C}\mathfrak{S}_3 \cong M^{1\times 1} \oplus M^{2\times 2} \oplus M^{1\times 1}.$$

Consider three irreducible representations for $\mathfrak{S}_3$ defined by

$$\begin{aligned} \rho_{(3)}(213) &= \begin{pmatrix} 1 \end{pmatrix} & \rho_{(3)}(132) &= \begin{pmatrix} 1 \end{pmatrix} \\ \rho_{(2,1)}(213) &= \begin{pmatrix} -1 & 0 \\ -1 & 1 \end{pmatrix} & \rho_{(2,1)}(132) &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ \rho_{(1,1,1)}(213)) &= \begin{pmatrix} -1 \end{pmatrix} & \rho_{(1,1,1)}(132) &= \begin{pmatrix} -1 \end{pmatrix}. \end{aligned}$$

Now consider the excedance function on $\mathfrak{S}_3$. We can identify this function with the group algebra element

$$\mathrm{Exc} = 0 \cdot 123 + 1 \cdot 132 + 1 \cdot 213 + 2 \cdot 231 + 1 \cdot 312 + 1 \cdot 321.$$

Since Exc is a 1-local function, we expect that the $\rho_{(1,1,1)}$-component of $\Psi(\mathrm{Exc})$ will be zero. Indeed, we can verify that

$$\begin{aligned} \Psi(\mathrm{Exc})_{(1,1,1)} &= 0 \cdot \rho_{(1,1,1)}(123) + 1 \cdot \rho_{(1,1,1)}(132) + 1 \cdot \rho_{(1,1,1)}(213) \\ &\quad + 2 \cdot \rho_{(1,1,1)}(231) + 1 \cdot \rho_{(1,1,1)}(312) + 1 \cdot \rho_{(1,1,1)}(321) \\ &= 0 \cdot \begin{pmatrix} 1 \end{pmatrix} + 1 \cdot \begin{pmatrix} -1 \end{pmatrix} + 1 \cdot \begin{pmatrix} -1 \end{pmatrix} + 2 \cdot \begin{pmatrix} 1 \end{pmatrix} + 1 \cdot \begin{pmatrix} 1 \end{pmatrix} + 1 \cdot \begin{pmatrix} -1 \end{pmatrix} \\ &= \begin{pmatrix} 0 \end{pmatrix}. \end{aligned}$$

Localness is not particularly interesting when $n = 3$ since there are only three irreducible representations up to isomorphism. In general, the number of nonzero matrices in the image of a 1-local function is 2 and does not grow with n. This is remarkable as it means that the number of coefficients needed to describe a 1-local function grows with n^2 rather than $n!$.

We have defined k-localness for functions on $\mathfrak{S}_n$ by defining k-localness for elements of $\mathbb{C}\mathfrak{S}_n$. This definition extends to elements of $\mathbb{C}\mathfrak{S}_n$-modules as well. Understanding local modules can help us understand the entire k-local space.

Definition 2.12. Let N be a $\mathbb{C}\mathfrak{S}_n$ module with

$$N \cong \bigoplus_{\lambda \vdash n} a_\lambda S^\lambda$$

where the a_λ are nonnegative integers. We say that N is a *k-local module* if $a_\lambda = 0$ for λ with $\lambda_1 < n - k$. We say that N is a *full k-local module* if $a_\lambda = 0$ if and only if $\lambda_1 < n - k$. Furthermore, $f \in N$ is *k-local* if and only if the projection of f into the isotypic subspace of N associated with λ is zero when $\lambda_1 < n - k$.

Example 2.16. The modules $S^{(n)} \oplus S^{(n-1,1)} \oplus S^{(n-2,2)} \oplus S^{(n-2,1,1)}$ and $3S^{(n)} \oplus 2S^{(n-1,1)} \oplus S^{(n-2,2)} \oplus S^{(n-2,1,1)}$ are both full 2-local modules, but $3S^{(n)} \oplus 2S^{(n-1,1)} \oplus S^{(n-2,2)}$ is not because it does not contain any copies of $S^{(n-2,1,1)}$.

Remark. Why don't we care about the multiplicities of the irreducible modules in a full k-local module N? The answer lies in the guarantee of a *symmetry adapted basis*; see Fässler and Stiefel (1992). For a group G, any $\mathbb{C}G$-module can be decomposed as a direct sum of irreducible submodules. We can form a basis for the original module by taking the union of bases for the irreducible submodules. Such a basis is symmetry adapted when a representation with respect to the basis can be written as a block diagonal matrix so that every block is an irreducible $\mathbb{C}G$-representation and representations corresponding to isomorphic irreducible submodules are exactly the same. Thus we only care about whether a block appears at all, rather than how many times it appears.

Our goal will be to find a basis for the k-local space. The k-local space, defined as the set of k-local functions on $\mathfrak{S}_n$, may not be as easy to work with as we would hope. To get around this, we can use an injective homomorphism from the k-local space to the endomorphism ring of a full k-local module. As a consequence, the image of this map will then be isomorphic to the k-local space and so we can work with it instead or working directly with the k-local space.

2.5 A Change in Perspective

How can we bridge the combinatorial and algebraic definitions of k-localness? There is a happy medium which makes use of algebraic tools but has easier combinatorics than the irreducible representations of $\mathfrak{S}_n$. To accomplish this, we will look at the action of $\mathfrak{S}_n$ on two combinatorial structures: tabloids and tuples.

2.5.1 Action on Tabloids

If we let $\mathfrak{S}_n$ act on the combinatorial objects called tabloids, we get a full k-local module. First, let us understand what tabloids are.

Definition 2.13. Define an equivalence relation on tableaux such that two tableaux are equivalent if and only if their row contents is the same. A *tabloid* is an equivalence class under this relation.

Example 2.17. The tableaux

$$\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 5 & 4 \\ \hline 3 \\ \cline{1-1} \end{array}, \quad \begin{array}{|c|c|}\hline 2 & 1 \\ \hline 5 & 4 \\ \hline 3 \\ \cline{1-1} \end{array}, \quad \begin{array}{|c|c|}\hline 1 & 2 \\ \hline 4 & 5 \\ \hline 3 \\ \cline{1-1} \end{array}, \quad \begin{array}{|c|c|}\hline 2 & 1 \\ \hline 4 & 5 \\ \hline 3 \\ \cline{1-1} \end{array}$$

are all equivalent under our relation. The equivalence class of these tableaux is the tabloid

$$\begin{array}{cc}\hline 1 & 2 \\ \hline 4 & 5 \\ \hline 3 \\ \cline{1-1} \end{array}.$$

The notation indicates that we can write the row elements in any order, but if we moved elements between rows, then we would have a different tabloid.

Let $\mathbb{T}(\lambda)$ denote the set of all tabloids of shape λ. We can define an action of $\mathfrak{S}_n$ on $\mathbb{T}(\lambda)$ given by replacing all of the entries i in a tabloid with $\sigma(i)$. For example,

$$(1\,2\,3\,4) \cdot \begin{array}{cccc}\hline 3 & 1 & 5 & 7 \\ \hline 2 & 6 \\ \cline{1-2} 4 \\ \cline{1-1} \end{array} = \begin{array}{cccc}\hline 4 & 2 & 5 & 7 \\ \hline 3 & 6 \\ \cline{1-2} 1 \\ \cline{1-1} \end{array}$$

because $(1\,2\,3\,4) \cdot 1 = 2$, $(1\,2\,3\,4) \cdot 2 = 3$, $(1\,2\,3\,4) \cdot 3 = 4$, $(1\,2\,3\,4) \cdot 4 = 1$ and the other elements are fixed.

Notation. We are using a different notation for permutations here. Until now, we have been using one-line notation to write our permutations, where the position of each number is its preimage. Sometimes, we will also use cycle notation, in which each number appears directly before its image. We typically omit one-cycles, which are fixed points of the permutation. For instance, $(1\,2\,3\,4)$ is the cycle notation for the permutation 2341567.

Example 2.18. Suppose $\lambda = (n-1, 1)$. Since almost all of the numbers are in the same row, when σ acts on $t \in \mathbb{T}((n-1,1))$, the only thing we need to track is how the element in the second row is moved. This means that we can think of the action of $\mathfrak{S}_n$ on $\mathbb{T}((n-1,1))$ as the action of $\mathfrak{S}_n$ on $[n]$ via $\sigma \cdot i = \sigma(i)$.

Example 2.19. The action of $\mathfrak{S}_n$ on tabloids generalizes the action of $\mathfrak{S}_n$ on k-tuples without repetition. If we focus on the action of $\mathfrak{S}_n$ on tabloids of shape $(n-k, 1^k)$, then we recover the action of $\mathfrak{S}_n$ on k-tuples without repetition by focusing on the entries not in the first row.

Given this group action of $\mathfrak{S}_n$ on the set $\mathbb{T}(\lambda)$, we can linearize to create a $\mathbb{C}\mathfrak{S}_n$-permutation module M^λ with basis $\mathbb{T}(\lambda)$. Associated with this module is a representation with rows and columns indexed by the set $\mathbb{T}(\lambda)$.

Example 2.20. Consider the action of $\mathfrak{S}_3$ on $\mathbb{T}((2,1))$. The three tabloids of shape $(2,1)$ are

$$\begin{array}{cc}\hline 2 & 3 \\ \hline 1 & \\ \cline{1-1}\end{array}, \begin{array}{cc}\hline 1 & 3 \\ \hline 2 & \\ \cline{1-1}\end{array}, \begin{array}{cc}\hline 1 & 2 \\ \hline 3 & \\ \cline{1-1}\end{array}.$$

When $\sigma = (1\ 2)$ acts on $\mathbb{T}((2,1))$, it swaps the first two tabloids above and fixes the third one. We can encode this with the matrix

$$[(1\ 2)]_{\mathbb{T}((2,1))} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The first column means that

$$(1\ 2) \cdot \begin{array}{cc}\hline 2 & 3 \\ \hline 1 & \\ \cline{1-1}\end{array} = \begin{array}{cc}\hline 1 & 3 \\ \hline 2 & \\ \cline{1-1}\end{array},$$

the second column means that

$$(1\ 2) \cdot \begin{array}{cc}\hline 1 & 3 \\ \hline 2 & \\ \cline{1-1}\end{array} = \begin{array}{cc}\hline 2 & 3 \\ \hline 1 & \\ \cline{1-1}\end{array},$$

and the third column means that the action of $(1\ 2)$ fixes

$$\begin{array}{cc}\hline 1 & 2 \\ \hline 3 & \\ \cline{1-1}\end{array}.$$

Notation. Call the permutation representation associated with M^λ with tabloids as basis vectors and the action as described above ρ^λ. Call the linear extension of this map to a DFT D^λ.

The module M^λ is well-studied, and its decomposition into Specht modules is governed by a combinatorial rule dealing with semistandard tableaux. A semistandard tableau is a tableau whose contents are strictly increasing along columns and weakly increasing along rows.

Definition 2.14. The *Kostka number* $K_{\lambda\mu}$ counts semistandard tableaux of shape λ and content μ.

Example 2.21. Say $\mu = (3, 2, 1, 1, 1)$ and $\lambda = (4, 2, 2)$. Then the content we need to fill into the semistandard tableau is $(1, 1, 1, 2, 2, 3, 4, 5)$. There are three 1s because $\mu_1 = 3$, two 2s because $\mu_2 = 2$, and one each of $3, 4, 5$ because $\mu_3 = \mu_4 = \mu_5 = 1$. Then we have the following semistandard tableaux of content μ and shape λ:

$$\begin{array}{|c|c|c|c|}\hline 1&1&1&2\\\hline 2&3\\\cline{1-2} 4&5\\\cline{1-2}\end{array},\ \begin{array}{|c|c|c|c|}\hline 1&1&1&2\\\hline 2&4\\\cline{1-2} 3&5\\\cline{1-2}\end{array},\ \begin{array}{|c|c|c|c|}\hline 1&1&1&3\\\hline 2&2\\\cline{1-2} 4&5\\\cline{1-2}\end{array},\ \begin{array}{|c|c|c|c|}\hline 1&1&1&4\\\hline 2&2\\\cline{1-2} 3&5\\\cline{1-2}\end{array},\ \begin{array}{|c|c|c|c|}\hline 1&1&1&5\\\hline 2&2\\\cline{1-2} 3&4\\\cline{1-2}\end{array}$$

and so $K_{\lambda\mu} = 5$.

Theorem 2.8 (Young's Rule)**.** *The multiplicity of S^λ in M^μ is $K_{\lambda\mu}$, i.e.,*

$$M^\mu \cong \bigoplus_{\lambda \vdash n} K_{\lambda\mu} S^\lambda.$$

How does this help us? The big idea is that the information we care about is contained in the modules S^λ with $\lambda_1 \geq n - k$. However, M^μ is an easier module to work with than S^λ, so we want to show that there is a choice of μ for which M^μ contains all of the information we need, i.e., that it is a full k-local module.

Proposition 2.9. *$M^{(n-k,1^k)}$ is a full k-local module.*

Proof. We first show that if $\lambda_1 \geq n - k$, then $K_{\lambda(n-k,1^k)}$ is at least 1. To do so, we need to construct a semistandard tableau of shape λ and content $(n - k, 1^k)$. Since $\lambda_1 \geq n - k$, we can fill all of the $n - k$ 1s into the first row. We can then fill in the remaining entries starting in the first row, moving left to right and then down the rows. Because we are filling in the numbers in order from left to right and top to bottom, all of the columns will be strictly increasing and all rows, but the first will be strictly increasing as well. The first row will still be weakly increasing, so we have constructed a semistandard tableau of shape λ and content $(n-k, 1^k)$.

Next we show that if $\lambda_1 < n-k$, then $K_{\lambda(n-k,1^k)} = 0$. Since the columns of a semistandard tableau must be strictly increasing, all of the ones must be placed in the first row. Creating a semistandard tableau of shape λ and content $(n-k, 1^k)$ would therefore require us to place $n-k$ ones into $\lambda_1 < n-k$ boxes, which is impossible. Thus $K_{\lambda(n-k,1^k)} = 0$. □

Now that we know that $M^{(n-k,1^k)}$ is a full k-local module, we can work with $M^{(n-k,1^k)}$ which gives us the benefit of an algebraic toolbox with the interpretability of its combinatorial definition.

2.5.2 Action on Tuples

Another approach we can take is to work with k-tuples instead of tabloids. Let $X_{k,n}$ denote the k-tuples with letters in $[n]$ and repetition allowed. Define the action of $\mathfrak{S}_n$ on $X_{k,n}$

$$\sigma \cdot (\ell_1, \ell_2, \ldots, \ell_k) \mapsto (\sigma(\ell_1), \sigma(\ell_2), \ldots, \sigma(\ell_k)).$$

When we linearly extend this action to $\mathbb{C}\mathfrak{S}_n$ and $\mathbb{C}X_{k,n}$ we get a module M^k. Denote the permutation representation induced by this action by ρ^k and denote the associated DFT by D^k. See, for example, Fig. 2.2.

Remark. The module M^k is actually isomorphic to the module $M^{(n-1,1)}$ tensored with itself k times. See Doty (2021) for more on this perspective.

To make use of this module, we must show that it is a full k-local module.

Proposition 2.10. *M^k is a full k-local module.*

Proof. M^k is k-local by a result in Hamaker and Rhoades (2022) which implies that tensoring a 1-local module with itself k times gives a k-local module. Furthermore, notice that M^k contains a copy of $M^{(n-k,1^k)}$ if we examine only the tuples without repetition. Since M^k is k-local and contains a full k-local module, M^k is a full k-local module. □

In Chapter 4, we will finish showing that the k-local space is isomorphic to the images of $D^{(n-k,1^k)}$ and D^k and use these images to find a basis for the k-local space.

	11	12	13	14	21	22	23	24	31	32	33	34	41	42	43	44
11						1										
12								1								
13							1									
14					1											
21														1		
22																1
23															1	
24													1			
31										1						
32												1				
33											1					
34									1							
41		1														
42				1												
43			1													
44	1															

Figure 2.2 Representation of 4132 acting on ordered pairs, i.e., $\rho^2(4132)$.

Chapter 3

Reduced Words and RSK Tableaux

There are lots of ways to represent permutations. In this section, we focus on two of these representations: reduced words and RSK tableaux. The motivation for this work is to better understand how longest increasing subsequences appear across different representations of permutations. Schensted showed that the length of the first row of the RSK tableaux of a permutation is the same as the length of the longest increasing subsequence of the permutation; see Schensted (1961). More recently, Gunawan et al. showed that the minimum number of runs over all reduced words for a permutation plus the length of the longest increasing subsequence in the permutation is always equal to n; see Gunawan et al. (2022). For boolean permutations, Gunawan et al. also give an algorithm for constructing a minimal run word, a reduced word which achieves this minimum. The word produced by this algorithm is called a canonical reduced word. Our goal in this section is to extend this algorithm to all permutations and show that the resulting word is a minimal run word. The technique for showing the word is a minimal run word is to relate it to the RSK tableaux. The approach is to show that the runs in the canonical reduced word for a permutation are in bijection with boxes not in the first row of the RSK tableaux of the permutation.

3.1 Canonical Reduced Words

As previously discussed, reduced words are not unique. Gunawan et al. define a canonical reduced word for boolean permutations which is a minimal run word; see Gunawan et al. (2022). This word is the one produced by Algorithm 2. Our goal is to extend this algorithm to all permutations.

Recall that we can move between reduced words for a single permutation using the commutativity and braid relations, i.e., we can swap two letters i and j as long as $|i-j| > 1$ and that $i(i+1)i = (i+1)i(i+1)$. For example, we can move from 234321 to 432123 via

$$243421 \xrightarrow{24=42} 423241 \xrightarrow{41=14} 423214 \xrightarrow{232=323} 432314 \xrightarrow{31=13} 432134.$$

Algorithm 2 Algorithm for Finding a Canonical Reduced Word for a Boolean Permutation

Require: w, a reduced word with each letter appearing at most once
  if $w = []$ **then return** w
  end if
  Let $a := \min(\mathrm{supp}(w))$
  if $a + 1 \notin \mathrm{supp}(w)$ **then**
    Use the commutativity relation to push a to the left of w and recurse on the remaining word
  else if $a + 1$ appears to the left of a in w **then**
    Let b be the largest number such that if $b > c \geq a$, then $c + 1$ appears to the left of c in w
    Use the commutativity relation to push the run $b \cdots (a+1)a$ to the left of w and recurse on the remaining word
  else if $a + 1$ appears to the right of a in w **then**
    Let b be the largest number such that if $b > c \geq a$, then $c + 1$ appears to the right of c in w
    Use the commutativity relation to push the run $a(a+1)\cdots b$ to the right of w and recurse on the remaining word
  end if

Boolean permutations are permutations whose reduced words consist of all distinct letters. The idea behind Algorithm 2 is to consolidate runs in a permutation by pushing decreasing runs to the left and increasing runs to the right. For example, the word 142 has three runs, but we can consolidate the 1 and 2 into a run by pushing 1 to the right, giving the word 412. Similarly, if we started with the word 241, we could create the decreasing run 21 by pushing 1 to the left to get 214.

Notation. Given an increasing run $r = a(a+1)(a+2)\cdots(a+k)$ or a decreasing run $r = (a+k)\cdots(a+2)(a+1)a$, define $\min(r) = a$ and $\max(r) = a+k+1$. We define the maximum so that it is the biggest number which is not fixed by r.

Example 3.1. We give an example of applying Algorithm 2 to a boolean permutation. Suppose we start with the word $w = 67154983$. The minimum letter in the word is 1. Since the word does not contain 2, we simply push 1 to the left to get $1 \cdot 6754983$. We can do this because 1 commutes with 6 and 7. We now recurse on 6754983. The new minimum letter is 3. Since 4 appears to the left of 3, we will push a run to the left. In this case $b = 6$, so we decompose 6754983 into $6543 \cdot 798$ and the full word is $1 \cdot 6543 \cdot 798$. We now recurse on 798. Since 7 is the minimum letter and 8 appears to the right of 7, we will push a run to the right. In this case, $b = 8$, so we decompose 798 as $9 \cdot 78$ and recurse on 9. Since 9 is now the minimum letter we push it to the left and recurse on the empty word, which remains the same. This leaves us with the final canonical word $1 \cdot 6543 \cdot 9 \cdot 78$.

Theorem 3.1 (Gunawan et al. (2022))**.** *The canonical reduced word for a boolean permutation produced by Algorithm 2 is unique and is a minimal run word.*

In extending this algorithm to words in which letters may appear more than once, one feature we would like to preserve is that, at the end, all of the runs in the run decomposition of the canonical word have unique minima. Guaranteeing unique minima is a little trickier in the general case, since the minimum of the support of the word may appear more than once in our starting word. The following lemma guarantees the existence of a word in which the minimum of the support appears exactly once.

Lemma 3.2. *Given a permutation σ, there exists a reduced word for σ in which* $\min(\mathrm{supp}(\sigma))$ *appears exactly once.*

Proof. We use induction on the length of the reduced words for σ. Let w be an arbitrary reduced word for σ. If w has length 1, then the minimum letter in w appears exactly once.

Let $a = \min(\mathrm{supp}(\sigma))$. If $w = w_0w_1\dots w_n$ has length n, then split w into two words: $w' = w_0w_1\dots w_i$ and $w'' = w_{i+1}\dots w_n$ so that w_i is the leftmost occurrence of a. If w'' does not contain a, then we are finished. If it does, then by the inductive hypothesis there exists an equivalent reduced word for w'' in which a appears exactly once. Concatenate this word with w' to obtain a new reduced word for σ, denoted w^*, in which a occurs exactly twice.

Let r be the word we get if we take the letters between the two occurrences of a. The word r must contain $a+1$ at least once. If it did not, then a would commute with everything between the two occurrences of a and we could cancel the two a's, which would contradict the assumption that we started with a reduced word. Also, the word r cannot contain any letters less than $a+1$, because this would contradict either a being minimal or w^* containing exactly two occurrences of a. Thus $a+1$ is the minimum element of r, which is a shorter word than s and so there exists an equivalent word r^* which contains $a+1$ exactly once. Replace r with r^* and call the resulting word w^{**}. The word w^{**} is equivalent to w. Because a commutes with all of the letters in w^{**} which are not $a+1$, we can use the commutativity relation to move each occurrence of a to either side of $a+1$. From there, we can use the braid relation to convert $a(a+1)a$ into $(a+1)a(a+1)$. This final word is equivalent to w. □

We illustrate the procedure described in the proof with an example.

Example 3.2. Suppose we start with the reduced word $w = 21234231231$. We first split this word into $w' = 21$ and $w'' = 234231231$. Since w'' contains 1 twice, we need to recurse. We can split w'' into $p' = 34321$ and $p'' = 231$. Since p'' already contains 1 only once, we can move on to using our relations to eliminate one of the 1s in w''. The word between the two occurrences of 1 in w'' is 23, which only contains 2 once. Thus we can replace w'' with 234232123 by first switching the 1 and 3 at the end and using the braid relation on 121. Then $w^* = 21234232123$ contains 1 exactly twice. The word between the two occurrences of 1 is $r = 234232$, which contains 2 three times. Split r into $r' = 2$ and $r'' = 34232$. We can replace r'' with 34323 to get $r^* = 234323$. The word between the two 2s is 343, which we can replace with 434 to get 243423. Using commutativity, we move the 2s next to the 3 to get 423243 and using another braid relation gives us 432343. Substituting this into w^* for r results in $w^{**} = 21432343123$. We can now move the 1s next to the 2: 24312134323. Finally, we apply the braid relation once more to obtain 24321234323.

Another choice we need to make in generalizing this algorithm is if $a+1$ appears on both sides of a, which direction do we push the run? My intuition is that we always want to produce the longest run possible. Using this strategy and pushing ties to the left yields Algorithm 3.

Example 3.3. We give an example of applying Algorithm 3. Let $w = 1234325436$. The minimum letter in w is 1. Since w contains 2, we need to

Algorithm 3 Algorithm for Finding a Canonical Reduced Word for All Permutations

Require: w, a reduced word
  if $w = []$ **then return** w
  end if
  Let $a := \min(\text{supp}(w))$
  Let w be an equivalent reduced word which contains a exactly once
  if $a + 1 \notin \text{supp}(w)$ **then**
    Use the commutativity relation to push a to the left of w and recurse on the remaining word
  else
    Let b_ℓ be the largest number such that $b_\ell, \ldots, a+1, a$ is a (not necessarily consecutive) strictly decreasing subsequence of w
    Let b_r be the largest number such that $a, a+1, \ldots, b_r$ is a (not necessarily consecutive) strictly increasing subsequence of w
    if $b_\ell \geq b_r$ **then**
      Push the run $b \cdots (a+1)a$ to the left of w and recurse on the remaining word
    else if $b_r > b_\ell$ **then**
      Push the run $a(a+1) \cdots b$ to the right of w and recurse on the remaining word
    end if
  end if

compute b_r and b_ℓ. Notice that $b_\ell = 1$, since there are no numbers to the left of 1. However, $b_r = 6$, since the longest increasing subsequence of the word starting at 1 is 123456 (highlighted in red), so we will push this run to the right:

$$1234325436 \xrightarrow{343=434} 1243425436 \xrightarrow[42=24]{124=412} 4123245436 \xrightarrow{232=323}$$
$$4132345436 \xrightarrow{13=31} 4312345436 \xrightarrow{454=545} 4312354536 \xrightarrow[53=35]{1235=5123}$$
$$4351234356 \xrightarrow{343=434} 4351243456 \xrightarrow{124=412} 4354123456.$$

We then recurse on the word 4354. The minimum letter is 3. Here, $b_r = b_\ell = 4$. Since we have a tie, we will push the run to the left: $43 \cdot 54$. We recurse on 54, which is already a run, so we are finished. Thus the word produced by our algorithm is $43 \cdot 54 \cdot 123456$.

Notice that if we apply Algorithm 3 to a boolean permutation, the procedure is exactly the same as in Algorithm 2. Many of the results in this section could be strengthened if one could prove the following conjecture.

Conjecture 3.3. *The canonical reduced word produced by Algorithm 3 is unique and is a minimal run word.*

Using an inductive strategy, the proof comes down to showing that the maximal increasing or decreasing consecutive subsequence containing the minimum letter in the word is the same across all equivalent words. This would involve showing that we cannot extend such a sequence using the commutativity and braid relations.

3.2 Some New Terminology

To prove that their canonical word is a minimal run word, Gunawan et al. prove that RSK tableaux can be read directly off of their canonical reduced words for boolean permutations. We extend this result in special cases where the shape of the RSK tableaux of a permutation is a hook, i.e., the RSK tableaux have shape $(n-k, 1^k)$. Because the tableaux must be standard, we need only identify which elements are in the first row and which are not.

We focus on a simplified case in which all of the runs are either increasing or decreasing. Our goal for the remainder of this chapter will be to characterize hook-shaped RSK tableaux and write them down directly from the reduced words of their permutations. To state our main result, we first need the following definitions.

Definition 3.1. We say a reduced word is *monotone* when its run decomposition has only increasing runs with strictly decreasing minima or has only decreasing runs with strictly increasing minima.

We can now define what we mean by a nested permutation.

Definition 3.2. Suppose w is a monotone word with run decomposition $w = r_1 \cdot r_2 \cdots r_k$. If the r_i are all decreasing, then w is a *nested word* if r_i is a subword of r_{i-1} for all $i > 1$; that is $\max(r_i) \leq \max(r_{i-1})$ for all $i > 1$. If the r_i are all increasing, then w is a *nested word* if r_i is a subword of r_{i+1} for all $i < k$; that is $\max(r_i) \leq \max(r_{i+1})$ for all $i < k$. We call a permutation *nested* if it has a nested reduced word.

Note that the strictness condition on the minima guaranteed by the word being monotone ensures that the runs will always be proper subwords of one another.

Section 3.4 will be devoted to proving the following result:

Proposition 3.4. *If σ has a nested reduced word which decomposes into k runs, then* $\mathrm{RSK}(\sigma) = (n-k, 1^k)$.

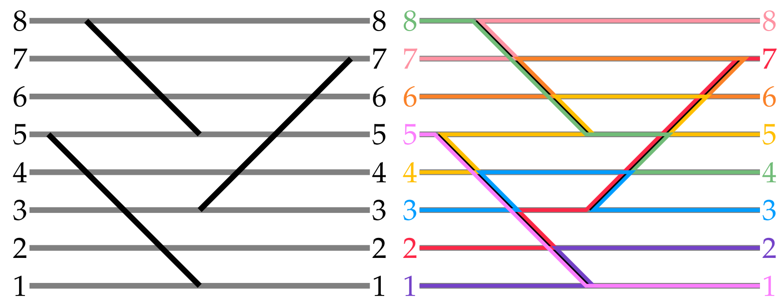
This result also implies that nested words are always minimal run words.

3.3 Visualizing Canonical Run Decompositions

In this section, we introduce a way to visualize reduced words using their run decompositions. These diagrams are not necessary to an understanding of the result, but I want to include them because they provide intuition for how these objects work and offer a visual framework that links permutations to their reduced words. Additionally, these diagrams eliminate some of the asymmetry of reduced words because they don't bias towards the smaller element in a transposition. Finally, these diagrams are interesting to study in their own right. See some of the questions at the end of the section for ideas.

We only define these diagrams for particular kinds of reduced words, namely those that might have been produced by Algorithm 3. These words will have all of the decreasing runs on the left and all the increasing runs on the right, the minima of the runs will all be unique, the minima of the decreasing runs will be increasing and the minima of the increasing runs will be decreasing. In other words, if the word has run decomposition $r_1 \cdot r_2 \cdots r_k$, there exists an i such that $r_1 \cdots r_i$ is monotone decreasing and $r_{i+1} \cdots r_k$ is monotone increasing. While not every word satisfies this property, for any word there exists an equivalent word which does satisfy this condition. This is guaranteed by the result that super-Yamanouchi words exist for every permutation; see Section 3.5.

Given a word that satisfies the conditions in the above paragraph whose maximum letter is $n-1$, we create a grid of horizontal lines labeled $1, 2, \ldots, n$ with 1 at the bottom and n at the top. Add a diagonal line for each run r in the run decomposition between the lines labeled with $\min(r)$ and $\max(r)$ so that the lines have slope 1 if r is increasing and -1 if r is decreasing. Finally, place the lines in the grid so that the minima are aligned vertically; see Fig. 3.1a. See also Fig. 3.2a and Fig. 3.2c for examples of diagrams for nested decreasing and increasing words, respectively.



a. Diagram for the reduced word $4321 \cdot 765 \cdot 3456$.

b. Diagram for the reduced word $4321 \cdot 765 \cdot 3456$ with paths drawn.

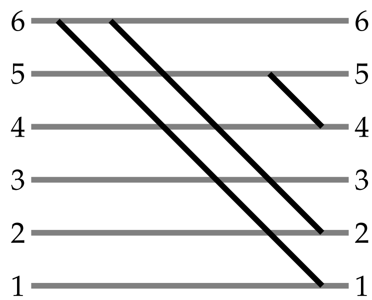
Figure 3.1 Diagram for the reduced word $4321 \cdot 765 \cdot 3456$, which is a reduced word for the permutation 51384627. The lines representing runs are drawn to intersect every line corresponding to a number which the run does not fix. The permutation can be read off of Fig. 3.1b by putting j in position i if and only if j is the same color on the left as i is on the right.

We can read the permutation off of these diagrams as well. When reading this explanation, it will be helpful to keep Fig. 3.1b, Fig. 3.2b, and Fig. 3.2d in mind.

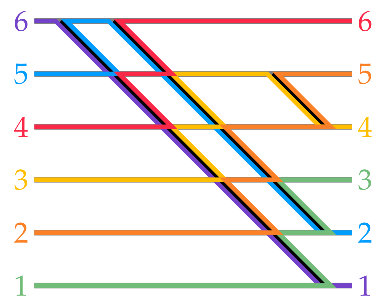
To read the permutation off of the diagram, think of the right endpoints of the grid as the domain of the permutation and the left endpoints as the codomain of the permutation. To determine where a letter a is mapped by the permutation, begin on the gridline labeled with a on the right. Follow this line until you encounter a run. If the intersection is the right endpoint of the run, follow the run all the way to the left endpoint and continue on that gridline. If the intersection not the right endpoint, follow the run one unit up or down to the right and follow that gridline until the next intersection. The final line you end up on is the image of the letter you started with under σ.

One way to reason about why this works involves a shift in perspective. Each run is an *adjacent cycle*; if it is a decreasing run, then it is a cycle of the form $(a+k+1 \ \cdots \ a+1 \ a)$. When this cycle is applied to letters in $\{a, a+1, \ldots, a+k\}$, then a will jump up to $a+k+1$ and everything else will shift down one unit. This is why we follow the run all the way up if we intersect its minimum and down one unit otherwise. The same reasoning can be applied to increasing runs.

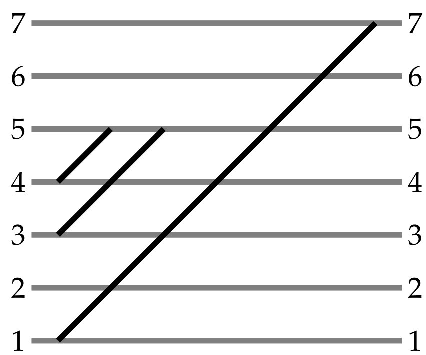
Through this process, we have traced a *path* over our diagram which



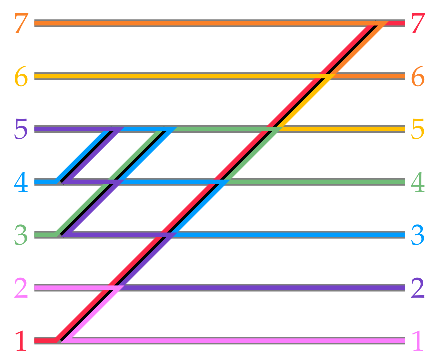
a. Diagram for the reduced word 54321 · 5432 · 4.



b. Diagram for 54321 · 5432 · 4 with paths drawn.



c. Diagram for the reduced word 4 · 34 · 123456.



d. Diagram for 4 · 34 · 123456 with paths drawn.

Figure 3.2 Diagrams with and without paths for nested reduced words.

moves from a number on the right to a number on the left following gridlines and runs. We call the point on the right at which the path begins the *right endpoint* of the path, and the point on the left at which the path ends the *left endpoint*. If a path intersects the rightmost point of some decreasing run, we call the path a *ladder* since it climbs all the way up the run. Analogously, we define *chutes* as paths which intersect the rightmost point of some increasing run. From here on out, we will focus on the monotone case. In this scenario, we will never have chutes and ladders in the same diagram. Paths in diagrams for decreasing monotone words which are not ladders are called *non-ladders*, and paths in diagrams for increasing monotone words which are not chutes are called *non-chutes*.

Now that we have defined these diagrams, we can start to prove things about them that will help us prove our main result.

Lemma 3.5. *Suppose w is a nested decreasing word with run decomposition $r_1 \cdot r_2 \cdots r_k$. If a path intersects r_i, then it also intersects but does not climb r_j for $j < i$.*

Proof. It is sufficient to show that the path intersects but does not climb r_{i-1}. Let g be the label of the gridline that the path is on after it intersects r_i. The highest gridline the path can be on after intersecting r_i is $\max(r_i) \leq \max(r_{i-1})$. Similarly, the lowest gridline the path can be on after intersecting r_i is $\min(r_i) > \min(r_{i-1})$. Thus

$$\min(r_{i-1}) < \min(r_i) \leq g \leq \max(r_i) \leq \max(r_{i-1}).$$

Since $\min(r_{i-1}) < g \leq \max(r_{i-1})$, the path will intersect r_{i-1} at a point which is not its right endpoint. Thus the path must intersect r_{i-1} without climbing it. □

The following lemma will help us determine the RSK tableaux from the diagrams by extracting information about increasing subsequences from the diagrams.

Lemma 3.6. *Let σ be a permutation with a nested decreasing reduced word w. Then the left endpoints of ladders form a decreasing subsequence of σ and the left endpoints of non-ladders form an increasing subsequence of σ.*

Proof. Suppose that w is nested with run decomposition $r_1 \cdot r_2 \cdots r_k$. We show that if $a < b$ are the right endpoints of ladders A and B then $\sigma(b) < \sigma(a)$. Let r_i, r_j be the runs with $\min(r_i) = a$ and $\min(r_j) = b$. Since $a < b$, $i < j$. Because $i < j$ and the runs are nested, $\max(r_i) \geq \max(r_j)$. By Lemma 3.5, the left endpoint of A is $\sigma(a) = \max(r_i) - (i-1)$ and the left endpoint of B is $\sigma(b) = \max(r_j) - (j-1)$, since there are $i-1$ runs to the left of r_i and $j-1$ runs to the left of r_j. Combining $\max(r_i) \geq \max(r_j)$ with $i < j$ gives $\max(r_i) - i > \max(r_j) - j$ which implies that

$$\sigma(a) = \max(r_i) - (i-1) > \max(r_j) - (j-1) = \sigma(b)$$

as desired.

We also show that if $a < b$ are right endpoints of non-ladders A and B, then $\sigma(a) < \sigma(b)$. Let r_i be the first run that A intersects and r_j be the first run B intersects, reading right to left. After A intersects r_i, it will be on the gridline $a-1$. After B intersects r_j, it will be on the gridline $b-1$. By

Lemma 3.5, the left endpoint of A is $\sigma(a) = a-1-(i-1) = a-i$. Similarly, the left endpoint of B is $\sigma(b) = b - j$. If $i \geq j$, then $\sigma(a) = a - i < b - j = \sigma(b)$ as desired. If $i < j$, the runs which intersect the b gridline but not the a gridline (not including r_j) can have minima between $a+1, a+2, \ldots, b-1$. Thus the maximum number of runs strictly between r_i and r_j is $j-i-1 \leq (b-1)-(a+1)$. We can rewrite this as $i \geq a - b + 1 + j$. Thus

$$\sigma(a) = a - i \leq a - (a - b + 1 + j) = b - j - 1 < b - j = \sigma(b).$$

□

Now all we need to do is relate left endpoints of ladders to left endpoints of non-ladders.

Lemma 3.7. *Suppose w is a nested reduced word for $\sigma \in \mathfrak{S}_n$. If $a < b$ so that a and $b+1$ are right endpoints of non-ladders and b is the right endpoint of a ladder, then $\sigma(a) < \sigma(b+1) < \sigma(b)$.*

Proof. Using Lemma 3.6, it is sufficient to show that $\sigma(b) > \sigma(b+1)$. Let B and B' be paths in the diagram for w such that $b, b+1$ are right endpoints of B and B', respectively. Let r be the run climbed by B. Since B' is a non-ladder, the first run it intersects must be r. Since B is on the max(r) gridline after intersecting r, B will be above B' both paths have intersected r. By Lemma 3.5, the left endpoint of B will remain greater than the left endpoint of B'. Thus $\sigma(b) > \sigma(b+1)$. □

3.4 From Reduced Words to RSK Tableaux

We can now state a more detailed version of Proposition 3.4. This version will give us the actual hook tableaux, not just the shapes. Since the tableaux are standard, it is sufficient to specify the entries not in the first row. See Fig. 3.3 for a visualization of how the diagrams for decreasing nested permutations relate to the insertion procedure for RSK tableaux.

Lemma 3.8. *If σ has a nested word, its RSK tableaux are hook-shaped.*

(1) *If σ has a decreasing nested reduced word*

 (a) *the entries not in the first row of the P tableau are the left endpoints of ladders.*

 (b) *the entries not in the first row of the Q tableau are the right endpoints of ladders plus one.*

(2) *If σ has a decreasing nested reduced word*

(a) *the entries not in the first row of the P tableau are the left endpoints of chutes plus one.*

(b) *the entries not in the first row of the Q tableau are the right endpoints of chutes.*

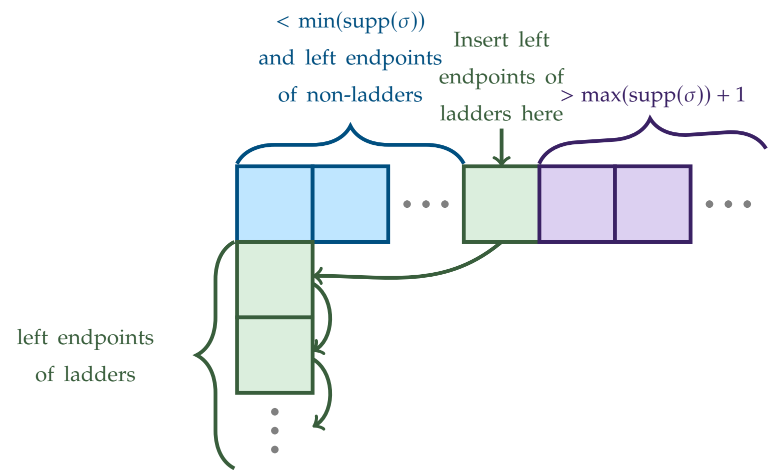


Figure 3.3 Visualization of how to read insertion tableaux directly from the diagram for a word with decreasing runs.

Proof. Call our permutation σ and let w be a decreasing nested word for σ with run decomposition $r_1 \cdot r_2 \cdots r_k$. We prove (1) directly and prove (2) using (1).

(1) Let $a = \min(r_1)$ and $b = \max(r_1)$. After filling in the letters $\{1, \ldots, a-1\}$, we will have partial RSK tableaux

$$P = \begin{array}{|c|c|c|}\hline 1 & \cdots & a-1 \\ \hline \end{array} \qquad Q = \begin{array}{|c|c|c|}\hline 1 & \cdots & a-1 \\ \hline \end{array}.$$

Next, we will fill in all of letters in positions $a, a+1, \ldots, b$. Since a is the right endpoint of the leftmost run and does not appear in any other runs, $\sigma(a) = b$. Thus the new partial tableaux are

$$P = \begin{array}{|c|c|c|c|}\hline 1 & \cdots & a-1 & b \\ \hline \end{array} \qquad Q = \begin{array}{|c|c|c|c|}\hline 1 & \cdots & a-1 & a \\ \hline \end{array}.$$

From this point forward, there are two possible states that the partial RSK tableaux can be in. Either we are in State A and all of the letters in the first row of P following $a-1$ are right endpoints of non-ladders, or we are in State B and all but the last letter are right endpoints of non-ladders and the last letter is the right endpoint of a ladder. Additionally, in either state, each of the remaining rows of P consists of exactly one element which is the left endpoint of a ladder and each of the remaining rows of Q consists of one element which is the right endpoint of a ladder plus 1. Finally, the most recent letter inserted will always be the last letter of the first row of P.

Notice that we begin in State B with the tableaux above. The condition on the rows following the first row are vacuously satisfied, since there is currently only one row. Now assuming we are in one of these states, we wish to show that as we continue inserting we will remain in either State A or State B.

Suppose we are in State A. If the next letter is the left endpoint of a non-ladder, then by Lemma 3.6 this element will be inserted at the end of the first row, and we will stay in State A. If the next letter is the left endpoint of a ladder, then by Lemma 3.7 it will be added to the end of the first row, putting us in State B. This move does not change the rows after the first row, and we inserted at the end of the first row.

Suppose that we are in State B. If the next letter is the left endpoint of a ladder, then by Lemma 3.6, this element will bump the last element into the next row, keeping us in State B if the conditions on the remaining rows are satisfied. If the next letter is the left endpoint of a non-ladder, then by Lemma 3.7 it will bump the last letter into the next row, putting us in State A if the conditions on the remaining rows are satisfied. Notice also that in both of these cases, the letter we inserted was inserted into the last spot in the first row.

By Lemma 3.6 the element in the second row will be bumped, if it exists, and it will in turn bump the element below it, etc. Thus P will retain its hook shape and the rows after the first will still consist of left endpoints of ladders, since the element we bumped was the left endpoint of a ladder.

Now observe that if a left endpoint of a ladder is inserted, it is immediately bumped by the next element. Since the shape of the P tableau had a single element added in a new row, we must add a new row to Q which will be filled with the right endpoint of the element we just added. Since the element we just added was directly after a ladder, this is the same as adding the right endpoint of the ladder plus 1 to Q.

Notice that inserting the left endpoint of a non-ladder always puts us in State A. Since $\max(\mathrm{supp}(\sigma)) + 1$ cannot be the minimum of any run, the last letter we insert cannot be the left endpoint of a ladder, so we will always end in State A.

After we insert the remaining letters which must be fixed by σ into the first row, we will have obtained the tableaux we were looking for.

(2) We now suppose σ has a nested reduced word with decreasing runs. Then σ^{-1} has a nested reduced word with increasing runs, since we can read the word backwards to get a reduced word for σ. Thus we can determine the P and Q tableaux for σ^{-1} from the first part of the claim. By Theorem 2.2, the P and Q tableaux for σ are the Q and P tableaux for σ^{-1}, respectively. Since the diagram for a nested reduced word for σ^{-1} can simply be obtained by flipping the diagram for the nested reduced word for σ over a vertical axis, the left and right endpoints are swapped in the statement and the proof is complete.

□

Example 3.4. Consider panel (b) of Figure 3.2. The yellow, blue, and purple paths are all ladders. The left endpoints of these lines are $6, 5, 3$ and the right endpoints are $4, 2, 1$. Thus the RSK tableaux for this permutation are

$$P = \begin{array}{|c|c|c|}\hline 1 & 2 & 4 \\ \hline 3 \\ \cline{1-1} 5 \\ \cline{1-1} 6 \\ \cline{1-1}\end{array} \qquad Q = \begin{array}{|c|c|c|}\hline 1 & 4 & 6 \\ \hline 2 \\ \cline{1-1} 3 \\ \cline{1-1} 5 \\ \cline{1-1}\end{array}.$$

Now consider panel (d) of Figure 3.2. The red, green, and blue lines are all chutes. Their left endpoints are $1, 3, 4$ and their right endpoints are $3, 4, 7$. Thus the RSK tableaux for the permutation are

$$P = \begin{array}{|c|c|c|c|}\hline 1 & 3 & 6 & 7 \\ \hline 2 \\ \cline{1-1} 4 \\ \cline{1-1} 5 \\ \cline{1-1}\end{array} \qquad Q = \begin{array}{|c|c|c|c|}\hline 1 & 2 & 5 & 6 \\ \hline 3 \\ \cline{1-1} 4 \\ \cline{1-1} 7 \\ \cline{1-1}\end{array}.$$

While it does seem that we need to compute the permutation to find the paths which traverse entire runs, we can actually be more efficient with a change of perspective. Since exactly one path will traverse the entirety of each run, we can use the run to predict the path and its endpoints. The entries we get from the minimum values of the runs, right in the decreasing case and left in the increasing case, are fairly straightforward. For each run, simply take its minimum value and add 1. In the decreasing case, this will determine the Q tableau. In the increasing case, this will determine the P tableau. To find the entries in the other tableaux, we need to subtract the number of runs the ladder intersects after climbing a run or the number of runs the chute intersects before sliding down a run.

Proposition 3.9. *Let σ be a permutation with a nested reduced word with run decomposition $r_1 \cdot r_2 \cdots r_k$. If all of the r_i are decreasing, then the RSK tableaux of σ have hook shapes and when the first rows are removed they are*

$$\tilde{P} = \begin{array}{|c|}\hline \max(r_k) \\ -(k-1) \\ \hline \vdots \\ \hline \max(r_2) \\ -1 \\ \hline \max(r_1) \\ \hline \end{array} \qquad \tilde{Q} = \begin{array}{|c|}\hline \min(r_1) \\ +1 \\ \hline \min(r_2) \\ +1 \\ \hline \vdots \\ \hline \min(r_k) \\ +1 \\ \hline \end{array}.$$

If all of the r_i are increasing, then the RSK tableaux of σ have hook shapes and when the first rows are removed they are

$$\tilde{P} = \begin{array}{|c|}\hline \min(r_k) \\ +1 \\ \hline \vdots \\ \hline \min(r_2) \\ +1 \\ \hline \min(r_1) \\ +1 \\ \hline \end{array} \qquad \tilde{Q} = \begin{array}{|c|}\hline \max(r_1) \\ -(k-1) \\ \hline \max(r_2) \\ -1 \\ \hline \vdots \\ \hline \max(r_k) \\ \hline \end{array}.$$

Proof. This follows directly from Lemma 3.8 and Lemma 3.5. □

This result allows us to make some progress towards our original goal: finding minimal run words.

Corollary 3.10. *If σ has a nested word, then that word is a minimal run word for σ.*

Proof. If the nested run word has k runs, then by Proposition 3.9 the first component of RSK(σ) is $n - k$. By Theorem 2.5, $\textsc{Run}(\sigma) = k$, so the word is a minimal run word. □

3.5 Future Work

I only started studying this in the spring, but it's been a lot of fun and I wish I had some more time with it. If I were to keep working on this project, here are some questions I would be thinking about.

Question 3.1. *Prove that Algorithm 3 is well-defined, or adjust it to make it well-defined.*

Algorithm 3 being well-defined opens the door to many other interesting questions.

Question 3.2. *Characterize all permutations σ with hook-shaped RSK tableaux. If the canonical word for a permutation is monotone but not nested, can we guarantee that* RSK(σ) *is not a hook? Can the canonical word for a permutation with hook-shaped* RSK *tableaux contain both increasing and decreasing runs?*

Question 3.3. *Prove that the canonical word of a permutation is a minimal run word for that permutation.*

The *super-Yamanouchi word* defined in by Assaf is simply a monotone decreasing word, but Assaf proved that every permutation has a unique super-Yamanouchi word; see Assaf (2019). From Proposition 3.4, we know that if the super-Yamanouchi word is nested, then the RSK tableaux of the corresponding permutation have hook shapes.

Question 3.4. *Under what conditions do our canonical words coincide with super-Yamanouchi words?*

Question 3.5. *Under what condition is the super-Yamanouchi word for a permutation a minimal run word?*

Question 3.6. *Can the RSK tableaux of permutations be written down directly from the super-Yamanouchi word, even when the runs are not nested?*

The following questions are a bit more difficult or open-ended.

Question 3.7. *Given some reduced word for σ is it possible to predict* RSK(σ)?

Question 3.8. *Given a permutation, find an algorithm for a minimal run word. Bonus points if you use the longest increasing subsequence.*

Question 3.9. *Develop a better understanding of diagrams for run words with both increasing and decreasing subsequences. For example, can paths be both chutes and ladders?*

More broadly, studying minimal run words and their frequencies could make for an interesting project. Relatedly, Run is a quirky statistic, and studying it would be worthwhile.

Question 3.10. *Give a combinatorial proof of Gunawan et al.'s result that the length of the longest increasing subsequence plus the number of runs in a minimal run word is always n; see Gunawan et al. (2022).*

Question 3.11. *If we randomly generate diagrams for reduced words, what kind of distribution do we get on $\mathfrak{S}_n$?*

This work was really motivated by finding interpretations of the longest increasing subsequence function for other representations of permutations.

Question 3.12. *What other representations have nice connections to long increasing subsequences?*

Chapter 4

A Basis for the k-Local Space

So far, we have defined the k-local space combinatorially and using representation theory. We have shown that many permutation statistics of interest are k-local and given a spanning set for the k-local space. This chapter is devoted to progress towards proving the following result.

Proposition 4.1. *$\mathcal{B}_k$ is a basis for L_k.*

Here, L_k is the k-local subalgebra of $\mathbb{C}\mathfrak{S}_n$ and $\mathcal{B}_k$ is the set of projections of permutations containing an increasing subsequence of length $n-k$ into L_k.

In this section, we show that Proposition 4.1 boils down to showing that $\rho^{(n-k,1^k)}$ or ρ^k evaluated at permutations with increasing subsequences gives a linearly independent set of matrices.

4.1 Utilizing Full Local Modules

To prove Proposition 4.1, it will be easier to study modules than to look at k-local functions themselves. The k-local functions on $\mathfrak{S}_n$ form a subalgebra of $\mathbb{C}\mathfrak{S}_n$ which we denote by L_k. To verify this, the only tricky part is showing closure under multiplication by elements of $\mathbb{C}\mathfrak{S}_n$. However, we see that

$$\sigma \cdot \mathbf{1}_{(I,J)} = \mathbf{1}_{(\sigma \cdot I, \sigma \cdot J)},$$

where $\sigma \cdot (\ell_1, \ell_2, \ldots, \ell_k) = (\sigma(\ell_1), \sigma(\ell_2), \ldots, \sigma(\ell_k))$, is still k-local and so the set of k-local functions is closed under multiplication by elements in $\mathbb{C}\mathfrak{S}_n$.

We use the following notation for the remainder of this section. Let N be a full k-local module and ρ be a representation for N. Linearly extend ρ to a

DFT

$$D : \mathbb{C}\mathfrak{S}_n \to \mathrm{End}_{\mathbb{C}}(N) \qquad \text{where} \qquad \sum_{\sigma \in \mathfrak{S}_n} c_\sigma \sigma \mapsto \sum_{\sigma \in \mathfrak{S}_n} c_\sigma \rho(\sigma).$$

D is a $\mathbb{C}$-algebra homomorphism, but not an isomorphism. By Theorem 2.7, we can define an analogous map which is restricted to L_k

$$D_k : L_k \to \mathrm{im}(D) \qquad \text{by} \qquad \sum_{\sigma \in \mathfrak{S}_n} c_\sigma \sigma \mapsto \sum_{\sigma \in \mathfrak{S}_n} c_\sigma \rho(\sigma).$$

Our goal in this section is to show that D_k a bijection. If this is the case, then $\mathrm{im}(D)$ will be isomorphic to L_k.

We first prove this useful lemma.

Lemma 4.2 (Hamaker and Rhoades (2022))**.** *Define*

$$\Psi_k : L_k \to \bigoplus_{\substack{\lambda \vdash n \\ \lambda_1 \geq n-k}} M^{f^\lambda \times f^\lambda} \qquad \textit{by} \qquad \sum_{\sigma \in \mathfrak{S}_n} c_\sigma \sigma \mapsto \bigoplus_{\substack{\lambda \vdash n \\ \lambda_1 \geq n-k}} \sum_{\sigma \in \mathfrak{S}_n} c_\sigma \rho_\lambda(\sigma)$$

for some irreducible representations ρ_λ. Then Ψ_k is an isomorphism.

Proof. By Wedderburn's decomposition theorem, there exists an isomorphism

$$\Psi : \mathbb{C}\mathfrak{S}_n \to \bigoplus_{\lambda \vdash n} M^{f^\lambda \times f^\lambda} \quad \text{defined by} \quad \sum_{\sigma \in \mathfrak{S}_n} c_\sigma \sigma \mapsto \bigoplus_{\lambda \vdash n} \sum_{\sigma \in \mathfrak{S}_n} c_\sigma \rho_\lambda(\sigma)$$

where the ρ_λ are irreducible representations such that they match the ρ_λ in the statement of the result when $\lambda_1 \geq n-k$.

Ψ_k is injective. Suppose $f, g \in L_k$ have $\Psi_k(f) = \Psi_k(g)$. Then $\Psi(f) = \Psi(g)$, because the matrices indexed by λ with $\lambda_1 \geq n-k$ are identical since $\Psi_k(f) = \Psi_k(g)$ and the matrices indexed by λ with $\lambda_1 < n-k$ are zero for both f and g by Theorem 2.7. Because Ψ is injective, $f = g$ and so Ψ_k is injective as well. Ψ_k is also surjective. Given some

$$\tilde{m} \in \bigoplus_{\substack{\lambda \vdash n \\ \lambda_1 \geq n-k}} M^{f^\lambda \times f^\lambda},$$

we can turn it into $m \in \bigoplus_{\lambda \vdash n} M^{f^\lambda \times f^\lambda}$ by appending zero matrices for all for all of the partitions λ with $\lambda_1 < n-k$. Since Ψ is an isomorphism,

$\Psi^{-1}(m) \in \mathbb{C}\mathfrak{S}_n$, and by Theorem 2.7, $\Psi^{-1}(m)) \in L_k$. Because the matrices indexed by λ with $\lambda_1 \geq n-k$ in m are precisely the matrices in $\tilde{m}$,

$$\Psi_k(\Psi^{-1}(m)) = \tilde{m},$$

so Ψ_k is surjective.

The fact that Ψ_k is a homomorphism follows directly from the fact that the ρ_λ are representations. □

We can now prove that $L_k \cong \mathrm{im}(D)$.

Lemma 4.3. *D_k is an isomorphism.*

Proof. Since N is a full k-local module, there exist nonzero coefficients c_λ such that

$$N \cong \bigoplus_{\substack{\lambda \vdash n \\ \lambda_1 \geq n-k}} c_\lambda S^\lambda.$$

There exists a change of basis T so that $\tilde{\rho} = T^{-1}\rho T$ is an equivalent representation written with respect to a symmetry adapted basis for N. Then $\tilde{\rho}(\sigma)$ is a block-diagonal matrix such that each block is an irreducible representation ρ_λ for σ and $\rho_\lambda(\sigma)$ appears as a block c_λ times for all λ with $\lambda_1 \geq n-k$. Thus there exists an isomorphism

$$\Theta : \mathrm{im}(D) \to \bigoplus_{\substack{\lambda \vdash n \\ \lambda_1 \geq n-k}} M^{f^\lambda \times f^\lambda} \quad \text{defined by} \quad \sum_{\sigma \in \mathfrak{S}_n} c_\sigma \rho(\sigma) \mapsto \bigoplus_{\substack{\lambda \vdash n \\ \lambda_1 \geq n-k}} \sum_{\sigma \in \mathfrak{S}_n} c_\sigma \rho_\lambda(\sigma).$$

where f^λ is the dimension of the irreducible module associated with λ and ρ_λ is the irreducible representation associated with λ which appears as a block in $\tilde{\rho}$.

We can write

$$D_k = \Theta^{-1} \circ \Psi_k$$

where Ψ_k is the isomorphism defined in Lemma 4.2 such that the irreducible representations are those that appear in the images of Θ. Since Θ^{-1} and Ψ_k are isomorphisms, D_k is an isomorphism. □

We can now define what we mean by the projection of σ into the k-local subalgebra.

Definition 4.1. For $f \in \mathbb{C}\mathfrak{S}_n$, define the projection of f into L_k by

$$\tilde{f} = D_k^{-1} \circ D(f).$$

This is a projection since D restricted to the k-local subalgebra is simply D_k. This also allows us to more precisely define $\mathcal{B}_k$.

Definition 4.2. Let

$$\mathbf{1}_\sigma : \mathfrak{S}_n \to \mathbb{C} \qquad\qquad \mathbf{1}_\sigma(\tau) = \begin{cases} 1 & \tau = \sigma \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$\mathcal{B}_k = \{\tilde{\mathbf{1}}_\sigma \mid \sigma \in \mathfrak{S}_n \text{ has an increasing subsequence of length } n-k\}.$$

The following lemma is a direct consequence of how we defined our projection, and will allow us to prove that $\mathcal{B}_k$ is a basis for L_k by showing that the set of $\rho(\sigma)$ where σ has an increasing subsequence of length $n-k$ forms a basis for $\operatorname{im}(D) \cong L_k$.

Lemma 4.4. *For any $\sigma \in \mathfrak{S}_n$, $D_k(\tilde{\mathbf{1}}_\sigma) = \rho(\sigma)$.*

Proof. By Definition 4.1

$$D_k(\tilde{\mathbf{1}}_\sigma) = D_k \circ D_k^{-1} \circ D(\mathbf{1}_\sigma) = D(\mathbf{1}_\sigma) = \rho(\sigma).$$

□

With this machinery, proving our conjecture now simply boils down to showing that

$$D(\mathcal{B}_k) = \{\rho(\sigma) \mid \tilde{\mathbf{1}}_\sigma \in \mathcal{B}_k\}$$

is a $\mathbb{C}$-basis for $\operatorname{im}(D)$.

Using the full k-local module M^λ, I proved the result for $k = 1, 2$. Using the full k-local module M^k, Doty proved the general case. Recall that D^k is the linear extension of ρ^k to a DFT, where ρ^k is the representation which arises from the action of $\mathfrak{S}_n$ on k-tuples with repetition.

Theorem 4.5 (Doty (2021))**.** *$D(\mathcal{B}_k)$ is a $\mathbb{C}$-basis for $\operatorname{im}(D^k)$.*

This result allows us to easily prove the desired result.

Proposition 4.1. *$\mathcal{B}_k$ is a basis for L_k.*

Proof. By Proposition 2.10, M^k is a full k-local module. By Lemma 4.3, $\mathrm{im}(D^k) \cong L_k$. By Theorem 4.5, $D^k(\mathcal{B}_k)$ is a basis for $\mathrm{im}(D^k)$. Thus, $\mathcal{B}_k$ is a basis for L_k. □

The next section gives an independent proof of the dimension of L_k. The following two sections contain my work on this problem for the $k = 1, 2$ cases. While my methods are difficult to generalize to larger k, I think the proof technique yields some insight into how we can use the structure of permutation representations of $\mathfrak{S}_n$.

4.2 Dimension

Showing that $\mathcal{B}_k$ has the correct number of vectors is straightforward using the Robinson-Schensted correspondence:

Lemma 4.6 (Hamaker and Rhoades (2022))**.** *The dimension of L_k as a vector space is $|\mathcal{B}_k|$.*

Proof. By Lemma 4.2,

$$L_k \cong \bigoplus_{\substack{\lambda \vdash n \\ \lambda_1 \geq n-k}} M^{f^\lambda \times f^\lambda}.$$

Thus

$$\dim(L_k) = \sum_{\substack{\lambda \vdash n \\ \lambda_1 \geq n-k}} (f^\lambda)^2.$$

By Theorem 2.6, f^λ is the number of standard Young tableaux of shape λ. Thus $(f^\lambda)^2$ is the number of ordered pairs of standard Young tableaux of shape λ. Thus

$$\dim(L_k) = \sum_{\substack{\lambda \vdash n \\ \lambda_1 \geq n-k}} (f^\lambda)^2$$

is the number of pairs of standard Young tableaux of the same shape such that the first component of their shapes is at least $n - k$. But by Theorem 2.3, this is the number of permutations with a longest increasing subsequence of length at least $n - k$, which is $|\mathcal{B}_k|$. Thus $\dim(L_k) = |\mathcal{B}_k|$. □

4.3 1-Local Case

Since we know that $\mathcal{B}_1$ has the correct size, our goal is to show that the matrices in $D^{(n-1,1)}(\mathcal{B}_1)$ are linearly independent, where $D^{(n-1,1)}$ is the $\mathbb{C}$-linear extension of the representation $\rho^{(n-1,1)}$, which is the representation arising from the action of $\mathfrak{S}_n$ on tabloids of shape $(n-1, 1)$. Our technique is to begin with a linear combination of the matrices in $D^{(n-1,1)}(\mathcal{B}_1)$ which is equal to zero, and show that all coefficients must be zero. We can do so by showing that each matrix has a 1 in a unique spot, so that its coefficient must be zero. However, this method will not quite work as is, so we must apply it in stages. First, remove the matrices that have ones where no other matrices do. This will not cover all the matrices, but once we have removed those we can remove another batch.

Before we generalize, let's take a look at this process for $n = 3$. Our goal is to show that $\{\rho^{(2,1)}(\sigma) \mid \sigma \in \{123, 132, 213, 231, 312\}\}$ are linearly independent. We begin with the $\mathbb{C}$-linear combination

$$c_{123}\overset{123}{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}} + c_{132}\overset{132}{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}} + c_{213}\overset{213}{\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}} + c_{231}\overset{231}{\begin{pmatrix} 0 & 0 & \boxed{1} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}} + c_{312}\overset{312}{\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \boxed{1} & 0 & 0 \end{pmatrix}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Immediately, we know that $c_{231} = c_{312} = 0$, because the 231 matrix is the only matrix with a nonzero entry in the first row and third column and the 312 matrix is the only matrix with a 1 in the third row and first column. Both of these entries are boxed in the above sum. This leaves us with the following linear combination.

$$c_{123}\overset{123}{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}} + c_{132}\overset{132}{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & \boxed{1} \\ 0 & \boxed{1} & 0 \end{pmatrix}} + c_{213}\overset{213}{\begin{pmatrix} 0 & \boxed{1} & 0 \\ \boxed{1} & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

From here, we can see that $c_{132} = c_{213} = 0$, because the boxed entries are the only nonzero entries in those positions in the sum. In this situation, the choice might seem arbitrary, but as we move into larger n, it will seem more natural. Finally, we are left with

$$c_{123} \overset{123}{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

which implies that $c_{123} = 0$. Since all the c_σ were zero, we can conclude that $D^{(2,1)}(\mathcal{B}_1)$ is a linearly independent set.

The important thing to note here is that as we were choosing which entries to focus on to remove matrices, we worked our way from the corners of the matrices in towards the diagonal. We will first remove representations of permutations containing an entry $i \mapsto j$ where there is a large difference between i and j. This will continue into the 2-local case as well and might be a useful technique for larger k. Our goal is to show that if a permutation has an increasing subsequence of sufficient length and $i \mapsto j$ with $|i-j|$ being sufficiently large, then it is the only permutation with an increasing subsequence of that length which maps i to j. The intuition behind why this is true is that if j is so far from where it is supposed to be, then it cannot possibly be in an increasing subsequence.

Lemma 4.7. *Suppose $\sigma \in \mathfrak{S}_n$ and $i, j \in n$ with $|i-j| \geq a$. Suppose also that $\sigma(i) = j$. Then the longest increasing subsequence of σ containing j has length at most $n-a$.*

Proof. We will construct the sequence in three parts: the part before j, j, and the part after j. We wish to bound the length of the longest possible increasing subsequence before j such that the largest value is less than j. Similarly, we want the longest increasing subsequence after j such that all the values are greater than j. The maximal increasing subsequence occurring before j is bounded above by $i-1$, the number of available spots, and $j-1$, the number of values less than j. Similarly, the length of the maximal increasing subsequence occurring after j is bounded above by $n-i$, the number of available spots, and $n-j$, the number of values greater than j. Thus the length of the longest increasing subsequence including j is

$$m \leq \min(i-1, j-1) + 1 + \min(n-i, n-j)$$

where 1 accounts for j itself. If $i < j$, then this becomes

$$m \leq i - 1 + 1 + n - j = n - (j-i) = n - |j-i| \leq n - a.$$

Similarly, if $j \leq i$, then

$$m \leq j - 1 + 1 + n - i = n - (i - j) = n - |i - j| \leq n - a.$$

□

The following statistic will be useful when stating conditions in many of the results that follow.

Definition 4.3. Define the *maximum excedance* of $\sigma \in \mathfrak{S}_n$ as the largest difference between a number and its image under σ. Define the function $\textsc{Mexc} : \mathfrak{S}_n \to \mathbb{C}$ by

$$\textsc{Mexc}(\sigma) = \max\{|\sigma(i) - i| \mid i \in [n]\}.$$

We can now show that if σ has a long increasing subsequence which contains a mapping of maximum excedance, then many of the numbers in $[n]$ are forced to be fixed points.

Lemma 4.8. *Suppose $\sigma \in \mathfrak{S}_n$ satisfies*

(1) $\textsc{Mexc}(\sigma) = k$

(2) *either*

  (*a*) $\sigma(i - k) = i$ *for some* $i \in \{k + 1, \dots n\}$, *or*
  (*b*) $\sigma(i + k) = i$ *for some* $i \in \{1, 2, \dots, n - k\}$, *and*

(3) σ *contains an increasing subsequence of length* $n - k$ *which includes* i.

Then σ satisfies $\sigma(\ell) = \ell$

(*a*) *for all* $\ell \in \{1, \dots, i - 2k\} \cup \{i + k, \dots, n\}$ *or*

(*b*) *for all* $\ell \in \{1, \dots, i - k\} \cup \{i + 2k, \dots, n\}$.

Proof. We prove (a) and omit the proof of (b), since it is essentially the same. Suppose $\sigma(i - k) = i$. There are $i - 1$ numbers less than i, but only $i - k - 1$ spots before i. Thus there must be k numbers less than i following i. These cannot be in the increasing subsequence of length $n - k$, since the sequence must include k. Thus all the numbers larger than i must be in increasing order after i and all the numbers with position less than $i - k$ must be in increasing order to achieve an increasing subsequence of length $n - k$ containing i.

Suppose $\ell \in \{1, \dots, i-2k\}$ and $\sigma(\ell) > \ell$. Let $j \in \{1, \dots, \ell\}$. Then j must appear before i because $j \leq \ell \leq i-2k$ and the positions after i are at least $i-k+1$, so the difference between j and the positions after i is at least $i-k+1-(i-2k) = k+1$. Since $\ell < \sigma(\ell)$, all of the numbers in $[\ell]$ must appear before $\sigma(\ell)$ in σ, because the portion of σ before i has to be increasing. But this is impossible because there are only $\ell - 1$ positions before ℓ and at least ℓ numbers that must fit there. If $\sigma(\ell) < \ell$, you can make a similar argument by showing that everything in $\{\ell, \ell+1, \dots, i-2k\}$ has to appear between positions $\ell+1$ and $i-2k$ (inclusive) and there are only $i-2k-\ell-1$ spots for $i-2k-\ell$ numbers. Thus $\sigma(\ell) = \ell$.

Now suppose that $j < i$ appears after i. Because $j \leq i-1$ the biggest position j can have is $i-1+k$. This means that all of the elements in positions $i+k, \dots, n$ must be greater than i. Then they must be in increasing order. Now suppose there is some $\ell \in \{i+k, \dots, n\}$ with $\sigma(\ell) \neq \ell$. If $\sigma(\ell) > \ell$, then there are more positions after ℓ than there are numbers greater than ℓ, so the subsequence cannot be increasing. If $\sigma(\ell) < \ell$, we also have a contradiction because $\sigma(\ell), \ell > i$, so everything in $\{\ell, \dots, n\}$ must appear after $\sigma(\ell)$, but the only available positions are $\{\ell+1, \dots, n\}$. Since there are fewer positions than values, this is impossible. Thus $\sigma(\ell) = \ell$. □

We can now use this lemma to show that if $i \mapsto j$ and $|i-j| \geq 2$, then there is exactly one permutation mapping i to j which has an increasing subsequence of length $n-1$. This will allow us to conclude that all the coefficients of permutations containing mappings $i \mapsto j$ with $|i-j| \geq 2$ must be zero.

Lemma 4.9. *Suppose $i, j \in [n]$ and that $|i-j| \geq 2$. There is exactly one permutation σ containing an increasing subsequence of length $n-1$ with $\sigma(i) = j$.*

Proof. We give a construction. Fix $\sigma(i) = j$ and arrange all the other numbers in $[n]$ in increasing order around j. Since they are in increasing order and there are $n-1$ of them, they form an increasing subsequence of length $n-1$.

Now suppose some other permutation τ satisfies $\tau(i) = j$. First, τ does not have an increasing subsequence of length $n-1$ which excludes j, because otherwise it would be equal to σ. Thus if τ had an increasing subsequence of length $n-1$, it would have to include j. But by Lemma 4.7, an increasing subsequence containing j has cannot be longer than $n-2$.

Thus σ is the only permutation with $\sigma(i) = j$ containing an increasing subsequence of length $n-1$. □

Now that all permutations with $\textsc{Mexc} \geq 2$ are eliminated, the remaining permutations contain only mappings $i \mapsto j$ with $|i-j| < 2$. This next lemma allows us to zero out the coefficients of the remaining non-identity permutations: the adjacent transpositions. The second condition is the reason that we must zero out all of the other coefficients first.

Lemma 4.10. *Suppose $i, j \in [n]$ and that $|i-j| = 1$. There is exactly one permutation σ satisfying*

(1) $\sigma(i) = j$

(2) $\textsc{Mexc}(\sigma) = 1$

(3) *σ contains an increasing subsequence of length $n-1$.*

Proof. We prove the statement when $j = i+1$. The $j = i-1$ case is essentially the same. The goal is to prove that there is only permutation σ satisfying $\sigma(i) = i+1$, $|\ell - \sigma(\ell)| \leq 1$ for all $\ell \in [n]$, and σ contains an increasing subsequence of length $n-1$. Suppose σ satisfies all of these criteria. There are $i-1$ positions before i, but i numbers which are less than $i+1$. This means that some number in $[i]$ must occur after $i+1$, forming a decreasing subsequence of length 2. Further, the only element in $[i]$ that can occur after $i+1$ is i, because otherwise we would violate condition (2). This also means that $\sigma(i+1) = i$. Since $i+1$ appears before i, and σ contains an increasing subsequence of length $n-1$, all the other elements must form an increasing subsequence of length $n-2$ around $i, i+1$. This fixes $\sigma(\ell) = \ell$ for all $\ell \neq i, i+1$. Thus σ is the adjacent transposition $(i\ i+1)$. □

After removing the adjacent transpositions, we are left with the identity and so we are finished. We can now put everything together to prove that $\mathcal{B}_1$ is a basis for L_1.

Proposition 4.11. *The set $\mathcal{B}_1$ is a basis for L_1.*

Proof. By Lemma 4.6, $\mathcal{B}_1$ has the correct size, so it is sufficient to show that it is a linearly independent set. By Proposition 2.9, $M^{(n-1,1)}$ is a full 1-local module. By Lemma 4.3, showing $\mathcal{B}_1$ is linearly independent is equivalent to showing that $D^{(n-1,1)}(\mathcal{B}_k)$ is linearly independent.

Thus it is sufficient to show that if

$$\sum_{\tilde{\mathbf{1}}_\sigma \in \mathcal{B}_k} c_\sigma \rho^{(n-1,1)}(\sigma) = 0,$$

then all the $c_\sigma = 0$. Recall that the $\rho^{(n-1,1)}(\sigma)$ are $n \times n$ matrices with a 1 in the ith column and jth row if and only if $\sigma(i) = j$. If σ is the only permutation such that $\rho^{(n-1,1)}(\sigma)$ has a nonzero value in a particular entry, then the coefficient of $\rho^{(n-1,1)}(\sigma)$ must be zero for the right-hand side to remain zero.

Suppose $|i - j| \geq 2$ for some $i, j \in [n]$. By Lemma 4.9, there is exactly one permutation σ containing an increasing subsequence of length $n - 1$ such that $\sigma(i) = j$. This means that $\rho^{(n-1,1^k)}(\sigma)$ is the only matrix in our sum with a nonzero value in the ji-entry. Thus c_σ must be zero.

We have now given zero coefficients to all permutations which contain a mapping where the index and image differ by at least 2. In other words, all of the remaining matrices have all of their nonzero entries either on the diagonal or directly adjacent to the diagonal. Now our sum is

$$\sum_{\substack{\tilde{\mathbf{1}}_\sigma \in \mathcal{B}_k \\ \textsc{Mexc}(\sigma) \leq 1}} c_\sigma \rho^{(n-1,1)}(\sigma) = 0.$$

Consider now some $i, j \in [n]$ with $|i - j| = 1$. By Lemma 4.10, there is exactly one permutation σ in the sum with $\sigma(i) = j$. Thus $\rho^{(n-1,1)}(\sigma)$ is the only permutation with a 1 in the ji entry, so c_σ must be zero.

We have now eliminated every permutation with $\textsc{Mexc}(\sigma) > 0$. The only permutation with $\textsc{Mexc}(\sigma) = 0$ is the identity. Since this is now the only permutation left in our sum, its coefficient must be zero as well. Thus the vectors in $\mathcal{B}_1$ are linearly independent and $\mathcal{B}_1$ forms a basis for L_1. □

4.4 2-Local Case

The 2-local case is a bit more complex than the 1-local case and relies heavily on the 1-local result. In the 1-local case, we only had to consider mappings with differences 0, 1, or 2+. In other words, we eliminated matrices with nonzero entries far from the diagonal, then the ones with nonzero entries adjacent to the diagonal, and finally the identity. In the 2-local case, we need to consider mappings with differences 0, 1, 2, 3+. This means that instead of eliminating all of the matrices with nonzero entries far from the diagonal at once, we do them in two chunks; see Figure 4.1. As in the 1-local case, the 0, 1, and 3+ cases are fairly straightforward, but the 2 case is more tricky.

At first glance, Figure 4.1 seems a bit misleading, because the structure of the 1 and 2-local representations are different. However, our proof technique

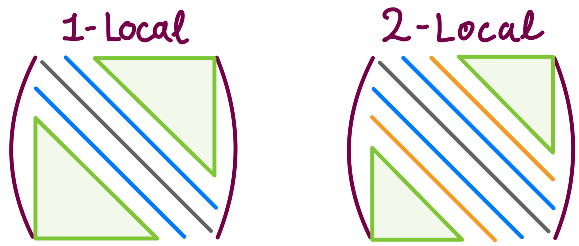


Figure 4.1 Visualization of the proof techniques for showing $\mathcal{B}_1, \mathcal{B}_2$ are linearly independent. In the 1-local case, we eliminate matrices with nonzero entries ij with $|i-j| \geq 2$, then those with $|i-j| = 1$, and finally the identity. The 2-local case is almost the same, but the $|i-j| \geq 2$ is split into sub-cases: $|i-j| \geq 3$, which is similar to the $|i-j| \geq 2$ case in the 1-local proof, and $|i-j| = 2$, which is more difficult. This technique is difficult to generalize to larger k because the number of cases, represented by lines, increases with k.

for the 2-local case has a different focus. Where in the 1-local case, we focused on zeroing out particular entries, in the 2-local case, our aim will instead be to zero out entire blocks using the 1-local result. To make sense of this, we need to understand what we mean by a block.

Definition 4.4. Define the *ji-block* of $\rho^{(n-2,1,1)}$ as the submatrix formed by taking all of the rows indexed by tabloids with j in the second row and all of the columns indexed by tabloids with i in the second row.

Remark. While we defined $\rho^{(n-2,1,1)}$ as the representation arising from the action of $\mathfrak{S}_n$ on tabloids of shape $(n-2,1,1)$, we can think of $\mathfrak{S}_n$ as acting on ordered pairs without repetition, where the ordered pair comes from the second and third rows of the tabloid.

The next lemma will help us use the 1-local result.

Lemma 4.12. *If $\sigma(i) = j$, then ji-block of the 2-local representation of σ is the 1-local representation of σ with j removed and then reduced to a permutation in $\mathfrak{S}_{n-1}$.*

Proof. Consider the ji-block of the 2-local representation of σ. Call this matrix A. The rows are indexed by the ordered pairs $(j,1),\ldots,(j,j-1),(j,j+1),\ldots,(j,n)$ and the columns are indexed by the ordered pairs $(i,1)\ldots,(i,i-1),(i,i+1),\ldots,(i,n)$. Relabel the rows and columns so that they are both indexed by $1, 2, \ldots, n-1$. Then the matrix A has a 1 in the ℓm entry if and only if $\sigma(\overline{m}) = \overline{\ell}$ where

$$\overline{\ell} = \begin{cases} \ell & \text{if } \ell < j \\ \ell + 1 & \text{else.} \end{cases} \qquad \overline{m} = \begin{cases} m & \text{if } m < i \\ m + 1 & \text{else.} \end{cases}$$

Now suppose we remove j from σ and reduce, i.e., relabel all of the letters so that the result $\tilde{\sigma}$ is a permutation in $\mathfrak{S}_{n-1}$ which has the same relative order as σ when j is removed. Then $\tilde{\sigma}(m) = \ell$ if and only if $\sigma(\overline{m}) = \overline{\ell}$. Thus A is the 1-local representation of $\tilde{\sigma}$. □

Example 4.1. Here is the 2-local representation of 4132 with zeros omitted for clarity.

$$\begin{array}{c} \\ 12 \\ 13 \\ 14 \\ 21 \\ 23 \\ 24 \\ 31 \\ 32 \\ 34 \\ 41 \\ 42 \\ 43 \end{array}
\begin{array}{c}
\begin{array}{cccccccccccc} 12 & 13 & 14 & 21 & 23 & 24 & 31 & 32 & 34 & 41 & 42 & 43 \end{array} \\
\left(\begin{array}{ccc|ccc|ccc|ccc}
 & & & & & 1 & & & & & & \\
 & & & & 1 & & & & & & & \\
 & & & 1 & & & & & & & & \\
\hline
 & & & & & & & & & & 1 & \\
 & & & & & & & & & & & 1 \\
 & & & & & & & & & 1 & & \\
\hline
 & & & & & & & 1 & & & & \\
 & & & & & & & & 1 & & & \\
 & & & & & & 1 & & & & & \\
\hline
1 & & & & & & & & & & & \\
 & & 1 & & & & & & & & & \\
 & 1 & & & & & & & & & &
\end{array}\right)
\end{array}$$

Notice that if we remove 4 from 4132, then we get the permutation 132, which has 1-local representation

$$\begin{pmatrix} 1 & & \\ & & 1 \\ & 1 & \end{pmatrix}$$

which is submatrix the in the lower left corner of the matrix, in the $1 \mapsto 4$ block. If we remove 1, instead, from 4132, then we are left with 432. After we reduce, we get 321, which has 1-local representation

$$\begin{pmatrix} & & 1 \\ & 1 & \\ 1 & & \end{pmatrix}$$

which can be found in the $2 \mapsto 1$ block of the 2-local representation of 4132. The same is true with the remaining two blocks when 2 and 3 are removed.

The idea behind the proof for the two-local case is to remove blocks by showing that if we remove the mapping associated with the block from all permutations containing that mapping and reduce, then they will still have increasing subsequences of the same length, $n-2$, but the reduced permutations will now be permutations in $\mathfrak{S}_{n-1}$. This means that they are linearly independent by Proposition 4.11, which forces the coefficient for the entire matrix to be zero. Thus the problem becomes this: for a given mapping and permutation, can we construct an increasing subsequence of length $n-2$ which does not contain the mapping?

This is straightforward in the situation where $|i-j| \geq 3$.

Lemma 4.13. *Suppose $\sigma \in \mathfrak{S}_n$, $i, j \in [n]$, and $|i-j| \geq 3$. If σ contains an increasing subsequence of length $n-2$ and $\sigma(i) = j$, then σ contains an increasing subsequence of length $n-2$ which does not contain j.*

Proof. By Lemma 4.7, the longest increasing subsequence containing j has length at most $n-3$, so if σ contains an increasing subsequence of length $n-2$, then the subsequence must not contain j. □

Unfortunately, this does not always hold when $|i-j| = 2$, even if the maximum excedance in the permutation is 2. For example, $241567 \in \mathfrak{S}_7$ does not contain an increasing subsequence of length 5 which excludes 4. However, it does contain one which excludes 1, so if we can remove all permutations with that mapping first, we won't need to worry about this one when we get to the $2 \mapsto 4$ mapping. This means we cannot eliminate all of the mappings $i \mapsto j$ with difference 2 in one fell swoop: we must do it sequentially. It turns out the best strategy is to start from the ends.

Lemma 4.14. *Suppose $\sigma \in \mathfrak{S}_n$, $i, j \in [n]$, $|i-j| = 2$, and i or j is equal to 1 or n. If $\sigma(i) = j$, $\textsc{Mexc}(\sigma) \leq 2$, and σ contains an increasing subsequence of length $n-2$, then σ contains an increasing subsequence of length $n-2$ which does not contain j.*

Proof. We prove the statement when $i = 1$ and $j = 3$, but the proof can easily be adapted to the other cases. It is sufficient to show that if 3 is in the increasing subsequence, then there is another increasing subsequence which does not contain 3. By Lemma 4.8, if $\ell \geq 5$, when $\sigma(\ell) = \ell$. This means that σ has an increasing subsequence of length $n - 4$ consisting of $5, 6, \ldots, n$. The only numbers we still need to place are $1, 2$, and 4, since 3 and everything greater than 4 is fixed. The only permutation of three numbers which does not have an increasing subsequence of length 2 is 421, but this places 1 in position 4, which contradicts the condition that $|\sigma(\ell) - \ell| \leq 2$. Thus there must be some increasing subsequence formed by the numbers $1, 2, 4$, which we can combine with the subsequence $5, 6, \ldots, n$ to construct an increasing subsequence of length $n - 2$ that does not contain 3. □

To get this result for general mappings $i \mapsto j$ with $|i - j| = 2$, we need to impose an additional condition.

Lemma 4.15. *Suppose $\sigma \in \mathfrak{S}_n$, $i, j \in [n]$, and $|i - j| = 2$. Suppose also that $\sigma(i) = j$,* Mexc$(\sigma) = \ell$*, and σ contains an increasing subsequence of length $n - 2$. If*

(*a*) *$i < j$ and $\sigma(i + 1) \neq j - 3$*

(*b*) *$i > j$ and $\sigma(i - 1) \neq j + 3$*

then σ contains an increasing subsequence of length $n - 2$ which does not contain j.

Proof. We prove (a). Again, it is sufficient to show that if σ contains an increasing subsequence of length $n - 2$ that includes j, then it contains one that does not include j. By Lemma 4.8, $1, 2, \ldots, j - 4, j + 2, \ldots n$ forms an increasing subsequence of length $n - 5$. Thus we just need to show that the values $j - 3, j - 2, j - 1, j + 1$ are arranged such that they contain an increasing subsequence of length 3. The available positions are $i - 1, i + 1, i + 2, i + 3$, which is $j - 3, j - 1, j, j + 1$ in terms of j. Accounting for the fact that Mexc$(\sigma) = 2$, this is equivalent to finding all permutations in $\mathfrak{S}_4$ which avoid the mappings $1 \mapsto 4$, $4 \mapsto 1$, $3 \mapsto 1$, $4 \mapsto 2$ and $2 \mapsto 1$. These permutations are 1234, 1243, 1324, and 1423. All of these permutations contain an increasing subsequence of length 3. Thus we can merge this increasing subsequence with $1, 2, \ldots, j - 4, j + 2, \ldots n$ to form an increasing subsequence of σ with length $n - 2$ which does not contain j.

□

Example 4.2. When a permutation σ satisfies $\textsc{Mexc}(\sigma) = 2$ it is not always the case that for all $i \in [n]$ with $|\sigma(i) - i| = 2$ σ contains an increasing subsequence of length $n-2$ which does not contain $\sigma(i)$. For example, take the permutation $\sigma = 1253746 \in \mathfrak{S}_7$. The unique longest increasing subsequence in this permutation is 12346. But this includes the mapping $6 \mapsto 4$. This means that if we remove 4 from the word, we will be left with a permutation in $\mathfrak{S}_6$ with a longest increasing subsequence of length 4, which means we cannot use the 1-local case to our advantage. However, not all hope is lost. We can show that the coefficient of this permutation must be zero by first examining either the map $5 \mapsto 7$. This gives some idea of where the extra condition in Lemma 4.15 comes from.

To zero out mappings where the image and preimage differ by 1, we use a similar trick as in the 1-local case, but we still need the following lemma.

Lemma 4.16. *Suppose σ satisfies $\textsc{Mexc}(\sigma) = 1$ and contains an increasing subsequence of length $n-2$. Then if $\sigma(i) = j$ and $|i-j| = 1$, σ has an increasing subsequence of length $n-2$ which does not contain j.*

Proof. Suppose σ contains an increasing subsequence of length $n-2$ which includes j. We prove the case where $i < j$, but when $i > j$, the proof is roughly the same. Since $i < j$ and $|i - j| = 1$, $j = i + 1$, so $\sigma(i) = i + 1$. The possible preimages of i are then $i - 1$ and $i + 1$. If $\sigma(i-1) = i$, then we have a contradiction because there are $i-2$ potential preimages before i and $i-1$ potential images they can map to without violating the $\textsc{Mexc}(\sigma) = 1$ condition. Thus $\sigma(i+1) = i$. But then σ has a descent at i, so $i, i+1$ cannot both be in the same increasing subsequence. Since their positions and values are adjacent, we can exchange i for $i+1$ in the increasing subsequence to obtain a new increasing subsequence of length $n-2$ which does not contain $i+1$. □

We now have enough information to prove the result for $k = 2$.

Proposition 4.17. *The set $\mathcal{B}_2$ is a basis for L_2.*

Proof. By Lemma 4.6, $\mathcal{B}_1$ has the correct size, so it is sufficient to show that it is a linearly independent set. By Proposition 2.9, $M^{(n-2,1,1)}$ is a full 2-local module. By Lemma 4.3, to show $\mathcal{B}_2$ is linearly independent, it is sufficient to show that $D^{(n-2,1,1)}(\mathcal{B}_2)$ is linearly independent. Thus we must show that if

$$\sum_{\mathbf{\tilde{1}}_\sigma \in \mathcal{B}_k} c_\sigma \rho^{(n-2,1,1)}(\sigma) = 0,$$

then $c_\sigma = 0$ for all σ. Recall that $\rho^{(n-2,1,1)}(\sigma)$ is an $n(n-1)\times n(n-1)$ matrix with rows and columns indexed by ordered pairs from the alphabet $[n]$ without repetition in lexicographic order, i.e., $(1,2),(1,3),\ldots,(1,n),(2,1),\ldots,(n,n-1)$ and that the $(i,j),(\ell,m)$ entry is 1 when $\sigma(i)=\ell$ and $\sigma(j)=m$ and zero otherwise.

We first consider the ji-blocks with $|i-j|\geq 3$. Let $S\subseteq \mathfrak{S}_n$ be the permutations mapping i to j which contain an increasing subsequence of length $n-2$. Let B_{ji} denote the set of ji-blocks of the $\rho^{(n-2,1,1)}$-representations of the permutations in S. By Lemma 4.12, the matrices in B_{ji} are 1-local representations of permutations formed by removing j from the permutations in S and reducing so that the permutations are in $\mathfrak{S}_{n-1}$. Then by Lemma 4.13, every permutation in S contains an increasing subsequence of length $n-2$ which does not contain j. Thus all the matrices in B_{ji} are 1-local representations of permutations in $\mathfrak{S}_{n-1}$ containing an increasing subsequence of length $n-2$. By Proposition 4.11, B_{ji} is a linearly independent set. To zero out the ji block, then the coefficients c_σ must be zero for all $\sigma\in S$.

Since this holds for all $i,j\in[n]$ with $|i-j|\geq 3$, we are left with

$$\sum_{\substack{\tilde{\mathbf{1}}_\sigma\in\mathcal{B}_k\\ \textsc{Mexc}(\sigma)\leq 2}} c_\sigma \rho^{(n-2,1,1)}(\sigma)=0.$$

Next, we will eliminate all permutations with maximum excedance 2. This is the most difficult step in the proof because we cannot consider a generic mapping; we must instead consider them one at a time. To see why, see Example 4.2. To see the progression of which blocks in the matrix are zeroed out, see Figure 4.2. We wish to show that if $\sigma(m)=m+2$ or $\sigma(m+2)=m$, then $c_\sigma=0$. We proceed by induction on m. We begin with the mappings $1\mapsto 3$ and $3\mapsto 1$. By Lemma 4.14 and the same reasoning as above, all the c_σ with $\sigma(1)=3$ or $\sigma(3)=1$ must be zero. Now suppose that $c_\sigma=0$ when $\sigma(m-1)=m+1$ and when $\sigma(m+1)=m-1$. Recall that $|\sigma(\ell)-\ell|\leq 2$ for all $\ell\in[n]$ and σ has an increasing subsequence of length $n-2$. If $c_\sigma\neq 0$ and $\sigma(m)=m+2$, by condition (a) of Lemma 4.15 σ has an increasing subsequence of length $n-2$ which does not contain $m+2$ since $c_\sigma=0$ when $\sigma(m+1)=m-1$. Similarly, if $\sigma(m+2)=m$, by condition (b) of Lemma 4.15 σ has an increasing subsequence of length $n-2$ which does not contain m since $c_\sigma=0$ when $\sigma(m-1)=m+1$. Then by the same reasoning as above, $c_\sigma=0$ if $\sigma(m)=m+2$ or if $\sigma(m+2)=m$.

Now we can rewrite our sum as

$$\sum_{\substack{\tilde{\mathbf{1}}_\sigma \in \mathcal{B}_k \\ \textsc{Mexc}(\sigma) \leq 1}} c_\sigma \rho^{(n-2,1,1)}(\sigma) = 0.$$

By Lemma 4.16 and the same logic as given above, c_σ must be zero for all σ such that $\textsc{Mexc}(\sigma) = 1$.

Finally, we can rewrite our sum as

$$\sum_{\substack{\tilde{\mathbf{1}}_\sigma \in \mathcal{B}_k \\ \textsc{Mexc}(\sigma) = 0}} c_\sigma \rho^{(n-2,1,1)}(\sigma) = 0.$$

which contains only the identity. Thus c_1 must be zero and so all of the coefficients are zero. Thus $\mathcal{B}_2$ is linearly independent and forms a basis for L_2. □

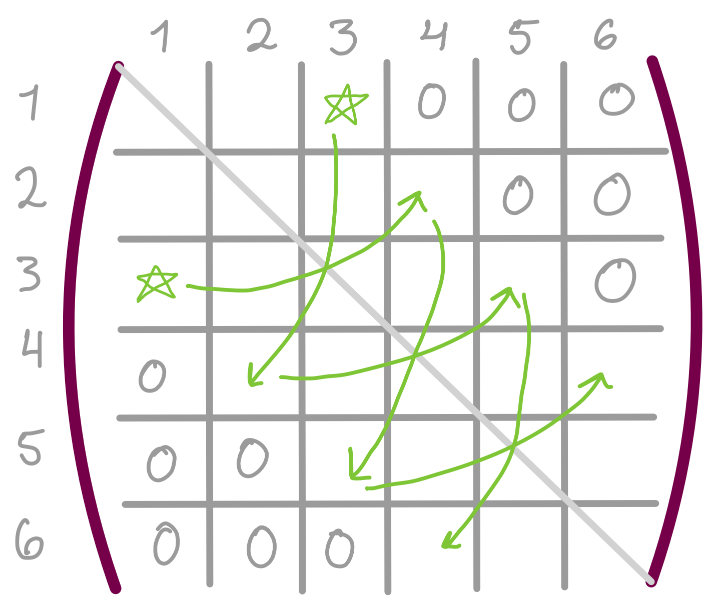


Figure 4.2 Visualization of the order in which permutations containing mappings with excedance 2 are zeroed out in the proof of Proposition 4.17.

4.5 Future Work

Lemma 4.15 is the heart of why the 2-local case is more complex than the 1-local case and why this approach does not scale well. As n grows, so does the region of the permutation which is not guaranteed to be fixed by Lemma 4.8. I have looked into the 3-local case a bit and creating an analog of Lemma 4.15 seems very difficult.

Luckily, the result has already been proven by Doty (2021) so it's okay if we don't prove it ourselves, and now we can ask lots of questions about our new basis!

Question 4.1. *It seems like when we write the* $\mathbf{1}_{(I,J)}$ *functions in this basis, we tend to get integer coefficients. Are they counting something*?

Question 4.2. *If we do always get integer coefficients, then it would mean that permutation statistics could also be written in this basis with integer coefficients. Are those coefficients counting something*?

Question 4.3. *What does the projection of a permutation into the k-local space even look like*? *Are there nice combinatorial descriptions for this function*?

Question 4.4. *In Hamaker and Rhoades* (2022)*, the authors list several open problems. One of these problems is to find a fast technique for projecting into the k-local space. Is it possible to do so with this basis for the k-local space*?

Question 4.5. *Lots of global functions have local analogs. How do the local analogs relate to the local projections, and can this basis help us understand this relationship*?

Question 4.6. *Permutations with long decreasing subsequences also form a basis for this space*! *What other bases are there for the k-local space*? *Is there a nice basis hiding inside the* $\mathbf{1}_{(I,J)}$ *functions*?

Conclusion

I've had a lot of fun thinking about these ideas in the last year, and I hope you enjoyed reading about them. At the beginning of this year, I did not think that I would end up studying any of the ideas in this document. But that's what makes long increasing subsequences so wonderful. A long increasing subsequence is not a complicated object, but you could spend years studying connections between long increasing subsequences and other algebraic and combinatorial ideas, and then spend more years doing that again. My hope is that through reading any part of this, you've been able to get a feel for some of the mathematics I've been working on and maybe been inspired to work on it yourself. I recognize that many of my explanations may not have resonated with you, or that you might have questions. If you do, or if you make any progress towards one of the questions I've listed, please feel free to contact me; I will be very happy to speak with you.

Bibliography

Assaf, Sami. 2019. An inversion statistic for reduced words. *Advances in Applied Mathematics* 107:1–21. doi:https://doi.org/10.1016/j.aam.2019.02.005. URL https://www.sciencedirect.com/science/article/pii/S0196885819300429.

Babson, Eric, and Einar Steingrímsson. 2000. Generalized permutation patterns and a classification of the Mahonian statistics. *Séminaire Lotharingien de Combinatoire [electronic only]* 44:B44b, 18 p.–B44b, 18 p. URL http://eudml.org/doc/120841.

Bóna, Miklós. 2012. *Combinatorics of Permutations, Second Edition*. Discrete Mathematics and Its Applications, Taylor & Francis. URL https://books.google.com/books?id=Op-nF-mBR7YC.

Bousquet-Mélou, Mireille, Anders Claesson, Mark Dukes, and Sergey Kitaev. 2010. (2+2)-free posets, ascent sequences and pattern avoiding permutations. *J Comb Theory, Ser A* 117:884–909.

Claesson, Anders, and Sergey Kitaev. 2008. Classification of bijections between 321- and 132-avoiding permutations. *FPSAC'08 - 20th International Conference on Formal Power Series and Algebraic Combinatorics* DMTCS Proceedings vol. AJ,... doi:10.46298/dmtcs.3594.

Diaconis, Persi, Jason Fulman, and Robert Guralnick. 2008. On fixed points of permutations. *Journal of Algebraic Combinatorics* 28(1):189–218. doi:10.1007/s10801-008-0135-2. URL https://doi.org/10.1007/s10801-008-0135-2.

Doty, Stephen. 2021. Doubly stochastic matrices and Schur-Weyl duality for partition algebras. *Electronic Journal of Combinatorics* 29.

Fässler, Albert, and Eduard Stiefel. 1992. *Group Theoretical Methods and Their Applications*. Birkhäuser.

Foulkes, H.O. 1980. Eulerian numbers, Newcomb's problem and representations of symmetric groups. *Discrete Mathematics* 30(1):3–49.

Gaetz, Christian, and Laura Pierson. 2022. Positivity of permutation pattern character polynomials. arXiv:2204.10633v1.

Gaetz, Christian, and Christopher Ryba. 2021. Stable characters from permutation patterns. *Selecta Mathematica* 27:1–13.

Garsia, A.M., and A. Goupil. 2009. Character polynomials, their q-analogs, and the Kronecker product. *Electronic Journal of Combinatorics* 16. doi: https://doi.org/10.37236/85.

Gessel, Ira M, and Christophe Reutenauer. 1993. Counting permutations with given cycle structure and descent set. *Journal of Combinatorial Theory, Series A* 64(2):189–215. doi:https://doi.org/10.1016/0097-3165(93)90095-P. URL https://www.sciencedirect.com/science/article/pii/009731659390095P.

Guillemot, Sylvain, and Daniel Marx. 2014. *Finding small patterns in permutations in linear time*, 82–101. doi:10.1137/1.9781611973402.7. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611973402.7. https://epubs.siam.org/doi/pdf/10.1137/1.9781611973402.7.

Gunawan, Emily, Jianping Pan, Heather M. Russell, and Bridget Eileen Tenner. 2022. Runs and RSK tableaux of boolean permutations. 2207.05119.

Hamaker, Zachary, and Brendon Rhoades. 2022. Characters of local and regular permutation statistics. arXiv:2206.06567v2.

Hultman, Axel. 2014. Permutation statistics of products of random permutations. *Adv Appl Math* 54:1–10.

Jacques, Alain. 1972. Nombre de cycles d'une permutation et caractères du groupe symétrique. In *Permutation: Actes Du Colloque Sur Les Permutations*, 93–96.

Kagey, Peter. 2021. Expected value of letters of permutations with a given number of k-cycles. URL https://arxiv.org/abs/2112.05281. arXiv:2112.05281v1.

Kitaev, S. 2011. *Patterns in Permutations and Words*. Monographs in Theoretical Computer Science. An EATCS Series, Springer Berlin Heidelberg. URL https://books.google.com/books?id=JgQHtgR5N60C.

Knuth, Donald E. 1968. *The Art of Computer Programming*, vol. 1. Addison-Wesley, 1st ed.

MacMahon, P.A. 1915. *Combinatory Analysis*, vol. 1. Cambridge University Press.

Marcus, Adam, and Gábor Tardos. 2004. Excluded permutation matrices and the Stanley–Wilf conjecture. *Journal of Combinatorial Theory, Series A* 107(1):153–160. doi:https://doi.org/10.1016/j.jcta.2004.04.002. URL https://www.sciencedirect.com/science/article/pii/S0097316504000512.

Mendes, Anthony, and Jeffrey Remmel. 2015. *Counting with symmetric functions*. Springer.

Romik, Dan. 2015. *The Surprising Mathematics of Longest Increasing Subsequences*. Institute of Mathematical Statistics Textbooks, Cambridge University Press. doi:10.1017/CBO9781139872003.

Rubey, Martin, Christian Stump, et al. 2022. FindStat - The combinatorial statistics database. http://www.FindStat.org. URL http://www.FindStat.org. Accessed: May 4, 2023.

Sagan, B. 2001. *The Symmetric Group: Representations, Combinatorial Algorithms, and Symmetric Functions*. Graduate Texts in Mathematics, Springer New York. URL https://books.google.com/books?id=dmrnR48_x38C.

Schensted, C. 1961. Longest increasing and decreasing subsequences. *Canadian Journal of Mathematics* 13:179–191. doi:10.4153/CJM-1961-015-3.

Schützenberger, Marcel Paul. 1963. Quelques remarques sur une construction de Schensted. *Mathematica Scandinavica* 12:117–128.

Stanley, Richard P., and Sergey Fomin. 1999. *Symmetric Functions*, *Cambridge Studies in Advanced Mathematics*, vol. 2, 286–560. Cambridge University Press.