RECTIFICATION OF COMPOSITION TABLEAUX

BY

MELISSA K. BECHARD

A Thesis Submitted to the Graduate Faculty of
WAKE FOREST UNIVERSITY GRADUATE SCHOOL OF ARTS AND SCIENCES
in Partial Fulfillment of the Requirements
for the Degree of
MASTER OF ARTS
Mathematics
May 2012
Winston-Salem, North Carolina

Approved By:
Sarah, K. Mason, Ph.D., Advisor
Ellen Kirkman, Ph.D., Chair
Edward E. Allen, Ph.D.
Acknowledgments

I would like to thank my advisor Dr. Sarah Mason for her support, motivation, encouragement and friendship. I would also like to thank the faculty and staff of the Department of Mathematics at Wake Forest. To the members of my thesis committee, I appreciate your willingness and assistance in reading this thesis. To the Math Department at James Madison University, your preparation assisted in my success. I would also like to thank Cameron Atkins for all his help verifying my algorithms were understandable and proof-reading various parts of my thesis. Finally, I would like to thank my family, friends, and fellow graduate students for always being supportive.
# Table of Contents

Acknowledgments .............................................................. ii

Abstract ....................................................................... iv

Chapter 1  Introduction....................................................... 1
  1.1 Partitions and Compositions ........................................... 3
  1.2 Symmetric Polynomials ................................................ 3
  1.3 Tableaux .............................................................. 4
  1.4 Schur Polynomials .................................................... 6
  1.5 Quasisymmetric Polynomials ......................................... 7
  1.6 Composition Tableaux ................................................ 8
  1.7 Bijection Between Composition Tableaux and Semi-Standard Young Tableaux ................................................ 10
  1.8 Skew Diagrams ....................................................... 11
  1.9 Rectification .......................................................... 12
  1.10 Evacuation ........................................................... 14

Chapter 2  Rectification of Composition Tableaux...................... 17
  2.1 Rectification of one cell in the first column ....................... 17
      2.1.1 Algorithm ...................................................... 17
  2.2 Rectification of any number of cells in the first column ........ 25
      2.2.1 Algorithm ...................................................... 26

Chapter 3  Evacuation of Composition Tableaux .......................... 40
  3.1 Algorithm ............................................................ 40
  3.2 Applications and Extensions ......................................... 43

Bibliography ................................................................... 52

Vita............................................................................ 54
Abstract

In this thesis we define an algorithm for rectifying one cell in a composition tableau. We then describe a generalization of this rectification process. The generalization is from one cell in the first column to any number of cells in the first column, provided they are bottom-justified. We show this rectification of a composition tableau commutes with the rectification of a reverse semi-standard Young tableau. Rectification of reverse semi-standard Young tableaux is used to define evacuation of reverse semi-standard Young tableaux. We define evacuation for composition tableaux using our rectification process for composition tableaux.
Chapter 1: Introduction

A symmetric function is a polynomial (possibly in infinitely many variables) which is invariant under the action of the symmetric group, $S_n$. The study of symmetric functions has connections to many branches of mathematics, including algebraic geometry, group theory, representation theory, and Lie algebras. There are several different bases for symmetric polynomials including elementary symmetric polynomials, monomial symmetric polynomials, power-sum symmetric polynomials, complete homogeneous symmetric polynomials, and Schur polynomials. The Schur polynomial basis is described through fillings of partition diagrams. Schur polynomials provide information about the multiplicative structure of the cohomology ring of the Grassmannian. In representation theory, Schur polynomials correspond to the characters of the general linear group. Combinatorially, the tableaux used to define Schur polynomials have interesting properties such as the Littlewood-Richardson Rule, evacuation, and the RSK algorithm [1, 2].

Quasisymmetric functions are generalizations of symmetric functions. Quasisymmetric functions relate to many algebraic structures. For example, the Hopf algebra of quasisymmetric functions is dual to the Hopf algebra of non-commutative symmetric functions. There are analogs to some symmetric functions bases for quasisymmetric functions bases. Similar to Schur polynomials, quasisymmetric Schur functions are a useful basis for quasisymmetric functions [3]. This basis is described through fillings of composition diagrams, called composition tableaux.

This thesis focuses on the rectification of composition tableaux. There are several reasons rectification is useful. Rectification of semi-standard Young tableaux provides a way to prove how a skew Schur function can be written as a sum of Schur func-
tions [10]. It is unknown when a skew quasisymmetric Schur function can be written as a sum of quasisymmetric Schur functions. Rectification might provide insight to this question. Rectification is also helpful in providing information on multiplication rules. There are known multiplications such as the multiplication of two Schur functions [7] as well as the multiplication of a quasisymmetric Schur function with a Schur function [4]. These multiplications can be represented using operations on tableaux. There are several multiplications that are unknown such as: multiplication of two quasisymmetric Schur functions, multiplication of a skew quasisymmetric Schur function with a Schur function, and multiplication of two skew quasisymmetric Schur functions. In the case of the multiplication of two quasisymmetric Schur functions, information is lost in the representation by composition tableaux, since when the product is expanded there are some negative coefficients; rectification might be useful in keeping track of the sign change. Additionally, since evacuation is an algorithm defined around the rectification of a cell, rectification is necessary in order to carry out this process. Evacuation is an involution, and in fact can be used to describe certain situations which occur when using the RSK algorithm [5, 6]. Refer to section 3.3 for more information. The RSK algorithm provides a bijection between N-matrices and pairs of semistandard Young tableaux of the same shape $\lambda$. There is also a bijection between N-matrices and composition tableaux which rearrange the same shape. It is unknown which matrices correspond to pairs of composition tableaux of the same shape. Rectification of a composition tableau provides the foundation for evacuation of a composition tableau, since it should behave similarly to the evacuation of semistandard Young tableaux. We provide an algorithm for evacuation of composition tableaux in this thesis.
1.1 Partitions and Compositions

A partition of a positive integer $n$ is a way to write $n = \lambda_1 + \lambda_2 + ... + \lambda_k$ where $\lambda_i > 0$. If two sums differ only in their order, then they are considered to be the same partition; if order matters then we call this a composition. For example, the partitions of 3 are (3), (2, 1) and (1, 1, 1). On the other hand, the compositions of 3 are (3), (2, 1), (1, 2), and (1, 1, 1). Notice, (2, 1) and (1, 2) differ only in their order, and thus represent the same partition. However, as compositions these are distinct. We denote a partition by $\lambda = (\lambda_1, \lambda_2, ..., \lambda_n)$ where $\lambda_i \geq \lambda_{i+1}$ for all $i$. The length of a partition, $l(\lambda)$, is the number of non-zero parts of the partition. For example, when $\lambda = (4, 3, 3, 1)$, $l(\lambda) = 4$. The order of a partition is $|\lambda| = \sum_{i=1}^{l(\lambda)} \lambda_i$. For example, when $\lambda = (4, 3, 3, 1)$, $|\lambda| = 11$.

1.2 Symmetric Polynomials

A symmetric polynomial, $f(x)$, where $x = (x_1, x_2, ..., x_n)$, is invariant under all the permutations of the indices, where the permutations are from the symmetric group, $\mathfrak{S}_n$. This means if $\sigma f(x_1, x_2, ..., x_n) = f(x_{\sigma(1)}, x_{\sigma(2)}, ..., x_{\sigma(n)}) = f(x_1, x_2, ..., x_n)$ for all $\sigma \in \mathfrak{S}_n$, then $f(x)$ is a symmetric polynomial. For example, $x_1^2 x_2 + x_1 x_2^2$ is a symmetric polynomial in two variables. We can think of the action of $\sigma$ on $f(x)$ as permuting the exponents of the polynomial. In our example 2, 1 and 1, 2 are the exponents, and the permutations of the numbers 1 and 2 are (1, 2) and (2, 1). Since we have all the permutations of the exponents, 1 and 2, this is a symmetric polynomial. An example of a polynomial in three variables that is not symmetric is $y(x) = x_1 x_2^2 x_3^3 + x_1 x_2^3 x_3^2 + x_1^2 x_2^3 x_3 + x_1^2 x_2 x_3^2$. We would need to add the monomials $x_1^2 x_2 x_3^2$ and $x_1^2 x_2^3 x_3$ to $y(x)$ in order to have a symmetric polynomial. The ring of symmetric
polynomials is denoted $Sym$. Partitions form the indexing set for a basis for $Sym$. The monomial symmetric basis, $m_\lambda(x)$ where $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ (some possibly zero) and $x = (x_1, x_2, ..., x_n)$, is $m_\lambda = \sum_{\beta \in B} x^\beta$ where $\beta = (\beta_1, \beta_2, ..., \beta_m) \in B$ and $B$ is the set of all of the distinct permutations of $\lambda$. See equation (1.1) for an example of a monomial symmetric polynomial in three variables.

$$m_{21} = x_1^2x_2 + x_1^2x_3 + x_2^2x_3 + x_1x_2^2 + x_1x_3^2 + x_2x_3^2$$  

(1.1)

In equation (1.1), $B = \{(2,1,0), (2,0,1), (0,2,1), (1,2,0), (1,0,2), (0,1,2)\}$, which is all the distinct permutations of 2, 1 and 0. Notice we have a zero exponent since the $l(\lambda)$ is less than the number of variables. We can see the sum of all the monomials in three variables with the exponents containing all the permutations of 2, 1, 0 are represented in $m_{21}$.

### 1.3 Tableaux

A Young diagram is an array of left-justified cells that gives a visual representation of a partition $\lambda = (\lambda_1, \lambda_2, \lambda_3, ..., \lambda_n)$, where $\lambda_i$ gives the number of cells in the $i^{th}$ row of the diagram. The cells are filled with positive integers so that the entries in each row are weakly increasing, while the entries in each column are strictly increasing. The result is a semi-standard Young tableau, often abbreviated SSYT or Young tableau. See Figure 1.1(a) for an example of a Young tableau of shape $\lambda = (4, 3, 3, 1)$.

The frequency of each number in a tableau, $T$, is the weight of $T$; an example of the weight of a tableau is given in Figure 1.1. We can look at a tableau and determine the weight, and from this we can find the associated Schur polynomial (see section 1.4). A standard Young tableau (SYT), is a Young tableau whose entries are numbered 1 through $n$, where each number is used exactly once; refer to Figure 1.1(b). A Corner
of a partition $\lambda$ is a cell that can be removed such that the resulting diagram is still a Young diagram; refer to Figure 1.2. We focus on reverse semistandard Young tableau abbreviated RSSYT, which are analogous to semistandard Young tableaux; RSSYT make the proofs easier in this thesis. For further details see [2].

\begin{tabular}{cccc}
2 & 2 & 4 & 5 \\
4 & 5 & 7 & 13 \\
5 & 6 & 8 & 6 \\
7 & 9 & 10 & 7 \\
\end{tabular}

\begin{tabular}{cccc}
1 & 3 & 6 & 10 \\
2 & 5 & 8 & 7 \\
4 & 7 & 11 & 8 \\
9 & 10 & 11 & 9 \\
\end{tabular}

(a) SSYT \hspace{2cm} (b) SYT

Figure 1.1: A SSYT of weight (0, 2, 0, 2, 3, 1, 2, 1) and a SYT both of shape 4331

\begin{tabular}{|c|c|c|}
\hline
\multicolumn{3}{|c|}{Figure 1.2: A SSYT with highlighted corners} \\
\hline
\end{tabular}

A reverse semistandard Young tableau (RSSYT) is a filling of a partition diagram with positive integers such that:

1. Row entries are weakly decreasing left to right,

2. Column entries are strictly decreasing from top to bottom.

\begin{tabular}{cccc}
9 & 8 & 6 & 4 \\
7 & 7 & 5 & 1 \\
5 & 4 & 2 & 1 \\
3 & 2 & 1 & 1 \\
1 & & & 1 \\
\end{tabular}

Figure 1.3: A RSSYT of weight (4, 3, 1, 2, 2, 1, 2, 1, 1)
The reading word of a RSSYT, $T$, is the sequence of entries of $T$ obtained by concatenating the rows of $T$ from top to bottom. The tableau $T$ in Figure 1.3 has reading word 98642775115423211.

1.4 Schur Polynomials

*Schur polynomials* are symmetric polynomials that form an additive basis for the ring of symmetric polynomials. The Schur polynomials are used to record information about the multiplicative structure of groups and the classification of permutation groups. A Schur polynomial relates to the character of the general linear group, $GL(n)$, of $n \times n$ invertible matrices and is easily created from tableaux.

**Definition 1.** The set of Schur polynomials is defined by

\[
s_\lambda(x_1, x_2, \ldots, x_n) = \sum_{T \in \text{SSYT}} x^T = \sum_{T \in \text{SSYT}} x_1^{t_1}x_2^{t_2}\cdots x_n^{t_n}
\]

where we sum over all SSYT $T$ of shape $\lambda$, where $T$ has weight $(t_1, t_2, \ldots, t_n)$.

Below is an example of a Schur polynomial $s_{21}(x_1, x_2, x_3)$ of shape $\lambda = (2, 1)$. To find $s_{21}$ we sum the the ways to fill a semi-standard Young tableau of shape $\lambda$ with the integers $\{1, 2, 3\}$.

**Example 1.4.1.** A Schur polynomial, $s_{21}$, and the associated fillings.

\[
\begin{array}{c}
x_1^2x_2 \leftrightarrow 2 \begin{array}{c}1 \\
1
\end{array}, \quad x_1^2x_3 \leftrightarrow 3 \begin{array}{c}1 \\
1
\end{array}, \quad x_2^2x_3 \leftrightarrow 3 \begin{array}{c}2 \\
2
\end{array}
\end{array}
\]

\[
\begin{array}{c}
x_1x_2^2 \leftrightarrow 2 \begin{array}{c}2 \\
1
\end{array}, \quad x_1x_3^2 \leftrightarrow 3 \begin{array}{c}3 \\
1
\end{array}, \quad x_2x_3^2 \leftrightarrow 3 \begin{array}{c}3 \\
2
\end{array}
\end{array}
\]
\[ s_{21} = (x_1^2 x_2 + x_1^2 x_3 + x_2^2 x_3) + (x_1 x_2^2 + x_1 x_3^2 + x_2 x_3^2) + (2 x_1 x_2 x_3) \quad (1.2) \]

Notice we have grouped \( s_{21} \) in a very specific way. Using equation (1.1) we can see \( s_{21} = m_{21} + 2m_{111} \).

### 1.5 Quasisymmetric Polynomials

A **quasisymmetric function** is a bounded degree formal power series \( F \in \mathbb{Q}[\mathbb{[}x_1, x_2, ...]\) such that for all \( k \) the coefficient of \( x_{i_1}^{a_1} x_{i_2}^{a_2} ... x_{i_k}^{a_k} \) is equal to the coefficient of \( x_1^{a_1} x_2^{a_2} ... x_k^{a_k} \) for all \( i_1 < i_2 < ... < i_k \) and for all compositions \((a_1, a_2, ..., a_k)\) [3]. A quasisymmetric function with finitely many variables is a **quasisymmetric polynomial**. The following are examples of quasisymmetric polynomials: \( f(x_1, x_2, x_3) = x_1^2 x_2 + x_1^2 x_3 + x_2^2 x_3 \), and \( g(x_1, x_2, x_3) = x_1^2 x_2 x_3^3 \). Examples of polynomials that are not quasisymmetric include \( h(x_1, x_2) = x_1^2 \) and \( k(x_1, x_2, x_3) = x_1 x_2^2 + x_1 x_3^2 \). Unlike monomial symmetric polynomials where we sum the monomials over all possible permutations of the exponents, quasisymmetric polynomials must at least sum over all permutations of the zero exponents. Notice, the exponent sequences of the monomials occurring in \( f \) are \( (2, 1, 0) \) and \( (2, 0, 1) \) and \( (0, 2, 1) \). Notice the 2 and 1 remain in the same order, while the zero exponent was permuted. Since the polynomial contains all permutations of the zero exponents, it is a quasisymmetric polynomial. The second example, \( g \), has the exponent sequence \( (2, 1, 3) \); since there are no zero exponents, no permutations are needed to make this a quasisymmetric polynomial. It is important to note that adding any monomial which uses all the variables to a quasisymmetric polynomial results in a quasisymmetric polynomial since this monomial is itself quasisymmetric, and adding any two quasisymmetric polynomials in the same number of variables
results in a quasisymmetric polynomial. For example, adding the monomial $x_1x_2^2x_3^2$ to $g$ gives $x_1^2x_2x_3^3 + x_1x_2^3x_3^2$, which remains quasisymmetric. On the other hand, if we add $f$ and $g$ together we have $x_1^2x_2 + x_1^2x_3 + x_2^2x_3 + x_1^2x_2x_3^2$, which is also quasisymmetric. We denote the ring of quasisymmetric polynomials by $Qsym$. Also notice that quasisymmetric polynomials do not have to be symmetric, as $g$ is not, but all symmetric polynomials are quasisymmetric. Thus, $Sym \subseteq Qsym$. A monomial quasisymmetric polynomial $M_\alpha(x_1, x_2, ..., x_n)$, is the sum of all the monomials $x^\beta$, where $\beta$ runs over all the compositions of $\alpha$. See equation (1.3) for two examples of monomial quasisymmetric polynomials.

$$M_{21} = x_1^2x_2 + x_1^2x_3 + x_2^2x_3$$

$$M_{12} = x_1x_2^2 + x_1x_3^2 + x_2x_3^2$$

(1.3)

Notice that $m_{21}$ in equation (1.1) can be written in terms of quasisymmetric monomials from equation (1.3) as $m_{21} = M_{21} + M_{12}$. We can also write $s_{21}$ in equation (1.2) as a sum of quasisymmetric monomials. Since we saw above that $s_{21} = m_{21} + 2m_{111}$, we have $s_{21} = M_{21} + M_{12} + 2M_{111}$.

### 1.6 Composition Tableaux

A composition diagram is an array of left-justified cells that gives a visual representation of a composition. The following is an example of a composition diagram, $C$, of shape $(3,1,2)$:

**Example 1.6.1.** A composition diagram representing the composition of 6 into $(3,1,2)$.
A Composition Tableau (CT) is a filling of a composition diagram with positive integers, satisfying the following properties:

1. Entries in the first column are strictly increasing from top to bottom,

2. Row entries are weakly decreasing from left to right,

3. Given any cell $a$ directly to the right of any cell $c$, and some cell $b$ that is below cell $a$ in the same column, but not necessarily directly below, if $a \leq b$ then $b > c$. (We think of empty cells as containing the entry 0.)

Example 1.6.2. A composition tableau

$$
\begin{array}{ccc}
0 & 1 & 2 \\
1 & 2 \\
3 & 3 & 3 & 1 \\
5 & 4 & 2 \\
\end{array}
$$

Analogous to the Schur function basis for $Sym$, there is a basis for $Qsym$ called the quasisymmetric Schur function basis.

Definition 2. The set of quasisymmetric Schur polynomials is defined by

$$QS_\alpha = \sum_{F \in CT(\alpha)} x^F$$

where we sum over all CT $F$ of shape $\alpha$.

Below is an example of a quasisymmetric Schur polynomial $QS_{121}$ of shape $\alpha = (1, 2, 1)$. To find $QS_\alpha(x_1, x_2, x_3, x_4)$ we sum the ways to fill a CT of shape $\alpha$ with the integers $\{1, 2, 3, 4\}$. 

Example 1.6.3. A quasisymmetric Schur polynomial, \( QS_{121} \), and the associated fillings.

\[
\begin{align*}
x_1x_2^2x_3 & \leftrightarrow \begin{array}{ccc}
1 & 2 & 2 \\
2 & 3 & \\
\end{array}, &
x_1x_2^2x_4 & \leftrightarrow \begin{array}{ccc}
1 & 2 & 2 \\
2 & 4 & \\
\end{array}, &
x_1x_2x_3x_4 & \leftrightarrow \begin{array}{ccc}
1 & 3 & 2 \\
4 & & \\
\end{array}
\end{align*}
\]

\[
\begin{align*}
x_1x_2^2x_3 & \leftrightarrow \begin{array}{ccc}
1 & 3 & 3 \\
4 & & \\
\end{array}, &
x_2x_3^2x_4 & \leftrightarrow \begin{array}{ccc}
2 & 3 & 3 \\
4 & & \\
\end{array}
\end{align*}
\]

\[
QS_{121} = x_1x_2^2x_3 + x_1x_2^2x_4 + x_1x_2x_3x_4 + x_1x_3^2x_4 + x_2x_3^2x_4
\]  (1.4)

1.7 Bijection Between Composition Tableaux and Semi-Standard Young Tableaux

At this point we have described partitions, a way to visually represent partitions, RSSYT, and the analogous compositions and a way to visually represent compositions, CT. We now want to define a bijection between RSSYT and CT that will be useful in proving properties of CT that were previously unknown.

Theorem 1.7.1. [3] There exists a bijection, \( \rho \), between composition tableaux and reverse semistandard Young tableaux.

Given a CT \( U \), \( \rho \) arranges the column entries in decreasing order to produce a RSSYT \( T \). Given a RSSYT \( T \), \( \rho^{-1} \) arranges the first column in increasing order from top to bottom. Then in decreasing order, \( \rho^{-1} \) places each entry of the second column of \( T \) in the highest possible position so that the row entries are weakly decreasing; once a cell is taken, no other entry may be placed in that cell. This insertion is repeated for each subsequent column of \( T \), until the final result is a CT. Below is an example of this bijection, where \( T \) is the RSSYT, and \( U \) is the CT.
Example 1.7.2.

\[
U = \begin{array}{cccc}
2 & 2 & 2 & 2 \\
3 & 1 & & \\
4 & 4 & 4 & 3 \\
6 & 5 & 5 & 1 \\
7 & 7 & 3 \\
\end{array}
\quad \xrightarrow{\rho} \quad T = \begin{array}{cccc}
7 & 7 & 5 & 3 \\
6 & 5 & 4 & 2 \\
4 & 4 & 3 & 1 \\
3 & 2 & 2 \\
2 & 1 \\
\end{array}
\]

1.8 Skew Diagrams

Given two diagrams, one of shape \( \lambda \) and one of shape \( \mu \), we say \( \mu \subseteq \lambda \) if \( \mu_i \leq \lambda_i \) \( \forall i \). If \( \mu \subseteq \lambda \) then the skew diagram, \( \lambda/\mu \), consists of the cells contained in \( \lambda \) but not contained in \( \mu \). The following is an example of a skew diagram.

Example 1.8.1.

\[
\lambda = \begin{array}{cccc}
\text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } \\
\end{array}, \quad \mu = \begin{array}{cccc}
\text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } \\
\end{array}, \quad \lambda/\mu = \begin{array}{cccc}
\text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } \\
\end{array}
\]

We say a skew diagram is a horizontal strip if no two cells lie in the same column, and is a vertical strip if no two cells lie in the same row. In the above example \( \lambda/\mu \) is neither a vertical strip nor a horizontal strip. Below is an example of both a horizontal strip, \( \lambda \), and a vertical strip, \( \mu \).

Example 1.8.2.

\[
\lambda = \begin{array}{cccc}
\text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } \\
\end{array}, \quad \mu = \begin{array}{cccc}
\text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } \\
\text{ } & \text{ } & \text{ } & \text{ } \\
\end{array}
\]

Horizontal strip \hspace{1cm} Vertical strip
An inside corner of a skew diagram, $\lambda/\mu$ is a corner of $\mu$. See Example 1.8.3 for an illustration of the inside corners of the skew diagram from Example 1.8.1.

**Example 1.8.3.** Inside corners of a skew diagram

A skew Schur function denoted $s_{\lambda/\mu}$ is a Schur function obtained from summing over all SSYT $T$ of shape $\lambda/\mu$.

### 1.9 Rectification

Rectification of a tableau $T$ is a procedure that straightens a skew tableau, i.e. makes $T$ a RSSYT. This procedure gives a way to multiply Young tableaux. It is important to know when a skew Schur function is equal to a Schur function. Rectification plays a major role in proving when a skew Schur function is equal to a Schur function. The process of rectifying a RSSYT of shape $\lambda/(1)$ is provided below, where the cell of the removed entry is considered an empty cell.

1. Slide the larger of the two neighbors below and to the right of the empty cell into the empty cell. If the two neighbors have the same entry, then slide the lower entry into the empty cell. Whichever neighbor slid into the empty cell’s spot, that neighbor’s spot is now the new empty cell. Consider any empty cell as a zero.

2. Continue this process until there are no more neighbors to compare.

We denote the process of rectifying a RSSYT by $\psi$. Below is an example of $T$ and the rectification of $T$. 

12
Example 1.9.1. Rectification of RSSYT

We can generalize this process to rectification of more than one cell. The idea is to begin with a skew semi-standard Young tableau and execute a sequence of slides so that the final result is a semi-standard Young tableau. In the above example we have a skew shape of $\lambda/(1)$. Since the larger of the two neighbors slides into the empty cell, which is positioned to the right of and/or higher than the entry’s original position, the rows remain in weakly decreasing order, as required by the properties of RSSYT. Additionally, when the two neighbors are equal, the lower entry slides up, which guarantees the entries in each column are distinct, as also required by the properties of RSSYT. Thus, through the sequence of slides the final result is a RSSYT. It is known that the choice of sliding steps is independent of the choice of inside corners [2]. That means, if we began with a skew shape $(5441)/(321)$ we would have three
choices of inside corners to begin our process of rectification, all resulting in the same RSSYT. Thus, there is a unique rectification of any RSSYT. Below is a RSSYT of shape $T = (5441)/(321)$. The example illustrates the three initial choices of inside corners and one possible order in which they may be chosen which is highlighted in white. The algorithm for rectification begins with the cell with a 1 highlighted in white.

Example 1.9.2.

$$T = \begin{array}{c} \hline & & 3 & 4 & 5 \\ \hline 1 & 3 & 2 \\ 2 & 2 & 2 & 1 \\ 4 \end{array} \quad |\mu| = 6$$

The following example illustrates how rectification gives a way to multiply RSSYT. Here we are multiplying $\lambda = (21)$ with $\mu = (2, 2)$.

Example 1.9.3. Multiplication of two RSSYT one of shape (2, 1) and the other of shape (2, 2) using rectification

Here we have $(x_2x_4^2)(x_1^2x_3^2) = x_1x_2^2x_3x_4^2$.

1.10 Evacuation

*Evacuation* is a reverse sliding algorithm and is an involution on RSSYT. Let $n = l(\lambda) + 1$ and fill a RSSYT with entries less than $n$. The following is the algorithm for evacuation of a reverse semistandard Young tableau of shape $\lambda$:  

14
1. Remove the largest entry in column one. Call this entry \( r \).

2. Rectify the tableau.

3. Start a new RSSYT of the same shape \( \lambda \).

4. In the corner that was removed after the rectification, fill in the new RSSYT with \((n - r)\).

5. Repeat until there are no more cells left to rectify in the original RSSYT.

**Example 1.10.1.** RSSYT of shape \( \lambda = (3, 2, 1) \)

\[
T = \begin{array}{ccc}
5 & 5 & 4 \\
3 & 2 & \\
1 & & \\
\end{array}
\]

new RSSYT

\[
\begin{array}{ccc}
&& \\
& 1 & \\
\end{array}
\]

\[
\begin{array}{ccc}
5 & 4 & \\
3 & 2 & \\
1 & & \\
\end{array}
\rightarrow
\begin{array}{ccc}
5 & 4 & \\
3 & 2 & \\
1 & & \\
\end{array}
\rightarrow
\begin{array}{ccc}
&& \\
& 1 & \\
\end{array}
\]

\[
\begin{array}{ccc}
4 & \\
3 & 2 & \\
1 & & \\
\end{array}
\rightarrow
\begin{array}{ccc}
4 & 2 & \\
3 & & \\
1 & & \\
\end{array}
\rightarrow
\begin{array}{ccc}
&& \\
& 1 & \\
\end{array}
\]

\[
\begin{array}{ccc}
2 & \\
3 & 2 & \\
1 & & \\
\end{array}
\rightarrow
\begin{array}{ccc}
3 & 2 & \\
1 & & \\
\end{array}
\rightarrow
\begin{array}{ccc}
&& \\
& 1 & \\
\end{array}
\]

\[
\begin{array}{ccc}
2 & \\
1 & & \\
\end{array}
\rightarrow
\begin{array}{ccc}
2 & & \\
1 & & \\
\end{array}
\rightarrow
\begin{array}{ccc}
3 & 1 & \\
1 & & \\
2 & & \\
\end{array}
\]

15
Since evacuation is an involution, we can apply the process again to the above example and see that the result is the original RSSYT.
2.1 Rectification of one cell in the first column

The set of reverse semi-standard Young tableaux with the binary operation multiplication is an associative monoid. In other words, the empty tableau is the multiplicative identity, and multiplication is closed and associative. There are two common ways to multiply RSSYT, one of which uses rectification of RSSYT, and is described in further detail later. However, it is unknown whether or not composition tableaux have similar properties. Since rectification gives a way to multiply tableaux, rectification of composition tableaux may help provide information about the algebraic structure of the set of composition tableaux. It is unknown when a skew quasisymmetric Schur function can be written as a sum of quasisymmetric Schur functions. There are several multiplication rules that are unknown such as: multiplication of two quasisymmetric Schur functions, multiplication of a skew quasisymmetric Schur function with a Schur function, and multiplication of two skew quasisymmetric Schur functions. Rectification of a composition tableau, which was previously unknown, will help provide insight to these questions. Additionally, evacuation for composition tableaux was previously unknown. We provide an algorithm for evacuation for composition tableaux in this thesis.

2.1.1 Algorithm

We number our columns from left to right, so that the 1\textsuperscript{st} column is the leftmost non-empty column. We consider the cell of a removed entry as an empty cell. Let \((c, r)\) denote the position of an entry in column \(c\) and row \(r\). The following is the process for the rectification of a CT of shape \(\alpha/(1)\):
The algorithm, $\phi$, to rectify a composition tableau:

1. If there is an entry, $a$, in the cell directly to the right of the removed entry, move entry $a$ into the first column in such a way that the column remains strictly increasing. Moving $a$ into the $i^{th}$ row forces the first $(i−1)$ rows to shift up. We refer to $a$’s original cell as the new removed cell, in position $(c, r)$.

2. Move the entry from $(c + 1, r)$ into column $c$, in the highest cell possible so that the rows remain weakly decreasing. This entry may bump any entry of smaller value. If an entry is bumped, that entry’s spot is replaced by the new entry, and the bumped entry must find a new cell.

3. Move the bumped entry to the next highest cell in that column such that the corresponding row of that cell remains weakly decreasing. Bumped entries are also allowed to bump entries of smaller value.

4. Repeat steps 3 and 4 for each subsequent column.

5. If there is no cell directly to the right of the removed cell then stop.
Example 2.1.1. Rectification of CT

The following definitions and lemmas are used to prove this algorithm gives a rectification of composition tableaux and commutes with the rectification of RSSYT. We denote $c_{ij}$ as the cell in column $i$ from the left and row $j$ from the top, and $f(c_{ij})$ as the entry in cell $c_{ij}$.

**Definition 3.** Any entry in column $c$ that shifts into column $(c-1)$ during rectification is called a *shifting entry*. 

19
Lemma 2.1.2. If there are no shifting entries of the $i^{th}$ column, then there are no shifting entries of the $m^{th}$ column, where $i < m$.

Proof. Let $T$ be a RSSYT. Assume there are no shifting entries of the $i^{th}$ column. This means during the rectification of $T$, none of the entries in the $i^{th}$ column move into the $(i - 1)^{th}$ column. Therefore, there are no empty cells in the $i^{th}$ column for the $(i + 1)^{th}$ column entries to shift into. Thus, there are no shifting entries of the $(i + 1)^{th}$ column. In fact, there are no empty cells for any of the successive column entries to shift into. Hence, there are no shifting entries of the $m^{th}$ column for $i < m$. \hfill \Box

Definition 4. In a RSSYT, an entry $f(c_{ij})$ has diagonal dominance if

$$f(c_{ij}) > f(c_{(i-1)(j+1)})$$

Figure 2.1: Diagonal dominance of $f(c_{ij})$ provided $f(c_{ij}) > f(c_{(i-1)(j+1)})$

Lemma 2.1.3. If an entry, $f(c_{ij})$, is a shifting entry, then $f(c_{ij})$ must have diagonal dominance.

Proof. Let $T$ be a RSSYT. During the rectification of $T$, comparisons are made between the neighbor directly below a removed entry and the neighbor directly to the right of a removed entry. Let $f(c_{ij})$ be a shifting entry. This means at some point during the rectification of $T$ the cell $c_{(i-1)j}$ must be empty in order for $f(c_{ij})$ to shift into column $(c - 1)$. When cell $c_{(i-1)j}$ is empty, the neighbor, $f(c_{(i-1)(j+1)})$, directly below the empty cell $c_{(i-1)j}$ is compared to the neighbor, $f(c_{ij})$, directly to the right of
the empty cell \( c_{(i-1)j} \). Since \( f(c_{ij}) \) is a shifting entry, we know during this comparison \( f(c_{ij}) \) must satisfy \( f(c_{ij}) > f(c_{(i-1)(j+1)}) \) in order to shift into the empty cell \( c_{(i-1)j} \). Thus, \( f(c_{ij}) > f(c_{(i-1)(j+1)}) \), which means \( f(c_{ij}) \) has diagonal dominance.

**Lemma 2.1.4.** All of the shifting entries from a RSSYT, \( T \), are positioned in the bottom row of the associated CT by \( \rho^{-1} \).

**Proof.** Let \( T \) be a RSSYT, where some cells of \( T \) might be empty; we think of empty cells as containing the entry 0. Let \( f(c_{IJ}) \) be the leftmost entry to have diagonal dominance where if there are two diagonal dominant entries in the same column, \( f(c_{IJ}) \) is the largest of the two. Note, in the leftmost column that contains a diagonal dominant entry, there is only one largest diagonally dominant entry in that column since the column entries are ordered in strictly decreasing order.

All entries \( f(c_{Ij}) \) where \( j < J \), in column \( I \) of \( T \) larger than \( f(c_{IJ}) \), satisfy \( f(c_{Ij}) \leq f(c_{(I-1)(j+1)}) \) since \( f(c_{IJ}) \) is the largest entry to have diagonal dominance, see diagram (2.1). Thus, \( f(c_{Ij}) \) is not diagonally dominant. So, there are \((j + 1)\) many cells in column \((I - 1)\) that contain an entry larger than \( f(c_{I(j-1)}) \), and at the time of the placement of \( f(c_{I(j-1)}) \) only \((j - 1)\) many have already been placed by \( \rho^{-1} \), which leaves two cells available when placing \( f(c_{I(j-1)}) \). Since \( \rho^{-1} \) requires each entry must take the highest cell available, none of these entries will be placed in the bottom row.

\[
\begin{array}{c|c}
\hline
& C_{I1} \\
\hline
C_{I2} & \vdots \\
\hline
& C_{I(J-1)} \\
\hline
C_{(I-1)J} & C_{IJ} \\
\hline
C_{(I-1)(J+1)} & \hline
\end{array}
\]

(2.1)

In this table \( f(c_{Ij}) \) for \( j < J \) is bolded
After $f(c_{(I-1)J})$ has been placed by $\rho^{-1}$, the next entry to be placed is $f(c_{IJ})$. Since $f(c_{IJ}) > f(c_{(I-1)(J+1)})$ by diagonal dominance, and the $J$-many entries larger than $f(c_{(I-1)(J+1)})$ already have one of the $f(c_{IJ})$'s to their right excluding the entry in the bottom row of column $(I-1)$, we know $f(c_{IJ})$ must be placed in the bottom row. We must now consider the next column to ensure each shifting entry is positioned in the bottom row of the CT by $\rho^{-1}$. We will be doing an inductive argument.

Let $f(c_{(I+1)j})$ be the largest diagonally dominant entry such that $j \geq J$. We have two cases. The first is none of the entries $f(c_{(I+1)k})$ where $k < j$ in column $(I+1)$ of $T$ larger than $f(c_{(I+1)j})$ have diagonal dominance. In this case we apply the same argument as above, and thus we know $f(c_{(I+1)j})$ is positioned in the bottom row of the CT by $\rho^{-1}$. The second case is we have some entry, $t$, in column $(I+1)$, that is larger than $f(c_{(I+1)j})$ and is diagonally dominant. This is possible since we required $f(c_{(I+1)j})$ to be the largest diagonally dominant entry such that $j \geq J$. Since $t$ must be higher than $f(c_{IJ})$, there must be some entry in column $I$ that is to the left of $t$ in $T$, and therefore, $t$ may be placed next to this entry in the CT by $\rho^{-1}$ since $t$ is less than or equal to this entry and $t$ is placed before $f(c_{(I+1)j})$ by $\rho^{-1}$. Thus, $t$ does not need to be placed in the bottom row beside $f(c_{IJ})$ since $t$ must take the highest cell available. We now know all the entries above $f(c_{(I+1)j})$ may be placed by $\rho^{-1}$ beside some entry other than $f(c_{IJ})$ in the bottom row. We then have the same argument as with $f(c_{IJ})$, and $f(c_{(I+1)j})$ cannot find a position available except in the bottom row next to $f(c_{IJ})$. This argument is continued for each subsequent column. Thus, we have shown the shifting entries are in the bottom row of the CT.

**Lemma 2.1.5.** Moving an entry directly to the right of the removed cell of a CT, into the first column into the $i$th row shifts the first $(i - 1)$ rows up.

**Proof.** Let $T$ be a RSSYT, where some cells may be empty; recall that we think of
empty cells as containing the entry 0. Let \( f(k_{2,j}) \) be the largest entry in the second column to have diagonal dominance.

During the rectification of \( T \), \( f(k_{2,j}) \) moves into the 1\(^{st} \) column, causing the 1\(^{st} \) column to have at least one more entry than the 2\(^{nd} \) column. When we apply \( \rho^{-1} \) we re-order the first column in increasing order. Then, any entry in column 2 that was larger than \( f(k_{2,j}) \) in the RSSYT must be placed lower than the row where \( f(k_{2,j}) \) is placed in the CT, \( U \), in order to preserve weakly decreasing row entries. Any entry in column 2 that was smaller than \( f(k_{2,j}) \) in the \( T \) must be placed higher than the row where \( f(k_{2,j}) \) is placed in the \( U \) since \( \rho^{-1} \) requires entries to take the highest cell available in decreasing order, and we already showed in the previous lemma that there is at least one other entry in the 1\(^{st} \) column that each of the entries may be placed beside so that they are not placed beside \( f(k_{2,j}) \). Therefore, moving \( f(k_{2,j}) \) into the 1\(^{st} \) column, say in row \( j \), causes the highest \((j - 1)\) rows to shift up.

Note, by the contrapositive of Lemma 2.1.3, if there is no entry in the second column with diagonal dominance, then there are no shifting entries of the second column, and the remaining entries in the 1\(^{st} \) column shift up in the \( T \). Thus, only the largest entry in the 1\(^{st} \) column has been removed. \( \square \)

**Theorem 2.1.6.** The algorithm \( \phi \) gives the rectification of the composition tableaux skewed by (1). This process commutes with the rectification of a RSSYT in the sense that the following diagram commutes:

\[
\begin{array}{ccc}
CT & \xrightarrow{\phi} & \phi(CT) \\
\rho & \downarrow & \downarrow \rho \\
RSSYT & \xrightarrow{\psi} & \psi(RSSYT)
\end{array}
\]

**Proof.** We prove this theorem by showing that \( \phi \) produces a CT and that the above diagram commutes.
Notice property 1 of a composition tableau is satisfied since the entry directly to the right of the removed box in the CT is moved into the first column by \( \phi \) in the highest cell allowing the column to be strictly increasing. Every shifting entry shifts one column to the left, and is placed in the highest cell in that column such that the row into which it is placed remains weakly decreasing, which satisfies property 2 of a composition tableau. Additionally, bumping satisfies property 2 of a composition tableau since an entry may only bump an entry of smaller value so that the row in which it is placed remains weakly decreasing. The map \( \rho^{-1} \) fixes the entries in each of the columns. This is because \( \rho^{-1} \) is defined to re-order each column keeping the associated entries within that column. So, since an entry can only bump an entry of smaller value so that the row in which it is placed remains weakly decreasing, bumping in \( \phi \) obeys the insertion rule, and property 3 must hold. Therefore, we do in fact have a valid map from a composition tableau to a composition tableau.

We now show this process commutes with the rectification of a RSSYT. Refer to diagram (2.1) for a visual representation of the following inequalities. The rules of a reverse semi-standard Young tableau imply \( f(c_{IJ}) < f(c_{(I-1)J}) \) and
\[
f(c_{IJ}) \leq f(c_{(I-1)J})\text{ since column entries are strictly decreasing and row entries are weakly decreasing. We also know } f(c_{(I-1)J}) \leq f(c_{(I-1)(J-1)}) \text{ since } f(c_{IJ}) \text{ is the largest entry in column } I \text{ to have diagonal dominance. This gives us } \]
\[
f(c_{IJ}) < f(c_{(I-1)J}) \leq f(c_{(I-1)(J-1)}) \text{. So, } f(c_{IJ}) < f(c_{(I-1)J}) \text{. We also know that } \]
f(c_{IJ}) > f(c_{(I-1)(J+1)}) \text{ by diagonal dominance. Since each column has distinct entries in the RSSYT, and } f(c_{IJ}) < f(c_{(I-1)J}) \text{ and } f(c_{IJ}) > f(c_{(I-1)(J+1)}) \text{, this means } f(c_{IJ}) \text{ cannot be the same as any entry in the } (I-1)^{th} \text{ column. This argument holds for all of the shifting entries. So, column entries of the rectified CT are distinct. By Lemma 2.1.4 we have shown the shifting entries are positioned in the bottom row of the composition tableau by } \rho^{-1}. \text{ Only these entries in the bottom row of the CT}
shift columns by $\phi$. Thus, the column entries are the same in the rectified CT and
the rectified RSSYT. Hence, this process does indeed give the same tableau, i.e.,
$\rho(\phi(T)) = \psi(\rho(T))$.

Therefore, $\phi$ gives a rectification of CT that commutes with the rectification of
RSSYT. \hfill $\square$

Below is an example of a composition tableau, $U$, and its associated reverse semi-
standard Young tableau, $T$, followed by both of their rectifications. We can see that
$\rho(\phi(T)) = \psi(\rho(T))$.

**Example 2.1.7.**

\[
U = \begin{array}{cccc}
2 & 2 & 2 & 1 \\
3 & 1 \\
4 & 4 & 4 & 3 \\
6 & 5 & 5 & 1 \\
7 & 3
\end{array} \quad \phi \quad \begin{array}{cccc}
2 & 2 & 2 & 1 \\
3 & 3 \\
4 & 4 & 4 & 3 \\
6 & 5 & 5 & 1 \\
7 & 1
\end{array}
\]

\[
T = \begin{array}{cccc}
7 & 5 & 3 & 1 \\
6 & 5 & 4 & 2 \\
4 & 4 & 3 & 1 \\
3 & 2 & 2 \\
2 & 1
\end{array} \quad \psi \quad \begin{array}{cccc}
7 & 5 & 5 & 3 & 1 \\
6 & 4 & 4 & 2 \\
4 & 3 & 2 & 1 \\
3 & 2 \\
2 & 1
\end{array}
\]

**2.2 Rectification of any number of cells in the first column**

The following is an algorithm, $\phi$, to rectify a CT of shape $\lambda/(1^k)$, that is, $k$ many
cells that are adjacent and bottom-justified are removed from the first column of $\lambda$.
The algorithm is followed by an example to help illustrate.
2.2.1 Algorithm

Algorithm $\phi$:

1. Swap all entries directly right of the $k$-removed cells with the $k$-removed cells.

2. Reorder the rows so that the first column entries are strictly increasing, fixing the row entries. Consider the empty cells as part of the row.

3. Start with the largest entry in column 3 to the right of a removed box, and insert this entry into column 2 into the highest cell possible so that the rows remain weakly decreasing. This entry may bump any entry of smaller value. Bumped entries are moved into the next highest cell in that column so that the rows remain weakly decreasing (bumping if necessary). Repeat for the next largest entry in column 3 to the right of a removed box. Continue until there are no more entries to the right of the removed boxes in column 3. The cells of the entries from column 3 that were inserted into column 2 are thought of as the new removed boxes.

4. Repeat step 4 for each subsequent column.

5. If there are no cells directly right of the removed cells then stop.
Example 2.2.1. Generalized rectification of CT

The following definitions and lemmas are used in an upcoming proof.

Definition 5. If $\psi$ is applied to an inside corner of a RSSYT of shape $\lambda/\mu$ $k$ times, then we say that we have $k$ stages of rectification, where $g_k$ denotes the $k^{th}$ stage of rectification.

We are going to show that for a RSSYT $\lambda/1^k$, the empty cell from the $n^{th}$ row is the starting point for $g_{(k-n+1)}$. 

27
Lemma 2.2.2. Let $T$ be a RSSYT skewed by $1^k$. For each column, $c$, the $n^{th}$ largest shifting entry shifts during $g_{(k-n+1)}$.

Proof. Let $T$ be a RSSYT skewed by $1^k$. We know we have a total of $k$ stages of rectification. If there are no shifting entries then we are done. Assume we have at least one shifting entry, $a_{2k}$, that shifts during $g_{(k-n+1)}$. Recall, the empty cell from the $n^{th}$ row is the starting point for $g_{(k-n+1)}$. We need to show $a_{2k}$ is the $n^{th}$ largest shifting entry for the second column.

First we compare $a_{1(n+1)}$ to $a_{2n}$. If $a_{2n} > a_{1(n+1)}$ then $a_{2n} = a_{2k}$ since $a_{2k}$ is the shifting entry for $g_{(k-n+1)}$. If not, then $a_{1(n+1)}$ shifts up, and $a_{1(n+2)}$ is compared to $a_{2(n+1)}$. We keep making these comparisons until $a_{2k}$ shifts into the first column. The following is to illustrate that $a_{2k}$ shifts during $g_{(k-n+1)}$.

First we compare $a_{1(n+1)}$ to $a_{2n}$. If $a_{2n} > a_{1(n+1)}$ then $a_{2n} = a_{2k}$ since $a_{2k}$ is the shifting entry for $g_{(k-n+1)}$. If not, then $a_{1(n+1)}$ shifts up, and $a_{1(n+2)}$ is compared to $a_{2(n+1)}$. We keep making these comparisons until $a_{2k}$ shifts into the first column. The following is to illustrate that $a_{2k}$ shifts during $g_{(k-n+1)}$.

![Diagram](image)

(2.2)

In the next stage of rectification, $g_{(k-n+2)}$, cell $(1, n-1)$ is the empty box. If there is no shifting entry during this stage then all the entries below cell $(1, n-1)$ shift up in the first column. Recall, $a_{2k}$ is now in the first column. In order for $a_{2k}$ to shift up it must satisfy $a_{2k} \geq a_{2(k-1)}$. But, $a_{2(k-1)} > a_{2k}$ by properties of RSSYT. Thus, there must be a shifting entry of the second column, that is strictly larger than $a_{2k}$, that shifts during $g_{(k-n+2)}$. In diagram (2.3) we have illustrated this next stage of rectification. The illustration shows that no matter when $a_{2k}$ has shifted into the first column, $a_{2k}$ will be compared to $a_{2(k-1)}$ during $g_{(k-n+2)}$ when there is no shifting.
entry of the second column during this stage of rectification. Again, this means there must be a shifting entry of the second column, that is strictly larger than $a_{2k}$, that shifts during $g(k-n+2)$.

\[
\begin{array}{c|cc}
 a_{1(n+1)} & a_{2(n-1)} & a_{2n} \\
 a_{1(n+2)} & \vdots & \\
 \vdots & & \\
 a_{2k} & a_{2(k-1)} & \\
\end{array}
\]

(2.3)

We have now shown once we have a shifting entry from the second column for $g(k-n+1)$, we in fact have a shifting entry from the second column for all of the stages of rectification after the $g(k-n+1)$. We have shown each of these shifting entries is strictly larger than the previous shifting entry, although not necessarily adjacent. Hence, we have a shifting entry for each of $g(k-n+i)$ for $1 \leq i \leq n$. This means we have a total of $n$-many shifting entries from the second column, and $a_{2k}$ is the $n^{th}$ shifting entry.

A similar argument can be made for each subsequent column. When $a_{2k}$ shifts into the first column, if we have a shifting entry from the third column, $a_{3m}$, that shifts during $g(k-n+1)$, we then need to show $a_{3m}$ is the $n^{th}$ largest shifting entry of the third column. To do this, we must consider the next stage of rectification. During $g(k-n+2)$, we know there is a shifting entry from the second column. If we assume there is no shifting entry of the third column during this stage of rectification, then we arrive at the same contradiction. We find $a_{3m}$ is compared to an entry in the third column that is strictly larger than it, and thus that entry must shift into the second column. Hence, if $a_{3m}$ shifts during $g(k-n+1)$, there must be $n$-many shifting entries in the third column, where each of these shifting entries is strictly larger than
their previous shifting entry, and not necessarily adjacent, to the previous shifting entry. This process can be extended to any number of columns in a RSSYT. Notice, by Lemma 2.1.2 this process may terminate. Therefore, the shifting entries during $g_{(k-n+1)}$ are the $n^{th}$ largest shifting cells in each column.

We can order the entries of a RSSYT in such a way that the entries which move columns during rectification are apparent. We call this ordering *eviction*. We define the eviction ordering as follows: Starting with the second column, in which the entries remain in strictly decreasing order, we align the entries remaining in the first column so that they are placed as high as possible on the left side of the second column with the rows weakly decreasing. The entries of the second column that have no entry of the first column beside them are removed and the remaining entries are used to align beside the third column. We continue this process through each of the columns. Below is an example of a RSSYT applying the eviction order.

**Example 2.2.3.** Eviction

\[
\begin{array}{ccc}
8 & 6 & \\
5 & 3 & \\
4 & 1 & \\
5 & 2 & \\
3 & & \\
\end{array}
\to
\begin{array}{ccc}
8 & & \\
5 & 5 & \\
4 & & \\
3 & 2 & \\
& & \\
\end{array}
\Rightarrow
\begin{array}{ccc}
8 & & \\
\end{array}
\]

8 and 4 are shifting entries for the 2\textsuperscript{nd} column.

Next, we align 5 and 2 among the third column.

\[
\to
\begin{array}{ccc}
6 & & \\
5 & 3 & \\
2 & 1 & \\
& & \\
\end{array}
\Rightarrow
\begin{array}{ccc}
6 & & \\
\end{array}
\]

6 is the only shifting entry for the 3\textsuperscript{rd} column.

If there was a fourth column then 3 and 1 would be aligned beside that column.

**Definition 6.** Let $f_k(c_{ij})$ be the entry in the cell $c_{ij}$ at the start of the $k^{th}$ stage of rectification. The entry $f_k(c_{ij})$ has *diagonal dominance over* the entry $f_k(c_{(i-1)m})$ during the $k^{th}$ stage of rectification if $f_k(c_{ij}) > f_k(c_{(i-1)m})$. \[30\]
We now have two lemmas that use the eviction ordering to show which entries are shifting entries of a RSSYT and where these shifting entries are positioned in the CT after applying $\rho^{-1}$.

**Lemma 2.2.4.** In a RSSYT, $T$, an entry $f(c_{ij})$ has no entry $f(c_{(i-1)m})$ positioned to the left of it during eviction if and only if the entry $f(c_{ij})$ is a shifting entry of the $i^{th}$ column.

**Proof.** To prove the reverse implication, we first show $n$ many shifting entries of the $i^{th}$ column, that are all adjacent, each do not have an entry from the $(i-1)^{th}$ column positioned to the left of them during eviction. We then show $n$ many shifting entries of the $i^{th}$ column, that are not all adjacent, each do not have an entry from the $(i-1)^{th}$ column positioned to the left of them during eviction by using the previous argument.

Let $n = k - j + 1$. Let $f_n(c_{ij}), f_{n-1}(c_{(i+1)}), ..., f_p(c_{ik})$ be $n$ many adjacent shifting entries for the $i^{th}$ column, where $f_p(c_{ik})$ is the first entry to shift from the $i^{th}$ column, and $f_n(c_{ij})$ is the last. Notice, we cannot say the first entry to shift is $f_1(c_{ik})$ since the first $(p-1)$ stages of rectification may not have any shifting entries from the $i^{th}$ column, hence we have labeled the first shifting entry of the $i^{th}$ column $f_p(c_{ik})$. Let $f_p(c_{ik})$ have diagonal dominance over $a$ where $a = f_p(c_{(i-1)(k+1)})$. Note, $a$ may be further down in column $(i-1)$ in the original RSSYT, but at the start of the $p^{th}$ stage of rectification $a$ is positioned directly below $f_p(c_{ik})$ in the $(i-1)^{th}$ column. Then we know $f_p(c_{ik}) > a$, and during eviction $a$ cannot move next to or higher than $f_p(c_{ik})$.

The next shifting entry is $f_{(p+1)}(c_{i(k-1)})$. We know $f_{(p+1)}(c_{i(k-1)}) > f_p(c_{ik})$ since these entries are adjacent in the $i^{th}$ column. We also know $f_p(c_{ik}) > a$, since $f_p(c_{ik})$ has diagonal dominance over $a$. Thus, $f_{(p+1)}(c_{i(k-1)}) > f_p(c_{ik}) > a$, which means $a$ could never move next to $f_p(c_{ik})$ to be
compared to \( f_{p+1}(c_{i(k-1)}) \). So, since \( f_{p+1}(c_{i(k-1)}) \) is a shifting entry it must have had diagonal dominance over some entry that is strictly larger than \( a \), since \( a \) is strictly below \( f_p(c_{ik}) \) and cannot be compared to \( f_{p+1}(c_{i(k-1)}) \). Since \( f_{p+1}(c_{i(k-1)}) \) and \( f_p(c_{ik}) \) are adjacent entries and \( f_p(c_{ik}) \) shifts first, we know that \( f_{p+1}(c_{i(k-1)}) \) is compared to \( f_p(c_{ik}) \). This is because in order for \( f_{p+1}(c_{i(k-1)}) \) to be the next shifting entry, \( f_p(c_{ik}) \) must have shifted already, and no other entry would be able to fill that cell afterwards. Thus, since \( f_{p+1}(c_{i(k-1)}) \) is compared to \( f_p(c_{ik}) \) and \( f_{p+1}(c_{i(k-1)}) > f_p(c_{ik}) \), then \( f_{p+1}(c_{i(k-1)}) \) is diagonally dominant over \( f_p(c_{ik}) \). In fact, since \( f(c_{ij}) > f_p(c_{ik}) \) for all \( j < k \), we know the remaining entries larger than \( f_p(c_{ik}) \) in the \( i^{th} \) column would shift over too. During eviction we have already showed \( a \) would remain below \( f_p(c_{ik}) \) in the \((i-1)^{th}\) column; refer diagram (2.4). We just showed all of the entries above \( f_p(c_{ik}) \) in the \( i^{th} \) column are compared to the entry that was positioned below them in the \( i^{th} \) column, and thus no entry of the \((i-1)^{th}\) column will be beside any of the shifting entries of the \( i^{th} \) column during eviction; refer to diagram (2.4).

\[
\begin{array}{c|c|c}
\hline
 f_n(c_{ij}) & \cdots & \hline
 f_{n-1}(c_{i(j+1)}) & \cdots & \hline
 f_p(c_{ik}) & \cdots & \hline
 a & \cdots & \hline
\end{array}
\]

(2.4)

The \((i-1)^{th}\) and \(i^{th}\) column during eviction

We must now look at the case where the shifting entries of the \(i^{th}\) column are not adjacent. Let \( f_p(c_{ik}) \) be the first entry to shift from the \(i^{th}\) column, and assume \( f(c_{i(k-1)}) \) is not a shifting entry. Let \( f_p(c_{ik}) \) have diagonal dominance over \( a \) where
\[ a = f_p(c_{(i-1)(k+1)}). \] Notice again, \( a \) may be further down in column \((i - 1)\) in the original RSSYT, but at the start of the \( p^{th} \) stage of rectification \( a \) is positioned directly below \( f(c_{ik}) \) in the \((i - 1)^{th} \) column. We know from our previous argument that during eviction \( a \) cannot move next to or higher than \( f_p(c_{ik}) \); refer to diagram (2.5).

Since \( f(c_{(i-1)}) \) is not a shifting entry for the \( i^{th} \) column there must be some entry \( b \), during the \( p^{th} \) stage of rectification, where \( b \) is compared to \( f(c_{(i-1)}) \) and \( b \geq f_p(c_{(i-1)}) \). We also know that \( b > a \). This is because the column entries remain in decreasing order and since \( f(c_{(i-1)}) > f_p(c_{ik}) > a \), we know \( a \) can never move next to \( f_p(c_{ik}) \) or any higher, requiring some entry larger than \( a \) to be compared to \( f(c_{(i-1)}) \).

Assume \( f_{(p+1)}(c_{iq}) \) is the next shifting entry of the \( i^{th} \) column. We know \( f_{(p+1)}(c_{iq}) \) must shift over after \( f_p(c_{ik}) \). We also know that \( b \) is positioned beside \( f(c_{(i-1)}) \) after the \( p^{th} \) stage of rectification. If \( f(c_{(i-2)}) = f_{(p+1)}(c_{iq}) \) then we are done, and we know \( f(c_{(i-2)}) \) is the second shifting entry to shift into the \((i - 1)^{th} \) column. If \( f(c_{(i-2)}) \neq f_{(p+1)}(c_{iq}) \), then \( f(c_{(i-2)}) \) cannot shift over since \( f_{(p+1)}(c_{iq}) \) is the next shifting entry of the \( i^{th} \) column. Since \( f(c_{(i-2)}) \) cannot shift over we know that there must be some entry \( c \), during the \((p+1)^{th} \) stage of rectification, where \( c \) is compared to \( f_{(p+1)}(c_{(i-2)}) \) and \( c \geq f_{(p+1)}(c_{(i-2)}) \). We also know that \( c > b \) by the same argument as above. We can make this same argument for every entry above \( f(c_{(i-2)}) \) in the \( i^{th} \) column, until we find \( f_{(p+1)}(c_{iq}) \). Thus, we have shown \( f(c_{(i-1)}), f(c_{(i-2)}) \) and all other non-shifting entries must have an entry beside them during eviction. We have in fact shown the shifting entries will not have an entry beside them during eviction; refer to diagram (2.5). This is because \( a \) cannot move higher than \( f_p(c_{ik}) \), and all the entries larger than \( a \) have been compared to the entries in the \( i^{th} \) column until an entry of the \( i^{th} \) column was large enough to stop these entries from sliding up. Hence,
the shifting entries have created these gaps in the diagonal comparisons, where an entry that was too small in comparison could no longer slide up. Therefore, a shifting entry of the $i^{th}$ column has no entry beside it in the $(i - 1)^{th}$ column during eviction, see diagram (2.5).

\[
\begin{array}{c|c}
\vdots & \vdots \\
\vdots & f(p+1)(c_{iq}) \\
\vdots & \\
f(c_{i(k-2)}) & \vdots \\
f(c_{i(k-1)}) & \\
f_p(c_{ik}) & \\
a & \vdots & \\
\vdots & \\
\end{array}
\]

(2.5)

The $(i - 1)^{th}$ and $i^{th}$ column during eviction

For the forward direction, assume in a RSSYT, $T$, there is only one entry in the $i^{th}$ column, $f(c_{ij})$, that does not have an entry beside it during eviction. Our goal is to show $f(c_{ij})$ is a shifting entry for the $i^{th}$ column. Assume there is no shifting entry of the $(i - 1)^{th}$ column. When we align column $(i - 1)$ to the left of column $i$ during eviction there are no gaps. This is because no entry has shifted from column $(i - 1)$ and therefore, both columns $i$ and $(i - 1)$ are unchanged from the original tableau $T$, which itself has no gaps. However, $f(c_{ij})$ does not have an entry beside it during eviction which is a contradiction. There must be a shifting entry, $f(c_{(i-1)m})$, for the $(i - 1)^{th}$ column. Assume there is no shifting entry of the $i^{th}$ column. This means once $f(c_{(i-1)m})$ has shifted into the $(i - 2)^{th}$ column all of the entries smaller than $f(c_{(i-1)m})$ in the $(i - 1)^{th}$ column must shift up in order to guarantee there is no
shifting entry of the $i^{th}$ column. Thus, during eviction when the $(i - 1)^{th}$ column is
aligned to the left of the $i^{th}$ column there are no gaps. This is because all of the entries
larger than $f(c_{(i-1)m})$ in the $(i - 1)^{th}$ column are unchanged from the original tableau
$T$, which itself has no gaps. Then, once $f(c_{(i-1)m})$ has shifted into the $(i - 2)^{th}$ column
and all the entries smaller than $f(c_{(i-1)m})$ have shifted up in the $(i - 1)^{th}$ column,
those entries that have shifted up in the $(i - 1)^{th}$ column fill in the only gap that
would have been possible. Which means there are no gaps during eviction, which is a
contradiction since $f(c_{ij})$ has no entry beside it during eviction. Thus, there must be
a shifting entry for the $i^{th}$ column. We know by Lemma 2.1.3 a shifting entry must
have diagonal dominance. Assume for contradiction that $f(c_{it})$ is the shifting entry
for the $i^{th}$ column, where $t < j$. We then know that $f(c_{it}) > f(c_{(i-1)p})$ for some entry
in the $(i - 1)^{th}$ column. We also know in order for $f(c_{it})$ to shift into the $(i - 1)^{th}$
column there must be a gap directly to the left of $f(c_{it})$ at some point during the
rectification of $T$. This means during eviction, when the $(i - 1)^{th}$ column is aligned
to the left of the $i^{th}$ column there will be a gap to the left of $f(c_{it})$. Yet, $f(c_{ij})$ is the
only entry of the $i^{th}$ column that does not have an entry beside it during eviction.
Therefore, $f(c_{it})$ for $t < j$ cannot be the shifting entry for the $i^{th}$ column. Assume
for contradiction that $f(c_{it})$ is the shifting entry for the $i^{th}$ column, where $t > j$. We
arrive at the same contradiction above and thus, $f(c_{it}) = f(c_{ij})$. Therefore, $f(c_{ij})$ is
a shifting entry for the $i^{th}$ column.

Assume $n$ entries in the $i^{th}$ column do not have an entry from the $(i - 1)^{th}$ column
beside them during eviction, and that these $n$ entries are shifting entries. We want to
show the $(n + 1)$ entries in the $i^{th}$ column that do not have an entry from the $(i - 1)^{th}$
column beside them during eviction are shifting entries. Given $(n + 1)$ many entries
in the $i^{th}$ column that do not have an entry of the $(i - 1)^{th}$ column beside them during
eviction, we know by the previous lemma that during $g_{(k-i+1)}$ the $i^{th}$ largest shifting
entry shifts columns. Then by our assumption, during the last \( n \) stages of rectification of the \( T \) we have \( n \)-many shifting entries of the \( i^{th} \) column that do not have an entry of the \((i - 1)^{th}\) column beside them during eviction. Next, we perform the last stage of rectification of \( T \). Notice, we are back at our original case. Therefore, by the base case, the \((n + 1)^{th}\) entry of the \( i^{th} \) column that does not have an entry of the \((i - 1)^{th}\) column besides it during eviction is a shifting entry for the \( i^{th} \) column.

Now that we know which entries are shifting entries in a RSSYT, we want to know where these shifting entries are positioned in the CT after applying \( \rho^{-1} \).

**Lemma 2.2.5.** The shifting entries of a RSSYT, \( T \), found via eviction, are positioned in the same rows of the CT, \( U \), as the removed boxes of \( U \).

**Proof.** Let \( T \) be a RSSYT. Consider the shifting entry \( f(c_{2j}) \) of \( T \). Since \( f(c_{2j}) \) is a shifting entry there is no entry from the 1\(^{st}\) column beside it during eviction. Applying \( \rho^{-1} \) to \( T \), ignoring those entries which have been removed from column 1 during eviction, causes \( f(c_{2j}) \) to have nowhere to be placed. This is because \( \rho^{-1} \) inserts each of the entries from the 2\(^{nd}\) column in decreasing order, as high as possible so that the row entries remain weakly decreasing, and during eviction the remaining entries of column 1 have been placed as high as possible in decreasing order among the entries of column 2 so that each row is weakly decreasing. This implies that since \( f(c_{2j}) \) had no entry beside it during eviction, there cannot be any entry from the first column for \( f(c_{2j}) \) to be placed beside by \( \rho^{-1} \). Thus, \( f(c_{2j}) \) must be placed beside one of the removed entries of column 1. Similarly, given a shifting entry, \( f(c_{ij}) \), of the \( i^{th} \) column, those entries of the \((i - 1)^{th}\) column have been positioned as high as possible in the eviction process, and since \( f(c_{ij}) \) has no entry from the \((i - 1)^{th}\) column beside it, \( f(c_{ij}) \) must be placed beside one of the shifting entries of the \((i - 1)^{th}\) column which is placed in the same row as one of the removed boxes in the CT. \( \square \)
We will now use the previous lemmas to prove the following theorem:

**Theorem 2.2.6.** The algorithm $\phi$ gives the rectification of composition tableaux skewed by $1^k$ for any $k$. This process commutes with the rectification of a RSSYT.

*Proof.* By Lemma 2.2.2 we know entries that shift columns during the RSSYT rectification are associated in decreasing order with the stages of rectification. We also know by Lemma 2.2.4 that eviction gives a natural way to find these entries. We then know by Lemma 2.2.5 the shifting entries are positioned in the same rows as the removed boxes in the CT. We can then use $\phi$, which is defined to shift each of these entries exactly one column to the left, with some rearranging and possible bumping, as in the RSSYT, which preserves each of the columns of the RSSYT and CT. Since the columns are preserved, the bijection $\rho$ tells us that there is a unique CT for each RSSYT and vice versa, and hence the RSSYT and CT do commute. Thus, all that is left to check is that $\phi$ produces a CT.

Step 2 of the algorithm requires the first column to remain strictly decreasing. Notice, the column entries remain distinct even after the shifting entries have moved columns. We can see this is true since initially the entries in each column are distinct, and the same entries shift in both the CT and RSSYT, which implies the column entries must be distinct.

Step 3 of algorithm $\phi$ rearranges each column, placing the entries in decreasing order so that the row entries are weakly decreasing. This placement follows directly from the mapping of $\rho^{-1}$. Thus, again our only concern is if there are duplicate entries in any of the columns, since these columns must have distinct entries. We actually have the same argument as before. Since initially the column entries are distinct, and the same entries shift in both the RSSYT and CT, the column entries remain distinct. So, now property 1 and property 2 of a composition tableau are both satisfied. The
insertion of $\phi$ follows directly from the mapping of $\rho^{-1}$, which gives a bijection between RSSYT and CT. Thus, property 3 of a composition tableau is satisfied. We know this is true since any two entries, $a, b$, of the $i^{th}$ column where $a < b$, we know $b$ is placed before $a$ as high as possible so that the row entries are weakly decreasing. If $b$ is positioned below $a$, this means $b$ must have been larger than the entry directly to the right of where $a$ was placed, since $b$ is placed as high as possible. Thus, the third property of a composition tableau is satisfied. Therefore, our algorithm does give a map from a skew composition tableau to a rectified composition tableau that commutes with the RSSYT and the rectified RSSYT respectively.

Notice, if we rectify only one cell in a composition tableau we have reduced to the case in Theorem 2.1.4.

Also, notice that the rectification of a RSSYT skewed by $1^k$ for any $k$ requires $k$ stages of rectification, while the rectification of a composition tableau skewed by $1^k$ for any $k$ requires only one stage of rectification. The rectification of a composition tableau skewed by $1^k$ for any $k$ does not quite give us a way to multiply a composition tableau with shape $1^k$ and a composition tableau of any shape, but this rectification does gives us a step in the right direction. Below is an example of why the rectification of a CT of shape $\alpha/1^k$ does not give us a nice multiplication.
**Example 2.2.7.** An example of why the multiplication of a CT of shape $1^k$ and a CT of any shape does not work.

\[
U = \begin{array}{cccc}
2 \\
3 \\
5 \\
6 \\
\end{array} \ 
\begin{array}{cccc}
& & & \\
& & & \\
& & & 3 & 3 & 3 & 2 \\
& 4 \\
& 6 & 6 & 4 \\
& 7 & 5 & 2 \\
\end{array}
\]

In this example, we can see if we apply $\phi(U)$ the first column will no longer have distinct entries.

We of course would like to find a way to multiply two composition tableaux of any shape. With this generalization we would be able to figure out whether or not composition tableaux have a nice algebraic structure.
Chapter 3: Evacuation of Composition Tableaux

Since evacuation of a reverse semistandard Young tableau is defined by rectification of one cell we now have the ability to define evacuation for composition tableaux using algorithm \( \phi \). In this chapter we provide an algorithm for evacuation of a composition tableau of shape \( \alpha \).

3.1 Algorithm

Let \( n = l(\alpha) + 1 \) and fill a CT with entries less than \( n \). The following is an algorithm for evacuation of a composition tableau of shape \( \alpha \):

Algorithm for evacuation of a composition tableau:

1. Remove the largest entry in column one of the CT. Call the removed entry \( r \).
2. Rectify the tableau.
3. Let \( k = (n - r) \).
4. Insert \( k \) into the set \( S_i \), where \( i \) is the column in the CT that lost an entry during rectification.
5. Repeat steps 1 – 4 until there are no more cells left to rectify in the original composition tableau.
6. Arrange \( S_1 \) in strictly increasing order from top to bottom to form a CT \( T' \).
7. Insert the elements of \( S_i \), starting with \( S_2 \), in decreasing order into \( T' \), so that each entry is inserted in the highest possible position so that the row entries are weakly decreasing from left to right.
Example 3.1.1. Evacuation of a composition tableau of shape $\gamma = (2, 1, 3)$

\[
\begin{array}{cccc}
2 & 1 \\
3 \\
\end{array}
\begin{array}{cccc}
5 & 4 & 2
\end{array}
\]

\[
\begin{array}{cccc}
2 & 1 \\
3 \\
\end{array}
\begin{array}{cccc}
4 & 2
\end{array}
\rightarrow
\begin{array}{cccc}
2 & 2 \\
3 & 1 \\
\end{array}
\begin{array}{cccc}
4
\end{array}
\rightarrow
1 \text{ in } 3^{rd} \text{ column } \rightarrow \quad S_3 = \{1\}
\]

\[
\begin{array}{cccc}
2 & 2 \\
3 & 1 \\
\end{array}
\begin{array}{cccc}
\end{array}
\rightarrow
\begin{array}{cccc}
2 & 2 \\
3 & 1 \\
\end{array}
\begin{array}{cccc}
\end{array}
\rightarrow
2 \text{ in } 1^{st} \text{ column } \rightarrow \quad S_1 = \{2\}
\]

\[
\begin{array}{cccc}
2 & 2 \\
3 & 1 \\
\end{array}
\begin{array}{cccc}
\end{array}
\rightarrow
\begin{array}{cccc}
1 & 2 \\
2 & 2 \\
\end{array}
\rightarrow
3 \text{ in } 2^{nd} \text{ column } \rightarrow \quad S_2 = \{3\}
\]

\[
\begin{array}{cccc}
1 & 2 \\
\end{array}
\rightarrow
\begin{array}{cccc}
1 & 2 \\
\end{array}
\rightarrow
4 \text{ in } 2^{nd} \text{ column } \rightarrow \quad S_2 = \{3, 4\}
\]

\[
\begin{array}{cccc}
1 \\
\end{array}
\rightarrow
\begin{array}{cccc}
1 \\
\end{array}
\rightarrow
4 \text{ in } 1^{st} \text{ column } \rightarrow \quad S_1 = \{2, 4\}
\]

\[
\begin{array}{cccc}
\end{array}
\rightarrow
\begin{array}{cccc}
\end{array}
\rightarrow
5 \text{ in } 1^{st} \text{ column } \rightarrow \quad S_1 = \{2, 4, 5\}
\]

$S_1 = \{2, 4, 5\}, \quad S_2 = \{3, 4\}, \quad S_3 = \{1\} \rightarrow
\begin{array}{cccc}
2 \\
4 & 4 & 1 \\
5 & 3
\end{array}$

Notice, the shape of the CT, $T$, is not necessary the same as the shape of the evacuation of $T$; the evacuation of RSSYT is the same shape as the original RSSYT. However, the shape of the evacuation of $T$ is a rearrangement of the same partition of $T$ since the rows are permuted during the evacuation of $T$.  

41
Theorem 3.1.2. Suppose $T$ is a CT. The algorithm for evacuation of a composition tableau commutes with the evacuation of RSSYT, in the sense that the following diagram commutes:

$$
\begin{array}{ccc}
T & \xrightarrow{\text{evac}} & \text{evac}(T) \\
\rho \downarrow & & \downarrow \rho \\
\rho(T) & \xrightarrow{\text{evac}} & \text{evac. of } \rho(T)
\end{array}
$$

Proof. By construction, #7 in the algorithm for evacuation of CT satisfies the properties of a composition tableau. This is true since we can construct a CT if we know the entries for each column. Our only concern is that the column entries are distinct. So, we must guarantee that each of the $S_i$'s have distinct elements. Since the elements are defined by $(n - r)$ where $r$ is the removed entry and $n = l(\alpha) + 1$, and the column entries are distinct in the original CT and each rectification of the CT, the removed entry $r$ will never be the same for any column. Thus, the elements of each $S_i$ are distinct, and hence, the evacuation of the CT will have distinct column entries.

We know by Theorem 2.1.6 that the same entry is removed from $T$ and from $\rho(T)$ during their respective rectifications. We also know by Theorem 2.1.6 that the rectifications commute, hence the column that is missing a cell after rectification is indeed the same for both $T$ and $\rho(T)$. Thus, the entries that appear in column $j$ of evac($\rho(T)$) are the same entries that appear in column $j$ of evac($T$). Since $\rho$ guarantees a distinct $\rho^{-1}(T)$ for every $T$, and we know the entries in each of the columns of the $T$ and $\rho(T)$ correspond after evacuation, the algorithm for evacuation
of CT commutes with the evacuation of RSSYT.

Below is an illustration of how the evacuation of a CT, $U$, and the evacuation of a RSSYT, $T$, commute. We can see $\text{evac}(U) = \rho^{-1}(\text{evac}(\rho(U))).$

**Example 3.1.3.**

\[
\begin{array}{c|c|c}
U & \text{evac} & T \\
\hline
2 & 1 & 2 \\
3 & \downarrow \rho & 4 \\
5 & 4 & 1 \\
4 & 2 & 5 \\
\end{array}
\]

**Remark 3.1.4.** Evacuation of composition tableaux is an involution. Since the evacuation of composition tableaux commutes with the evacuation of RSSYT, and we know the evacuation of RSSYT is an involution, the evacuation of CT is also an involution.

### 3.2 Applications and Extensions

Evacuation was first used in the theory of the Robinson–Schensted–Knuth (RSK) algorithm [8]. The *RSK algorithm* is a bijection between matrices with non-negative integer entries and pairs $(P, Q)$ of semi-standard Young tableaux of the same shape. The tableau $P$ is the inserting tableau, while the tableau $Q$ is the recording tableau. Given a matrix, we must first find the corresponding *biword* of the matrix by recording in two-line notation, the row of any nonzero entry on the top line, and the column
of that entry on the bottom line. The entry denotes the number of times to write that \textit{biletter}. For consistency with [8] we will use standard Young tableaux, which of course could be translated to reverse standard Young tableaux. The following is an example of a matrix and its corresponding biword:

\textbf{Example 3.2.1.}

\[
\begin{pmatrix}
0 & 1 & 0 \\
1 & 0 & 2 \\
0 & 1 & 0
\end{pmatrix}
\]

\text{biword} = \begin{pmatrix} 1 & 2 & 2 & 2 & 3 \\ 2 & 1 & 3 & 3 & 2 \end{pmatrix} \quad \text{biletters} = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix}

The RSK algorithm then requires insertion of the bottom row into the insertion tableau, \( P \), recording new cells using the bottom row in the recording tableau, \( Q \). The following example gives the process for finding \( P \) and \( Q \) for the above example.

\textbf{Example 3.2.2. RSK algorithm}

\begin{align*}
\text{P (insertion)} & \quad \quad \quad \quad \text{Q (recording)} \\
2 & \quad \quad \quad \quad 1 \\
2 & \quad \quad \quad \quad 2 \\
1 & \quad \quad \quad \quad 1 \\
2 & \quad \quad \quad \quad 2 \quad 1 \\
1 \quad 3 & \quad \quad \quad \quad 1 \quad 2 \\
2 & \quad \quad \quad \quad 2 \quad 1 \\
1 \quad 3 \quad 3 & \quad \quad \quad \quad 1 \quad 2 \quad 2 \\
2 \quad 3 & \quad \quad \quad \quad 2 \quad 3 \\
1 \quad 2 \quad 3 & \quad \quad \quad \quad 1 \quad 2 \quad 2
\end{align*}
Similarly, there is a bijection between matrices with non-negative integer entries and pairs \((P, Q)\) of composition tableaux of rearrangements of the same shape. It is unknown which matrices correspond to pairs \((P, Q)\) of composition tableaux of the same shape. However, evacuation is used in the theory of the RSK algorithm, which gives a bijection between matrices with non-negative integers entries and pairs \((P, Q)\) of SSYT of the same shape, so evacuation of CT may provide the answer to which matrices correspond to pairs of composition tableaux of the same shape. The following theorem is an application where evacuation is used in the RSK algorithm. In order to state the theorem we must first define a few definitions. A reading word, \(w\), of a standard Young tableaux, is a permutation of \(S_n\), thus, we can write \(w = w_1w_2 \cdots w_n\). We define \(w^\# \in S_n\) by \(w^\# = (n+1-w_n) \cdots (n+1-w_2) (n+1-w_1)\). For example, if \(w = 3725164\), then \(w^\# = 4273615\).

**Theorem 3.2.3.** [9] If the RSK algorithm applied to the word \(w\) gives the pair \((P, Q)\), then the RSK algorithm applied to the word \(w^\#\) gives the pair \((\text{evac}(P), \text{evac}(Q))\).

**Example 3.2.4.** Let \(w = 3725164 \Rightarrow w^\# = 4273615\).

\[
(RSK \leftrightarrow \begin{pmatrix}
3 & 7 \\
2 & 5 \\
1 & 4 & 6
\end{pmatrix}, \begin{pmatrix}
5 & 7 \\
3 & 4 \\
1 & 2 & 6
\end{pmatrix})
\]

\[
(RSK \leftrightarrow \begin{pmatrix}
4 & 7 \\
2 & 6 \\
1 & 3 & 5
\end{pmatrix}, \begin{pmatrix}
6 & 7 \\
2 & 4 \\
1 & 3 & 5
\end{pmatrix})
\]
Another application where evacuation is used in the RSK algorithm is a corollary of the previous theorem. Recall, $w^\# = (n + 1 - w_n) \cdots (n + 1 - w_2) (n + 1 - w_1)$. We can write $w^\# = w_0 w w_0$, where $w_0$ denotes the permutation $n n - 1 \cdots 2 1$ [9].

**Corollary 3.2.5.** If the RSK algorithm with a reading word $w = w_1 \cdots w_n$ gives a pair $(P, Q)$, then the RSK algorithm with the reading word $ww_0 = w_n \cdots w_1$ gives the pair $(P^T, evac(Q)^T)$, where $T$ denotes the transpose of the tableau.

To illustrate using the above example, $w = 3725164$ and $ww_0 = 4615273$. We get the following for our pair of tableaux by using the RSK algorithm:

| $P$ (insertion) |
|---|---|
| 4 | |
| 4 6 | 1 |
| 4 1 6 | 1 2 |
| 4 6 1 5 | 3 1 2 |
| 6 4 5 1 2 | 3 4 1 2 |
| 6 4 5 1 2 7 | 3 4 1 2 6 |
| 6 4 5 7 1 2 3 | 3 4 7 1 2 6 |

| $Q$ (recording) |
|---|---|
| 1 | |
| 1 2 | 5 |
| 3 1 2 | 5 4 3 1 2 |
| 3 4 1 2 6 | 5 3 4 7 1 2 6 |

$$(\hat{P}, \hat{Q}) = \begin{pmatrix} 6 & 5 \\ 4 5 7 & 3 4 7 \\ 1 2 3 & 1 2 6 \end{pmatrix}$$
We can see $\hat{P} = P^T$ and $\hat{Q} = \text{evac}(Q)^T$, for $P$ and $Q$ from the previous example.

Evacuation can be generalized to linear extensions of posets and has a nice geometric interpretation connected with flag varieties [8]. A linear extension of a poset is a total order on the elements of the extension, that preserves the order for the respective poset. Evacuation on a linear extension of a poset is defined by promotion of a linear extension. Promotion is the following: Let $P$ be a $p$-element poset. Remove label 1, then replace 1 with the smallest label, $a$, that covers 1. Replace $a$ with the smallest label covering $a$. Continue this process until reaching a maximal element. Label the maximal element with $(p + 1)$. Then decrease every label by 1. Evacuation of a linear extension of a poset, $P$, is the following: apply promotion to $P$, then fix the largest label, $m$. Apply promotion to $P \setminus m$, and then fix the largest label. Repeat until until every label of $P$ has been fixed. Below is an example of evacuation of linear extension of a poset. The circle represents the fixed element in each step of evacuation.
Example 3.2.6. Evacuation of a linear extension

Stanley extended evacuation from standard Young tableaux to arbitrary linear extensions of posets that preserve the ordering of standard Young tableaux [8]. We now have an algorithm for evacuation of composition tableaux, which may provide insight as to whether or not we can extend evacuation to an arbitrary total order that preserves composition tableaux shape.

We will provide several definitions before stating a nice fact. *Self-evacuating linear extensions* are linear extensions, \( f \), of a poset, \( P \), such that \( \text{evac}(f) = f \). An *order ideal* of \( P \) is a subset \( I \) such that if \( t \in I \) and \( s < t \), then \( s \in I \). A *P-domino tableaux* is a chain of order ideals, \( I_i \), of \( P \) where \( I_i - I_{i-1} \) is a two-element chain, for \( 2 \leq i \leq r \).
Stanley proved the number of self-evacuating linear extensions of \( P \) is equal to the number of \( P \)-domino tableaux [8].

There are several directions we can go to further our generalization of rectifying composition tableau. We have looked at rectifying an entry that is not in the first column, rectifying part of a column other than the first column, as well as rectifying part of a row. Several patterns have emerged, but we are still looking for a concrete proof. We have reason to believe if we knew something about rectifying punctured diagrams, then we may have a better approach to show these generalizations. The process of rectifying a punctured diagram is not well-defined since this rectification depends on the order of the punctures in the steps of rectification. Below is an example of a punctured RSSYT diagram.

**Example 3.2.7.** A RSSYT punctured diagram

![Punctured RSSYT diagram]

The reason punctured diagrams may help us is \( \rho^{-1} \) reorders the column entries so that entries that were once adjacent in the RSSYT may no longer be adjacent in the CT, and vice versa. So, if we knew how to approach rectification for punctured RSSYT, then we may be able to apply that process to the rectification of CT. Below is an example of CT, \( U \), with adjacent removed entries and \( \rho(U) \) exhibiting where these removed entries are positioned in the RSSYT, \( T \).
Example 3.2.8. Punctured RSSYT

\[ U \xrightarrow{\rho} T = \]

\[
\begin{array}{cccc}
1 & 1 & & \\
3 & 3 & 3 & 3 \\
4 & & & 1 \\
8 & 7 & 4 & 1 \\
9 & 5 & 2 & \\
\end{array}
\]

Another direction would be to look into fixing a reading order on reverse semi-standard Young tableaux in order to help with rectification of punctured diagrams. The reason this approach may help us is the choice of sliding steps is independent of the choice of inside corners in the rectification of a RSSYT; thus, there may be more than one way to rectify a RSSYT. See the example below for a possible reading order on the inside corners of a RSSYT, \( T \).

Example 3.2.9. A RSSYT of shape \((5, 4, 4, 3, 1)/(3, 2, 1)\), where the inside corners are highlighted in the order in which the sliding steps may be executed.

\[
T = \begin{array}{cccc}
& & & 2 \\
& & & \\
& 1 & & \\
1 & & & \\
\end{array}
\]

This ambiguity causes problems for rectifying CT since the shape skewed out may no longer be connected. So, in the CT it is unknown what order the cells should be considered in the stages of rectification. The following example illustrates an order on the inside corners of a RSSYT, \( T \), and the associated entries of the CT, \( U \), with the respective order on \( T \).
Example 3.2.10. RSSYT, $T$, with an order on the inside corners, and the associated CT, $U$, with the respective order on $T$.

\[ U = \begin{array}{cccc}
2 & 2 & 2 & 2 \\
3 & 3 & 3 & \\
1 & & & \\
\end{array} \quad \rho \quad \begin{array}{c}
\end{array} \quad T = \begin{array}{ccc}
2 & 5 & \\
3 & 3 & 2 \\
1 & 3 & 2 \\
3 & 2 & 1 \\
2 & & \\
\end{array} \]

The above example illustrates a very particular order on the entries in which the sliding steps are applied. We can see this order gives a not-so-obvious order on the respective CT.
Bibliography


52

Vita

Melissa Bechard

Date of Birth:  February 3, 1988

Place of Birth:  Newark, Delaware

Education:

- Master of Arts in Mathematics,
  Wake Forest University, expected May 2012.
  Thesis title:  *Rectification of Composition Tableaux*.
  Advisor:  Sarah K. Mason

- Bachelor of Science in Mathematics, Minor in Dance,
  James Madison University, May 2010.

Experience and Awards:

- Graduate Teaching Assistant, Wake Forest University

- Undergraduate Research, James Madison University, Harrisonburg, VA,
  “Microarray and Meta-analysis of *B. avium*”, collaborative research with a biology student, biology professor, and mathematics professor.

- Research Day Poster Presentation, Wake Forest University, 2012

- MAA Conference Poster Presentation, Virginia State University, 2010

- SUMS Conference Awarded Best Poster Presentation, James Madison University, 2009