ABSTRACT

HIRD, JOHN THOMAS, JR. Codes and Shifted Codes of Partitions and Compositions. (Under the direction of Naihuan Jing and Ernest Stitzinger.)

Codes of partitions were originally introduced by Stig Comét in 1959. Recently Carrell and Goulden have found a formula for the action of Bernstein's operator on the Schur functions which has a simple representation in terms of codes. In this work, we prove this formula in a new way that we then extend to Schur Q-functions. We determine explicit formulas for the analogous vertex operators acting on Schur Q-function and Hall-Littlewood polynomials. We also give a combinatorial algorithm to express any sequence of Bernstein operators in terms of a Schur function. We show how codes of Schur functions can be used to study Schur Q-functions and vice versa. As another application, we show the connection between Bernstein operators and Schur functions indexed by compositions.

Codes and Shifted Codes of Partitions and Compositions

by

John Thomas Hird, Jr.

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Mathematics

Raleigh, North Carolina

2012

APPROVED BY:

Patricia Hersh	Kailash Misra
Naihuan Jing Co-chair of Advisory Committee	Ernest Stitzinger Co-chair of Advisory Committee

DEDICATION

In memory of Susan Janet Hird. Miss you, Mom.

BIOGRAPHY

John Thomas "J.T." Hird, Jr. was born on December 31, 1984 in Youngstown, Ohio to Susan and Jack Hird. He grew up in Poland, Ohio with younger brother Kevin, and attended Holy Family School and Poland Seminary High School. After graduating, he attended Youngstown State University, majoring in mathematics. He then went to North Carolina State University, where he served as a graduate student teaching assistant, while working toward master's and doctorate degrees in mathematics.

J.T.'s first memory of the joy of mathematics is of his dad teaching him how to divide before his teachers had a chance. His uncle Howie predicted a future in mathematics (in between being a fighter pilot and the president of the United States) after he made a multiplication table to help teach a younger cousin how to multiply. A long string of exceptional teachers pointed him toward his eventual career in math and made that prediction come true.

ACKNOWLEDGEMENTS

First, I would like to thank my advisors Ernest Stitzinger and Naihuan Jing for all their help and patience and my parents Susan and Jack for giving me the opportunities and support that have led me here.

I would also like to thank all the people who helped me realize how much I love math and led me to this career. Mrs. Wargo at Holy Family taught me to love math contests. Mrs. Farina at Poland let me get an early start on math each day. Dr. Buoni taught me for more semesters than I can count and was always ready with advice. Brian Hamilton convinced me that taking discrete math with Dr. Fabrykowski was more important than organic chemistry. Dr. Fabrykowski taught the discrete math class that finally made up my mind about a major, and helped prepare me for the Putnam and other contests with Dr. Smotzer. Dr. Flowers taught me to love algebra and group theory and was my undergraduate advisor. Dr. Stitzinger continued both of these roles, teaching me more algebra and later becoming my graduate advisor. He helped me get through the trials and tribulations of being a grad student and helped me find my second advisor, Dr. Jing, who finally led me to the current body of work in combinatorics and algebra.

TABLE OF CONTENTS

List of Figures . . . . **vi**

Chapter 1 Introduction **1**
- 1.1 History of Codes 1
- 1.2 Preliminaries 2

Chapter 2 Bernstein Operators and Analogs **5**
- 2.1 Introduction 5
- 2.2 Codes of Partitions 7
- 2.3 Bernstein Operators 9
- 2.4 Schur Q-functions 12
- 2.5 Littlewood-Richardson Rule 19

Chapter 3 Equivalence Relations on Codes **23**
- 3.1 Introduction 23
- 3.2 Partitions and Code Models 24
- 3.3 A Relation on Codes 26
- 3.4 Codes of Compositions 28
- 3.5 Schur Functions Indexed by Compositions 34
- 3.6 Schur Q-functions 36
- 3.7 Shifted Codes 41
- 3.8 Reverse Shifted Codes 46
- 3.9 The Schur – Schur Q Correspondence 50

Chapter 4 Hall-Littlewood Polynomials **54**
- 4.1 Hall-Littlewood Analog 54
- 4.2 Restricting to Schur and Schur Q-functions 57

References **61**

LIST OF FIGURES

Figure 1.1 Young diagram and code of $\lambda = (4, 2, 2, 1)$. 2
Figure 1.2 Shifted Young diagram and shifted code of $\lambda = (4, 2, 1)$. 3

Figure 2.1 Young diagram and code of $\lambda = (4, 2, 1)$. 7
Figure 2.2 Change in the code from λ to $\lambda^{(3)}$. 8
Figure 2.3 Change in the code from λ to $\lambda^{[1]}$. 14
Figure 2.4 Change in the code from λ to $\lambda^{[2]}$. 15
Figure 2.5 Change in the code from λ to $\lambda^{[3]}$. 16
Figure 2.6 Shifted Young diagram and shifted code of $\lambda = (5, 4, 2)$. 16
Figure 2.7 Shifted Young diagram and shifted code of $\lambda = (6, 4, 3, 1)$. 17
Figure 2.8 Change in the shifted code from λ to $\lambda^{[2]}$. 18
Figure 2.9 Semistandard Young tableau of shape λ/μ. 20

Figure 3.1 Young diagram and code for $\lambda = (4, 2, 2, 1)$. 25
Figure 3.2 Young diagram and code for $\mu = (2, 3, 1, 4)$. 25
Figure 3.3 Change in the code from commuting $B_n B_m$. 27
Figure 3.4 Alternate version of the change in the code from commuting $B_n B_m$. . . . 28
Figure 3.5 Change in the code from commuting $Y_n Y_m$. 38
Figure 3.6 Shifted Young diagram and shifted code of $\lambda = (4, 2, 1)$. 42
Figure 3.7 Young diagram and code of $\lambda = \phi$. 42
Figure 3.8 Shifted Young diagram and preshifted code of $\lambda = \phi$. 42
Figure 3.9 Shifted Young diagram and shifted code of $\lambda = \phi$. 43
Figure 3.10 Shifted Young diagram and shifted code of $\mu = (2, 3, 1)$. 43
Figure 3.11 Change in the shifted code from commuting $Y_n Y_m$. 44
Figure 3.12 All five code variants for $\lambda = (0, 0, 0, 0)$. 47
Figure 3.13 Reverse-shifted Young diagram and reverse-shifted code of $\mu = (3, 1, 2)$. . 48
Figure 3.14 Change in the reverse-shifted code from commuting $B_n B_m$. 49
Figure 3.15 Correspondences between codes. 51
Figure 3.16 Correspondences between relations on codes. 52

Chapter 1

Introduction

1.1 History of Codes

Codes of partitions were originally introduced by Stig Comét [2] in 1959 in the study of hook lengths and were further developed by J. B. Olsson [10] in 1987. In these first appearances codes were defined as infinite sequences of zeros and ones with only zeros left of some point and only ones right of some point. Then each such sequence uniquely determines a partition. The code can also be recovered by tracing along the bottom-right edge of the Young diagram associated to the partition and writing 0 for each up step and 1 for each right step. This is the realization we will use throughout this paper, though we will use U and R instead of 0 and 1.

Codes are also equivalent to Maya diagrams [3], one of the oldest combinatorial descriptions of partitions. Maya diagrams are also defined as sequences of zeros and ones, except that in this setting the entries are indexed by the integers. So shifting every element of the sequence of zeros and ones one position to the right would give a different Maya diagram since the indices of the entries change, but it would give the same code since codes only keep track of the sequence itself not its position.

Several combinatorial structures similar to codes have also been used by Andrei Okounkov in studying random matrices. These structures include another description of Maya diagrams in [9] and the profile of the partition, which is the piecewise-linear function obtained by tracing out the code of the partition (see [8]).

Most recently Sean Carrell and Ian Goulden have found a formula for the action of Bernstein's operator on the Schur functions, which has a simple interpretation in terms of codes [1]. It is the goal of this work to generalize and elaborate on this result.

1.2 Preliminaries

A partition $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l)$ of n is a sequence of nonnegative integers satisfying $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_l$ whose sum is $\lambda_1 + \lambda_2 + \cdots + \lambda_l = |\lambda| = n$. A Young diagram of shape λ is an array of left aligned boxes with λ_i boxes in the i^{th} row from the top. A shifted Young diagram is a Young diagram in which the i^{th} row has been shifted $i-1$ positions to the right.

Definition 1.2.1. Define the *code* of a partition λ to be the doubly infinite sequence of letters R and U obtained by tracing along the bottom-right edge of the Young diagram of shape λ in the fourth quadrant of the xy-plane together with the negative y- and positive x-axes, where R corresponds to a unit right step and U corresponds to a unit up step.

Example 1.2.2. For the partition $\lambda = (4, 2, 2, 1)$, the path described above is shown in bold in Figure 1.1. Then the code of λ is given by $\alpha = \ldots UUURURUURRURRR\ldots$

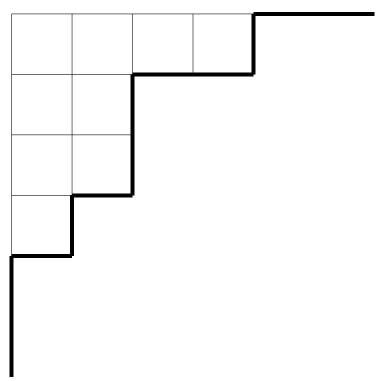


Figure 1.1: Young diagram and code of $\lambda = (4, 2, 2, 1)$.

Definition 1.2.3. Define the *shifted code* of a strict partition λ to be the infinite sequence of letters R and U obtained by tracing along the bottom-right edge of the shifted Young diagram of shape λ in the fourth quadrant together with the positive x-axis, starting at the bottom-right corner of the leftmost box on the bottom row of the diagram.

Example 1.2.4. For the strict partition $\lambda = (4, 2, 1)$, the path described above is shown in bold in Figure 1.2. Then the shifted code of λ is given by $\alpha = UURURRR\ldots$

Let $\Lambda = \mathbb{C}[p_1, p_2, p_3, \ldots] = \oplus_{n=0}^{\infty} \Lambda_n$ be the ring of symmetric functions, where p_n is the power sum symmetric function of degree n. As a graded vector space Λ has several distinguished bases,

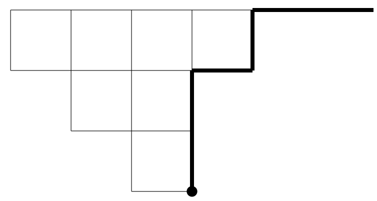


Figure 1.2: Shifted Young diagram and shifted code of $\lambda = (4, 2, 1)$.

such as the power sum symmetric functions p_λ and the Schur functions s_λ [7, 13] both indexed by partitions.

One way to construct the Schur functions is as the images of the Bernstein operators B_n, whose generating function $B(t)$ is a vertex operator:

$$B(t) = \sum_{n \in \mathbb{Z}} B_n t^n = exp\left(\sum_{k \geq 1} \frac{t^k}{k} p_k\right) exp\left(-\sum_{k \geq 1} t^{-k} \frac{\partial}{\partial p_k}\right)$$

which acts on the space Λ. Bernstein also showed the following two results, known as Bernstein's Theorem [15]:

$$B_n B_m = -B_{m-1} B_{n+1}, \tag{1.1}$$

$$s_\lambda = B_{\lambda_1} B_{\lambda_2} \cdots B_{\lambda_l} \cdot 1, \tag{1.2}$$

where s_λ is the Schur polynomial indexed by the partition $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l)$. For convenience we also write B_μ for $B_{\mu_1} B_{\mu_2} \cdots B_{\mu_l}$ for any composition μ.

Carrell and Goulden [1] have used codes of partitions to compute the action of the Bernstein operators on the Schur function s_λ:

$$B(t) s_\lambda = \sum_{i \geq 1} (-1)^{|\lambda| - |\lambda^{(i)}| + i - 1} t^{|\lambda^{(i)}| - |\lambda|} s_{\lambda^{(i)}}, \tag{1.3}$$

where $\lambda^{(i)}$ is another partition defined in terms of the code of the partition λ. They then use this formula to prove Bernstein's Theorem, (1.1) and (1.2). Their proof is combinatorial and they also show that their identity can be used in Plücker relations and KP hierarchies.

In Chapter 2 we show that Carrell and Goulden's formula follows directly from the commutation relation (1.1) satisfied by the Bernstein operators, and show how the same approach can be used to find a formula for the action of the Bernstein operators in any order. We also show the analogous statements to Carrel and Goulden's formula and our generalization for Schur Q-functions [12, 7] and the operator Y_n [6], which have an intuitive description in terms of either codes or shifted codes of partitions. We conclude this chapter with an application of codes to the Littlewood-Richardson Rule and the Pieri Rules.

Chapter 2 is an algebraic study of codes: using algebraic properties to prove identities and then writing the result in an often more intuitive way using codes. Chapter 3 is a purely combinatorial study of codes. We define several equivalence relations on codes (and shifted codes) which allow us to prove the main results from Chapter 2 using only the codes of the partitions involved. In addition, we show the application of these results to Schur functions indexed by compositions and introduce another new combinatorial object *reverse-shifted codes* to help illustrate the relationship between Schur functions and Schur Q-functions.

As Naihuan Jing pointed out in [6], this relationship between the Schur functions and Schur Q-functions is one instance of the celebrated Boson-Fermion correspondence. From this standpoint Schur functions and Schur Q-functions can be viewed as untwisted and twisted pictures of the Fock space representations respectively, and the vertex operators B_n and Y_n come from two different realizations of affine Lie algebras.

Finally, in Chapter 4 we generalize the main theorems in Chapters 2 and 3 on Schur functions and Schur Q-functions to Hall-Littlewood polynomials and show how the previous results can be interpreted as special cases of this more general theory.

Chapter 2

Bernstein Operators and Analogs

2.1 Introduction

In this chapter, we prove that Carrell and Goulden's formula for the action of the Bernstein operator on a Schur function follows directly from algebraic properties of the Bernstein operator. We then show how the relation between the Bernstein operator and the vertex operator $X(t)$ allows us to prove this formula in another way. This same approach can also be generalized to let us study the equivalent operator, $Y(t)$, on Schur Q-functions. To simplify this problem we generalize the combinatorial object *codes* of partitions to the new combinatorial object *shifted codes* of partitions. We then show how codes can be used to study the Littlewood-Richardson and Pieri Rules to show some of the strengths and applications of codes of partitions.

Let $\Lambda = \mathbb{C}[p_1, p_2, p_3 \ldots] = \oplus_{n=0}^{\infty} \Lambda_n$ be the ring of symmetric functions, where p_n is the power sum symmetric function of degree n. As a graded vector space, Λ has several linear bases such as the power sum symmetric functions p_λ and Schur functions s_λ [7, 13] indexed by partitions. One way to construct Schur functions is to realize them as images of Bernstein operators B_n, whose generating function $B(t)$ is a variant of vertex operator [15]:

$$B(t) = \sum_{n \in \mathbb{Z}} B_n t^n = exp\left(\sum_{k \geq 1} \frac{t^k}{k} p_k\right) exp\left(-\sum_{k \geq 1} t^{-k} \frac{\partial}{\partial p_k}\right)$$

which acts on the space Λ. In this construction the Schur function s_λ is easily given by $s_{(\lambda_1,\ldots,\lambda_l)} = B_{\lambda_1} \cdots B_{\lambda_l} \cdot 1$. The operator B_n is a graded linear transformation of degree n defined via its action on the power sum function p_μ. We can also define the operator B_n on the basis of Schur functions. It turns out the action of B_n on Schur functions has a close relationship with Maya diagrams [3], one of the oldest configurations of partitions. Recently Carrell and Goulden [1] have formulated the action of $B(t)$ in terms of codes of partitions, which are certain combinatorial description of Maya diagrams. Similar combinatorial structures have also been used in Okounkov's work on random matrices [9].

Carrell and Goulden use codes of partitions to compute the action of Bernstein operators on the Schur function s_λ:

$$B(t)s_\lambda = \sum_{i\geq 1}(-1)^{|\lambda|-|\lambda^{(i)}|+i-1}t^{|\lambda^{(i)}|-|\lambda|}s_{\lambda^{(i)}}, \tag{2.1}$$

where $\lambda^{(i)}$ is another partition defined in terms of the *code* of the partition λ. They then use this formula to prove Bernstein's Theorem, that $s_{(\lambda_1,\ldots,\lambda_l)} = B_{\lambda_1}\cdots B_{\lambda_l}\cdot 1$ and $B_mB_n = -B_{n-1}B_{m+1}$. Their proof is combinatorial and they also show that their identity can be used in Plücker relations and KP hierarchies.

In this paper we will show that Carrell and Goulden's formula (2.1) can be easily obtained from the classical results using algebraic properties satisfied by vertex operators. We will also generalize the combinatorial structures to the case of Schur Q-functions and derive a similar but simpler combinatorial formulation for the associated vertex operator. The Schur Q-functions are certain distinguished linear bases in the subring of symmetric functions:

$$\Lambda^- = \mathbb{C}[p_1, p_3, p_5\ldots] = \oplus_{n=0}^{\infty}\Lambda_n^-.$$

These symmetric functions were defined by I. Schur in his seminal work [12] on projective representations of the symmetric group S_n (see also [7]). As pointed out in [6] Schur functions and Schur Q-functions are two examples of the celebrated Boson-Fermion correspondence, in which they can be roughly viewed as untwisted and twisted pictures of the Fock space representations respectively, and the vertex operators for Schur and Schur Q-functions come from two different realizations of affine Lie algebras. Taking the advantage of this grand picture we can give a unified approach to derive the action of vertex operators on Schur and Schur Q-functions.

First we can compute the action of Bernstein operator by using the commutation relations:

$$B_mB_n = -B_{n-1}B_{m+1}. \tag{2.2}$$

The combinatorial structure of codes then follows easily from the algebraic structure.

When we tensor the ring Λ by the group algebra of one-dimensional lattice $\mathbb{Z}$, the commutation relations (2.2) can be improved into the exact anti-commutation relations of the vertex operators $X(t)$:

$$X_mX_n = -X_nX_m, \tag{2.3}$$

thus we obtain our second and even simpler proof of Carrell-Goulden's formula (2.1). Using the same idea we can generalize this to the twisted Fock space $\Lambda^- = \mathbb{C}[p_1, p_3, \ldots]$ and again we use the similar antisymmetry of the components of the vertex operator $Y(z)$ (see [6]) to study the action of the Schur Q-functions.

We also formulate the action of the twisted vertex operators in terms of *shifted codes*. In this way we have unified codes and shifted codes in the context of vertex operators and Boson-Fermion correspondence. We also show how these combinatorial objects can help us derive the Littlewood-Richardson Rule and the Pieri Rules.

2.2 Codes of Partitions

Let $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l)$ be a decreasing sequence of positive integers, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_l > 0$. We say that λ is a *partition* of n, denoted $\lambda \vdash n$, if $\lambda_1 + \lambda_2 + \cdots + \lambda_l = n$. We also say that the *weight* of the partition λ is $|\lambda| = n$, and the *length* of the partition is $l(\lambda) = l$.

The *Young diagram* of a partition λ is the left-justified arrangement of boxes with λ_i boxes in the i^{th} row from the top. Since the parts of λ are weakly decreasing, the number of boxes in each row will be less than or equal to the number of boxes in each row above it.

Define the *code* of a partition λ to be the doubly infinite sequence of letters R and U obtained from the Young diagram of shape λ as follows. Consider the Young diagram top and left aligned in the 4^{th} quadrant of the xy-plane together with the negative y-axis and the positive x-axis. Trace up the negative y-axis to the bottom of the Young diagram, then along the bottommost edge of the Young diagram, then right along the positive x-axis. The code of the partition is the sequence of R's and U's obtained from this path, where R corresponds to a unit right step and U corresponds to a unit up step.

Example 2.2.1. Let $\lambda = (4, 2, 1)$. Then the Young diagram of shape λ in the 4^{th} quadrant of the xy-plane is shown in Figure 2.1, with the path described above in bold.

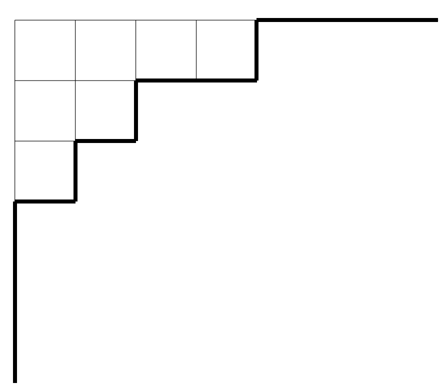


Figure 2.1: Young diagram and code of $\lambda = (4, 2, 1)$.

The path consists of infinitely many U's at the beginning - corresponding to tracing up the negative y-axis, then RURURRU - corresponding to tracing the bottommost edge of the Young diagram, then infinitely many R's at the end - corresponding to tracing right along the positive x-axis. Thus the sequence ...UUURURURRURRR... is the code of the partition $\lambda = (4, 2, 1)$.

Note that the code of any partition will always have infinitely many U's at the beginning of the code, and infinitely many R's at the end of the code (corresponding respectively to the negative y-axis and the positive x-axis).

Define the partition $\lambda^{(i)}$ to be the partition obtained by turning the i^{th} R from the left in the code of λ to a U. Equivalently, $\lambda^{(i)}$ is the partition obtained by looking at the lower-right edge of the associated Young diagram (together with the positive x-axis and negative y-axis, where the Young diagram is considered to be in the 4^{th} quadrant) and turning the i^{th} horizontal edge from the left into a vertical edge (and shifting the resulting path into the 4^{th} quadrant).

Example 2.2.2. Let $\lambda = (4, 2, 1)$. Then to find $\lambda^{(3)}$, the third right step from the left becomes an up step. The changed edge is shown in bold below.

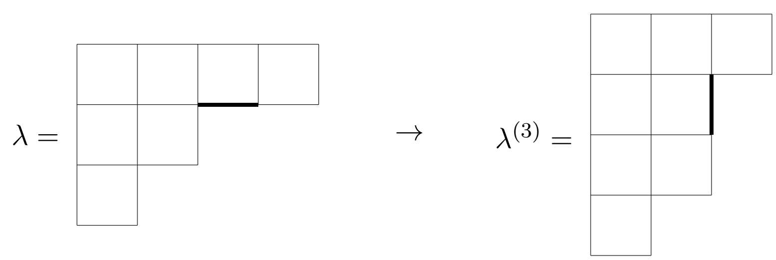


Figure 2.2: Change in the code from λ to $\lambda^{(3)}$.

Thus $\lambda^{(3)} = (3, 2, 2, 1)$.

A simple formula for the partition $\lambda^{(i)}$ is

$$\lambda^{(i)} = (\lambda_1 - 1, \lambda_2 - 1, \ldots, \lambda_j - 1, i - 1, \lambda_{j+1}, \ldots, \lambda_l), \tag{2.4}$$

where $\lambda_j \geq i > \lambda_{j+1}$ (with the convention that $\lambda_{l+1} = 0$ and $\lambda_0 = \infty$).

Given the code of a partition λ, let $u_i(\lambda)$ be the number of U's in the code of λ to the right of the i^{th} R from the left, and let $r_i(\lambda)$ be the number of R's in the code of λ to the left of the i^{th} U from the right. This means that $u_i(\lambda)$ is equal to the number of parts of λ of size at least i, and $r_i(\lambda) = \lambda_i$.

2.3 Bernstein Operators

Recall that the Bernstein operators $B(t)$, and B_n are given by

$$B(t) = \sum_{n\in\mathbb{Z}} B_n t^n = exp\left(\sum_{k\geq 1} \frac{t^k}{k} p_k\right) exp\left(-\sum_{k\geq 1} t^{-k} \frac{\partial}{\partial p_k}\right),$$

which acts on the ring of polynomials $\Lambda = \mathbb{C}[p_1, p_2, p_3, \ldots]$. Here the power sum p_n acts as a multiplication on Λ. Bernstein's primary result with these operators was Bernstein's formula, which states:

$$s_\lambda = B_{\lambda_1} B_{\lambda_2} \cdots B_{\lambda_l} \cdot 1, \tag{2.5}$$

where $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l)$, and s_λ is the Schur polynomial indexed by λ. For convenience, we will often denote this composition as $B_{\lambda_1} B_{\lambda_2} \cdots B_{\lambda_l} \cdot 1 = B_{\lambda_1, \lambda_2, \ldots, \lambda_l} \cdot 1$. Another key relation satisfied by Bernstein operators is the following:

$$B_n B_m = -B_{m-1} B_{n+1}. \tag{2.6}$$

We will now use these two results, Eqs.(2.5) and (2.6), to prove a formula given in [1] which gives the action of Bernstein's operators on the Schur polynomials.

Theorem 2.3.1. *For any partition λ,*

$$B(t) s_\lambda = \sum_{i\geq 1} (-1)^{|\lambda| - |\lambda^{(i)}| + i - 1} t^{|\lambda^{(i)}| - |\lambda|} s_{\lambda^{(i)}}.$$

This result was originally proved in [1] using some combinatorial considerations and the dual action of the Schur functions. In this section we will give two simpler proofs to this result, which will motivate our later generalization to the case of the Schur Q-functions.

Proof. Since $B(t) = \sum_{n\in\mathbb{Z}} B_n t^n$, we only need to determine the action of B_n on s_λ. By equation (2.5),

$$B_n s_\lambda = B_n B_{\lambda_1} B_{\lambda_2} \cdots B_{\lambda_l} \cdot 1.$$

Case 1: If $n \geq \lambda_1$, then by equation (2.5),

$$B_{n, \lambda_1, \lambda_2, \ldots, \lambda_l} \cdot 1 = s_{(n, \lambda_1, \lambda_2, \ldots, \lambda_l)} = s_{\lambda^{(n+1)}},$$

where this term in the summation on the right has a t term of $|\lambda^{(n+1)}| - |\lambda|$. Since $n \geq \lambda_1$, turning the $(n+1)^{\text{th}}$ horizontal edge to a vertical edge creates a new first row of size n. So

$|\lambda^{(n+1)}| - |\lambda| = n$ which is the exponent of t associated with B_n.

Case 2: If $n = \lambda_j - j$ for some j, $1 \leq j \leq l$, then by equation (2.6),

$$\begin{aligned} B_{n,\lambda_1,\lambda_2,\ldots,\lambda_l} \cdot 1 &= (-1)\ B_{\lambda_1-1,n+1,\lambda_2,\ldots,\lambda_l} \cdot 1 \\ &= (-1)^2\ B_{\lambda_1-1,\lambda_2-1,n+2,\lambda_3,\ldots,\lambda_l} \cdot 1 \\ &\vdots \\ &= (-1)^{j-1}\ B_{\lambda_1-1,\lambda_2-1,\ldots,\lambda_{j-1}-1,n+j-1,\lambda_j,\lambda_{j+1},\ldots,\lambda_l} \cdot 1, \end{aligned}$$

but $n = \lambda_j - j$, so $n + j - 1 = \lambda_j - 1$. From equation (2.6), $B_{i,i+1} = -B_{i,i+1}$ for all i, which implies $B_{i,i+1} = 0$ for all i. Since $B_{n+j-1,\lambda_j} = B_{\lambda_j-1,\lambda_j}$ is such a term, this product is zero.

Case 3: If $\lambda_{j+1} - (j+1) < n < \lambda_j - j$ for some j, $1 \leq j < l$, then similarly,

$$\begin{aligned} B_{n,\lambda_1,\lambda_2,\ldots,\lambda_l} \cdot 1 &= (-1)^j\ B_{\lambda_1-1,\lambda_2-1,\ldots,\lambda_j-1,n+j,\lambda_{j+1},\ldots,\lambda_l} \cdot 1 \\ &= (-1)^j\ s_{(\lambda_1-1,\lambda_2-1,\ldots,\lambda_j-1,n+j,\lambda_{j+1},\ldots,\lambda_l)} \\ &= (-1)^{(-n)+(n+j+1)-1}\ s_{\lambda^{(n+j+1)}} \\ &= (-1)^{(|\lambda|-|\lambda^{(n+j+1)}|)+(n+j+1)-1}\ s_{\lambda^{(n+j+1)}}, \end{aligned}$$

by equation (2.5), since $\lambda_1 - 1 \geq \lambda_2 - 1 \geq \cdots \geq \lambda_j - 1 \geq n + j \geq \lambda_{j+1} \geq \cdots \geq \lambda_l$. Note that $|\lambda| - |\lambda^{(n+j+1)}| = -n$ since $\lambda^{(n+j+1)}$ removes the last box from each of the first j rows of λ's Young diagram and then adds a row of size $n + j$. Also note that the exponent of t associated with $s_{\lambda^{(n+j+1)}}$ is $|\lambda^{(n+j+1)}| - |\lambda| = n$, the same exponent associated with B_n.

Case 4: If $n < \lambda_l - l$, then similarly,

$$B_{n,\lambda_1,\lambda_2,\ldots,\lambda_l} \cdot 1 = (-1)^l\ B_{\lambda_1-1,\lambda_2-1,\ldots,\lambda_l-1,n+l} \cdot 1.$$

We further break this into two cases:

- If $n + l \geq 0$, then by equation (2.5),

$$\begin{aligned} (-1)^l\ B_{\lambda_1-1,\lambda_2-1,\ldots,\lambda_l-1,n+l} \cdot 1 &= (-1)^{(-n)+(n+l+1)-1}\ s_{\lambda^{(n+l+1)}} \\ &= (-1)^{|\lambda|-|\lambda^{(n+l+1)}|+(n+l+1)-1}\ s_{\lambda^{(n+l+1)}}, \end{aligned}$$

  because $|\lambda| - |\lambda^{(n+l+1)}| = -n$ since $\lambda^{(n+l+1)}$ removes the last box from each of the l rows of λ's Young diagram and then adds a row of size $n + l$. Again note that the exponent of t associated with $s_{\lambda^{(n+l+1)}}$ is $|\lambda^{(n+l+1)}| - |\lambda| = n$, the same exponent associated with B_n.

- If $n+l<0$, then by equation (2.6):

$$B_{-1,0} = -B_{-1,0} = 0$$

$$B_{-a,0} = -B_{-1,-a+1} = B_{-1,-1,-a+2} = \cdots = (-1)^a \; B_{-1,-1,\ldots,-1,0} = 0,$$

  for all $a \in \mathbb{Z}^+$, since $B_0 \cdot 1 = 1$. This implies that

$$(-1)^l \; B_{\lambda_1-1,\lambda_2-1,\ldots,\lambda_l-1,n+l} \cdot 1 = (-1)^l \; B_{\lambda_1-1,\lambda_2-1,\ldots,\lambda_l-1,-a} \cdot 1 = 0.$$

This proves the theorem. □

We can also prove this theorem using vertex operators. This method will be particularly interesting to us because the same approach can be used to analyze the Schur Q-functions.

To see the symmetry of the indices of the Schur functions, we use a modified version of Bernstein's operator from [6]. Let $\mathbb{C}[\mathbb{Z}]$ be the group algebra of $\mathbb{Z}$ generated by e^p, meaning $\mathbb{C}[\mathbb{Z}] = \oplus_{n\in\mathbb{Z}} \mathbb{C}e^{np}$. Consider the two operators e^p and t^{∂_p} on $\mathbb{C}[\mathbb{Z}]$ defined by

$$\begin{aligned} e^p \cdot e^{np} &= e^{(n+1)p} \\ t^{\partial_p} \cdot e^{np} &= t^n e^{np}. \end{aligned}$$

Following [J1], the vertex operator $X(t)$ is defined on $\Lambda \otimes \mathbb{C}[\mathbb{Z}]$ by

$$X(t) = B(t^{-1}) e^p t^{\partial_p} = \sum_{n\in\mathbb{Z}} X_n t^{-n}.$$

The following result was proved in [4]: the product of the vertex operator $X(t)$ is antisymmetric, so $X_n X_m = -X_m X_n$, and we have the following theorem, which is a modified version from [4].

Theorem 2.3.2. *1. For any $l \in \mathbb{N}$, one has*

$$X_{t_1} \cdots X_{t_l} = (-1)^{l(\sigma)} X_{t_{\sigma(1)}} \cdots X_{t_{\sigma(l)}},$$

for all σ in S_l, where $l(\sigma)$ is the number of inversions in the permutation σ.

2. For any partition $\mu = (\mu_1, \ldots, \mu_l)$, we have

$$X_{-\mu_1} \cdots X_{-\mu_l} \cdot e^{mp} = s_{\mu-\delta+l\mathbb{1}} e^{(m+l)p},$$

where $\delta = (l-1, \ldots, 2, 1, 0)$ *and* $\mathbb{1} = (1, \ldots, 1) \in \mathbb{N}^l$.

In particular, this means that

$$X_{-\mu_1} \cdots X_{-\mu_l} \cdot e^{-lp} = s_{\mu-\delta+l\mathbb{1}}.$$

For simplicity, we removed the index shift of $\frac{1}{2}$ in the definition of $X(t)$ (see [6, 4]).

We can now give a simpler proof of Theorem 2.3.1.

Proof. For simplicity, we will denote the composition as $X_{-\mu_1}X_{-\mu_2}\cdots X_{-\mu_l} = X_{-\mu_1,-\mu_2,\ldots,-\mu_l}$.

$$\begin{aligned}
B(t)s_\lambda &= B(t)X_{-(\lambda_1-1),-(\lambda_2-2),\ldots,-(\lambda_l-l)} \cdot e^{-lp} \\
&= X(t)(e^p t^{\partial_p})^{-1} X_{-(\lambda_1-1),-(\lambda_2-2),\ldots,-(\lambda_l-l)} \cdot e^{-lp} \\
&= X(t)X_{-(\lambda_1-2),-(\lambda_2-3),\ldots,-(\lambda_l-l-1)} \cdot (e^p t^{\partial_p})^{-1} t^{-l} e^{-lp} \\
&= X(t)X_{-(\lambda_1-2),-(\lambda_2-3),\ldots,-(\lambda_l-l-1)} \cdot e^{-(l+1)p} \\
&= \sum_{n\in\mathbb{Z}} X_{-n}X_{-(\lambda_1-2),-(\lambda_2-3),\ldots,-(\lambda_l-l-1)} \cdot e^{-(l+1)p}t^n \\
&= \sum_{n\neq\lambda_k-k-1} (-1)^j X_{-(\lambda_1-2),\ldots,-(\lambda_j-j-1),-n,-(\lambda_{j+1}-j-2),\ldots,-(\lambda_l-l-1)} \cdot e^{-(l+1)p}t^n \\
&= \sum_{n\neq\lambda_k-k-1} (-1)^j X_{-(\lambda_1^{(i)}-1),\ldots,-(\lambda_{l+1}^{(i)}-l-1)} \cdot e^{-(l+1)p}t^n s_{\lambda^{(i)}},
\end{aligned}$$

where $\lambda_j - j - 1 > n > \lambda_{j+1} - j - 2$ and $i = n + j + 1$, so $\lambda_j > i \geq \lambda_{i+1} - 1$. This definition of i also implies that $n = |\lambda^{(i)}| - |\lambda|$, $j = |\lambda| - |\lambda^{(i)}| + i - 1$, and $\lambda^{(i)} = (\lambda_1 - 1, \ldots, \lambda_j - 1, i - 1, \lambda_{j+1}, \ldots, \lambda_l) = (\lambda_1 - 1, \ldots, \lambda_j - 1, n + j, \lambda_{j+1}, \ldots, \lambda_l)$. With this identification this last line becomes the following:

$$B(t)s_\lambda = \sum_{i\geq 1} (-1)^{|\lambda|-|\lambda^{(i)}|+i-1} t^{|\lambda^{(i)}|-|\lambda|} s_{\lambda^{(i)}}$$

and Theorem 2.3.1 is proved. □

2.4 Schur Q-functions

We will next state and prove a similar result for the Schur Q-functions, Q_λ, where λ is a *strict partition*, i.e. $\lambda_1 > \lambda_2 > \cdots > \lambda_l$ and $\lambda_i \in \mathbb{N}$.

For any partition $\mu = (1^{m_1(\mu)}2^{m_1(\mu)}\cdots)$, we define $z_\mu = \prod_{i\geq 1} i^{m_i(\mu)} m_i(\mu)!$. We consider the ring of symmetric functions in $x_1, x_2, \ldots$, but restrict ourselves to polynomials in odd degree power sums

$$p_{2k+1} = \sum_{i \geq 1} x_i^{2k+1}, \qquad k \in \mathbb{Z}_+.$$

Let $\mathcal{O}P$ denote the set of partitions with odd parts, and let Λ^- be the ring of symmetric functions generated by $p_{2k+1}, k \in \mathbb{Z}_+$. Under the inner product

$$< p_\lambda, p_\mu >= 2^{-l(\lambda)} \delta_{\lambda,\mu} z_\lambda, \qquad \lambda, \mu \in \mathcal{OP},$$

the space Λ^- has Q_λ (λ strict) as a distinguished orthogonal basis of symmetric polynomials [12, 7]. They play a fundamental role in the construction of projective representations of the symmetric group S_n.

On the space Λ^- we recall the definition of the twisted vertex operator [6]:

$$Y(t) = \sum_{n \in \mathbb{Z}} Y_n t^{-n} = exp\left(\sum_{k \geq 1} \frac{2t^{-2k+1}}{k} p_{2k-1}\right) exp\left(-\sum_{k \geq 1} t^{2k-1} \frac{\partial}{\partial p_{2k-1}}\right),$$

which acts on the ring of polynomials $\Lambda^- = \mathbb{C}[p_1, p_3, p_5, \ldots]$, and the power sum p_{2k-1} acts as a multiplication on Λ^-.

From [6], we have that the following two results hold:

$$Q_\lambda = Y_{-\lambda_1} Y_{-\lambda_2} \cdots Y_{-\lambda_l} \cdot 1, \tag{2.7}$$

where $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l)$, and Q_λ is the Schur Q-function indexed by λ. Again, we will often denote this composition as $Y_{\lambda_1} Y_{\lambda_2} \cdots Y_{\lambda_l} \cdot 1 = Y_{\lambda_1, \lambda_2, \ldots, \lambda_l} \cdot 1$. The second result is:

$$Y_n Y_m = -Y_m Y_n. \tag{2.8}$$

Theorem 2.4.1. *For any strict partition λ,*

$$Y(t) Q_\lambda = \sum_{n \neq \lambda_j} (-1)^i t^n Q_{(\lambda_1, \lambda_2, \ldots, \lambda_i, n, \lambda_{i+1}, \ldots, \lambda_l)}.$$

Proof. Recall that $Y(t) = \sum_{n \in \mathbb{Z}} Y_n t^{-n}$, then use equations (2.7) and (2.8):

$$\begin{aligned} Y(t) Q_\lambda = \sum_{n \in \mathbb{Z}} Y_n t^{-n} Q_\lambda &= \sum_{n \in \mathbb{Z}} t^n Y_{-n} Y_{-\lambda_1} Y_{-\lambda_2} \cdots Y_{-\lambda_l} \cdot 1 \\ &= \sum_{n \neq \lambda_j} (-1)^i t^n Y_{-\lambda_1, -\lambda_2, -\ldots, -\lambda_i, -n, -\lambda_{i+1}, \ldots, -\lambda_l} \cdot 1 \\ &= \sum_{n \neq \lambda_j} (-1)^i t^n Q_{(\lambda_1, \lambda_2, \ldots, \lambda_i, n, \lambda_{i+1}, \ldots, \lambda_l)}, \end{aligned}$$

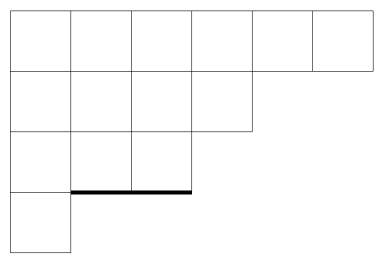
where $\lambda_i > n > \lambda_{i+1}$, because by equation (2.8), $Y_{-n}Y_{-\lambda_j} = Y_{-n}Y_{-n} = 0$ if $n = \lambda_j$. $\square$

We can also interpret the result in terms of codes of strict partitions, but we first need to reinterpret how codes behave for strict partitions.

Definition 2.4.2. Define the partition $\lambda^{[i]}$ to be the partition obtained from the code of a strict partition λ by inserting a U between the i^{th} pair of consecutive R's (with the convention that three consecutive R's counts as two pairs, four consecutive R's counts as three pairs, and so on). Equivalently, $\lambda^{[i]}$ is the partition obtained from the code of λ by inserting a U after the i^{th} R which is immediately followed by an R.

Example 2.4.3. For example, if $\lambda = (6, 4, 3, 1)$, the first pair of consecutive R's in the code of λ is shown in bold in Figure 2.3, with the new edge inserted between them to get $\lambda^{[1]}$ also shown in bold.

$\lambda =$



$\rightarrow$

$\lambda^{[1]} =$

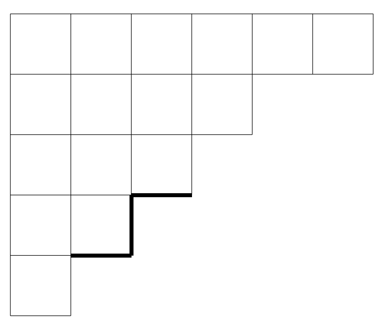


Figure 2.3: Change in the code from λ to $\lambda^{[1]}$.

To get $\lambda^{[2]}$, we insert a U between the second pair of consecutive R's in the code of λ. Again the pair of right steps corresponding to those R's are shown below in bold in Figure 2.4, along with the up step inserted between them.
To get $\lambda^{[3]}$, we insert a U between the third pair of consecutive R's in the code of λ. This works the same way as the previous examples, except that the third pair of R's are in the part of the

$\lambda =$ $\rightarrow$

$\lambda^{[2]} =$

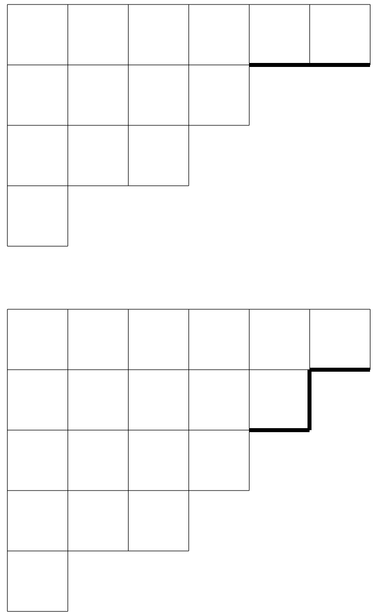


Figure 2.4: Change in the code from λ to $\lambda^{[2]}$.

code corresponding to the positive x-axis. Again the pair of right steps corresponding to those R's are shown in bold in Figure 2.5, along with the up step inserted between them.

Another way to think about $\lambda^{[i]}$ is the following. With this definition $\lambda^{[i]}$ is the strict partition with the i^{th} smallest possible integer inserted into the partition λ. This means that $\lambda^{[1]}$ is the strict partition with the smallest possible integer inserted into λ. For $\lambda = (6, 4, 3, 1)$, the smallest integer that can be inserted to still have a strict partition is 2, so $\lambda^{[1]} = (6, 4, 3, 2, 1)$. The second smallest integer that can be inserted into $\lambda = (6, 4, 3, 1)$ is 5, so $\lambda^{[1]} = (6, 5, 4, 3, 1)$. Similarly, $\lambda^{[3]} = (7, 6, 4, 3, 1)$, $\lambda^{[4]} = (8, 6, 4, 3, 1)$, and so on.

Often strict partitions are associated with *shifted Young diagrams* [14] rather than Young diagrams. A shifted Young diagram of shape λ, where λ is a strict partition, is an arrangement of boxes with λ_i boxes in the i^{th} row, with the leftmost box in each row one unit to the right of the leftmost box of the row above it. This is sometimes more intuitive since the rightmost edge of a shifted Young diagram of shape λ, where λ is a strict partition, follows the same rules of a Young diagram of shape μ, where μ is any partition, namely that the rightmost edge moves weakly left as you go from top to bottom. We can use this correlation to reinterpret $\lambda^{[i]}$ using the analogue of our existing machinery for codes on a shifted Young diagram of shape λ.

Definition 2.4.4. Define the *shifted code* of a strict partition λ to be the infinite sequence of letters R and U obtained from the shifted Young diagram of shape λ as follows. Consider

$\lambda =$ $\rightarrow$

$\lambda^{[3]} =$

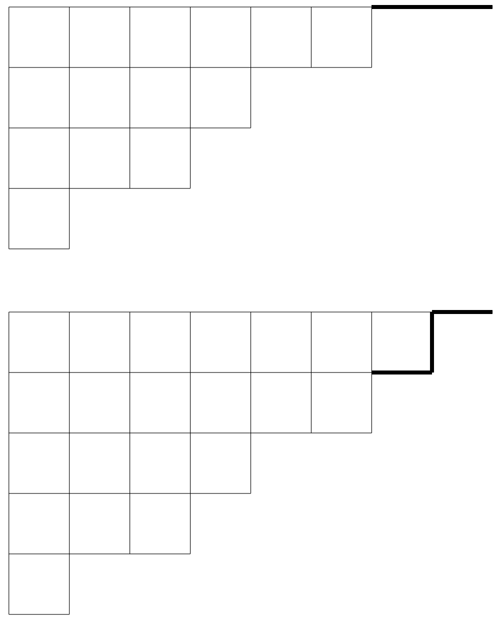


Figure 2.5: Change in the code from λ to $\lambda^{[3]}$.

the shifted Young diagram top and left aligned in the 4^{th} quadrant of the xy-plane together with the positive x-axis. Starting at the bottom right corner of the leftmost box in the last row, trace along the rightmost edge of the shifted Young diagram, then right along the positive x-axis. Equivalently, start the code at the lowest place where the line $y = -x$ intersects the shifted Young diagram. The shifted code of the strict partition is the sequence of R's and U's obtained from this path, where R corresponds to a unit right step and U corresponds to a unit up step.

Example 2.4.5. Let $\lambda = (5, 4, 2)$. Then the shifted Young diagram of shape λ in the 4^{th} quadrant of the xy-plane is shown in Figure 2.6, with the path described above in bold.

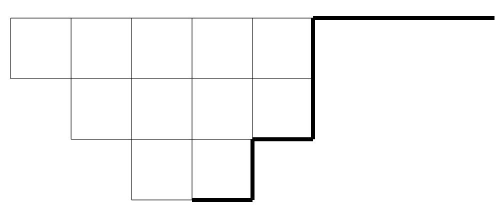


Figure 2.6: Shifted Young diagram and shifted code of $\lambda = (5, 4, 2)$.

Example 2.4.6. Let $\lambda = (6, 4, 3, 1)$. Then the shifted Young diagram of shape λ in the 4^{th} quadrant of the xy-plane is shown in Figure 2.7, with the path described above in bold.

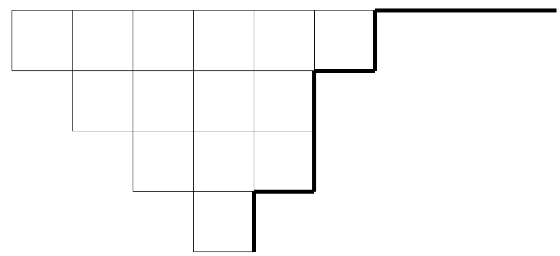


Figure 2.7: Shifted Young diagram and shifted code of $\lambda = (6, 4, 3, 1)$.

Note that the shifted code is not doubly infinite like the code of an arbitrary partition, since it has a fixed starting point. It does however retain the property that there are infinitely many R's at the end of the code.

Using shifted codes we can reinterpret our definition of $\lambda^{[i]}$. For a strict partition λ, $\lambda^{[i]}$ is obtained from the shifted code of λ by turning the i^{th} R in the shifted code to a U. This is since either method inserts the i^{th} smallest possible integer into the partition λ to still have a strict partition, or since the number of pairs of consecutive R's between two U's is the number of consecutive R's minus one, which is the number of R's in the shifted code corresponding to the same row.

Example 2.4.7. We return to our example $\lambda = (6, 4, 3, 1)$. Then we can find $\lambda^{[2]}$ by turning the second right step from the left in the shifted code of λ into an up step. The changed edge is shown in bold in Figure 2.8.

Given the code of a strict partition λ, let $\tilde{u}_i(\lambda)$ be the number of U's in the code of λ to the right of the i^{th} pair of consecutive R's from the left, which is the number of U's in the shifted code of λ to the right of the i^{th} R from the left.

This means that $\tilde{u}_i(\lambda)$ is equal to the number of parts of λ greater than the i^{th} smallest possible integer that can be inserted into λ, which is equal to the number of parts of λ of size at least $|\lambda^{[i]}| - |\lambda|$. Then the number of parts of λ of size at least $|\lambda^{[i]}| - |\lambda|$ is the length of λ minus the number of parts of size less than $|\lambda^{[i]}| - |\lambda|$. But the number of parts less than $|\lambda^{[i]}| - |\lambda|$ is the number of integers less than $|\lambda^{[i]}| - |\lambda|$ minus the number of integers less than $|\lambda^{[i]}| - |\lambda|$ that are not in λ, which is $(|\lambda^{[i]}| - |\lambda| - 1) - (i - 1) = |\lambda^{[i]}| - |\lambda| - i$. So $\tilde{u}_i(\lambda) = l(\lambda) - (|\lambda^{[i]}| - |\lambda| - i) = l(\lambda) + |\lambda| - |\lambda^{[i]}| + i$.

We can now use $\lambda^{[i]}$ to reinterpret Theorem 2.4.1.

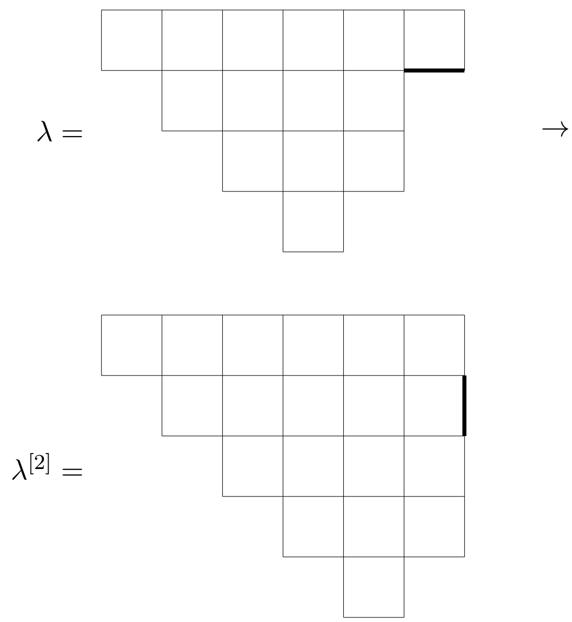


Figure 2.8: Change in the shifted code from λ to $\lambda^{[2]}$.

Theorem 2.4.8. *For any strict partition λ,*

$$Y(t)Q_\lambda = \sum_{i \geq 1} (-1)^{l(\lambda)+|\lambda|-|\lambda^{[i]}|+i} t^{|\lambda^{[i]}|-|\lambda|} Q_{\lambda^{[i]}}.$$

Proof. By Theorem 2.4.1, we know that

$$Y(t)Q_\lambda = \sum_{n \neq \lambda_j} (-1)^k t^n Q_{(\lambda_1,\lambda_2,\ldots,\lambda_k,n,\lambda_{k+1},\ldots,\lambda_l)}.$$

But $\lambda^{[i]}$ is the partition with the i^{th} smallest possible integer that can be inserted into the partition λ. Thus $(\lambda_1, \lambda_2, \ldots, \lambda_k, n, \lambda_{k+1}, \ldots, \lambda_l) = \lambda^{[i]}$ for $i = n - k$, since n is the i^{th} smallest possible integer that can be inserted into λ, so $n = |\lambda^{[i]}| - |\lambda|$. Then k is the number of parts of λ greater than the i^{th} smallest possible integer that can be inserted into λ, so by definition

$k = \tilde{u}_i(\lambda)$. Thus

$$\begin{aligned} Y(t)Q_\lambda &= \sum_{n \neq \lambda_j} (-1)^k t^n Q_{(\lambda_1,\lambda_2,\ldots,\lambda_k,n,\lambda_{k+1},\ldots,\lambda_l)} \\ &= \sum_{i \geq 1} (-1)^{\tilde{u}_i(\lambda)} t^{|\lambda^{[i]}|-|\lambda|} Q_{\lambda^{[i]}} \\ &= \sum_{i \geq 1} (-1)^{l(\lambda)+|\lambda|-|\lambda^{[i]}|+i} t^{|\lambda^{[i]}|-|\lambda|} Q_{\lambda^{[i]}} \end{aligned}$$

since we know $\tilde{u}_i(\lambda) = k = l(\lambda) + |\lambda| - |\lambda^{[i]}| + i$, $|\lambda^{[i]}| - |\lambda| = n$, and $\lambda^{[i]} = (\lambda_1, \lambda_2, \ldots, \lambda_i, n, \lambda_{i+1}, \ldots, \lambda_l)$. □

2.5 Littlewood-Richardson Rule

One application for codes of partitions is the following theorem, which gives a new way to compute Littlewood-Richardson coefficients [7, 11], using only the codes of the partitions involved.

A skew-partition λ/μ is a *horizontal n-strip* if no column in the Young diagram of λ/μ has more than one box. Equivalently, λ/μ is a horizontal n-strip if $\lambda_{i+1} \leq \mu_i \leq \lambda_i$ for all $1 \leq i \leq l(\mu)$, where $l(\mu)$ is the length of μ.

A skew-partition λ/μ is a *vertical n-strip* if no row in the Young diagram of λ/μ has more than one box. Equivalently, λ/μ is a horizontal n-strip if $\lambda_i - 1 \leq \mu_i \leq \lambda_i$ for all $1 \leq i \leq l(\mu)$, where $l(\mu)$ is the length of μ.

Theorem 2.5.1. *(The Littlewood-Richardson Rule)*

$$s_\mu s_\nu = \sum_\lambda c^\lambda_{\mu,\nu} s_\lambda = \sum_{(\mu=\mu^0,\mu^1,\mu^2,\ldots,\mu^l)} s_{\mu^l},$$

where $l = u_1(\nu)$. Given the code of the partition μ^{i-1}, μ^i is obtained as follows:

- *Starting with the U left of the leftmost R in the code of λ and working to the right, move the U's to the right a total of $r_i(\nu)$ places by switching a UR to RU in the code $r_i(\nu)$ times (so no U can move past the starting point of the next U in the code).*
- *Let $k(i,j)$ be the number of UR switches made using the last j U's. Then $k(i,0) = 0$, for all i.*
- *$k(i,j) \leq k(i-1,j-1)$, for all $i, j \geq 0$.*

Note that this proposition implies that $c^\lambda_{\mu,\nu}$ is equal to the number of sequences $(\mu = \mu^0, \mu^1, \mu^2, \ldots, \mu^l = \lambda)$.

Proof. This theorem just follows the computational way to calculate Littlewood-Richardson coefficients, with the only difference being that we use different notation. The sequences $(\mu = \mu^0, \mu^1, \mu^2, \dots, \mu^l = \lambda)$ are in 1-1 correspondence to the semistandard Young tableaux of shape λ/μ with 1's in the boxes in μ^1/μ^0, 2's in the boxes in μ^2/μ^1, $\dots$, and i's in the boxes in μ^i/μ^{i-1} for all $1 \le i \le l = u_1(\nu) = l(\nu)$. The restriction that no U can move past the next U means that for all i, μ^i/μ^{i-1} is a horizontal n-strip, so the corresponding Young tableau is indeed semistandard. The number of i boxes is the number of boxes added to get from μ^{i-1} to μ^i, which is equal to the total number of UR to RU switches made in this step, which is $r_i(\nu)$. This means that the Young tableau obtained has shape λ/μ and weight ν. The requirement $k(i,j) \le k(i-1, j-1)$ means that the number of i's in the first j rows have is less than the number of $(i-1)$'s in the first $(j-1)$ rows for all i and j. This is equivalent to saying that the reverse-row word is a lattice permutation. $\square$

To better illustrate the correspondence between sequences of partitions of the form $(\mu = \mu^0, \mu^1, \mu^2, \dots, \mu^l = \lambda)$ with the preceding conditions and semistandard Young tableaux, we give the following example.

Example 2.5.2. Consider the following: $\lambda = (4,3,2)$, $\mu = (2,1)$, and $\nu = (3,2,1)$, and the sequence $\mu^0 = \mu = (2,1)$, $\mu^1 = (4,1,1)$, $\mu^2 = (4,3,1)$, and $\mu^3 = \lambda = (4,3,2)$. It is straightforward though tedious to verify that this sequence does satisfies the above conditions and hence contributes to $c^{\lambda}_{\mu,\nu}$. If we follow the algorithm in the proof of the theorem and put i's in each box in μ^i/μ^{i-1}, we get the semistandard Young tableau in Figure 2.9:

		1	1
	2	2	
1	3		

Figure 2.9: Semistandard Young tableau of shape λ/μ.

We can also understand this using only the codes of these partitions. Using the algorithm for finding such a sequence, we would find the codes of these partitions (not the partitions themselves) and have the following sequence (omitting leading U's and trailing R's): $\mu^0 = \mu =$ RURU, $\mu^1 =$ RUURRRU, $\mu^2 =$ RURRURU, $\mu^3 = \lambda =$ RRURURU. To get from the code of μ^0 to the code of μ^1 the rightmost U has to move past two R's (since the number of R's between this U and the next rightmost U increases by two). This means that we have to add two boxes to the first row of μ in the first step, which are represented in the semistandard Young tableau with 1's. Similarly, the second U from the right does not have to move past any R's, so there are no boxes added to the second row in the first step thus there are no 1's in the second row in the tableau. Again, the third U from the right must move past one R, so one box is added in the third row and is represented by a 1 in the third row of the tableau. Repeating this same proceedure to get from μ^1 to μ^2 gives us the boxes added in the second step which are represented by 2's in the tableau. Continuing in this way we can find the same semistandard Young tableau using only the codes of the partitions.

Using the codes of partitions we can realize the *Pieri Rules* in a new way.

Corollary 2.5.3. *(The Pieri Rules)*

1. *If $\nu = (n)$, then*
$$s_\mu s_\nu = \sum_\lambda c^\lambda_{\mu,\nu} s_\lambda = \sum_\lambda s_\lambda,$$
where the sum is over all λ such that λ/μ is a horizontal n-strip.

2. *If $\nu = (1^n) = (1, 1, \ldots, 1)$, then*
$$s_\mu s_\nu = \sum_\lambda c^\lambda_{\mu,\nu} s_\lambda = \sum_\lambda s_\lambda,$$
where the sum is over all λ such that λ/μ is a vertical n-strip.

Proof. For part (1), $s_\mu s_\nu = \sum_{(\mu=\mu^0,\mu^1=\lambda)} s_\lambda$, where λ is obtained from the code of μ by moving the U's to the right a total of n places, with no U moving past the starting point of the next U in the code. Thus for any R in the code of μ, at most one U is moved past this R. But since the number of U's moved past the i^{th} R from the left is the number of boxes added to the i^{th} column, this implies that no two of the added boxes are above each other, so λ/μ is a horizontal n-strip. Since $l = u_1(\nu) = 1$, each sequence has length 2, so the third condition in the theorem, $k(i, j) \le k(i-1, j-1)$, is satisfied trivially. For each λ, the multiplicity of s_λ in the summation is the number of sequences $(\mu = \mu^0, \mu^1 = \lambda)$ which is one. Therefore $c^\lambda_{\mu,\nu} = 1$ if λ is a horizontal n-strip, and $c^\lambda_{\mu,\nu} = 0$ otherwise.

For part (2), $s_\mu s_\nu = \sum_{(\mu=\mu^0,\mu^1,\ldots,\mu^n=\lambda)} s_\lambda$, where μ^i is obtained from the code of μ^{i-1} by switching one UR to RU, and $k(i, j) \le k(i-1, j-1)$, for all $i, j \ge 0$. This restriction on $k(i, j)$ implies that the U moved to get from μ^i to μ^{i+1} is left of the U moved to get from μ^{i-1} to μ^i. But since the number of R's moved past the i^{th} U from the right is the number of boxes added to the i^{th} row of μ, this implies that no two of the added boxes are in the same row, so λ/μ is a vertical n-strip. For each λ such that λ/μ is a vertical n-strip, the only way for the sequence $(\mu = \mu^0, \mu^1, \ldots, \mu^n = \lambda)$ to end with the partition λ is for the rightmost box in λ/μ to be added first, then the next furthest right, and so on. Since there is only one way to do this, the multiplicity of s_λ in the summation is one. Therefore $c^\lambda_{\mu,\nu} = 1$ if λ is a vertical n-strip, and $c^\lambda_{\mu,\nu} = 0$ otherwise. $\square$

Chapter 3

Equivalence Relations on Codes

3.1 Introduction

In Chapter 2 we showed that Carrell and Goulden's formula for the action of B_n on any Schur function can be derived algebraically from known properties of B_n. In this chapter we define a combinatorial model of codes and show that the commutation relation satisfied by the Bernstein operators induces a natural relation on codes. We then show that this relation implies Carrell and Goulden's formula as well as a formula for the action Bernstein operators in any order. This provides a natural generalization of Schur functions to be indexed by compositions and we use this to prove the analog of Bernstein's theorem in this setting. We also show the analogous statements for Schur Q-functions and the operator Y_n using both codes and *shifted codes* of partitions and compare these results to those for Bernstein operators.

Bernstein defined the operators $B(t)$ and B_n on the ring of symmetric functions $\Lambda = \mathbb{C}[p_1, p_2, p_3, \ldots]$ by

$$B(t) = \sum_{n \in \mathbb{Z}} B_n t^n = exp\left(\sum_{k \geq 1} \frac{t^k}{k} p_k\right) exp\left(-\sum_{k \geq 1} t^{-k} \frac{\partial}{\partial p_k}\right).$$

He showed the following two results, often referred to as Bernstein's Theorem [15]:

$$B_n B_m = -B_{m-1} B_{n+1}, \tag{3.1}$$

$$s_\lambda = B_{\lambda_1} B_{\lambda_2} \cdots B_{\lambda_l}.1, \tag{3.2}$$

where s_λ is the Schur polynomial indexed by the partition $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l)$. For convenience we also write B_μ for $B_{\mu_1} B_{\mu_2} \cdots B_{\mu_l}$ for any composition μ.

The code of a partition λ is defined to be the sequence of letters R and U obtained by tracing right and up along the outside edge of the Young diagram of shape λ in the fourth quadrant. Carrell and Goulden showed that

$$B(t)s_\lambda = \sum_{i\geq 1}(-1)^{|\lambda|-|\lambda^{(i)}|+i-1}t^{|\lambda^{(i)}|-|\lambda|}s_{\lambda^{(i)}}, \tag{3.3}$$

where $\lambda^{(i)}$ is a particular partition defined in terms of the code of λ [1].

We extend the above model of codes to allow a new *left* move. In the extended model a composition μ is the sequence of letters R, L, and U obtained in the same way using the Young diagram of shape μ, including steps right, left, and up. Using (3.1) we can define an equivalence relation on the codes of compositions by equality of their Bernstein operators. Using this identification we can find a simple formula for B_μ using only the code of μ. In the same spirit the formula (3.3) follows easily.

Schur functions can be defined for any composition as indicated by (3.1). Our new combinatorial model of codes provides a new explanation for the reason behind this. Using the aforementioned results we can state the analog of Bernstein's Theorem (3.2) for Schur functions indexed by compositions. This shows the relationship between Schur functions and our relation on codes.

In Chapter 2 we showed the analog of Carrell and Goulden's formula for the vertex operator Y_n defined in [6] on any Schur Q-function both in terms of codes and in terms of *shifted codes* of partitions. We show how these results follow from the relation induced on either codes or shifted codes by the commutation relation satisfied by Y_n and show the similarities between these two approaches and the corresponding approach for Bernstein operators.

3.2 Partitions and Code Models

A partition $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l)$ of n is a sequence of nonnegative integers satisfying $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_l$ whose sum is n. A Young diagram of shape λ is an array of left aligned boxes with λ_i boxes in the i^{th} row from the top.

Definition 3.2.1. Define the *code* of a partition λ to be the doubly infinite sequence of letters R and U obtained by tracing along the bottom-right edge of the Young diagram of shape λ in the fourth quadrant of the xy-plane together with the negative y- and positive x-axes, where R corresponds to a unit right step and U corresponds to a unit up step.

Example 3.2.2. For the partition $\lambda = (4, 2, 2, 1)$, the path described above is shown in bold in Figure 3.1. Then the code of λ is given by $\alpha = \ldots UUURURUURRURRR\ldots$

We now introduce an extended code model.

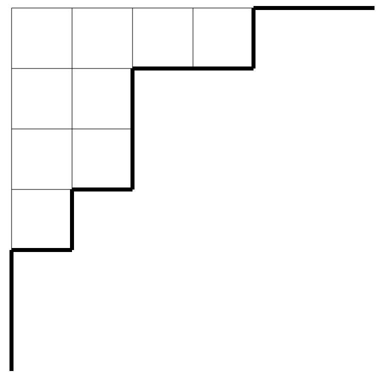


Figure 3.1: Young diagram and code for $\lambda = (4, 2, 2, 1)$.

Definition 3.2.3. Define the *code* of a composition μ to be the doubly infinite sequence of letters R, L, and U obtained by tracing along the rightmost edge of the Young diagram of shape λ in the fourth quadrant of the xy-plane together with the negative y- and positive x-axes, where R corresponds to a unit right step, L corresponds to a unit left step and U corresponds to a unit up step.

Note that the code of a composition μ will contain L's exactly when μ has an exceedance, $\mu_i < \mu_{i+1}$.

Example 3.2.4. For the composition $\mu = (2, 3, 1, 4)$, the path described above is shown in bold in Figure 3.2. Then the code of μ is given by $\alpha = \ldots UUURRRRULLLURRULURRR \ldots$

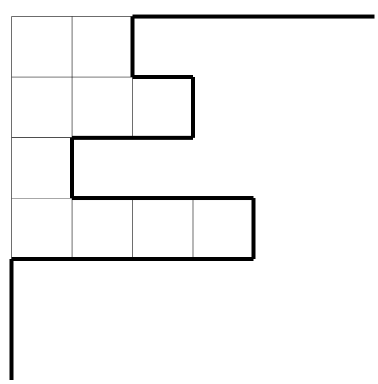


Figure 3.2: Young diagram and code for $\mu = (2, 3, 1, 4)$.

We will often write codes multiplicatively. For instance, we might write R^4 rather than

$RRRR$ in a code. As codes in this setting are a special type of word, we also use the terminology *prefix*, *suffix*, and *subword* in the standard way.

The Bernstein operators $B(t)$ and B_n are defined on the ring of symmetric functions $\Lambda = \mathbb{C}[p_1, p_2, p_3, \ldots]$ by

$$B(t) = \sum_{n\in\mathbb{Z}} B_n t^n = exp\left(\sum_{k\geq 1} \frac{t^k}{k} p_k\right) exp\left(-\sum_{k\geq 1} t^{-k} \frac{\partial}{\partial p_k}\right),$$

where p_k is the k^{th} power sum symmetric function $p_k = \sum_{i\geq 1} x_i^k$. Bernstein showed the following two relations for these operators, often referred to as Bernstein's Theorem:

$$B_n B_m = -B_{m-1} B_{n+1} \tag{3.4}$$

$$s_\lambda = B_{\lambda_1} B_{\lambda_2} \cdots B_{\lambda_l} \cdot 1 \tag{3.5}$$

where $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l)$ is a partition, and s_λ is the Schur polynomial indexed by λ. For convenience we will write B_μ for $B_{\mu_1} B_{\mu_2} \cdots B_{\mu_l}$ for any composition μ.

3.3 A Relation on Codes

Define an equivalence relation $\sim$ on the set of signed codes of compositions by $\alpha \sim \pm\beta$ if and only if $B_{\mu_1} B_{\mu_2} \cdots B_{\mu_l} = \pm B_{\nu_1} B_{\nu_2} \cdots B_{\nu_l}$, and $\alpha \sim 0$ if and only if $B_{\mu_1} B_{\mu_2} \cdots B_{\mu_l} = 0$, where α is the code of μ and β is the code of ν. This is an equivalence relation since there is a one to one correspondence between a composition μ and its code α. For convenience of notation, we will write $\alpha_1 \sim \pm\beta_1$ if α_1 is a subsequence of α and β_1 is a subsequence of β such that $\alpha_0\alpha_1\alpha_2 = \alpha \sim \pm\beta = \pm\beta_0\beta_1\beta_2$ where $\alpha_0 = \beta_0$ and $\alpha_2 = \beta_2$.

Proposition 3.3.1. *For any positive integer k, we have*

$$R\left(L^k U R^{k-1}\right) \sim \left(L^{k-1} U R^{k-2}\right) R \tag{3.6}$$

$$U\left(L^k U R^{k-1}\right) \sim -\left(L^{k-1} U R^{k-2}\right) U. \tag{3.7}$$

Proof. To prove relation (3.6), notice that $LR \sim RL \sim \phi$, the empty set of no letters. In other words, any consecutive L's and R's will cancel since they leave the path and the composition unchanged. The proof is then immediate. Throughout this paper we will assume that all codes have been reduced, meaning any possible LR or RL cancellations have already been made.

Relation (3.7) is actually a version the commutation relation (3.4) in terms of codes. Consider the composition $\mu = (n, m)$, where $n < m$. Using the equation $B_n B_m = -B_{m-1} B_{n+1}$, we get the change in the code of μ shown in Figure 3.3, with the altered path in bold.

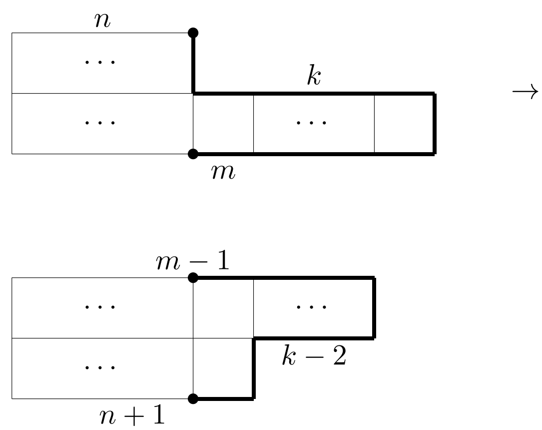


Figure 3.3: Change in the code from commuting $B_n B_m$.

Since the bold path above begins and ends at the same point at the same point horizontally, and since the codes of these two compositions are related from the definition of $\sim$, we have that

$$R^k U L^k U \sim -RUR^{k-2}UL^{k-1}. \tag{3.8}$$

Now multiply both sides of relation (3.8) by L^k on the left and by R^{k-1} on the right and again use the fact that $LR \sim RL \sim \phi$ to obtain (3.7). If we insert the prefix L^k before both of the subsequences in (3.8) and cancel any consecutive LR's, we get exactly identity (3.7). $\square$

We can also realize relation (3.7) directly from the commutation identity of the Bernstein functions (3.4) by following the bold line in Figure 3.4.

In this construction our relation on codes in (3.7) gives exactly the path along the rightmost edge of the diagram for $\mu = (n, m)$ from the bottom-right corner of the bottom row to the point two units up and one unit left of the starting position. The subsequence on the right-hand side of (3.7) is exactly the path along the rightmost edge of the diagram for $\nu = (m-1, n+1)$ which starts and ends at those same points. So relation (3.7) follows directly from the fact that the Bernstein functions indexed by these two compositions are related by (3.4).

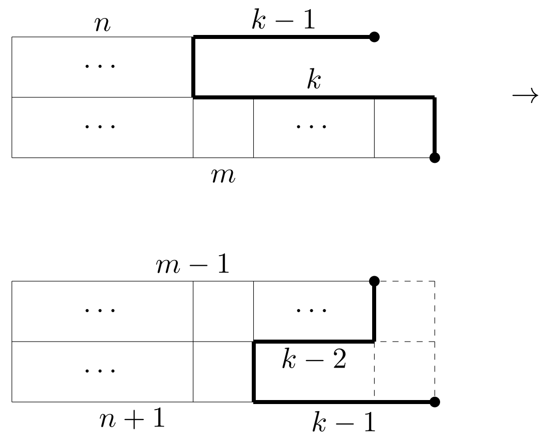


Figure 3.4: Alternate version of the change in the code from commuting B_nB_m.

This interpretation is less intuitive than the original construction, but it serves to show the deep connection that still exists between this relation on codes and the commutation relation of the Bernstein functions.

Notice that the special case $k = 1$ in (3.7) gives that

$$ULU = U(L^1UR^0) \sim -(L^0UR^{-1})U = -ULU,$$

so $ULU \sim 0$, since any composition whose code contains this subword must contain a subsequence $(n+1, n)$ and $B_{n+1}B_n = -B_nB_{n+1} = 0$.

3.4 Codes of Compositions

We now want to use the relation on codes of compositions we developed in Section 3.3 to study B_μ where μ is a composition.

Lemma 3.4.1. *Suppose that the codes α and β of two compositions μ and ν differ only by one of the relations (3.6) or (3.7). Then μ and ν have the same number of components, l, and the same sum, $\mu_1 + \mu_2 + \cdots + \mu_l = \nu_1 + \nu_2 + \cdots + \nu_l$.*

Proof. If α and β differ by relation (3.6), notice the two sides of (3.6) are two different descriptions of the same path, $L^{k-1}UR^{k-1}$, so we actually have that $\mu = \nu$.

If α and β differ by relation (3.6), notice that both sides of (3.7) have a net shift of one unit leftward and two units upward. This implies that μ and ν are the same composition except for the two components determined by the two upward steps in the changed subword. In other words, both μ and ν have length l and $\mu_i = \nu_i$ for $i = 1, 2, \ldots, j-1, j+2, \ldots, l$ for some $1 \leq j \leq l-1$.

It remains only to show that the two components of μ that are changed to get ν have the same sum. Notice that this case corresponds exactly to the picture above, so the corresponding components of μ and ν are (n, m) and $(m-1, n+1)$ for some integer n with $m = n + k$. $\square$

Theorem 3.4.2. *Let μ be any composition of m with code α. Suppose that α can be written in the form*

$$\alpha = \ldots \beta_3\beta_2\beta_1 L^k U \gamma_1\gamma_2\gamma_3 \ldots$$

where $\beta = \ldots \beta_3\beta_2\beta_1$ consists only of R's and U's and $\beta_1 = U$.

- *If $\beta_k = U$, then $B_\mu = 0$.*
- *If $\beta_k = R$, then $B_\mu = (-1)^j B_\nu$, where j is the number of U's in $\beta_{k-1} \ldots \beta_2\beta_1$ and ν is the composition of m with code given by*

$$\ldots \beta_{k+1} U \beta_{k-1} \ldots \beta_3\beta_2\beta_1 L^{k-1} \gamma_1\gamma_2\gamma_3 \ldots$$

Proof. First, rewrite α in the form:

$$\begin{aligned} \alpha &= \ldots \beta_3\beta_2\beta_1 L^k U \gamma_1\gamma_2\gamma_3 \ldots \\ &\sim \ldots \beta_3\beta_2\beta_1 \left(L^k U R^{k-1}\right) L^{k-1} \gamma_1\gamma_2\gamma_3 \ldots \end{aligned}$$

By Proposition 3.3.1, every time we permute $\left(L^k U R^{k-1}\right)$ left past a letter of β, k decreases by one, and the sign changes if that letter was a U. Thus if we permute $\left(L^k U R^{k-1}\right)$ past $k-1$ letters, we get that

$$\begin{aligned} \alpha &\sim (-1)^j \ldots \beta_{k+1}\beta_k \left(L^1 U R^0\right) \beta_{k-1} \ldots \beta_2\beta_1 L^{k-1} \gamma_1\gamma_2\gamma_3 \ldots \\ &= (-1)^j \ldots \beta_{k+1}(\beta_k L U)\beta_{k-1} \ldots \beta_2\beta_1 L^{k-1} \gamma_1\gamma_2\gamma_3 \ldots \end{aligned}$$

where j is the number of U's in $\beta_{k-1} \ldots \beta_2\beta_1$. If $\beta_k = U$, then α is related to a code with the subword $\beta_k L U = U L U \sim 0$, thus $B_\mu = 0$. If $\beta_k = R$, then $\beta_k L U = R L U \sim U$, so

$$\alpha \sim (-1)^j \ldots \beta_{k+1} U \beta_{k-1} \ldots \beta_2\beta_1 L^{k-1} \gamma_1\gamma_2\gamma_3 \ldots \tag{3.9}$$

Thus $B_\mu = (-1)^j B_\nu$, where the code of ν is given by the right hand side of relation (3.9). Since in each step we used only relation (3.7), by Lemma 3.4.1 ν is also a composition of m. $\square$

Another way to understand this theorem is to notice that the code of ν is obtained from the code α of μ by replacing the letter $\beta_k = R$ which is k positions left of the leftmost L in α with U and by replacing L^kU with L^{k-1}, letting j be the number of U's between these two positions.

Corollary 3.4.3. *Let $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l)$ be a partition and n be any integer with $n < \lambda_1$. Let $k = \lambda_1 - n$, and let ζ be the letter $k-1$ positions left of the rightmost U in the code of λ.*

- *If $\zeta = U$, then $B_n B_\lambda = 0$.*
- *If $\zeta = R$, then $B_n B_\lambda = (-1)^{j+1} B_\nu$, where j is the number of U's between the rightmost U and ζ, and where ν is the partition whose code is given by replacing ζ by U.*

Proof. The corollary follows immediately from Theorem 3.4.2, for the special case where $\mu = (n, \lambda_1, \lambda_2, \ldots, \lambda_l)$. In this case, β_1 is the rightmost U in the code of λ, $\beta_k = \zeta$, $\gamma = \gamma_1\gamma_2\gamma_3\ldots = RRR\ldots$ and $\beta\gamma = \ldots\beta_3\beta_2\beta_1\gamma_1\gamma_2\gamma_3\ldots$ is exactly the code of λ. In this case, when we replace L^kU by L^{k-1}, it cancels with the first $k-1$ R's in γ, so our result will actually be a partition. □

Corollary 3.4.4. *Let μ be any composition of m with code α. Then either $B_\mu = 0$ or $B_\mu = \pm B_\lambda$ for some partition λ of m with the same length as μ.*

The proof of this corollary follows directly and immediately from Theorem 3.4.2 by using induction on the number of L's in α. However, we will present the proof using induction on the number of U's to the right of the leftmost L in α to better generalize to our later results.

Proof. Consider the leftmost sequence of consecutive L's in α. If there are no U's right of this sequence, then all the L's cancel, and μ is a partition. So assume there is at least one U after the leftmost L and assume α is written in reduced form. Then in the notation of the theorem, we have

$$\alpha = \ldots\beta_3\beta_2\beta_1 L^k U \gamma_1\gamma_2\gamma_3\ldots$$

where $\beta = \ldots\beta_3\beta_2\beta_1$ consists only of R's and U's. By Theorem 3.4.2, either $\alpha \sim 0$ or

$$\alpha = \ldots\beta_{k+1} U \beta_{k-1}\ldots\beta_3\beta_2\beta_1 L^{k-1}\gamma_1\gamma_2\gamma_3\ldots$$

In the later case the number of U's to the right of the leftmost L has decreased by one (or more if L^{k-1} cancels completely with R's in γ) so the result holds by induction. □

In particular, this corollary provides a simple way to compute $B_\mu \cdot 1$ in terms of Schur functions for any composition μ, since $B_\lambda \cdot 1 = s_\lambda$ by equation (3.5) for any partition λ.

In fact, given any composition μ, the number of times we have to apply Theorem 3.4.2 to get $B_\mu = \pm B_\lambda$ for some partition λ is less than or equal to the number of U's right of the leftmost L in the code of μ. Equivalently, the maximum number of steps is the largest i such that $\mu_i < \mu_{i+1}$, i.e. the position of the last exceedance in μ. From the remark before the proof, we can also say that the number of steps is less than or equal to the total number of L's in the code of μ.

We now present a purely combinatorial approach to computing the partition λ such that $B_\mu = \pm B_\lambda$ for a given composition μ, as well as the sign itself.

Proposition 3.4.5. *Given a composition μ with code α, $B_\mu = (-1)^j B_\lambda$, where j and the code of λ are obtained by reading then deleting letters left to right starting with the leftmost L in α and keeping track of a position in the code, starting with the same L.*

- *Every time an L or R is read, move one position in that direction.*
- *If a U is read and the letter in the current position is a U, $B_\mu = 0$.*
- *If a U is read and the letter in the current position is an R, increase j by the number of U's between these two positions and replace the R with a U. Move right one position.*

Stop when the current position indicates the next letter to be read. If there are still any L's in the code, repeat this process.

Note that in this method you must keep track of two positions, the position which is being read and the "current position" which indicates which letter will be changed by any U's which are read. Note also that this proposition holds even if α is not in reduced form.

Proof. Suppose that the leftmost sequence of consecutive L's in α is L^k and that this is followed by a U (this will always happen if α is in reduced form). Then by Theorem 3.4.2, $\alpha \sim 0$ if the letter k positions left of L^k is a U and if this letter is an R, replace it by U, replace L^kU by L^{k-1}, and change the sign by the number of U's between these positions. Following the algorithm in the proposition, we read k L's so we move left k positions. Since the next letter is a U, we perform the same change to the sign and the letter in the current position and will next consider the letter one position to the right of the changed position, which corresponds to L^kU being replaced by L^{k-1}.

If the letter after L^k is R, then $L^kR \sim L^{k-1}$ so we will next consider the letter one position further to the right. If the letter after L^k is L, then $L^kL \sim L^{k+1}$ so we will next consider the letter one position further to the left. We stop this process when all the L's in L^k have been cancelled. □

Example 3.4.6. Consider the composition $\mu = (1, 3, 1, 6, 2)$. Then the proposition gives the following. In each step below we read off letters until a U is reached and underline these letters, while the arrows indicate how the current position changes. Note that we delete the underlined letters after each step.

$$\begin{array}{rcll}
\alpha & = & & \ldots UUURRURRRRU\underline{LLLLLU}RRULLURRRR\ldots \\
& \to & (-1)^1 & \ldots UUURRUURRRU\underline{RRU}LLURRRR\ldots \\
& \to & (-1)^{1+1} & \ldots UUURRUURRUU\underline{LLU}RRRR\ldots \\
& \to & (-1)^{1+1+2} & \ldots UUURRUURUUU\underline{RR}RR\ldots \\
\alpha & \sim & (-1)^{1+1+2} & \ldots UUURRUURUUURR\ldots
\end{array}$$

We stop since the current position is the same as the next letter to be read. This final code has no remaining L's so we are done. This code is the code of the partition $\lambda = (3, 3, 3, 2, 2)$, so $B_{(1,3,1,6,2)} = (-1)^4 B_{(3,3,3,2,2)} = B_{(3,3,3,2,2)}$. In particular, this means that $B_1 B_3 B_1 B_6 B_3 \cdot 1 = B_\mu \cdot 1 = B_\lambda \cdot 1 = s_\lambda = s_{(3,3,3,3,3)}$.

Definition 3.4.7. Define $r_i(\lambda)$ to be the number of R's in the code of the partition λ left of the i^{th} U from the right in the code of λ.

Note that $r_i(\lambda) = \lambda_i$, the i^{th} component of λ.

Definition 3.4.8. For any partition λ, define $\lambda^{(i)}$ to be the partition obtained from the code of λ by replacing the i^{th} R from the left in the code of λ by U.

In particular, this means that $\lambda^{(i)} = (\lambda_1 - 1, \lambda_2 - 1, \ldots, \lambda_j - 1, i - 1, \lambda_{j+1}, \ldots, \lambda_l)$, where $\lambda_j - 1 \geq i - 1 \geq \lambda_{j+1}$, so $\lambda_j \geq i > \lambda_{j+1}$. By convention, we take $\lambda_0 = \infty$ and $\lambda_{l+1} = 0$, so the formula for $\lambda^{(i)}$ holds for $0 \leq j \leq l$.

Theorem 3.4.9. *For any partition λ,*

$$B(t)s_\lambda = \sum_{i \geq 1} (-1)^{|\lambda| - |\lambda^{(i)}| + i - 1} t^{|\lambda^{(i)}| - |\lambda|} s_{\lambda^{(i)}}. \tag{3.10}$$

Proof. From the definition of $B(t) = \sum_{n\in\mathbb{Z}} B_n t^n$, and equation (3.5), we have that $B(t)s_\lambda = \sum_{n\in\mathbb{Z}} B_n t^n (B_{\lambda_1} B_{\lambda_2} \cdots B_{\lambda_l} \cdot 1)$. So it is sufficient to consider the coefficient of t^n, $B_n B_{\lambda_1} B_{\lambda_2} \cdots B_{\lambda_l}$.

If $n \geq \lambda_1$, then $B_n B_\lambda = B_{(n,\lambda)}$ is already in decreasing order. In fact, since $(n, \lambda) = \lambda^{(n+1)}$, we can write $B_n B_\lambda \cdot 1 = B_{(n,\lambda)} \cdot 1 = B_{\lambda^{(n+1)}} \cdot 1 = s_{\lambda^{(n+1)}}$.

If $n < \lambda_1$, then using the notation of Theorem 3.4.2, we write the code of λ as

$$\alpha = \ldots \beta_3 \beta_2 \beta_1 L^k URRR\ldots$$

where $\beta_1 = U$. Let j be the number of U's in $\beta_k \beta_{k-1} \ldots \beta_2 \beta_1$, where $k = \lambda_1 - n$.

From Corollary 3.4.3, if $\beta_k = U$, then $\alpha \sim 0$, so $B_n B_\lambda = 0$. Since β_k is the j^{th} U from the right in $\beta_k\beta_{k-1}\ldots\beta_2\beta_1$, β_k is also the j^{th} U from the right in α. Since $r_i(\lambda) = \lambda_i$ is the number of R's left of the i^{th} U from the right in α, the number of R's between the $(i+1)^{\text{th}}$ U from the right and the i^{th} U from the right is $r_i(\lambda) - r_{i+1}(\lambda) = \lambda_i - \lambda_{i+1}$. In this case we can write

$$\alpha = \ldots\beta_{k+1}UR^{\lambda_{j-1}-\lambda_j}U\ldots UR^{\lambda_2-\lambda_3}UR^{\lambda_1-\lambda_2}URRR\ldots$$

So the k letters $\beta_k\beta_{k-1}\ldots\beta_2\beta_1$ consist of j U's and $(\lambda_1-\lambda_2)+(\lambda_2-\lambda_3)+\cdots+(\lambda_{j-1}-\lambda_j)$ R's. So we have:

$$\begin{aligned}
\lambda_1 - n \quad = \quad k &= j + (\lambda_1-\lambda_2)+(\lambda_2-\lambda_3)+\cdots+(\lambda_{j-1}-\lambda_j)\\
n+j &= \lambda_1 - (\lambda_1-\lambda_2)-(\lambda_2-\lambda_3)-\cdots-(\lambda_{j-1}-\lambda_j)\\
n+j &= \lambda_j\\
n &= \lambda_j - j
\end{aligned}$$

Thus $B_n B_\lambda = 0$ exactly when $n = \lambda_j - j$ for some $j = 1, 2, \ldots, l$.

If $n \neq \lambda_j - j$ for any j, then we must have that $\beta_k = R$. Hence by Corollary 3.4.3, $B_{(n,\lambda)} = (-1)^j B_\nu$, where j is the number of U's in $\beta_k\beta_{k-1}\ldots\beta_2\beta_1$ and ν is the partition whose code is the same as α except that $\beta_k = U$. Then the number of R's in $\beta_k\beta_{k-1}\ldots\beta_2\beta_1$ is $k-j$, and the number of R's in all of $\beta = \ldots\beta_{k+1}\beta_k\beta_{k-1}\ldots\beta_2\beta_1$ is $\lambda_1 = r_1(\lambda)$, which is the number of R's left of the rightmost U in α, β_1. Thus there are $\lambda_1-(k-j) = \lambda_1-k+j = \lambda_1-(\lambda_1-n)+j = n+j$ R's left of β_k in α. So ν is the partition obtained by replacing the $(n+j+1)^{\text{th}}$ R from the left in the code α of λ with a U. Thus $\nu = \lambda^{(n+j+1)}$ and $B_n B_\lambda \cdot 1 = (-1)^j B_\nu \cdot 1 = (-1)^j s_{\lambda^{(n+j+1)}}$.

Using the convention that $\lambda_0 = \infty$ and $\lambda_{l+1} = 0$, we obtain a cover of the integers greater than or equal to $-l$: $[-l,\infty) = \bigcup_{j=0}^{l}(\lambda_{j+1}-(j+1), \lambda_j - j]$. We know that if $n < -l$, then $B_n s_\lambda = (-1)^l B_{\lambda_1-1}B_{\lambda_2-1}\cdots B_{\lambda_l-1}B_{n+l}\cdot 1 = 0$ since $B_{-m}\cdot 1 = B_{-m}B_0\cdot 1 = B_{-1}B_{-1}\cdots B_{-1}B_0\cdot 1 = 0$. Also note that the right limits of this cover are of the form $n = \lambda_j - j$, so we know that $B_n B_\lambda = 0$. Thus:

$$B(t)s_\lambda = \sum_{n\in\mathbb{Z}} t^n B_n s_\lambda = \sum_{j=0}^{l}\sum_{n=\lambda_{j+1}-j}^{\lambda_j-j} t^n B_n s_\lambda = \sum_{j=0}^{l}\sum_{n=\lambda_{j+1}-j}^{\lambda_j-j-1}(-1)^j t^n s_{\lambda^{(n+j+1)}}.$$

But looking at the summation on the right hand side, $\lambda_{j+1} - j \leq n \leq \lambda_j - j - 1$, so $\lambda_{j+1}+1 \leq n+j+1 \leq \lambda_j$, so $\lambda_{j+1} < n+j+1 \leq \lambda_j$ so the indices of $\lambda^{(n+j+1)}$ in the summation cover all positive integers. Hence

$$B(t)s_\lambda = \sum_{j=0}^{l} \sum_{n=\lambda_{j+1}-j}^{\lambda_j-j-1} (-1)^j t^n s_{\lambda^{(n+j+1)}} = \sum_{i\geq 1} (-1)^{|\lambda|-|\lambda^{(i)}|+i-1} t^{|\lambda^{(i)}|-|\lambda|} s_{\lambda^{(i)}},$$

since $|\lambda| - |\lambda^{(i)}| + i - 1 = j$ and $|\lambda^{(i)}| - |\lambda| = n$ from the definition of $\lambda^{(i)}$. $\square$

3.5 Schur Functions Indexed by Compositions

In this section we will use the functions, *Schur functions indexed by compositions*, to show the usefulness of the results obtained in the previous section. In this section we will consider only compositions consisting of all nonnegative components.

Let $\delta = (l-1, l-2, \ldots, 2, 1, 0)$ and define $a_\mu = \det({x_i}^{\mu_j})_{1\leq i,j\leq l}$ for any composition μ of n of length l. Then one classical definition of the *Schur polynomials* is given by $s_\lambda(x_1, x_2, \ldots, x_l) = \dfrac{a_{\lambda+\delta}}{a_\delta}$ for any partition λ of n of length l. This definition can also be generalized to compositions as follows.

Definition 3.5.1. Let μ be a composition of n of length l. Define the *Schur polynomial indexed by the composition* μ to be $s_\mu(x_1, x_2, \ldots, x_l) = \dfrac{a_{\mu+\delta}}{a_\delta}$.

Lemma 3.5.2. *Let μ and ν be any two compositions of length l with codes α and β. Then $s_\mu(x_1, x_2, \ldots, x_l) = \pm s_\nu(x_1, x_2, \ldots, x_l)$ if and only if $\alpha \sim \pm\beta$ as described in Sections 3.3 and 3.4.*

Proof. By definition, $s_\mu(x_1, x_2, \ldots, x_l) = \dfrac{a_{\mu+\delta}}{a_\delta}$ and $s_\nu(x_1, x_2, \ldots, x_l) = \dfrac{a_{\nu+\delta}}{a_\delta}$, so it suffices to show that $a_{\mu+\delta} = a_{\nu+\delta}$ precisely when the codes of μ and ν are related. By the antisymmetry of the determinant, we have

$$\begin{aligned}
a_{\mu+\delta} &= \begin{vmatrix} {x_1}^{\mu_1+l-1} & \dots & {x_1}^{\mu_i+l-i} & {x_1}^{\mu_{i+1}+l-i-1} & \dots & {x_1}^{\mu_l+0} \\ {x_2}^{\mu_1+l-1} & \dots & {x_2}^{\mu_i+l-i} & {x_2}^{\mu_{i+1}+l-i-1} & \dots & {x_2}^{\mu_l+0} \\ \vdots & & \vdots & \vdots & & \vdots \\ {x_l}^{\mu_1+l-1} & \dots & {x_l}^{\mu_i+l-i} & {x_l}^{\mu_{i+1}+l-i-1} & \dots & {x_l}^{\mu_l+0} \end{vmatrix} \\
&= -\begin{vmatrix} {x_1}^{\mu_1+l-1} & \dots & {x_1}^{\mu_{i+1}+l-i-1} & {x_1}^{\mu_i+l-i} & \dots & {x_1}^{\mu_l+0} \\ {x_2}^{\mu_1+l-1} & \dots & {x_2}^{\mu_{i+1}+l-i-1} & {x_2}^{\mu_i+l-i} & \dots & {x_2}^{\mu_l+0} \\ \vdots & & \vdots & \vdots & & \vdots \\ {x_l}^{\mu_1+l-1} & \dots & {x_l}^{\mu_{i+1}+l-i-1} & {x_l}^{\mu_i+l-i} & \dots & {x_l}^{\mu_l+0} \end{vmatrix} \\
&= -\begin{vmatrix} {x_1}^{\mu_1+l-1} & \dots & {x_1}^{(\mu_{i+1}-1)+l-i} & {x_1}^{(\mu_i+1)+l-i-1} & \dots & {x_1}^{\mu_l+0} \\ {x_2}^{\mu_1+l-1} & \dots & {x_2}^{(\mu_{i+1}-1)+l-i} & {x_2}^{(\mu_i+1)+l-i-1} & \dots & {x_2}^{\mu_l+0} \\ \vdots & & \vdots & \vdots & & \vdots \\ {x_l}^{\mu_1+l-1} & \dots & {x_l}^{(\mu_{i+1}-1)+l-i-1} & {x_l}^{(\mu_i+1)+l-i-1} & \dots & {x_l}^{\mu_l+0} \end{vmatrix} \\
&= -\ a_{(\mu_1,\dots,\mu_{i-1},\mu_{i+1}-1,\mu_i+1,\mu_{i+2},\dots,\mu_l)+\delta}.
\end{aligned}$$

Hence, the indices, μ, of $a_{\mu+\delta}$ satisfy the same commutation relation as the indices of the Bernstein operators in equation 3.4. Thus, by the definition of the equivalence relation $\sim$ on signed codes of compositions, $a_{\mu+\delta} = a_{\nu+\delta}$ if and only if the codes of μ and ν are related, so the result holds. □

Definition 3.5.3. Let μ be a composition of n of length l. Define the *Schur function indexed by the composition* μ, s_μ, to be the unique symmetric function in $\oplus_{k=0}^{l}\Lambda_k$ whose restriction to $x_{l+1} = x_{l+2} = \cdots = 0$ is the Schur polynomial $s_\mu(x_1, x_2, \dots, x_l)$.

Note that if $\mu = \lambda$ is a partition, we obtain recover the classical definition of Schur functions. Note also that in Λ there is not a unique symmetric function whose restriction to $x_{l+1} = x_{l+2} = \cdots = 0$ is $s_\mu(x_1, x_2, \dots, x_l)$. In fact, given any such function s_μ, any element of the coset $s_\mu + < e_{l+1}, e_{l+2}, \dots >$ will satisfy this condition, where $e_m = \sum_{1\le i_1<i_2<\cdots<i_m}(x_{i_1}x_{i_2}\cdots x_{i_m})$ is the elementary symmetric function.

Theorem 3.5.4. *Let μ and ν be any two compositions of length l with codes α and β. Then $s_\mu = \pm s_\nu$ if and only if $\alpha \sim \pm\beta$ as described in Sections 3.3 and 3.4.*

Proof. Suppose that $s_\mu = \pm s_\nu$. Then restricting to $x_{l+1} = x_{l+2} = \cdots = 0$ we obtain that $s_\mu(x_1, x_2, \dots, x_l) = \pm s_\nu(x_1, x_2, \dots, x_l)$. Hence by Lemma 3.5.2, $\alpha \sim \pm\beta$.

Suppose that $\alpha \sim \pm\beta$. Then by Lemma 3.5.2,

$$s_\mu(x_1, x_2, \ldots, x_l) = \pm s_\nu(x_1, x_2, \ldots, x_l).$$

Now since the Schur functions s_μ and s_ν are both uniquely determined by the same Schur polynomial $s_\mu(x_1, x_2, \ldots, x_l) = \pm s_\nu(x_1, x_2, \ldots, x_l)$ (up to the sign), we have that $s_\mu = \pm s_\nu$. $\square$

Theorem 3.5.5. *Given any composition $\mu = (\mu_1, \mu_2, \ldots, \mu_l)$,*

$$s_\mu = B_{\mu_1} B_{\mu_2} \cdots B_{\mu_l} \cdot 1.$$

Proof. If $B_\mu = 0$, then the columns of $({x_i}^{\mu_j + \delta_j})$ will be linearly dependent and $a_{\mu+\delta} = 0$, so $s_\mu = 0$. Thus $B_\mu \cdot 1 = 0 = s_\mu$.

If $B_\mu \neq 0$, then by Corollary 3.4.4, $B_\mu = \pm B_\lambda$ for some partition λ. Hence $B_\mu \cdot 1 = \pm B_\lambda \cdot 1$. By equation (3.5) we know that $B_\lambda \cdot 1 = s_\lambda$. Finally, $s_\lambda = \pm s_\mu$ by Theorem 3.5.4, where the sign is the same as above, since both come from the relation between the codes of μ and λ. Therefore $B_\mu \cdot 1 = \pm B_\lambda \cdot 1 = \pm s_\lambda = s_\mu$. $\square$

This theorem tells us that when the Bernstein operators act in an arbitrary order on 1, i.e. when they are indexed by a composition, then the result is the Schur function indexed by that same composition. This generalizes Berstein's theorem (3.5), which gives the same result when the Bernstein operators act in nonincreasing order on 1, i.e. when they are indexed by a partition, obtaining the classical Schur functions (indexed by partitions) as the result.

3.6 Schur Q-functions

We now turn our attention to Schur Q-functions and show some analogous results using codes of *strict* partitions. The Schur Q-functions are denoted by Q_λ, where λ is a *strict partition*, i.e. $\lambda_1 > \lambda_2 > \cdots > \lambda_l$. The functions Q_λ where λ is a strict partition are an orthogonal basis of Λ^-, the ring of symmetric functions generated by the odd degree power sums p_{2k+1}. Then from [6] we have that the twisted vertex operator $Y(t)$ given by

$$Y(t) = \sum_{n\in\mathbb{Z}} Y_n t^{-n} = exp\left(\sum_{k\geq 1} \frac{2t^{-2k+1}}{k} p_{2k-1}\right) exp\left(-\sum_{k\geq 1} t^{2k-1} \frac{\partial}{\partial p_{2k-1}}\right),$$

which acts on Λ^-, satisfies the following two results. The first is that $Y_{-\lambda}$ generates Q_λ in the same way that the Bernstein operator B_λ generates the Schur function s_λ. That is

$$Q_\lambda = Y_{-\lambda_1} Y_{-\lambda_2} \cdots Y_{-\lambda_l} \cdot 1, \tag{3.11}$$

where $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l)$ is a strict partition and Q_λ is the Schur Q-function indexed by λ. The second result is that the operators Y_n anticommute.

$$Y_n Y_m = -Y_m Y_n \tag{3.12}$$

for any integers m and n. In particular, this means that $Y_n Y_n = -Y_n Y_n$, so $Y_n Y_n = Y_n^2 = 0$.

Now we use the relationship among the Y_n's to define a new equivalence relation on the set of signed codes of compositions. We define $\alpha \dot{\sim} \pm\beta$ if and only if $Y_{-\mu} = \pm Y_{-\nu}$, and $\alpha \dot{\sim} 0$ if and only if $Y_{-\mu} = 0$, where α is the code of the composition μ and β is the code of the composition ν. As in Section 3.3, this is an equivalence relation since there is a one to one correspondence between a composition μ and its code α. Throughout this section we will refer only to this new relation.

Since the operators Y_n anticommute, we know that $Y_\mu = 0$ whenever μ contains any repetitions, so we can restrict ourselves to compositions μ with distinct components, but we want to recapture this in terms of codes alone.

Proposition 3.6.1. *For any positive integer k, we have*

$$R\left(L^k U R^k\right) \;\dot{\sim}\; \left(L^{k-1} U R^{k-1}\right) R \tag{3.13}$$

$$U\left(L^k U R^k\right) \;\dot{\sim}\; -\left(L^k U R^k\right) U. \tag{3.14}$$

The proof of this proposition is similar to Proposition 3.3.1. Relation (3.14) follows from the commutation identity (3.12) and corresponds to the code given by the altered path in Figure 3.5.

Note that unlike Proposition 3.3.1, Proposition 3.6.1 says that in this setting we only decrease the index k when we permute $\left(L^k U R^k\right)$ past an R. However, like the previous case, the sign only changes when we permute $\left(L^k U R^k\right)$ past a U.

Lemma 3.6.2. *Suppose that the codes α and β of two compositions μ and ν differ only by one of the relations (3.13) or (3.14). Then μ and ν have the same number of components, l, and the same sum, $\mu_1 + \mu_2 + \cdots + \mu_l = \nu_1 + \nu_2 + \cdots + \nu_l$.*

The proof is identical to Lemma 3.4.1.

Theorem 3.6.3. *Let μ be any composition of m with code α. Suppose that α can be written in the form*

$$\alpha = \ldots \beta_3 \beta_2 \beta_1 L^k U \gamma_1 \gamma_2 \gamma_3 \ldots$$

where $\beta = \ldots \beta_3 \beta_2 \beta_1$ consists only of R's and U's and $\beta_1 = U$. Let j be the smallest integer such that $\beta_{k+j} \ldots \beta_2 \beta_1$ has k R's. Then $Y_{-\mu} = 0$ if $\beta_{k+j+1} = U$ and $Y_{-\mu} = (-1)^j Y_{-\nu}$ if $\beta_{k+j+1} = R$,

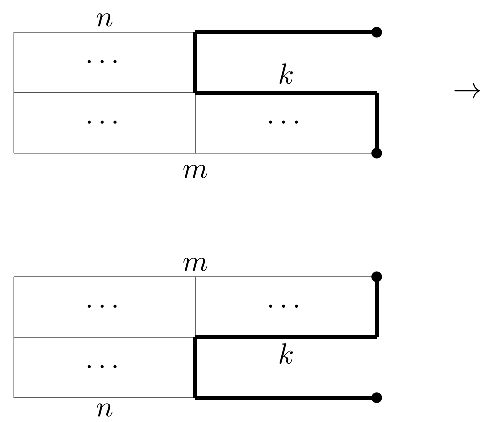


Figure 3.5: Change in the code from commuting Y_nY_m.

where ν is the composition of m with code given by

$$\ldots\beta_{k+j+1}U\beta_{k+j}\ldots\beta_2\beta_1L^k\gamma_1\gamma_2\ldots$$

Proof. By the minimality of j, we have that $\beta_{k+j}=R$. By applying Proposition 3.6.1 $k+j$ times we have that

$$\begin{aligned}\alpha &= \ldots\beta_3\beta_2\beta_1L^kU\gamma_1\gamma_2\gamma_3\ldots\\ &\dot\sim \ldots\beta_3\beta_2\beta_1\left(L^kUR^k\right)L^k\gamma_1\gamma_2\gamma_3\ldots\\ &\dot\sim (-1)^j\ldots\beta_{k+j+1}\left(L^0UR^0\right)\beta_{k+j}\ldots\beta_2\beta_1L^k\gamma_1\gamma_2\gamma_3\ldots\\ &= (-1)^j\ldots\beta_{k+j+1}U\beta_{k+j}\ldots\beta_2\beta_1L^k\gamma_1\gamma_2\gamma_3\ldots\end{aligned}$$

since j is the number of U's in $\beta_{k+j}\ldots\beta_2\beta_1$. If $\beta_{k+j+1}=U$, then the above code contains the subword $\beta_{k+j+1}U=UU\mathrel{\dot\sim}0$, so $\alpha\mathrel{\dot\sim}0$, so $Y_{-\mu}=0$. If $\beta_{k+j+1}=R$, then we have that $Y_{-\mu}=(-1)^jY_{-\nu}$ for the partition ν which satisfies the conditions of the theorem. In particular, ν is also a partition of m by Lemma 3.6.2. □

Corollary 3.6.4. *Let $\lambda=(\lambda_1,\lambda_2,\ldots,\lambda_l)$ be a strict partition and n be any integer with $n<\lambda_1$. Let $k=\lambda_1-n$, and let ζ be the letter immediately left of the k^{th} R left of the rightmost U in the code of λ.*

- *If $\zeta=U$, then $Y_{-n}Y_{-\lambda}=0$.*
- *If $\zeta=R$, then $Y_{-n}Y_{-\lambda}=(-1)^{j+1}Y_{-\nu}$, where j is the number of U's between the rightmost*

U and ζ, and ν is the strict partition whose code is the code of λ with U inserted after ζ.

This corollary is the analog of Corollary 3.4.3, and similarly follows from Theorem 3.6.3 since the code α of λ can be written in the form $\alpha = \beta\gamma$ in the notation of the theorem.

Corollary 3.6.5. *Let μ be any composition of m with code α. Then either $Y_{-\mu} = 0$ or $Y_{-\mu} = \pm Y_{-\lambda}$ for some strict partition λ of m with the same length as μ.*

The proof of this corollary is the same as the proof presented for Corollary 3.4.4, that is by induction on the number of U's right of the leftmost L in α. In this case though we can not use induction on the number of L's in α since the number of L's in the code do not decrease when we apply Theorem 3.6.3, unlike Theorem 3.4.2.

Similarly to Corollary 3.4.4, this corollary provides a simple way to compute $Y_{-\mu} \cdot 1$ in terms of Schur Q-functions for any composition μ, since $Y_{-\lambda} \cdot 1 = Q_\lambda$ by equation (3.11) for any strict partition λ.

In fact, given any composition μ, the number of times we have to apply Theorem 3.6.3 to get $Y_{-\mu} = \pm Y_{-\lambda}$ for some strict partition λ is less than or equal to the number of U's right of the leftmost L in the code of μ, which is the largest i such that $\mu_i < \mu_{i+1}$, i.e. the position of the last exceedance in μ.

The above corollary follows from the previous theorem, but since in this case we know that the Y_{-n} anticommute, we actually have the following stronger statement.

Proposition 3.6.6. *Let μ be any composition of m with length l. Then either $Y_{-\mu} = 0$ or $Y_{-\mu} = \operatorname{sgn}(\sigma) Y_{-\sigma(\mu)}$ for any permutation $\sigma \in S_l$.*

Proof. The proof of this statement is immediate from (3.12). In particular, $Y_{-\mu} = 0$ exactly when μ has a repeated index. The second case, $Y_{-\mu} = \operatorname{sgn}(\sigma) Y_{-\sigma(\mu)}$ follows from the fact that any permutation σ can be written as a sequence of adjacent transpositions $\sigma = \sigma_1\sigma_2\cdots\sigma_k$, and $\operatorname{sgn}(\sigma) = (-1)^k$. Then $Y_{-\mu} = -Y_{-\sigma_k(\mu)} = +Y_{-\sigma_{k-1}\sigma_k(\mu)} = \cdots = (-1)^k Y_{-\sigma_1\sigma_2\cdots\sigma_k(\mu)} = \operatorname{sgn}(\sigma) Y_{-\sigma(\mu)}$. □

This proposition follows immediately from known results. We include it only to show that Corollary 3.6.5 gives an only slightly less general version of this result using only codes.

Definition 3.6.7. For any strict partition λ, define $\lambda^{[i]}$ to be the strict partition obtained from the code of λ by inserting a U between the i^{th} pair of consecutive R's from the left.

In particular, this means that $\lambda^{[i]}$ is the strict partition with the i^{th} smallest positive integer not already in λ inserted into λ.

Theorem 3.6.8. *For any strict partition $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l)$,*

$$Y(t)Q_\lambda = \sum_{j=0}^{l} \sum_{n=\lambda_{j+1}+1}^{\lambda_j - 1} (-1)^j \, t^n \, Q_{(\lambda_1,\lambda_2,\ldots,\lambda_j,n,\lambda_{j+1}\ldots,\lambda_l)} \tag{3.15}$$

$$Y(t)Q_\lambda = \sum_{i\geq 0} (-1)^{l+|\lambda|-|\lambda^{[i]}|+i} \, t^{|\lambda^{[i]}|-|\lambda|} \, Q_{\lambda^{[i]}} \tag{3.16}$$

where we take the convention $\lambda_0 = \infty$ and $\lambda_{l+1} = -1$.

Proof. From the definition of $Y(t) = \sum_{n\in\mathbb{Z}} Y_n t^{-n}$, and equation (3.11), we have that $Y(t)Q_\lambda = \sum_{n\in\mathbb{Z}} Y_n t^{-n}(Y_{-\lambda_1} Y_{-\lambda_2} \cdots Y_{-\lambda_l} \cdot 1)$. So it is sufficient to consider the coefficient of t^{-n}, $Y_{-n}Y_{-\lambda_1}Y_{-\lambda_2}\cdots Y_{-\lambda_l}$.

If $n > \lambda_1$, then $Y_{-n}Y_{-\lambda} = Y_{-(n,\lambda)}$ is already in decreasing order. In fact, since $(n, \lambda) = \lambda^{(n-l)}$, we can write $Y_{-n}Y_{-\lambda} \cdot 1 = Y_{-(n,\lambda)} \cdot 1 = Y_{\lambda^{(n-l)}} \cdot 1 = Q_{\lambda^{(n-l)}}$.

If $n \leq \lambda_1$, then we write the code of λ in the form

$$\alpha = \ldots \beta_3\beta_2\beta_1 RRR \ldots$$

where $\beta_1 = U$. Let j be the smallest integer such that $\beta_{k+j} \ldots \beta_2\beta_1$ has k R's, where $k = \lambda_1 - n$.

From Corollary 3.6.4, if $\beta_{k+j+1} = U$, then $\alpha \dot\sim 0$, so $Y_{-n}Y_{-\lambda} = 0$. Since there are j U's in $\beta_{k+j} \ldots \beta_2\beta_1$, β_{k+j+1} is the $(j+1)^{\text{th}}$ U from the right in α. This naturally divides the $r_1(\lambda)$ R's left of β_1, the rightmost U in α, into $r_{j+1}(\lambda)$ R's left of $\beta_{k+j+1} = U$ and k R's right of β_{k+j+1}. Then we have that $n = \lambda_1 - k = r_1(\lambda) - k = (r_{j+1}(\lambda) + k) - k = r_{j+1}(\lambda) = \lambda_{j+1}$. Thus $Y_{-n}Y_{-\lambda} = 0$ exactly when $n = \lambda_i$ for some i, that is, when n is already a component of λ, as we expect from (3.11) and (3.12).

If $n \neq \lambda_i$ for any i, then from Corollary 3.6.4, $Y_{-n}Y_{-\lambda} = (-1)^j Y_{-\nu}$, where ν is the strict partition with code

$$\alpha' = \ldots \beta_{k+j+1} U \beta_{k+j} \ldots \beta_2\beta_1 RRR \ldots$$

Comparing this to the code α of λ, we have that $r_{j+1}(\nu) = n = \lambda_1 - k$, $r_i(\lambda) = r_i(\lambda)$ for all $i \leq j$, and $r_{i+1}(\nu) = r_i(\lambda)$ for all $i > j$. Thus $\nu = (\lambda_1, \lambda_2, \ldots, \lambda_j, n, \lambda_{j+1}, \ldots, \lambda_l)$, and thus (3.15) holds.

There are n R's left of the inserted U in α'. Each U corresponding to a component of ν will be immediately after an R (RU), since ν is a strict partition. Thus $l - j$ of the R's left of the inserted U will be immediately before a U, so $n - (l - j) = n - l + j$ of the R's left of the inserted U will be immediately before an R. This includes β_{k+j+1}, the R immediately before the inserted U, so this U is inserted into the $(n - l + j)^{\text{th}}$ position from the left. So $\nu = \lambda_1, \lambda_2, \ldots, \lambda_j, n, \lambda_{j+1}, \ldots, \lambda_l) = \lambda^{[n-l+j]}$.

Let $i = n - l + j$. Then $\lambda^{[i]} = \nu$, $|\lambda^{[i]}| - |\lambda| = n$, and $l + |\lambda| - |\lambda^{[i]}| + i = l - n + (n - l + j) = j$.

Finally, notice that as n runs over all summands which contribute to (3.15), i will run over all nonnegative integers ($i = 0$ corresponds to the case $n = 0$), so (3.16) holds. $\square$

In light of the results in this section, particularly Proposition 3.6.6 and Theorem 3.6.8, one might ask if there is an intuitive way to define a function Q_μ indexed by compositions such that Q_λ is the Schur Q-function when λ is a partition which satisfies analogous statements to Theorem 3.5.4 and Theorem 3.5.5. That is, can we generalize Schur Q-functions as we generalized Schur functions in Section 3.5?

Unfortunately the definition of the Schur Q-functions does not lend itself to generalization in this manner as the definition of the Schur functions we used in Section 3.5 did. However, even if we could generalize the definition of the Schur Q-functions in an intuitive way, the analogous result to Theorem 3.5.4 would tell us that $Q_\mu = \operatorname{sgn}(\sigma)Q_{\sigma(\mu)}$ for any permutation $\sigma \in S_l$ by Proposition 3.6.6. This implies that $Q_\mu = Q_\lambda$ exactly when μ is a rearrangement of λ. In other words, the result we would get from generalizing Schur Q-functions (unlike when we generalized Schur functions) would be trivial.

If we choose as our definition $Q_\mu = Y_{-\mu} \cdot 1$ for any composition μ, then we recover the two results mentioned in the previous paragraph immediately. Namely that $Q_\mu = \operatorname{sgn}(\sigma)Q_{\sigma(\mu)}$ for any permutation $\sigma \in S_l$ (by Proposition 3.6.6) and $Q_\mu = Q_\lambda$ exactly when μ is a rearrangement of λ. The drawback here is that we *define* Q_μ to satisfy the same defining relation as the Schur Q-functions, whereas in Section 3.5 we were able to generalize Schur functions and *prove* the relationship between the new function s_μ and $B_\mu \cdot 1$ in order to better understand the latter. So this definition, while valid, is not terribly useful or illustrative.

3.7 Shifted Codes

In this section we will relate codes to *shifted codes* of strict partitions and use these to study the analog of the Bernstein operators for Schur Q-functions.

Definition 3.7.1. Define the *shifted code* of a strict partition λ to be the infinite sequence of letters R and U obtained by tracing along the bottom-right edge of the shifted Young diagram of shape λ in the fourth quadrant together with the positive x-axis, starting at the bottom-right corner of the leftmost box on the bottom row of the diagram.

Example 3.7.2. For the strict partition $\lambda = (4, 2, 1)$, the path described above is shown in bold in Figure 3.6. Then the shifted code of λ is given by $\alpha = UURURRR\ldots$

Using the tools we developed to study codes of compositions, we will now show how the shifted code of a strict partition can be obtained directly from the code of that partition.

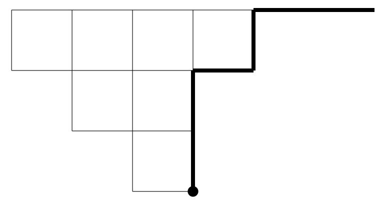


Figure 3.6: Shifted Young diagram and shifted code of $\lambda = (4, 2, 1)$.

Definition 3.7.3. Given a strict partition λ with code α, replace each U in α with UL and use the identity $LR \sim \phi$ to cancel wherever possible. Call the resulting sequence of letters R, L, and U the *preshifted code* of λ.

If the infinite prefix "$\ldots ULULU$" is removed from the preshifted code of a strict partition, then you obtain exactly the shifted code of that partition. The reason for this is that replacing each U with UL makes the diagram left aligned along the line $y = -x$ rather than the negative y-axis.

Example 3.7.4. For the empty partition $\lambda = \phi$, the code of λ is $\ldots UUURRR\ldots$ from the diagram in Figure 3.7. The preshifted code is $\ldots ULULURRR\ldots$ from the diagram in Figure 3.8. And the shifted code is $RRR\ldots$ from the diagram in Figure 3.9.

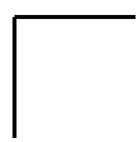


Figure 3.7: Young diagram and code of $\lambda = \phi$.

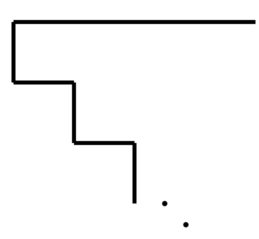


Figure 3.8: Shifted Young diagram and preshifted code of $\lambda = \phi$.

Figure 3.9: Shifted Young diagram and shifted code of $\lambda = \phi$.

Shifted codes can provide an alternative (but equivalent) definition of $\lambda^{[i]}$ to the one given in Definition 3.6.7.

Definition 3.7.5. For any strict partition λ, define $\lambda^{[i]}$ to be the strict partition obtained from the shifted code of λ by replacing the i^{th} R from the left in the shifted code of λ by U.

Notice the similarity to Definition 3.4.8, which defined $\lambda^{(i)}$ the exact same way using codes of partitions rather than shifted codes of strict partitions.

Definition 3.7.6. Given a composition μ with code α, replace each U in α with UL. Call the resulting sequence of letters R, L, and U the *preshifted code* of μ. Remove the prefix "$\ldots ULULU$" to obtain the *shifted code* of μ.

Example 3.7.7. For the composition $\mu = (2, 3, 1)$, the shifted code is obtained from the path shown in bold in Figure 3.10. Then the shifted code of μ is given by $\alpha = URULLURRR\ldots$

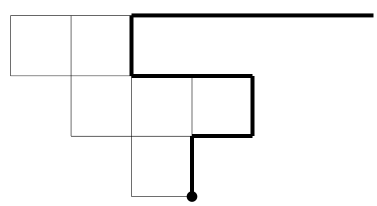


Figure 3.10: Shifted Young diagram and shifted code of $\mu = (2, 3, 1)$.

and the preshifted code of μ is $\ldots ULULUURULLURRR\ldots$

We can now use the relationship among the Y_n's to define yet another equivalence relation, this one on the set of signed shifted codes of compositions. We define $\alpha \ddot{\sim} \pm\beta$ if and only if $Y_{-\mu} = \pm Y_{-\nu}$, and $\alpha \ddot{\sim} 0$ if and only if $Y_{-\mu} = 0$, where α is the shifted code of the composition μ and β is the shifted code of the composition ν. As before, this will be an equivalence relation since there is a one to one correspondence between a composition μ and its shifted code α.

Proposition 3.7.8. *For any positive integer k, we have*

$$R\left(L^{k+1}UR^{k}\right) \;\; \ddot{\sim} \;\; \left(L^{k}UR^{k-1}\right)R \tag{3.17}$$

$$U\left(L^{k+1}UR^{k}\right) \;\; \ddot{\sim} \;\; -\left(L^{k}UR^{k-1}\right)U. \tag{3.18}$$

Again the proof is similar to Proposition 3.3.1. Relation 3.18 follows from the commutation identity (3.12) and corresponds to the shifted code given by the altered path in Figure 3.11, where $k = m - n$ as before.

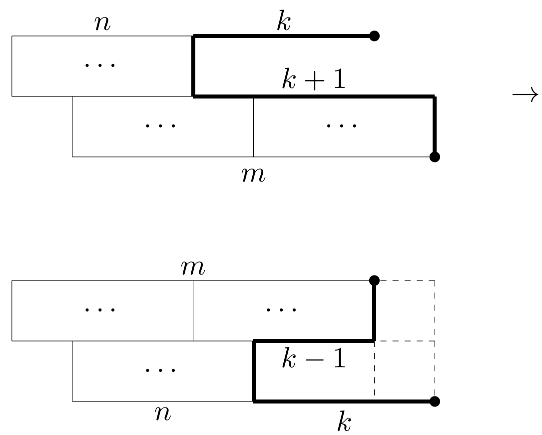


Figure 3.11: Change in the shifted code from commuting Y_nY_m.

Notice that the relation on shifted codes of compositions in Proposition 3.7.8 is identical to the relation on codes of compositions in Proposition 3.3.1. The two propositions give the same relation; the only difference being that the index k in Proposition 3.3.1 has been replaced by $k+1$ in Proposition 3.7.8 to preserve the identity $k = m - n$. This relationship will be studied in more depth in Section 3.9.

With this identification we can prove the shifted code analog of each result in Section 3.6 in exactly the same way as each corresponding result in Section 3.4. We include the statements of these results for completeness.

Lemma 3.7.9. *Suppose that the codes α and β of two compositions μ and ν differ only by one of the relations (3.17) or (3.18). Then μ and ν have the same number of components, l, and the same sum, $\mu_1 + \mu_2 + \cdots + \mu_l = \nu_1 + \nu_2 + \cdots + \nu_l$.*

The proof is identical to Lemma 3.4.1.

Theorem 3.7.10. *Let μ be any composition of m with shifted code α. Suppose that α can be written in the form*

$$\alpha = \beta_t \dots \beta_3\beta_2\beta_1 L^k U \gamma_1\gamma_2\gamma_3 \dots$$

where $\beta = \beta_t \dots \beta_3\beta_2\beta_1$ consists only of R's and U's and $\beta_1 = U$.

- *If $\beta_k = U$, then $Y_{-\mu} = 0$.*
- *If $\beta_k = R$, then $Y_{-\mu} = (-1)^j Y_{-\nu}$, where j is the number of U's in $\beta_{k-1} \dots \beta_2\beta_1$ and ν is the composition of m with shifted code given by*

$$\beta_t \dots \beta_{k+1} U \beta_{k-1} \dots \beta_3\beta_2\beta_1 L^{k-1}\gamma_1\gamma_2\gamma_3 \dots$$

The proof is identical to Theorem 3.4.2.

Corollary 3.7.11. *Let $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_l)$ be a strict partition and n be any integer with $n < \lambda_1$. Let $k = \lambda_1 - n$, and let ζ be the letter $k - 1$ positions left of the rightmost U in the shifted code of λ.*

- *If $\zeta = U$, then $Y_{-n}Y_{-\lambda} = 0$.*
- *If $\zeta = R$, then $Y_{-n}Y_{-\lambda} = (-1)^{j+1} Y_{-\nu}$, where j is the number of U's between the rightmost U and ζ, and ν is the partition obtained by replacing ζ by U.*

The proof is identical to Corollary 3.4.3. It follows directly from Theorem 3.7.10 for the special case where $\mu = (n, \lambda_1, \lambda_2, \dots, \lambda_l)$ and $\gamma = RRR\dots$ If $n = \lambda_j$ for some j, then we would have $\zeta = U$ and $Y_{-\mu} = 0$. So the case $\zeta = R$ corresponds to the *strict* partition $\nu = (\lambda_1, \lambda_2, \dots, \lambda_j, n, \lambda_{j+1}, \dots, \lambda_l)$.

Corollary 3.7.12. *Let μ be any composition of m with shifted code α. Then either $Y_{-\mu} = 0$ or $Y_{-\mu} = \pm Y_{-\lambda}$ for some strict partition λ of m with the same length as μ.*

The proof is identical to Corollary 3.4.4. As in that case, we can prove the result using induction on either the number of L's in α or on the number of U's to the right of the leftmost L in α. Notice that Corollary 3.6.5 gives the exact same result as this corollary, however in that case only the latter method of proof is intuitive. This is one example of the strength of using shifted codes to study this problem.

Theorem 3.7.13. *For any strict partition* $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l)$*,*

$$Y(t)Q_\lambda = \sum_{j=0}^{l} \sum_{n=\lambda_{j+1}+1}^{\lambda_j - 1} (-1)^j \, t^n \, Q_{(\lambda_1, \lambda_2, \ldots, \lambda_j, n, \lambda_{j+1} \ldots, \lambda_l)} \tag{3.19}$$

$$Y(t)Q_\lambda = \sum_{i \geq 0} (-1)^{l+|\lambda|-|\lambda^{[i]}|+i} \, t^{|\lambda^{[i]}|-|\lambda|} \, Q_{\lambda^{[i]}} \tag{3.20}$$

where we take the convention $\lambda_0 = \infty$ *and* $\lambda_{l+1} = -1$*.*

The proof is almost identical to the proof of Theorem 3.4.9. In the setting of shifted codes, we replace codes with shifted codes and $\lambda^{(i)}$ with $\lambda^{[i]}$. This means that the number, j, of U's in $\beta_k \ldots \beta_2 \beta_1$ will be $j = l + |\lambda| - |\lambda^{[i]}| + i$ rather than $j = |\lambda| - |\lambda^{(i)}| + i - 1$, since $\lambda^{[i]} = (\lambda_1, \lambda_2, \ldots, \lambda_j, i + (l - j), \lambda_{j+1}, \ldots, \lambda_l)$ and $\lambda^{(i)} = (\lambda_1 - 1, \lambda_2 - 1, \ldots, \lambda_j - 1, i - 1, \lambda_{j+1}, \ldots, \lambda_l)$. Finally, we must take the convention $\lambda_{l+1} = -1$ rather than zero in this case, since Y_0 represents the nontrivial insertion of zero into the strict partition, whereas B_0 acts as trivially on 1.

Notice that this gives the exact same result as Theorem 3.6.8, but the proof will follow that of the Schur function case.

Since the only difference between preshifted codes and shifted codes is the prefix $\ldots ULULU$, we can also state and prove each of these results in terms of preshifted codes, where we concern ourselves only with L's not in the prefix $\ldots ULULU$. The statements and proofs of these results will be otherwise identical to those using shifted codes.

3.8 Reverse Shifted Codes

In Section 3.7 we saw that the relations satisfied by shifted codes in the Schur Q-function case were identical to the relations satisfied by codes in the Schur function case. In other words, Propositions 3.3.1 and 3.7.8 have the same statement for their corresponding set of codes. Furthermore, we found that the shifted codes can be obtained by a translation of the codes in the Schur Q-function case.

In light of this correspondence between codes for Schur functions and $s$hifted codes for Schur Q-function, it is natural to ask what is the analog for Schur functions of unshifted codes for Schur Q-functions. Not surprisingly, the answer involves an inverse operation to that which gave us shifted codes.

We return now to Schur functions, and will define the following analog of shifted codes in this setting.

Definition 3.8.1. Given a partition λ with code α, replace each U in α with UR and use the identity $RL \sim \phi$ to cancel wherever possible. Call the resulting sequence of letters R, L, and

U the *pre-reverse-shifted code* of λ. Remove the infinite prefix "$\ldots URURU$" to obtain the *reverse-shifted code* of λ.

It is now convenient to introduce the anagrams S, PS, U, PRS, and RS to denote *shifted*, *preshifted*, *unshifted*, *pre-reverse-shifted*, and *reverse-shifted* codes, respectively, which we will use only when discussing several of these objects at the same time.

Example 3.8.2. For the partition $\lambda = (0, 0, 0, 0)$, the various kinds of codes are shown on the same graph in Figure 3.12, with the infinite portion of the preshifted and pre-reverse-shifted codes represented by dotted lines.

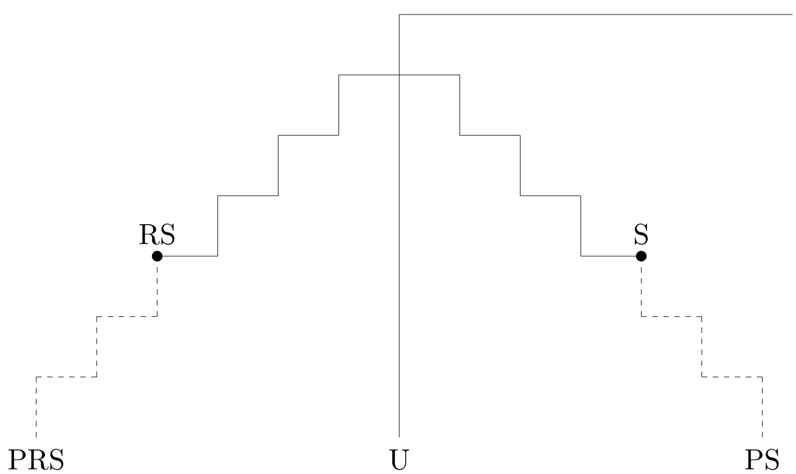


Figure 3.12: All five code variants for $\lambda = (0, 0, 0, 0)$.

The main insight this example provides is that since the Young diagram of $(0, 0, 0, 0)$ is empty, the codes of this partition represent the leftmost edges of each kind of diagram. As we have seen before, all unshifted codes are left aligned, so they have a vertical left edge. Preshifted codes are aligned along the diagonal line $y = -x$ and continue infinitely, whereas shifted codes stop at the bottom of the partition. Pre-reverse-shifted and reverse-shifted codes are aligned the same way along $y = x$ to the left.

As with shifted codes, we define the reverse-shifted (and pre-reverse-shifted) code of a composition μ using the same definition as for a partition λ.

Example 3.8.3. For the composition $\mu = (3, 1, 2)$, the reverse-shifted code is obtained from the path shown below in bold. So the reverse-shifted code of μ is $\alpha = RRUURRRRURRR\ldots$ and the pre-reverse-shifted code of μ is $\ldots URURURRUURRRRURRR\ldots$

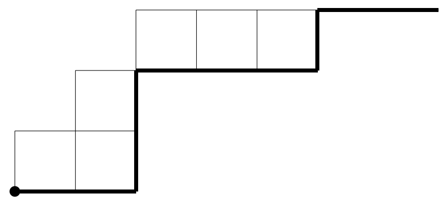


Figure 3.13: Reverse-shifted Young diagram and reverse-shifted code of $\mu = (3, 1, 2)$.

Note that in this setting we no longer necessarily have L's to denote an exceedance. If $\mu_{i+1} = \mu_i + 1$, as in this example $\mu_3 = \mu_2 + 1$, then the code has only a UU to denote this position.

We now define another equivalence relation on the set of signed reverse-shifted codes of compositions by $\alpha \overset{\cdot\cdot}{\approx} \pm\beta$ if and only if $B_\mu = \pm B_\nu$, and $\alpha \overset{\cdot\cdot}{\approx} \pm 0$ if and only if $B_\mu = 0$, where α is the reverse-shifted code of the composition μ and β is the reverse-shifted code of the composition ν. This will be an equivalence relation since there is a one to one correspondence between a composition μ and its reverse-shifted code α.

Proposition 3.8.4. *For any positive integer k, we have*

$$R\left(L^k U R^k\right) \overset{\cdot\cdot}{\approx} \left(L^{k-1} U R^{k-1}\right) R \tag{3.21}$$

$$U\left(L^{k-1} U R^{k-1}\right) \overset{\cdot\cdot}{\approx} -\left(L^{k-1} U R^{k-1}\right) U. \tag{3.22}$$

Again the proof is similar to Propositions 3.3.1, 3.6.1, and 3.7.8. Relation (3.21) follows from canceling $RL \overset{\cdot\cdot}{\approx} \phi$ and distributing, and Relation (3.22) follows from the commutation relation on the Bernstein operators (3.4), $B_n B_m = -B_{m-1} B_{n+1}$. In particular Relation (3.22) corresponds to the reverse-shifted code given by the altered path below, where again $k = m - n$.

Notice that these two relations on reverse-shifted codes for Schur functions are exactly the same as the two relations we had on codes for Schur Q-functions. The only difference between Proposition 3.8.4 and Proposition 3.6.1 is an index shift replacing $k-1$ by k in the second relation to preserve the identity $k = m - n$. This is the exact same as the result we obtained comparing Proposition 3.3.1 on codes for Schur functions to Proposition 3.7.8 on shifted codes for Schur Q-functions! We save the explanation for this until the next section, and will first state the analog of the main results from Sections 3.4, 3.6, and 3.7 for reverse-shifted codes.

Lemma 3.8.5. *Suppose that the codes α and β of two compositions μ and ν differ only by one of the relations (3.21) or (3.22). Then μ and ν have the same number of components, l, and the same sum, $\mu_1 + \mu_2 + \cdots + \mu_l = \nu_1 + \nu_2 + \cdots + \nu_l$.*

The proof is identical to Lemma 3.4.1.

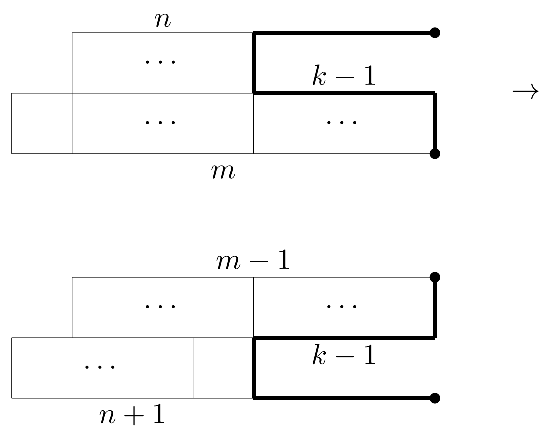


Figure 3.14: Change in the reverse-shifted code from commuting B_nB_m.

Theorem 3.8.6. *Let μ be any composition of m with code α. Suppose that α can be written in the form*

$$\alpha = \beta_t \dots \beta_3\beta_2\beta_1 L^k U \gamma_1\gamma_2\gamma_3 \dots$$

where $\beta = \beta_t \dots \beta_3\beta_2\beta_1$ consists only of R's and U's and $\beta_1 = U$. Let j be the smallest integer such that $\beta_{k+j} \dots \beta_2\beta_1$ has k R's. Then $B_\mu = 0$ if $\beta_{k+j+1} = U$ and $B_\mu = (-1)^j B_\nu$ if $\beta_{k+j+1} = R$, where ν is the composition of m with code given by

$$\beta_t \dots \beta_{k+j+1} U \beta_{k+j} \dots \beta_2\beta_1 L^k \gamma_1\gamma_2 \dots$$

The proof is identical to Theorem 3.6.3.

Corollary 3.8.7. *Let $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_l)$ be a partition and n be any integer with $n < \lambda_1$. Let $k = \lambda_1 - n$, and let ζ be the letter immediately left of the k^{th} R left of the rightmost U in the code of λ.*

- *If $\zeta = U$, then $B_nB_\lambda = 0$.*
- *If $\zeta = R$, then $B_nB_\lambda = (-1)^{j+1}B_\nu$, where j is the number of U's between the rightmost U and ζ, and ν is the strict partition whose code is the code of λ with U inserted after ζ.*

This gives the same result as Corollary 3.4.3 using reverse-shifted codes. The proof is identical to Corollary 3.6.4 and similarly follows from Theorem 3.8.6, corresponding to the case when $t = n$ and $\beta = RR \dots RR$ in the notation of the theorem.

Corollary 3.8.8. *Let μ be any composition of m with code α. Then either $B_\mu = 0$ or $B_\mu = \pm B_\lambda$ for some strict partition λ of m with the same length as μ.*

This corollary gives the same result as Corollary 3.4.4 and follows from Theorem 3.8.6 with the same proof given for that corollary.

Definition 3.8.9. For any partition λ, define $\lambda^{(i)}$ to be the partition obtained from the reverse-shifted code of λ by inserting a U between the i^{th} pair of consecutive R's from the left.

This definition is actually a new formulation equivalent to the existing definition in terms of codes of partitions, which we gave in Definition 3.4.8. This is because both definitions for $\lambda^{(i)}$ describe ways to insert a new component $i-1$ (and decrease the size of all larger components of λ by 1) in the i^{th} possible position from the bottom of λ. In terms of reverse-shifted codes, there must be an R both before and after the inserted component or else the result will not be a partition: two consecutive U's would give $\lambda_j + 1 = \lambda_{j+1}$, so λ is not in decreasing order.

Theorem 3.8.10. *For any partition $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l)$,*

$$B(t)s_\lambda = \sum_{j=0}^{l} \sum_{n=\lambda_{j+1}}^{\lambda_j - 1} (-1)^j t^n B_{(\lambda_1-1,\lambda_2-1,\ldots,\lambda_j-1,n,\lambda_{j+1},\ldots,\lambda_l)} \tag{3.23}$$

$$B(t)s_\lambda = \sum_{i\geq 1} (-1)^{|\lambda|-|\lambda^{(i)}|+i-1} t^{|\lambda^{(i)}|-|\lambda|} s_{\lambda^{(i)}}. \tag{3.24}$$

The proof is very similar to Theorem 3.6.8. The only differences are the following. The components before n in $\lambda^{(i)}$ decrease in size by one in this setting because inserting a U makes the code above it one unit closer to the left edge (because of the slant). This shift then changes the exponent of -1, which becomes $j = |\lambda| - |\lambda^{(i)}| + i - 1$, though it is still the position of the inserted entry. And finally, the limits of the second summation in (3.23) start at λ_{j+1} not $\lambda_{j+1}+1$ to correspond to consecutive U's. In Theorem 3.6.8 consecutive U's correspond to not being a strict partition so $Y_{-n}Y_{-\lambda} = 0$, whereas in this setting consecutive U's correspond to $B_n B_{n+1} = 0$, as described in the preceding paragraph.

3.9 The Schur – Schur Q Correspondence

The reason for the similarities between Propositions 3.8.4 and 3.6.1 and between Propositions 3.3.1 and 3.7.8 is the following. Consider a partition $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l)$ and a strict partition $\eta = (\eta_1, \eta_2, \ldots, \eta_l)$ with $\eta_i = \lambda_i - i + 1$ or $\eta = (\lambda_1, \lambda_2 - 1, \lambda_3 - 2, \ldots, \lambda_l - (l-1)) = \lambda + \delta - (l-1)\mathbb{1}$, where $\delta = (l-1, l-2, \ldots, 2, 1, 0)$ as in Section 3.5 and $\mathbb{1} = (1, 1, \ldots, 1)$. Then the reverse-shifted code of λ will be the same as the code of η and the code of λ will be the same as the shifted code of η, except for their corresponding prefixes, which will be the same for any partition of length l.

Example 3.9.1. Consider the partition $\lambda = (3,3)$. Then $\eta = (\lambda_1, \lambda_2 - 1) = (3,2)$. The correspondences described above are shown as dotted lines in Figure 3.15 with corresponding codes shown in bold. From left to right, the reverse-shifted code and code of λ are shown in the first row and the code and shifted code of η are shown in the second row. We also show the maps S and RS corresponding to shifting and reverse-shifting the codes.

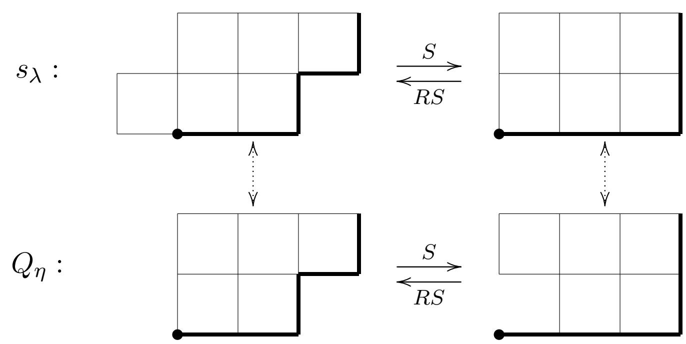


Figure 3.15: Correspondences between codes.

Note that in this example we chose the starting points of the shifted and reverse-shifted codes on the opposite side of the bottom-left corner of the tableau from where we started previously in this paper in order to match the two unshifted codes shown. This only means that the prefix which goes before this portion of the code would be RR for the reverse-shifted code of s_λ and LL for the shifted code of Q_η (for any λ and η), as opposed to something more intuitive. In most other situations our previous choice is more intuitive.

We can also summarize this result with Figure 3.16 relating the equivalence relations defined in this paper: $\sim$ on codes for Schur functions, $\dot{\sim}$ on codes for Schur Q-functions, $\ddot{\sim}$ on shifted codes for Schur Q-functions, and $\dddot{\sim}$ on reverse-shifted codes for Schur functions. Again the dotted lines correspond to changing the prefix, and S and RS correspond to shifting and reverse-shifting the associated codes.

It is important to note that the relation $\dot{\sim}$ corresponds to the anticommuting operators Y_{-n}, so the new relation $\dddot{\sim}$ on reverse-shifted codes will also be exactly anticommuting after truncating the slanted component on the left. Notice that the bijection between s_λ and Q_η described above is given by $\eta = \lambda + \delta - (l-1)\mathbb{1}$ or $\lambda = \eta - \delta + (l-1)\mathbb{1}$, and since the components of Q_η anticommute, the components of $s_{\eta-\delta+(l-1)\mathbb{1}}$ should as well (with respect to

$$
\begin{tikzcd}
s_\lambda : & \dddot{\sim} \arrow[r, shift left, "S"] \arrow[d, dotted, leftrightarrow] & \sim \arrow[l, shift left, "RS"] \arrow[d, dotted, leftrightarrow] \\
Q_\eta : & \dot{\sim} \arrow[r, shift left, "S"] & \ddot{\sim} \arrow[l, shift left, "RS"]
\end{tikzcd}
$$

Figure 3.16: Correspondences between relations on codes.

permuting components of η).

Then this strange combinatorial object, reverse-shifted codes of partitions (specifically when the left edge is truncated), indexes an anticommuting version of Bernstein's operator. Such an anticommuting operator was defined by Jing in [6] by

$$X(t) = B(t^{-1})e^p t^{\partial_p} = \sum_{n \in \mathbb{Z}} X_n t^{-n},$$

acting on the space $\Lambda \otimes \mathbb{C}[\mathbb{Z}]$, where e^p and t^{∂_p} act on $\mathbb{C}[\mathbb{Z}]$ by $e^p \cdot e^{np} = e^{(n+1)p}$ and $t^{\partial_p} \cdot e^{np} = t^n e^{np}$. Here we follow the notation of Chapter 2 and omit the index shift of $\frac{1}{2}$ used in [6] and [4].

Jing showed in [6] and [4] that the operator $X(t)$ also generates the Schur functions according to the equation

$$X_{-\mu_1} \cdots X_{-\mu_l} \cdot e^{-lp} = s_{\mu - \delta + l\mathbb{1}}, \tag{3.25}$$

and that these operators anticommute,

$$X_n X_m = -X_m X_n.$$

This commutation of X_n in equation (3.25) is exactly the identity satisfied by $s_{\eta - \delta + (l-1)\mathbb{1}}$ after an index shift of $\eta = \mu - \mathbb{1}$.

Then this operator induces the same relation as that given by $\dddot{\sim}$ on reverse-shifted codes of compositions. So we could have instead defined $\dddot{\sim}$ by $\alpha \dddot{\sim} \pm\beta$ if and only if $X_{-\mu} = \pm X_{-\nu}$, and $\alpha \dddot{\sim} 0$ if and only if $X_{-\mu} = 0$, where α is the reverse-shifted code of μ and β is the reverse-shifted code of ν.

Thus just as the indices of $B(t)$ can be naturally described combinatorially by codes of partitions and the indices of $Y(t)$ can be naturally described by codes of strict partitions, so too can the indices of $X(t)$ be naturally described by reverse-shifted codes of partitions.

With this identification, looking back at Chapter 2, we can see that Theorem 3.8.10 is the combinatorial version of the second proof of Theorem 2.3.1, just as Theorem 3.4.9 is the combinatorial version of the first proof, Theorem 3.6.8 is the combinatorial version of Theorem

2.4.1, and Theorem 3.7.13 is the combinatorial version of Theorem 2.4.8.

Chapter 4

Hall-Littlewood Polynomials

4.1 Hall-Littlewood Analog

In Chapters 2 and 3 we showed the action of the Bernstein operators $B(t) = \sum_{n\in\mathbb{Z}} B_n t^n$ on the Schur functions $s_\lambda = B_{\lambda_1}B_{\lambda_2}\cdots B_{\lambda_l}\cdot 1$ is given by

$$B(t)s_\lambda = \sum_{i\geq 1}(-1)^{|\lambda|-|\lambda^{(i)}|+i-1}t^{|\lambda^{(i)}|-|\lambda|}s_{\lambda^{(i)}},$$

for any partition $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l)$. Similarly we showed that the action of the vertex operator $Y(t) = \sum_{n\in\mathbb{Z}} Y_n t^{-n}$ on the Schur Q-function $Q_\lambda = Y_{-\lambda_1}Y_{-\lambda_2}\cdots Y_{-\lambda_l}\cdot 1$ is given by

$$Y(t)Q_\lambda = \sum_{n\neq\lambda_j}(-1)^i t^n Q_{(\lambda_1,\lambda_2,\ldots,\lambda_i,n,\lambda_{i+1},\ldots,\lambda_l)},$$

for any strict partition $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l)$.

We now turn our attention to studying with more generality the action of the vertex operator $H(z)$ on the Hall-Littlewood polynomials $Q_\lambda(t)$ defined in [5], where we have that $Q_\lambda(t) = H_{-\lambda_1}H_{-\lambda_2}\cdots H_{-\lambda_l}\cdot 1$ for

$$H(z) = \sum_{n\in\mathbb{Z}} H_{-n}z^n = exp\left(\sum_{n\geq 1}\frac{1-t^n}{n}p_n z^n\right) exp\left(-\sum_{n\geq 1}\frac{\frac{\partial}{\partial p_n}}{1-t^n}z^n\right),$$

for any partition $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l)$.

It follows from this that $Q_\lambda(t)$ is the coefficient of $z^\lambda = z_1^{\lambda_1}z_2^{\lambda_2}\cdots z_l^{\lambda_l}$ in $H(z_1)H(z_2)\cdots H(z_l)\cdot 1$, which then implies that $H_{-n}Q_\lambda(t)$ is the coefficient of $z_0^n z^\lambda$ in $H(z_0)H(z_1)H(z_2)\cdots H(z_l)\cdot 1$. However, if n is less than λ_1, then the term $H_{-n}H_{-\lambda_1}\cdots H_{-\lambda_l}\cdot 1$ which appears is no longer a Hall-Littlewood polynomial, since the formula $Q_\lambda(t) = H_{-\lambda_1}\cdots H_{-\lambda_l}\cdot 1$ only holds for λ a partition (decreasing parts).

The expression $H_{-n}H_{-\lambda_1}\cdots H_{-\lambda_l}\cdot 1$ can be interpreted as a linear combination of Hall-Littlewood polynomials by repeated applications of the commutation relation $[H_m, H_n]_t = -[H_{n+1}, H_{m-1}]_t$, or

$$H_mH_n - tH_nH_m = -(H_{n+1}H_{m-1} - tH_{m-1}H_{n+1}),$$

$$H_mH_n = tH_nH_m - H_{n+1}H_{m-1} + tH_{m-1}H_{n+1}.$$

The problem with this approach is that if n is significantly smaller than λ_1 (if $\lambda_1 - n$ is large), then the repeated applications yield an enormous number of Hall-Littlewood polynomials as summands. We therefore seek some other way to explicitly determine $H_{-n}Q_\lambda(t)$.

We would like to interpret $H_{-n}Q_\lambda$ as a linear combination of $Q_\mu(t) = H_{\mu_1}\cdots H_{\mu_{l+1}}\cdot 1$ for some partitions μ. However, $Q_\mu(t)$ is the coefficient of a power of $z_0, z_1, z_2, \ldots, z_l$ in $H(z_{i_0})H(z_{i_1})H(z_{i_2})\cdots H(z_{i_l})\cdot 1$, where $i_0, i_1, \ldots, i_l$ is some rearrangement of $0, 1, \ldots, l$. Thus we have to find some way to interpret $H_{-n}Q_\lambda(t)$ as the coefficient of $z_0^n z^\lambda$ in the rearranged product $H(z_{i_0})H(z_{i_1})\cdots H(z_{i_l})\cdot 1$.

For the remainer of this chapter, we will focus on the case where $H(z_{i_0})H(z_{i_1})\cdots H(z_{i_l})$ is of the form $H(z_1)H(z_2)\cdots H(z_j)H(z_0)H(z_{j+1})\cdots H(z_l)$. This means that z_0, the variable corresponding to the entry n in the composition $(n, \lambda_1, \lambda_2, \ldots, \lambda_l)$, is the only one to change relative position. In the notation of vertex operators, this means that we are restricting ourselves to the region where $|z_1| > |z_2| > \cdots > |z_j| > |z_0| > |z_{j+1}| > \cdots > |z_l|$.

Theorem 4.1.1. *The action of the vertex operator $H(z) = \sum_{n\in\mathbb{Z}} H_{-n}z^n$ on the Hall-Littlewood polynomial $Q_\lambda(t)$ is given by*

$$H_{-n}Q_\lambda(t) = \sum_{n_1,\ldots,n_j=0}^{\infty} c_{n_1,\ldots,n_j}(t)Q_{(\lambda_1-n_1,\ldots,\lambda_j-n_j,n+n_1+\cdots+n_j,\lambda_{j+1},\ldots,\lambda_l)}(t),$$

where $c_{n_1,\ldots,n_j}(t) = t^{n_1+\cdots+n_j-j+2k}(t^2-1)^{j-k}$, for k equal to the number of zeros in the set $\{n_1, n_2, \ldots, n_j\}$.

Proof. We have from [5] that

$$H(z_0)H(z_1)\cdots H(z_l) = \left(\prod_{0\le i<k\le l}\frac{z_j - z_i}{z_j - tz_i}\right) : H(z_0)H(z_1)\cdots H(z_l) : 1,$$

where $:\quad:$ is the normal ordering product, which is symmetric, i.e. $:\ H(n)H(m)\ : = :\ H(m)H(n)\ :$ for any n and m. Let K_1 denote the coefficient above, $K_1 = \left(\prod_{0\le i<k\le l}\frac{z_j - z_i}{z_j - tz_i}\right)$.

Similarly,

$$H(z_1)\cdots H(z_j)H(z_0)H(z_{j+1})\cdots H(z_l) = K_2 : H(z_1)\cdots H(z_0)\cdots H(z_l) : 1,$$

where $K_2 = \left(\prod_{i=1}^{j} \frac{z_i - z_0}{z_i - tz_0}\right)\left(\prod_{i=j+1}^{l} \frac{z_0 - z_i}{z_0 - tz_i}\right)\left(\prod_{1\le i<k\le l} \frac{z_i - z_k}{z_i - tz_k}\right)$.

Then since the normal ordering product is symmetric, we have that $H(z_0)H(z_1)\cdots H(z_l) = \frac{K_1}{K_2}H(z_1)\cdots H(z_0)\cdots H(z_l)$. Now simplify $\frac{K_1}{K_2}$ as follows:

$$\frac{K_1}{K_2} = \frac{\left(\prod_{0\le i<k\le l} \frac{z_j - z_i}{z_j - tz_i}\right)}{\left(\prod_{i=1}^{j} \frac{z_i - z_0}{z_i - tz_0}\right)\left(\prod_{i=j+1}^{l} \frac{z_0 - z_i}{z_0 - tz_i}\right)\left(\prod_{1\le i<k\le l} \frac{z_i - z_k}{z_i - tz_k}\right)} =$$

$$= \frac{\left(\prod_{i=1}^{l} \frac{z_0 - z_i}{z_0 - tz_i}\right)}{\left(\prod_{i=1}^{i} \frac{z_i - z_0}{z_i - tz_0}\right)\left(\prod_{i=j+1}^{l} \frac{z_0 - z_i}{z_0 - tz_i}\right)} = \prod_{i=1}^{j} \frac{\left(\frac{z_0 - z_i}{z_0 - tz_i}\right)}{\left(\frac{z_i - z_0}{z_i - tz_0}\right)} = \prod_{i=1}^{j}\left(-\frac{z_i - tz_0}{z_0 - tz_i}\right).$$

To simplify this, we introduce the notation $\tilde{z}_i = \frac{z_i}{z_0}$. Then we have

$$\begin{aligned}
\left(-\frac{z_i - tz_0}{z_0 - tz_i}\right) &= \frac{tz_0 - z_i}{z_0 - tz_i} = \frac{t - \frac{z_i}{z_0}}{1 - t\frac{z_i}{z_0}} = \frac{t - \tilde{z}_i}{1 - t\tilde{z}_i} \\
&= (t - \tilde{z}_i)(1 + t\tilde{z}_i + t^2\tilde{z}_i^2 + \cdots) \\
&= t + (t^2 - 1)\tilde{z}_i + (t^3 - t)\tilde{z}_i^2 + (t^4 - t^2)\tilde{z}_i^3 + \cdots \\
&= t + \sum_{n=1}^{\infty}(t^{n+1} - t^{n-1})\tilde{z}_i^n \\
&= t + \sum_{n=1}^{\infty} t^{n-1}(t^2 - 1)\tilde{z}_i^n.
\end{aligned}$$

This means that the coefficient $\frac{K_1}{K_2}$ is given by

$$\begin{aligned}
\frac{K_1}{K_2} &= \prod_{i=1}^{j}\left(t + \sum_{n=1}^{\infty} t^{n-1}(t^2 - 1)\left(\frac{z_i}{z_0}\right)^n\right) \\
&= \sum_{n_1,n_2,\ldots,n_j=0}^{\infty} c_{n_1,n_2,\ldots,n_j}(t)\left(\frac{z_1}{z_0}\right)^{n_1}\left(\frac{z_2}{z_0}\right)^{n_2}\cdots\left(\frac{z_j}{z_0}\right)^{n_j},
\end{aligned}$$

for $c_{n_1,n_2,\ldots,n_j}(t) = t^{m-j+2k}(t^2-1)^{j-k} = \sum_{i=0}^{j-k}(-1)^{j-k+i}\binom{j-k}{i}t^{m-j+2k+2i}$, where $m = n_1+n_2+\cdots+n_j$, and k is the number of zeros in the set $\{n_1, n_2, \ldots, n_j\}$. The reason for this is that each

$n_i = 0$ gives a factor of t in $c_{n_1,\ldots,n_j}(t)$, for a total power of t being t^k from all $n_i = 0$. Similarly each $n_i \neq 0$ gives a factor of $t^{n_i-1}(t^2-1)$, for a total factor of $t^{n_{i_1}-1}(t^2-1)\cdots t^{n_{i_{j-k}}-1}(t^2-1) = t^{n_1+\cdots n_i-(j-k)}(t^2-1)^{j-k} = t^{m-j+k}(t^2-1)^{j-k}$, since $j-k$ is the number of nonzero n_i's.

Thus $H_{-n}Q_\lambda(t)$ is the coefficient of $z_0^n z^\lambda$ in

$$\begin{aligned} H(z_0)\cdots H(z_l)\cdot 1 &= \frac{K_1}{K_2} H(z_1)\cdots H(z_0)\cdots H(z_l)\cdot 1 \\ &= \sum_{n_1,\ldots,n_j=0}^{\infty} c_{n_1,\ldots,n_j}(t)\left(\frac{z_1}{z_0}\right)^{n_1}\cdots\left(\frac{z_j}{z_0}\right)^{n_j} H(z_1)\cdots H(z_0)\cdots H(z_l)\cdot 1. \end{aligned}$$

For simplicity of notation, we again use $m = n_1 + \cdots n_j$ and assume the convention that $n_i = 0$ if $i > j$. Then $H_{-n}Q_\lambda(t)$ is the sum over $n_1,\ldots,n_j \in \mathbb{Z}_{\geq 0}$ of $c_{n_1,\ldots n_j}(t)$ times the coefficient of

$$z_0^{n+n_1+n_2+\cdots+n_j} z_1^{\lambda_1-n_1} z_2^{\lambda_2-n_2}\cdots z_i^{\lambda_j-n_j} z_{j+1}^{\lambda_{j+1}}\cdots z_l^{\lambda_l} = z_0^{n+m}\prod_{i=0}^{l} z_i^{\lambda_i-n_i}$$

in $H(z_1)\cdots H(z_0)\cdots H(z_l)\cdot 1$. But this coefficient is itself $Q_{(\lambda_1-n_1,\ldots,\lambda_j-n_j,n+n_1+\cdots+n_j,\lambda_{j+1},\ldots,\lambda_l)}(t)$. In other words:

$$\begin{aligned} H_{-n}Q_\lambda(t) &= \mathrm{coeff}_{z_0^n z^\lambda}\left[H(z_0)\cdots H(z_l)\cdot 1\right] \\ &= \mathrm{coeff}_{z_0^n z^\lambda}\left[\sum_{n_1,\ldots,n_j=0}^{\infty} c_{n_1,\ldots,n_j}(t)\left(\frac{z_1}{z_0}\right)^{n_1}\cdots\left(\frac{z_j}{z_0}\right)^{n_j} H(z_1)\cdots H(z_0)\cdots H(z_l)\cdot 1\right] \\ &= \sum_{n_1,\ldots,n_j=0}^{\infty} c_{n_1,\ldots,n_j}(t)\cdot \mathrm{coeff}_{z_0^{n+m}\prod_{i=0}^{l} z_i^{\lambda_i+n_i}}\left[H(z_1)\cdots H(z_0)\cdots H(z_l)\cdot 1\right] \\ &= \sum_{n_1,\ldots,n_j=0}^{\infty} c_{n_1,\ldots,n_j}(t) Q_{(\lambda_1-n_1,\ldots,\lambda_j-n_j,n+n_1+\cdots+n_j,\lambda_{j+1},\ldots,\lambda_l)}(t). \end{aligned}$$

This completes the proof. $\square$

4.2 Restricting to Schur and Schur Q-functions

One important property of the Hall-Littlewood polynomial $Q_\lambda(t)$ is that its restriction to $t = 0$ is the Schur function s_λ and its restriction to $t = -1$ is the Schur Q-function Q_λ [7]. In other words,

$$Q_\lambda(0) = s_\lambda, \tag{4.1}$$

$$Q_\lambda(-1) = Q_\lambda. \tag{4.2}$$

N. Jing has also shown in [5] that the corresponding vertex operators satisfy the same properties, namely that the restriction of H_{-n} to $t = 0$ is B_n and the restriction of H_{-n} to $t = -1$ is Q_{-n}. In other words,

$$H_{-n}|_{t=0} = B_n \tag{4.3}$$

$$H_{-n}|_{t=-1} = Y_{-n} \tag{4.4}$$

Using these four identities we can restrict the result of Theorem 4.1.1 to the case of Schur and Schur Q-functions. We will now prove the main theorems of Chapters 2 and 3 (in particular Theorems 2.3.1, 2.4.1, and 2.4.8 – which also appear in Chapter 3) as special cases of Theorem 4.1.1.

Theorem 4.2.1. *For any partition λ,*

$$B(t)s_\lambda = \sum_{i\geq 1}(-1)^{|\lambda|-|\lambda^{(i)}|+i-1}t^{|\lambda^{(i)}|-|\lambda|}s_{\lambda^{(i)}}. \tag{4.5}$$

Proof. From Theorem 4.1.1 we have that

$$H_{-n}Q_\lambda(t) = \sum_{n_1,\ldots,n_j=0}^{\infty} c_{n_1,\ldots,n_j}(t)Q_{(\lambda_1-n_1,\ldots,\lambda_j-n_j,n+n_1+\cdots+n_j,\lambda_{j+1},\ldots,\lambda_l)}(t), \tag{4.6}$$

where $c_{n_1,\ldots,n_j}(t) = t^{n_1+\cdots+n_j-j+2k}(t^2-1)^{j-k}$, for k equal to the number of zeros in the set $\{n_1, n_2, \ldots, n_j\}$. Using the notation of the previous theorem, we let $m = n_1 + \cdots + n_j$, so $c_{n_1,\ldots,n_j}(t) = t^{m-j+2k}(t^2-1)^{j-k}$.

Then $c_{n_1,\ldots,n_j}(0) = 0$ unless $m - i + 2k = 0$, in which case $c_{n_1,\ldots,n_j}(0) = (-1)^{j-k}$. But $j - k$ is the number of nonzero entries in the set $\{n_1, \ldots, n_j\}$, so

$$m - j + k = m - (j-k) = \sum_{\substack{i=1\\ n_i\neq 0}} n_i - (j-k) = \sum_{\substack{i=1\\ n_i\neq 0}} (n_i - 1) \geq 0.$$

So $m - j + 2k = (m - j + k) + k$ where both $(m - j + k)$ and k are greater than or equal to zero. So in the case $m - j + 2k = 0$, we have $m - j + k = 0$ and $k = 0$. Since $k = 0$, $n_i \neq 0$ for all $1 \leq i \leq j$ and $(-1)^{j-k} = (-1)^j$. Since $m - j + k = 0$, $\sum_{i=1}^{j}(n_i - 1) = 0$, so $n_1 = n_2 = \cdots = n_j = 1$. Thus $c_{n_1,\ldots,n_j}(0) = 0$ except for the single case $n_1 = n_2 = \cdots = n_j = 1$. So when we restrict equation (4.6) to $t = 0$ we have

$$\begin{array}{rccc}
(H_{-n}Q_\lambda(t))|_{t=0} = & \sum_{n_1,\ldots,n_j=0}^{\infty} c_{n_1,\ldots,n_j}(0) & Q_{(\lambda_1-n_1,\ldots,\lambda_j-n_j,n+n_1+\cdots+n_j,\lambda_{j+1},\ldots,\lambda_l)}(0) \\
H_{-n}|_{t=0}(Q_\lambda(0)) = & c_{1,1,\ldots,1}(0) & Q_{(\lambda_1-1,\ldots,\lambda_j-1,n+j,\lambda_{j+1},\ldots,\lambda_l)}(0) \\
B_n(s_\lambda) = & (-1)^j & s_{(\lambda_1-1,\ldots,\lambda_j-1,n+j,\lambda_{j+1},\ldots,\lambda_l)} \\
B_n s_\lambda = & (-1)^{|\lambda|-|\lambda^{(i)}|+i-1} & s_{\lambda^{(i)}},
\end{array}$$

where $i = n + j + 1$ and $\lambda^{(i)} = (\lambda_1 - 1, \ldots, \lambda_j - 1, i - 1, \lambda_{j+1}, \ldots, \lambda_l)$ as in Chapters 2 and 3. Therefore

$$\begin{aligned} B(t)s_\lambda &= \left(\sum_{n\in\mathbb{Z}} B_n t^n\right)(s_\lambda) \\ &= \sum_{n\in\mathbb{Z}} t^n (B_n s_\lambda) \\ &= \sum_{i\geq 1} t^n (-1)^j s_{\lambda^{(i)}} \\ &= \sum_{i\geq 1} (-1)^{|\lambda|-|\lambda^{(i)}|+i-1} t^{|\lambda^{(i)}|-|\lambda|} s_{\lambda^{(i)}}, \end{aligned}$$

where $j = |\lambda| - |\lambda^{(i)}| + i - 1$ and $n = |\lambda^{(i)}| - |\lambda|$, since i ranges over all positive integers as n ranges over $\mathbb{Z}$ as discussed in Theorem 2.3.1. □

Theorem 4.2.2. *For any strict partition $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l)$,*

$$Y(t)Q_\lambda = \sum_{j=0}^{l} \sum_{n=\lambda_{j+1}+1}^{\lambda_j - 1} (-1)^j \, t^n \, Q_{(\lambda_1,\lambda_2,\ldots,\lambda_j,n,\lambda_{j+1}\ldots,\lambda_l)} \tag{4.7}$$

$$Y(t)Q_\lambda = \sum_{i\geq 0} (-1)^{l+|\lambda|-|\lambda^{[i]}|+i} \, t^{|\lambda^{[i]}|-|\lambda|} \, Q_{\lambda^{[i]}} \tag{4.8}$$

where we take the convention $\lambda_0 = \infty$ and $\lambda_{l+1} = -1$.

Proof. From Theorem 4.1.1 we have that

$$H_{-n}Q_\lambda(t) = \sum_{n_1,\ldots,n_j=0}^{\infty} c_{n_1,\ldots,n_j}(t) Q_{(\lambda_1-n_1,\ldots,\lambda_j-n_j,n+n_1+\cdots+n_j,\lambda_{j+1},\ldots,\lambda_l)}(t), \tag{4.9}$$

where $c_{n_1,\ldots,n_j}(t) = t^{n_1+\cdots+n_j-j+2k}(t^2-1)^{j-k}$, for k equal to the number of zeros in the set $\{n_1, n_2, \ldots, n_j\}$. Again using the notation of Theorem 4.1.1, we let $m = n_1 + \cdots + n_j$, so $c_{n_1,\ldots,n_j}(t) = t^{m-j+2k}(t^2-1)^{j-k}$.

Then $c_{n_1,\ldots,n_j}(-1) = (-1)^{m-j+2k}\left((-1)^2 - 1\right)^{j-k}$, which is zero if $j - k > 0$. Hence we can assume $j - k = 0$, so $j = k$. Thus $n_1 = n_2 = \cdots = n_j = 0$, so $m = 0$. So $c_{0,0,\ldots,0}(-1) = (-1)^{m-j+2k} = (-1)^{0-j+2j} = (-1)^j$. So when we restrict equation (4.9) to $t = -1$ we have

$$\begin{array}{rcl} (H_{-n}Q_\lambda(t))|_{t=-1} = & \sum_{n_1,\ldots,n_j=0}^{\infty} c_{n_1,\ldots,n_j}(-1) & Q_{(\lambda_1-n_1,\ldots,\lambda_j-n_j,n+n_1+\cdots+n_j,\lambda_{j+1},\ldots,\lambda_l)}(-1) \\ H_{-n}|_{t=-1}(Q_\lambda(-1)) = & c_{0,0,\ldots,0}(-1) & Q_{(\lambda_1-0,\ldots,\lambda_j-0,n+0+\cdots+0,\lambda_{j+1},\ldots,\lambda_l)}(0) \\ Y_{-n}(Q_\lambda) = & (-1)^j & Q_{(\lambda_1,\ldots,\lambda_j,n,\lambda_{j+1},\ldots,\lambda_l)}. \end{array}$$

Now notice that since $(Y_m)^2 = 0$, this contributes exactly when n is not an element of the partition λ, or when $\lambda_{j+1} + 1 \leq n \leq \lambda_j - 1$ with the convention mentioned in the theorem.

Therefore

$$
\begin{aligned}
Y(t)Q_\lambda &= \left(\sum_{n\in\mathbb{Z}} Y_{-n}t^n\right)(Q_\lambda) \\
&= \sum_{n\in\mathbb{Z}} t^n (Y_{-n}Q_\lambda) \\
&= \sum_{j=0}^{l} \sum_{n=\lambda_{j+1}+1}^{\lambda_j - 1} (-1)^j\, t^n\, Q_{(\lambda_1,\lambda_2,\ldots,\lambda_j,n,\lambda_{j+1}\ldots,\lambda_l)}.
\end{aligned}
$$

So (4.7) holds. Then (4.8) holds by setting $i = n - k$ as in Theorem 2.4.8. □

Thus Theorem 4.1.1 is a generalization of Carrell and Goulden's formula (4.5) and the Schur Q-analog (4.7) or (4.8), which recaptures the original results as special cases at $t = 0$ and $t = -1$ respectively.

REFERENCES

[1] Carrell, S. R. and Goulden, I. P. "Symmetric functions, codes of partitions and the KP hierarchy." *J. Algebraic Combin.* 32 (2010): 211–226.

[2] Comét, S. "Über die Anwendung von Binärmodellen in der Theorie der Charaktere der symmetrischen Gruppen." *Numer. Math.* 1 (1959): 90–109.

[3] Date, E., M. Jimbo, A. Kuniba, T. Miwa, and M. Okado. "A new realization of the basic representation of $A_n^{(1)}$." *Lett. Math. Phys.* 17 (1989): no. 1, 51–54.

[4] Jing, N. "Symmetric polynomials and $U_q(\widehat{sl_2})$." *Rep. Theory* 4 (2000): 46–63.

[5] Jing, N. "Vertex operators and Hall-Littlewood symmetric functions." *Adv. Math.* 87 (1991): 226–248.

[6] Jing, N. "Vertex operators, symmetric functions, and the spin group Γ_n." *J. Algebra* 138 (1991): no. 2, 340–398.

[7] Macdonald, I. G. *Symmetric functions and Hall polynomials.* Clarendon Press, Oxford, 1995.

[8] Nekrasov, N. A. and A. Okounkov. "Seilberg-Witten theory and random partitions." *Prog. in Math.* 244, The Unity of Mathematics (2006).

[9] Okounkov, A. "Infinite wedge and random partitions." *Selecta Math. (N.S.)* 7 (2001): no. 1, 57–81.

[10] Olsson, J. B. "Frobenius symbols for partitions and degrees of spin characters." *Math. Scand.* 61 (1987): 223–247.

[11] Sagan, B. E. *The symmetric group: representations, combinatorial algorithms, and symmetric functions.* Springer, New York, 1991.

[12] Schur, I. "Über die Darstellung der symmetrischen und der alternierenden Gruppe durch gebrochene lineare Substitutionen." *J. Reine Angew. Math.* 139 (1911): 155–250.

[13] Stanley, R. P. *Enumerative combinatorics, Vol. 2.* Cambridge University Press, Cambridge, 1999.

[14] Stembridge, J. R. "Shifted tableaux and the projective representations of symmetric groups." *Adv. Math.* 74 (1989): no. 1, 87–134.

[15] Zelevinsky, A. *Representations of finite classical groups, A Hopf algebra approach.* LNM 869, New York, 1981.