SHIFTED $K$-THEORETIC POIRIER-REUTENAUER BIALGEBRA

ADAM KEILTHY, REBECCA PATRIAS, LILLIAN WEBSTER, YINUO ZHANG, SHUQI ZHOU

Abstract. We use shifted $K$-theoretic jeu de taquin to show that the weak $K$-Knuth equivalence relation introduced in [3] is compatible with the shifted Hecke insertion algorithm introduced in [9]. This allows us to define a $K$-theoretic analogue of the shifted Poirier-Reutenauer Hopf bialgebra developed by [6]. From this, we derive a new symmetric function that corresponds to $K$-theory of $OG(n, 2n+1)$ and prove a Littlewood-Richardson rule for these symmetric functions.

1. Introduction

In [11], Poirier and Reutenauer defined a Hopf algebra structure on the $\mathbb{Z}$-span of all standard Young tableaux called the Poirier-Reutenauer Hopf algebra. This Hopf algebra is spanned by elements $T$ indexed by standard Young tableaux, where

$$T = \sum_{P(w) = T} w.$$ 

In other words, $T$ is the formal sum of words that insert into $T$ using the Robinson-Schensted-Knuth insertion algorithm (see [14]) or equivalently, the formal sum of words Knuth equivalent to the row word of $T$.

Using the bialgebra structure and a bialgebra morphism sending $T$ to the Schur function indexed by the shape of $T$, one can prove a version of the Littlewood-Richardson rule for the cohomology rings of Grassmannians. In other words, it yields an explicitly positive description for the structure constants of the cohomology ring in the basis of Schubert classes.

In [6], Jing and Li constructed a version of the Poirier-Reutenauer algebra in the setting of shifted tableaux, using shifted Knuth equivalence due to Sagan [13] thus obtaining a Littlewood-Richardson rule for Schur $P$-functions.

Patrias and Pylyavskyy define a $K$-theoretic version of the Poirier-Reutenauer algebra in [10] using the Hecke insertion algorithm of [2] and the $K$-Knuth equivalence of [3]. Using this bialgebra structure, they obtain Littlewood-Richardson rules for $K$-theory of Grassmannians—structure constants for the product and coproduct of stable Grothendieck polynomials defined in [1]. An important property of the $K$-Knuth equivalence is that, in contrast with Knuth and shifted Knuth equivalence, distinct tableaux of straight shape may be $K$-Knuth equivalent; it is no longer true that each $K$-Knuth class of words contains the reading word of exactly one straight-shaped increasing tableau. To account for this, Patrias and Pylyavskyy make use of the notion of a unique rectification target [3], which is an increasing tableau that is the only tableau in its $K$-Knuth equivalence class.

This paper is concerned with the shifted $K$-theoretic setting. In [4], Clifford, Thomas, and Yong give a description of the Schubert structure constants for the $K$-theory of $OG(n, 2n+1)$, the Grassmannian of isotropic $n$-dimensional planes in $\mathbb{C}^{2n+1}$, using a shifted $K$-theoretic jeu de taquin algorithm. In [3], Buch and Samuel recover this same result in more generality.
However, there is no shifted version of stable Grothendieck polynomials in the literature to use to develop the combinatorial side of the story. For this reason, we define a new type of function that we conjecture to be symmetric.

We define our new functions as a weighted generating function over certain shifted tableaux, called weak set-valued shifted tableaux, which we define below.

**Definition** (Definition 7.4). A weak set-valued shifted tableau is a filling of the boxes of a shifted shape with finite, nonempty multisets of primed and unprimed positive integers with ordering $1' < 1 < 2' < 2 < \ldots$ such that:

1. The smallest number in each box is greater than or equal to the largest number in the box directly to the left of it, if that box exists.
2. The smallest number in each box is greater than or equal to the largest number in the box directly above it, if that box exists.
3. There are no primed entries on the main diagonal.
4. Each unprimed integer appears in at most one box in each column.
5. Each primed integer appears in at most one box in each row.

Given any weak set-valued shifted tableau $T$, we define $x^T$ to be the monomial $\prod_{i \geq 1} x_{a_i} i$, where $a_i$ is the number of occurrences of $i$ and $i'$ in $T$. We can now state the definition for our functions.

**Definition** (Definition 7.6). The shifted weak stable Grothendieck polynomial, $K_\lambda$, is defined by

$$K_\lambda = \sum_T x^T,$$

where the sum is over the set of weak set-valued tableaux $T$ of shape $\lambda$.

We conjecture here that the $K_\lambda$ are symmetric functions. A proof of this conjecture from Zachary Hamaker will appear in the arXiv preprint.

It is not obvious that this choice of definition is the appropriate shifted analogue to the stable Grothendieck polynomials. However, by developing the shifted $K$-theoretic Poirier-Reutenauer algebra, we obtain a Littlewood-Richardson rule for these functions that coincides with the structure constants found by Clifford, Thomas, and Yong in [4] and by Buch and Samuel in [3].

For this version of the Poirier-Regenauer algebra, we use shifted tableau as in [6]. However, we require that the labels are strictly increasing along the rows and down columns. Such tableaux are obtained from the shifted Hecke insertion algorithm given in [9]. The equivalence classes of words for this algebra are given by the weak $K$-Knuth equivalence relation of [3]. In order for the operations to be well-defined, we prove the following result using the shifted $K$-theoretic jeu de taquin introduced in [4].

**Proposition** (Corollary 4.29). If two words $u$ and $v$ have the same shifted Hecke insertion tableau, then $u$ is weak $K$-Knuth equivalent to $v$.

As in the case of the $K$-Knuth relation, a weak $K$-Knuth equivalence class may have multiple corresponding insertion tableaux. However, this result gives us enough to define an algebra.

We define $SKPR$ to be the $\mathbb{R}$-vector space spanned by all $[[h]]$, where $[[h]]$ denotes the formal sum of all words in the weak $K$-Knuth equivalence class of $h$. We give $SKPR$ a
bialgebra structure by defining a compatible product and coproduct similar to the product and coproduct given for the other Poirier-Reutenauer algebras. As in the case of [10], we make use of the notion of unique rectification targets to state our Littlewood-Richardson rules.

**Theorem (Theorem 9.1).** Let $T$ be a URT of shape $\mu$. Then we have

$$K_\lambda K_\mu = \sum_\nu c_{\lambda,\mu}^\nu K_\nu,$$

where $c_{\lambda,\mu}^\nu$ is given by the number of increasing shifted skew tableaux $R$ of shape $\nu/\lambda$ such that $P_{SK}(row(R)) = T$.

We also obtain the following rule for the coproduct.

**Theorem (Theorem 9.2).** Let $T_0$ be a URT of shape $\nu$. Then

$$\Delta(K_\nu) = \sum_{\lambda,\mu} d_{\lambda,\mu}^\nu K_\lambda \otimes K_\mu$$

where $d_{\lambda,\mu}^\nu$ is the number of ordered pairs of increasing shifted tableaux $T', T''$, of shapes $\lambda, \mu$ respectively, such that $P_{SK}(row(T') row(T'')) = T_0$.

The coefficients in these rules are the same as the structure constants found in [4] and the coefficients of [3], which tells us that the weak shifted stable Grothendieck polynomials are the proper polynomials for $K$-theory of $OG(n, 2n + 1)$, as desired.

1.1. **Plan of the paper.** In Section 2, we discuss in more detail the Poirier-Reutenauer Hopf algebra. We discuss the RSK insertion algorithm and the Knuth equivalence relation. We also define the product and coproduct operations and state the Littlewood-Richardson rule for Schur functions.

In Section 3, we discuss variations on the Poirier-Reutenauer. We summarize Jing and Li’s development of the shifted Poirier-Reutenauer algebra in [6] using Sagan-Worley insertion and the shifted Knuth equivalence relation. We then discuss Patrias and Pylyavskyy’s development of the $K$-theoretic Poirier-Reutenauer algebra in [10] using Hecke insertion and the $K$-Knuth equivalence relation.

In Section 4, we recall the shifted Hecke insertion algorithm given in [9]. We define the insertion tableau $P_{SK}(w)$ and the recording tableau $Q_{SK}(w)$ for a given word $w$. We discuss relevant results for shifted Hecke insertion. Then, we state the weak $K$-Knuth equivalence relation of [3] and recall a few characteristics of the equivalence. Finally, using the shifted $K$-theoretic jeu de taquin algorithm introduced in [4] we show that insertion respects the weak $K$-Knuth equivalence.

In Section 5, we define the vector space, product, and coproduct for the shifted $K$-Poirier Reutenauer bialgebra, called $SKPR$. We show that these operations are well-defined and compatible.

In Section 6, we define unique rectification targets as discussed in [3, 4]. We show two ways to create a URT of any shifted shape and demonstrate that the product and coproduct of classes corresponding to URTs are particularly simple to state.

In Section 7, we use descent sets to define a fundamental quasisymmetric function from a word. We define the $K_\lambda$ functions and show that they can be expressed as a sum of these fundamental quasisymmetric functions.
In Section 8, we construct a bialgebra morphism from $SKPR$ to quasisymmetric functions.

In Section 9, we use the morphism from Section 6 to give a Littlewood-Richardson rule for the product and coproduct $K_\lambda$.

2. Poirier-Reutenauer Hopf algebra.

Poirier and Reutenauer defined a Hopf algebra structure on the $\mathbb{Z}$-span of all standard Young tableaux in [11], which is later studied in [5, 12]. We now briefly restate those definitions and illustrate with a few examples, closely following the exposition of [10].

Given a partition $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ satisfying $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$, we associate a collection of boxes arranged in left-justified rows such that row $i$ has $\lambda_i$ boxes. We call this collection a Young diagram of shape $\lambda$. A Young tableau is a filling of these boxes with positive integers such that the entries increase across rows and down columns. We call a Young tableau standard if the entries are elements of $[k] = \{1, 2, \ldots, k\}$ for some $k$ where each element appears exactly once. The tableau shown below is a standard Young tableau of shape $(3, 3, 2, 1)$.

\[
\begin{array}{ccc}
1 & 2 & 5 \\
3 & 6 & 9 \\
4 & 8 & \\
7 & \\
\end{array}
\]

Given two partitions $\lambda$ and $\mu$ such that $\mu \subseteq \lambda$ (i.e. $\lambda_i \geq \mu_i$ for all $i$), we define the skew diagram $\lambda/\mu$ to be the set of boxes of $\lambda$ that are not contained in $\mu$. If $T$ has shape $\lambda/\mu$ where $\mu$ is the empty shape, then we say $T$ is of straight shape. We naturally extend the definitions of Young tableaux and standard Young tableaux to skew shapes. For example, the figure below shows a standard Young tableau of shape $(4, 3, 3, 1)/(3, 2)$.

\[
\begin{array}{cc}
2 & \\
4 & \\
1 & 3 & 6 \\
5 & \\
\end{array}
\]

Given a possibly skew Young tableau $T$, its row reading word, $\mathrm{row}(T)$ is obtained by reading the entries in the rows of $T$ from left to right starting with the bottom row and ending with the top row. For the first standard Young tableau above, $\mathrm{row}(T) = 748369125$ and for the standard Young tableau of skew shape, $\mathrm{row}(T) = 513642$.

Now, consider words with distinct letters on some ordered alphabet $A$. We will usually take $A$ to be a subset of the positive integers. We have the following Knuth relations:

\[acb \approx cab \text{ and } bca \approx bac \text{ whenever } a < b < c.\]

Given two words $w_1$ and $w_2$, we say they are Knuth equivalent, denoted $w_1 \approx w_2$, if $w_1$ can be obtained by a finite sequence of Knuth relations. For example, $53124 \approx 13542$ because $53124 \approx 51324 \approx 15324 \approx 15342 \approx 13542$. 

We say that two tableaux $T_1$ and $T_2$ are equivalent, denoted $T_1 \approx T_2$, if $\text{row}(T_1) \approx \text{row}(T_2)$. For example,

$$T_1 = \begin{array}{ccc}
1 & 2 & 4 \\
3 &  & \\
\end{array} \quad \approx \quad T_2 = \begin{array}{cccc}
2 &  &  & \\
1 & 3 &  & \\
\end{array}$$

From Theorem 5.2.5 of [8], any word with letters exactly $[k]$ is Knuth equivalent to $\text{row}(T)$ for a unique standard Young tableau $T$ of straight shape. To obtain this Young tableau, we use Robinson-Schensted-Knuth insertion, see [14]. For example, $316245 \approx \text{row}(T)$ for $T = \begin{array}{cccc}
1 & 2 & 4 & 5 \\
3 & 6 & & \\
\end{array}$

Given a standard Young tableau $T$, we define

$$T = \sum_{w \approx \text{row}(T)} w.$$ 

In other words, $T$ is the sum of words Knuth equivalent to $\text{row}(T)$. We define $PR$ to be the vector space of $\mathbb{R}$ generated by the set of $T$ for all standard Young tableaux.

Following [11], we describe a bialgebra structure on $PR$. To do this, we must describe product and coproduct operations on $PR$ that are compatible. We begin with two words $w_1$ and $w_2$ in $PR$, where $w_1$ has letters exactly the elements of $[n]$ for some positive integer $n$. Define $w_2[n]$ to be the word obtained by incrementing each letter of $w_2$ by $n$. For example, $3421[3] = 6754$. Now define the product $w_1 \cdot w_2$ to be $w_1 \sqcup w_2[n]$, the usual shuffle product of $w_1$ and $w_2[n]$. For example, $12 \cdot 21 = 12 \sqcup 43 = 1243 + 1423 + 4123 + 1432 + 4132 + 4312$.

For a word $w$ with no repeated letters, we define $\text{std}(w)$ to be the unique word on $|[w]|$ obtained by applying to the letters of $w$ the unique order-preserving injective mapping from the letters of $w$ onto $|[w]|$. For example, $\text{std}(3164) = 2143$. Now, we define the coproduct on $PR$ to be

$$\Delta(w) = \sum \text{std}(u) \otimes \text{std}(v)$$

where the sum is over all words $u$ and $v$ such that $w$ is the concatenation of $u$ and $v$. For example, $\Delta(321) = \emptyset \otimes 321 + 1 \otimes 21 + 1 \otimes 1 + 321 \otimes \emptyset$, where $\emptyset$ denotes the empty word. As shown in [11], the vector space $PR$ with product $\cdot$ and coproduct $\Delta$ extended linearly forms a bialgebra.

2.1. **Littlewood-Richardson Rule.** Using RSK insertion, the $PR$ bialgebra can be related to that of symmetric functions, and hence provide a Littlewood-Richardson rule for multiplication of Schur functions.

To see how this is done, we first state two theorems.

**Theorem 2.1** ([8] Theorem 5.4.3). Let $T_1$ and $T_2$ be two standard Young tableaux. Then we have

$$T_1 \cdot T_2 = \sum_{T \in T(T_1 \sqcup T_2)} T$$

where $T(T_1 \sqcup T_2)$ is the set of standard tableaux $T$ such that $T|n] = T_1$ and $T|[n+1,n+m] \approx T_2$. 

Given a tableau $T$, we define $\overline{T}$ to be the tableau of the same shape as $T$ with reading word $\text{std}(\text{row}(T))$. The following result is the coproduct analogue of Theorem 2.1 and is not difficult to prove with the methods of [8].

**Theorem 2.2.** Let $S$ be a standard Young tableau. We have
\[ \Delta(S) = \sum_{(T',T'') \in T(S)} T' \otimes T'', \]
where $T(S)$ is the set of pairs of tableaux $T', T''$ such that $\text{row}(T') \approx \text{row}(S)$. 

Let $\Lambda$ denote the ring of symmetric functions, and let $s_\lambda$ denote its basis of Schur functions. (See [14] for details.) Then $\Lambda$ has a bialgebra structure, see [17]. We would like a Littlewood-Richardson rule—a combinatorial rule for determining the coefficients $c^\nu_{\lambda,\mu}$ in the decomposition
\[ s_\lambda s_\mu = \sum_\nu c^\nu_{\lambda,\mu} s_\nu. \]
To do this, we define $\psi: PR \to \Lambda$ by
\[ \psi(T) = s_{\lambda(T)} \]
where $\lambda(T)$ denotes the shape of $T$.

**Theorem 2.3** ([8], Theorem 5.4.5). The map $\psi$ is a bialgebra morphism.

Now, we apply $\psi$ to the equalities in Theorem 2.1 and Theorem 2.2 to get the following two versions of the Littlewood-Richardson rule.

**Corollary 2.4** ([14], Theorem A1.3.1). Let $T$ be a standard Young tableau of shape $\mu$. Then the coefficient $c^\nu_{\lambda,\mu}$ is equal to the number of standard Young tableaux $R$ of skew shape $\nu/\lambda$ such that $\text{row}(R) \approx \text{row}(T)$.

**Corollary 2.5** ([14], Theorem 5.4.5). Let $S$ be a standard Young tableau of shape $\nu$. Then the coefficient $c^\nu_{\lambda,\mu}$ in the decomposition is equal to the number of standard Young tableaux $R$ of skew shape $\lambda \oplus \mu$ such that $\text{row}(R) \approx S$.


Generalizations of the approach outlined above have been used to provide similar rules for other classes of symmetric functions. In [6], Jing and Li develop the shifted Poirier-Reutenauer algebra, which provides a rule for multiplication of $P$-Schur functions.

In this setting, we associate a strict partition $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ where $\lambda_1 > \lambda_2 > \cdots > \lambda_k$ with an array of boxes in which the $i$th row has $\lambda_i$ boxes and is indented $i - 1$ units, called a shifted shape. A shifted tableau is a filling of a shifted shape with positive integers such that the rows are weakly increasing and the columns are strictly increasing. A shifted tableau of shape $\lambda$ is called standard if it contains each of the entries $1, \ldots, k$ exactly once for some positive integer $k$. Standard shifted tableau play the role of standard Young tableau for this algebra. The notion of reading words extends similarly to shifted shapes. The tableau on the left is a shifted tableau of shape $(4, 2, 1)$ with reading word 7351244 and the tableau on the right is a standard shifted tableau of shape $(4, 2, 1)$ with reading word 7361245.
As before, we consider words with distinct letters on some ordered alphabet. Sagan [13] and Worley [16] independently developed an insertion algorithm, Sagan-Worley insertion that produces standard shifted tableaux from words with letters exactly \([k]\). For a word \(w\), let \(P_{SW}(w)\) denote the Sagan-Worley insertion tableau of \(w\). This insertion algorithm corresponds to a generalization of the Knuth relations called the shifted Knuth equivalence relation, which is given by two local equivalences:

1. \(acb \equiv cab\) and \(bac \equiv bca\) if \(a < b < c\)
2. \(ab \equiv ba\) if \(a, b\) are the first two letters of the word.

In [13][Theorem 7.2], Sagan proves that the insertion algorithm characterizes shifted Knuth equivalent words. In other words, \(P_{SW}(u) = P_{SW}(v)\) if and only if \(u\) can be obtained from \(v\) by a finite number of applications of the equivalences above.

Given a shifted standard tableau \(T\), we define \(T_{\text{shifted}}\) to be the sum of words shifted Knuth equivalent to \(\text{row}(T)\). Jing and Li define \(SPR\) to be the vector space generated by the set of \(T_{\text{shifted}}\) for all shifted standard tableaux. Using the same product and coproduct operations as \(PR\), \(SPR\) is a bialgebra.

In [6], Jing and Li give a map \(\pi\) from \(SPR\) to symmetric functions which can be given by \(\pi(T_{\text{shifted}}) = P_\lambda\) where \(T\) is of shape \(\lambda\) and \(P_\lambda\) denotes the Schur \(P\)-function. From this, they derive the shifted Littlewood-Richardson rule for multiplication of Schur \(P\)-functions.

Similarly, in [10], Patrias and Pylyavskyy develop the \(K\)-theoretic Poirier-Reutenauer algebra using Hecke insertion, providing a rule for multiplication of the stable Grothendieck polynomials [1].

An increasing tableau is a Young diagram filled with positive integer such that the rows and columns are strictly increasing. Increasing tableaux serve as the analogue to a standard Young tableaux in this setting. Reading words are defined precisely as for Young tableaux. The following is an increasing tableau of shape \((3, 2)\).

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

Now, we consider the reading words of these tableaux. The Hecke insertion algorithm described in [2] produces increasing tableau and corresponds to the \(K\)-Knuth equivalence relation in [3], given by the following three local rules:

1. \(acb \equiv cab\) and \(bac \equiv bca\) if \(a < b < c\),
2. \(aa \equiv a\), and
3. \(aba \equiv bab\).

We say that two tableaux are \(K\)-Knuth equivalent if their row words are \(K\)-Knuth equivalent.

An important property of the \(K\)-Knuth equivalence is that, in contrast with Knuth and shifted Knuth equivalence, distinct tableaux of straight shape may be \(K\)-Knuth equivalent; it is no longer true that each \(K\)-Knuth class of words contains the reading word of exactly one straight -shaped increasing tableau. For example, the tableaux below are \(K\)-Knuth equivalent.
To account for this, Patrias and Pylyavskyy make use of the notion of a unique rectification target [3], which is an increasing tableau that is the only tableau in its $K$-Knuth equivalence class. We will also make use of this notion in developing our Littlewood-Richardson rule.

As a consequence, we may no longer define elements of the appropriate vector space as sums of words inserting into an increasing tableau $T$; this will not result in a closed product and coproduct as shown in [10]. Instead, we must define elements of the vector space using the $K$-Knuth relations. For a word $h$, we define

$$[[h]] = \sum_{h' \equiv h} h'.$$

That is, $[[h]]$ is the sum of all words $K$-Knuth equivalent to $h$. Patrias and Pylyavskyy define $KPR$ to be the vector space spanned by all such sums. The coproduct and product on this vector space are the natural extensions of those used for $PR$ and $SPR$, giving $KPR$ a bialgebra structure.

In [10], Patrias and Pylyavskyy give a bialgebra morphism $\phi$ from $KPR$ to the ring of symmetric functions. From this, they derive a Littlewood-Richardson rule for the multiplication of stable Grothendieck polynomials.

4. Shifted Hecke Insertion and Weak $K$-Knuth Equivalence

In [2], the authors show that the Hecke insertion algorithm respects the $K$-Knuth equivalence of words using the reverse Hecke insertion algorithm. Here, we show that the shifted analogue of Hecke insertion given in [9] respects the weak $K$-Knuth equivalence given in [3]. We start by introducing increasing shifted tableaux and shifted Hecke insertion.

4.1. Increasing shifted tableaux. Given a strict partition $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$, where by definition $\lambda_1 > \lambda_2 > \ldots \lambda_k$, the corresponding shifted shape is the arrangement of cells into $k$ rows, where row $i$ contains $\lambda_i$ cells and is indented $i - 1$ spaces. The shifted shape for strict partition $(6, 4, 3, 1)$ is shown below.

We will identify strict partitions with shifted shapes.

Given two strict partitions $\lambda$ and $\mu$ such that $\mu \subseteq \lambda$ (i.e. $\lambda_i \geq \mu_i$ for all parts $\mu_i$), we define the shifted skew shape $\lambda/\mu$ to be the set of boxes in $\lambda$ that do not belong to $\mu$. For example, the shape $(5, 3, 1)/(3, 1)$ is shown below.
We will fill each box of a shifted shape with a positive integer. We call a filling *increasing* if the labels strictly increase left to right along rows and top to bottom down columns. A shifted shape with an increasing filling is called an *increasing shifted tableau*. For example, the first two shifted shapes with fillings shown below are examples of increasing shifted tableaux, while the third one is not. Throughout, we will refer to shifted increasing tableaux simply as tableaux.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

|   | 1 | 3 | 6 | 7 |
|---|---|---|---|
| 4 | 7 |

|   | 1 | 2 | 4 | 6 |
|---|---|---|---|
| 3 | 4 |
| 5 |

For any tableau $T$, the *row word* of $T$, $\text{row}(T)$, is the word obtained by reading the entries of $T$ from left to right across rows starting with the bottom row and moving upward. For example, the row words for the three tableaux above are $635124$, $8471367$, and $5341246$.

**Lemma 4.1.** There are only finitely many increasing shifted tableaux filled with a given finite alphabet.

**Proof.** If the alphabet has $n$ letters, each row and column of the tableau can be no longer than $n$. □

Given an increasing shifted (possibly skew) tableau $T$, its *row reading word*, $\text{row}(T)$, is obtained by reading the entries in the rows of $T$ from left to right starting with the bottom row and ending with the top row. For the first increasing shifted tableau shown above, $\text{row}(T) = 635124$ and for the second, $\text{row}(T') = 8471367$.

### 4.2. Shifted Hecke Insertion

We now restate the rules for shifted Hecke insertion given in [9]. Hecke insertion gives a way to associate a word on positive integers with an increasing shifted tableau. It is a natural shifted analogue of Hecke insertion [2] and a natural $K$-theoretic analogue of Sagan-Worley insertion [13]. From this point on, “insertion” will always refer to shifted Hecke insertion unless stated otherwise.

First, we describe how to insert a positive integer $x$ to a given shifted increasing tableau $T$. We start with inserting $x$ to the first row of $T$. For each insertion, we assign a box to record where the insertion terminates. This notion appears when we define the recording tableau in Section 4.3. Our exposition will closely follow that of [9].

The rules for inserting $x$ to $T$ are as follows:

1. If $x$ is weakly larger than all integers in the row (resp. column) and adjoining $x$ to the end of the row (resp. column) results in an increasing tableau $T'$, then $T'$ is the resulting tableau. We say the insertion terminates at the new box.

**Example 4.2.** Inserting 5 into the first row of the left tableau gives us the right tableau below. The insertion terminates at box $(1, 4)$.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

|   | 1 | 2 | 4 | 5 |
|---|---|---|---|
| 3 | 5 |
| 6 |

2. If $x$ is weakly larger than all integers in the row (resp. column) and adjoining $x$ to the end of the row (resp. column) does not result in an increasing tableau, then $T' = T$. 

|   | 1 | 2 | 4 | 5 |
|---|---|---|---|
| 3 | 5 |
| 6 |
If $x$ is row inserted into a nonempty row, we say the insertion terminated at the box at the bottom of the column containing the rightmost box of this row. If $x$ is row inserted into an empty row, we say that the insertion terminated at the rightmost box of the previous row. If $x$ is column inserted, we say the insertion terminated at the rightmost box of the row containing the bottom box of the column $x$ could not be added to.

Example 4.3. Adjoining 5 to the second row of the left tableau does not result in an increasing tableau. Thus the insertion of 5 into the tableaux on the left terminates at (2,3) and gives us the tableau on the right.

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

Adjoining 2 to the (empty) second row of the tableau below does not result in an increasing tableau. The insertion ending in this failed row insertion terminates at (1,3).

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

Adjoining 3 to the end of the third column of the left tableau does not result in an increasing tableau. This insertion terminates at (1,3).

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

For the last two rules, suppose the row (resp. column) contains a box with label strictly larger than $x$, and let $y$ be the smallest such box.

(3) If replacing $y$ with $x$ results in an increasing tableau, then replace $y$ with $x$. In this case, $y$ is the output integer. If $x$ was inserted into a column or if $y$ was on the main diagonal, proceed to insert all future output integers into the next column to the right. If $x$ was inserted into a row and $y$ was not on the main diagonal, then insert $y$ into the row below.

Example 4.4. Row inserting 3 into the first row of the left tableau results in the tableau below on the right. This insertion terminates at (2,3)

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

Insert 3 into the second row of the left tableau below. Replace 4 with 3, and column insert 4 into the third column. The resulting tableau is on the right.

\[
\begin{array}{cccc}
1 & 2 & 3 & 5 \\
4 & 6 \\
8 \\
\end{array}
\quad \begin{array}{cccc}
1 & 2 & 3 & 5 \\
3 & 4 & 6 \\
8 \\
\end{array}
\]

(4) If replacing $y$ with $x$ does not result in an increasing tableau, then do not change the row (resp. column). In this case, $y$ is the output integer. If $x$ was inserted into a column or if $y$ was on the main diagonal, proceed to insert all future output integers into the next column to the right. If $x$ was inserted into a row, then insert $y$ into the row below.
Example 4.5. Replacing 5 with 3 in the first row of the tableau below does not create an increasing tableau. So, row insertion of 3 into the first row produces output integer 5, which is inserted into the second row. Replacing 6 with 5 in the second row does not create an increasing tableau. This produces output integer 6. Adjoining 6 to the third row does not result in an increasing tableau. Thus inserting 3 into the tableau below does not change the tableau. This insertion terminates at (2, 3).

\[
\begin{array}{ccc}
1 & 3 & 5 \\
4 & 6
\end{array}
\]

For any given word \( w = w_1w_2 \cdots w_n \), we define the \textit{insertion tableau} of \( w \), \( P_{SK}(w) \), to be \((\cdots ((\emptyset \xleftarrow{SK} w_1) \xleftarrow{SK} w_2) \cdots \xleftarrow{SK} w_n)\), where \( \emptyset \) denotes the empty shape and \( \xleftarrow{SK} \) denotes the insertion of a single letter.

Example 4.6. The sequence of tableaux obtained while computing \( P_{SK}(2115432) \) is shown below. The tableau on the right is \( P_{SK}(2115432) \).

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

For any interval \( I \), we define \( T|_I \) to be the tableau obtained from \( T \) by deleting all boxes with labels not in \( I \) and \( w|_I \) to be the word obtained from \( w \) by deleting all letters not in \( I \). Also, we use \([k]\) to denote the interval \([1, k] = \{1, 2, \ldots, k\}\). The following simple lemma will be useful.

Lemma 4.7. If \( P_{SK}(w) = T \), then \( P_{SK}(w)|_[k] = P_{SK}(w|_[k]) = T|_[k] \).

Proof. This follows from the insertion rules and the observation that letters greater than \( k \) never affect the placement or number of letters in \([1, 2, \ldots, k]\). \( \square \)

4.3. Recording tableaux. We next describe how to construct the recording tableau for shifted Hecke insertion of a word \( w \). We need the following definition.

Definition 4.8. [9, Definition 5.16] A \textit{set-valued shifted tableau} is defined to be a filling of the boxes of a shifted shape with finite, nonempty subsets of primed and unprimed positive integers such that:

1) The smallest number in each box is greater than or equal to the largest number in the box directly to the left of it, if that box exists.
2) The smallest number in each box is greater than or equal to the largest number in the box directly above it, if that box exists.
3) Any positive integer appears at most once, either primed or unprimed, but not both.
4) There are no primed entries on the main diagonal.

A set-valued shifted tableau is called \textit{standard} if the set of labels is exactly \([n]\) for some \( n \), each appearing either primed or unprimed exactly once. The tableaux below are set-valued shifted tableaux. The tableau on the right is standard.

\[
\begin{array}{ccc}
1 & 2 & 3' & 6 \\
4 & 8'9 & 5
\end{array}
\]
The recording tableau of a word \( w = w_1w_2\ldots w_n \), denoted \( Q_{SK}(w) \), is a standard set-valued shifted tableau that records where the insertion of each letter of \( w \) terminates. We define it inductively. Start with \( Q_{SK}(\emptyset) = \emptyset \). If the insertion of \( w_k \) added a new box to \( P_{SK}(w_1w_2\ldots w_{k-1}) \), then add the same box with label \( k \) (\( k' \) if this box was added by column insertion) to \( Q_{SK}(w_1w_2\ldots w_{k-1}) \). If \( w_k \) did not change the shape of \( P_{SK}(w_1w_2\ldots w_{k-1}) \), obtain \( Q_{SK}(w_1w_2\ldots w_k) \) from \( Q_{SK}(w_1\ldots w_{k-1}) \) by adding the label \( k \) (\( k' \) if it ended with column insertion) to the box where the insertion terminated. If insertion terminated when a letter failed to insert into an empty row, label the box where the insertion terminated \( k' \).

**Example 4.9.** Let \( w = 451132 \). We insert \( w \) letter by letter, writing the insertion tableau at each step in the top row and the recording tableau at each step in the bottom row.

\[
\begin{array}{cccc}
4 & 4 & 5 & \mid 1 & 4 & 5 & \mid 1 & 4 & 5 & \mid 1 & 3 & 5 & 4 & \mid 1 & 2 & 4 & 5 & = P_{SK}(w) \\
\hline
1 & 1 & 2 & \mid 1 & 2 & 3' & \mid 1 & 2 & 3' & 4 & \mid 1 & 2 & 3' & 4 & 5 & = Q_{SK}(w)
\end{array}
\]

In [9], a reverse insertion procedure is defined so that for each pair \((P_{SK}(w), Q_{SK}(w))\), the word \( w \) can be recovered. See [9] for details on reverse shifted Hecke insertion. This procedure gives the following result:

**Theorem 4.10.** [9, Theorem 5.19] There is a bijection between pairs consisting of an increasing shifted tableau and a standard set-valued shifted tableau of the same shape, \((P, Q)\), and words of positive integers, where the word \( w \) corresponds to the pair \((P_{SK}(w), Q_{SK}(w))\).

4.4. **Weak \( K \)-Knuth equivalence.** Recall that the Knuth equivalence relations allow us to determine which words have the same Robinson-Schensted-Knuth insertion tableau [14, Theorem A1.1.4]. We next define the shifted \( K \)-theoretic analogue called weak \( K \)-Knuth equivalence. As we will see in Example 4.30, weak \( K \)-Knuth equivalence is a necessary but not sufficient condition for two words to have the same shifted Hecke insertion tableau.

**Definition 4.11.** [3, Definition 7.6] Define the weak \( K \)-Knuth equivalence relation on the alphabet \( \{1,2,3,\ldots\} \), denoted by \( \equiv \), as the symmetric transitive closure of the following relations, where \( u \) and \( v \) are (possibly empty) words of positive integers, and \( a, b, \) and \( c \) are distinct positive integers:

1. \( (u,a,a,v) \equiv (u,a,v) \),
2. \( (u,a,b,a,v) \equiv (u,b,a,b,v) \),
3. \( (u,a,b,c,v) \equiv (u,a,c,b,v) \) if \( b < a < c \),
4. \( (u,b,c,a,v) \equiv (u,c,b,a,v) \) if \( b < a < c \), and
5. \( (a,b,u) \equiv (b,a,u) \).

We say that two words, \( w \) and \( w' \), are weak \( K \)-Knuth equivalent and write \( w \equiv w' \) if \( w' \) can be obtained from \( w \) by a finite sequence of weak \( K \)-Knuth equivalence relations.

**Example 4.12.** For instance, \( 1243 \equiv 442143 \) because

\[
1243 \equiv 2143 \equiv 21143 \equiv 21413 \equiv 42143 \equiv 442143.
\]

Note that in contrast to Knuth equivalence, each weak \( K \)-Knuth equivalence class has infinitely many elements and contains words of arbitrary length.

We will need the following lemma, which follows easily from the weak \( K \)-Knuth relations.
Lemma 4.13. If \( w \equiv w' \), then \( w|_I \equiv w'|_I \) for any interval \( I \), where \( w|_I \) is defined to be the subword of \( w \) restricted to the interval \( I \).

We end this section by extending the definition of weak \( K \)-Knuth equivalence to tableaux. Given shifted increasing tableaux \( T \) and \( T' \), we say that \( T \) is weak \( K \)-Knuth equivalent to \( T' \), denoted \( T \equiv T' \) if \( \text{row}(T) \equiv \text{row}(T') \).

4.5. Shifted \( K \)-theoretic jeu de taquin. We next describe the shifted \( K \)-theoretic jeu de taquin algorithm or shifted \( K \)-jdt introduced by Clifford, Thomas, and Yong in [4].

Definition 4.14. Given shifted skew shape \( \lambda/\mu \), where \( \mu \subset \lambda \) are two strict partitions, we say that two boxes in \( \lambda/\mu \) are adjacent if they share a common edge. We call a box in \( \lambda/\mu \) maximal if it has no adjacent boxes to the east or south and minimal if it has no adjacent boxes to the west or north.

For the following definition, let \( T(\alpha) \) denote the label of box \( \alpha \) in tableau \( T \).

Definition 4.15. [15, Section 4] We define the action of the \( K \)-theoretic switch operator \((i,j)\) on a tableau \( T \) as follows:

\[
((i,j)T)(\alpha) = \begin{cases} 
  j & \text{if } T(\alpha) = i \text{ and } T(\beta) = j \text{ for some box } \beta \text{ adjacent to } \alpha; \\
  i & \text{if } T(\alpha) = j \text{ and } T(\beta) = i \text{ for some box } \beta \text{ adjacent to } \alpha; \\
  T(\alpha) & \text{otherwise.}
\end{cases}
\]

In other words, we swap the labels of boxes labeled by \( i \) and the adjacent box (or boxes) labeled with entry \( j \). If no box labeled with entry \( j \) is adjacent to a box labeled by entry \( i \), we do nothing.

For the example of the switch operator below and for the definitions to follow, we will allow boxes to be labeled with the symbol \( \circ \) in addition to labels in \( \mathbb{N} \).

Example 4.16. Two examples of swaps may be found below.

\[
\begin{array}{ccc}
\circ & 2 & 3 \\
\circ & 2 & \circ \\
\end{array}
\xrightarrow{(2,\circ)}
\begin{array}{ccc}
2 & \circ & \circ \\
2 & \circ & 3 \\
\end{array}
\xrightarrow{(3,\circ)}
\begin{array}{ccc}
\circ & 2 & 3 \\
\circ & 2 & \circ \\
\end{array}
\]

The next step in defining shifted \( K \)-jdt is to define a shifted \( K \)-jdt slide. Let \( \Lambda \) denote the union of all shifted shapes \( \nu \): \( \Lambda = \bigcup \nu \). Informally, a complete \( K \)-jdt slide on skew tableau \( T \) consists of choosing an initial set of boxes in \( \Lambda/T \) to mark with \( \circ \) followed by a series of the switches described above. Boxes marked with \( \circ \) begin either all weakly northwest of \( T \) or all weakly southeast of \( T \) and will move across the \( T \) as swaps are performed during \( K \)-jdt. Each K-jdt move will be categorized as either a forward slide (where the boxes with \( \circ \) begin weakly northwest of \( T \)) or a reverse slide (where the boxes with \( \circ \) begin weakly southeast of \( T \)).

Definition 4.17. Let \( T \) be an increasing tableau of shifted skew shape \( \lambda/\mu \) with values in the interval \([a,b] \subset \mathbb{N} \). Let \( C \) be a subset of the maximal boxes of \( \mu \) and label each box in \( C \) with \( \circ \). The forward slide of \( T \) starting from \( C \), \( kjdt_C(T) \), is defined by

\[
kjdt_C(T) = (b,\circ)(b - 1,\circ) \ldots (a + 1,\circ)(a,\circ)(T)
\]
Similarly, if $\hat{C}$ is a subset of the minimal boxes of $\Lambda/\lambda$ labeled by $\circ$, the reverse slide of $T$ starting from $\hat{C}$, $k jdt_{\hat{C}}(T)$, is defined by

$$k jdt_{\hat{C}}(T) = (a, \circ)(a + 1, \circ) \ldots (b - 1, \circ)(b, \circ)(T)$$

where $\Lambda = \bigcup \nu$ is the union of all shifted shapes $\nu$.

**Example 4.18.** The example below gives shows one forward $K$-jdt slide and one reverse $K$-jdt slide.

Notice that one can use forward slides to transform a skew shape into a straight shape and reverse slides to do the opposite.

**Definition 4.19.** A *shifted $K$-rectification* of an increasing shifted skew tableau $T$ is any shifted tableau that can be obtained from $T$ by a series of forward slides.

It was shown in [4] that the shifted $K$-rectification of an increasing shifted tableau will be an increasing shifted tableau. However, it is not that case that the shifted $K$-rectification of a tableau is unique; the way in which we choose boxes to label with $\circ$ for each slide, called the *recification order*, may lead to different rectifications. See [4] for details. For this reason, when talking about $K$-rectification, it is often useful to communicate the subsets $C$ chosen for each slide. We will do so by marking such boxes with an underlined version of a positive integer as in [15]. The underlined boxes are labeled in reverse chronological order so that subset $C$ for the last shifted $K$-jdt slide is indicated by boxes labeled $1$, the second to last by $2$, and so on. See Example 4.20

We next describe a particular shifted $K$-rectification of a shifted skew tableau $T$ that we will use to establish the link between shifted Hecke insertion and the weak $K$-Knuth equivalence. As a preliminary step, we define the *superstandard tableau* of shifted shape $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ to be the tableau with first row labeled by $1, 2, \ldots, \lambda_1$, second row by $\lambda_1 + 1, \lambda_1 + 2, \ldots, \lambda_2$, and so on. The tableau below is superstandard.

Given an increasing shifted skew tableau $T$ of shape $\lambda/\mu$ with labels in the interval $[a, q]$, we define a rectification order by filling the shape $\mu$ so the resulting tableau is superstandard with entries $[1, p]$, where $p = |\mu|$. We refer to this marking order as the *superstandard rectification order* and refer to the resulting tableau as the *superstandard $K$-rectification*. To obtain the corresponding $K$-rectification, we perform the following sequence of switch operations, from left to right:
We refer to this sequence of pairs as the standard switch sequence.

Example 4.20. Given a skew increasing tableau

\[
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
5 & 6 & 2 & 3 \\
1 & 4 & & \\
\end{array}
\]

we obtain the shifted superstandard\(^{K}\)-rectification as follows:

\[
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
5 & 6 & 2 & 3 \\
1 & 4 & & \\
\end{array} \rightarrow \begin{array}{cccc}
1 & 2 & 3 & 4 \\
1 & 2 & 3 & 5 \\
4 & 6 & & \\
\end{array} \rightarrow \begin{array}{cccc}
1 & 2 & 3 & 4 \\
1 & 2 & 3 & 5 \\
4 & 6 & & \\
\end{array} \rightarrow \begin{array}{cccc}
1 & 2 & 3 & 4 \\
1 & 2 & 3 & 5 \\
4 & 6 & & \\
\end{array} \rightarrow \begin{array}{cccc}
1 & 2 & 3 & 4 \\
1 & 2 & 3 & 5 \\
4 & 6 & & \\
\end{array} \rightarrow \begin{array}{cccc}
1 & 2 & 3 & 4 \\
1 & 2 & 3 & 5 \\
4 & 6 & & \\
\end{array}
\]

The sequence of nontrivial switch operators used in this shifted \(^{K}\)-rectification is

\[(6, 1), (6, 4), (5, 1), (5, 2), (5, 3), (4, 1), (3, 1), (3, 3), (2, 1), (2, 2), (2, 4), (1, 1), (1, 2), (1, 3), (1, 4)\]

We see that the superstandard \(^{K}\)-rectification of \(T\) is

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

According to our definition, we must perform the switches in the order determined by the standard switch sequence. However, it turns out that there is some flexibility in this ordering. This motivates the following definition.

Definition 4.21. [15, Section 4] A viable switch sequence is a sequence of switch operators, with the following properties:

1. every switch \((i, j)\) occurs exactly once, for \(1 \leq i \leq p\) and \(1 \leq j \leq q\);
2. for any \(1 \leq i \leq p\), the pairs \((i, 1), \ldots, (i, q)\) occur in that relative order;
3. for any \(1 \leq j \leq q\), the pairs \((p, j), \ldots, (1, j)\) occur in that relative order.

Lemma 4.22. [4, Lemma 4.4] If \(i \neq j\) and \(r \neq s\) then the switch operators \((i, r)\) and \((j, s)\) commute.

Proposition 4.23. Any viable switch sequence can be used to calculate shifted \(^{K}\)-rectification.

Proof. It is straightforward to show that any viable switch sequence can be obtained from the standard switch sequence by repeated applications of Lemma 4.22. \qed

For any word \(w = w_1 \ldots w_n\), let \(T_w\) denote the shifted skew tableaux consisting of \(n\) boxes on the antidiagonal with reading word \(w\). The next theorem explains the relationship between shifted Hecke insertion and shifted \(^{K}\)-jdt.

Theorem 4.24. Shifted Hecke insertion of a word \(w\) gives the same tableau as shifted superstandard \(^{K}\)-theoretic jeu de taquin rectification of the antidiagonal tableau of \(T_w\).
Proof. We will closely follow the structure of the proof of [15, Theorem 4.2].

We proceed by induction. Let $P$ be the shifted tableau that results from shifted $K$-rectification of the antidiagonal tableau of $w_1w_2\ldots w_{n-1}$. By induction, $P$ was obtained by a viable switch sequence and is the same tableau as shifted Hecke insertion of $w_1w_2\ldots w_{n-1}$.

Now, shifted $K$-rectification of $T_w$ with this switch sequence gives the shifted shape $P^*\begin{array}{c}w_n\end{array}$, on the right below:

It now remains to show that the shifted $K$-rectification on $P^*\begin{array}{c}w_n\end{array}$ using the remaining underlined entries results in the same tableau as shifted Hecke insertion $P \leftarrow^{SK} w_n$.

Notice that any viable switch sequence for the underlined entries 1, $\ldots$, $t$ concatenated with the switch sequence we have by induction will give a viable switch sequence for $T_w$. Thus, by Proposition 4.23 once we construct a viable switch sequence such that the resulting shifted $K$-rectification is the same as the tableau obtained by insertion $P \leftarrow^{SK} w_n$, we complete the proof.

For $i \geq 1$, let $y_i$ be the letter that is inserted into row (resp column) $i$ in the shifted Hecke insertion of $w_n$ into $P$, so $y_1 = w_n$. Row $i$ is defined to be in normal form if:

(1) When $y_i$ has not yet moved from its initial position, the $i$-th row is of the form

\[
\begin{array}{cccccc}
t_1 & t_2 & \cdots & t_k & y_i & t_{k+1} & \cdots & t_m
\end{array}
\]

(2) After several switch operations, the $i$-th row of the tableau is of the form

\[
\begin{array}{cccccc}
t_1 & x_1 & \cdots & x_n & y_i & x_{n+1} & \cdots & x_{m-1}
\end{array}
\]

In other words, if we start with a row that has only one non-underlined entry and end with a row that has only one underlined entry in the leftmost box of the row, then we say this row is in normal form.

Starting with tableau $P^*\begin{array}{c}w_n\end{array}$ and row 1 in normal form, we construct a viable switch sequence involving local applications of the cases below to get the shifted $K$-rectification. There are four cases for row insertion and four cases for column insertion. During the
bumping step of shifted Hecke insertion, let \( z_i \) be the smallest number in row \( i \) that is greater than \( y_i \).

**R1)** \( y_i \) is inserted into row \( i \), bumping \( z_i \) into the next row: Since we will bump \( z_i \) into the next row, \( z_i \) was not on the main diagonal. Use switch sequence \((t, y_i), (t - 1, y_i), (t - 2, y_i), \ldots, (k, y_i)\) to move \( y_i \) to the left along row \( i \) until it is directly above \( z_i \) in row \( i + 1 \). And then, starting from the right, swap each box in row \( i \) having an underlined label with the one directly below, which has a non-underlined label using switch sequence \((t, x_t), (t - 1, x_{t-1}), \ldots, (k, x_k), (k - 1, x_{k-1}), \ldots, (2, x_2)\). We leave the underlined entry on the main diagonal in row \( i \). Then the local picture of row \( i \) and \( i + 1 \) is:

<table>
<thead>
<tr>
<th>( k - 1 )</th>
<th>( y_i )</th>
<th>( k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_{k-1} )</td>
<td>( z_i )</td>
<td>( x_k )</td>
</tr>
</tbody>
</table>

The result is row \( i + 1 \) normal form. Note that the non-underlined entries in row \( i \) we obtain after this process are the same as the entries in row \( i \) obtained after shifted Hecke insertion of the tableau \( P \), with \( y_i \) replacing \( z_i \) and \( z_i \) inserted into the next row, as desired. Notice also that the switches are in a valid order.

**R2)** \( y_i \) is not inserted into row \( i \) due to a horizontal violation (and no vertical violation):

Similarly with Case R1, \( y_i \) is moved to the left along row \( i \) until it is directly above \( z_i \) in row \( i + 1 \). In this case, a horizontal violation occurs if replacing \( z_i \) with \( y_i \) does not result in an increasing tableau. And since \( z_i \) is the smallest number greater than \( y_i \), the entry to the left of \( z_i \) must have value equal to \( y_i \); otherwise the horizontal violation could not occur. First, perform \((t, y_i)\) so \( t \) appears in both row \( i \) and \( i + 1 \). Since we consider no vertical violation in this case, in the middle picture, the entry \( a \) beneath \( t \) should be greater than \( z_i \). Then perform \((t, z_i)\). Finally, starting from the right, swap each box in row \( i \) having a underlined label with the one directly below with a non-underlined label as above, leaving the underlined entry on the main diagonal in row \( i \) in place. The local picture of row \( i \) and \( i + 1 \) is:

<table>
<thead>
<tr>
<th>( t )</th>
<th>( y_i )</th>
<th>( \cdot )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y_i )</td>
<td>( z_i )</td>
<td>( a )</td>
</tr>
<tr>
<td>( z_i )</td>
<td>( t )</td>
<td>( b )</td>
</tr>
</tbody>
</table>

The result is row \( i + 1 \) in normal form. Note that the non-underlined labels of row \( i \) we obtain after this process are the same as the entries in row \( i \) obtained after shifted Hecke insertion of the tableau \( P \), leaving \( z_i \) unchanged and inserting \( z_i \) into the next row. Furthermore, the switches used are in a valid order.

**R3)** \( y_i \) is not inserted into row \( i \) due to a vertical violation (and no horizontal violation):

When this insertion occurs in shifted Hecke insertion there must be a box with entry having value \( y_i \) directly above \( z_i \). Otherwise, it is impossible to have a vertical violation. Also note that when inserting a number into row \( i \) which also appears in row \( i - 1 \), the previous step could not be Case R1. In another word, the vertical violations will happen after an horizontal violation or an vertical violation. We refer readers to Example 4.25.
as the case of an vertical violation following an horizontal violation. Since there is no horizontal violation in this case, the number immediately to the left of $z_i$, say $a$, satisfies $a < y_i$. In the local picture in Case R2 above, we will not be able to get from the tableau in the middle to the tableau on the right because $a$ is in the box directly southwest of $z_i$. Hence we start at the tableau in the middle instead of the one on the right.

Perform switch operations $(t, a), (t, y_i), (t, z_i)$. Then, starting from the right, swap each box in row $i$ having an underlined label with the one directly below having a non-underlined label as above, leaving the underlined entry on the main diagonal in row $i$. The local picture of row $i - 1$, $i$ and $i + 1$ is:

```
... t ...
... y_i ...
... z_i ...
... a ...
```

The result is row $i + 1$ in normal form. Note that the non-underlined labels of row $i$ we obtain after this process are the same as the entries of row $i$ in shifted Hecke insertion tableau $P$, keeping row $i$ unchanged and inserting $z_i$ into the next row. Also, the switch operations used are in a valid order.

**R4)** $y_i$ is adjoined to the end of row $i$ and insertion terminates:

When this occurs in shifted Hecke insertion, $y_i$ is weakly larger than all elements in row $i$. Let $x_k$ be the last element in row $i$. First, consider the case where $y_i > x_k$. As in Case R1, switch $y_i$ to the left until it is directly above and to the right of $x_k$. Then, starting from the right, swap each box in row $i$ having an underlined label with the one directly below which has a non-underlined label. We leave the underlined on the main diagonal in row $i$. The local picture of row $i$ and $i + 1$ is:

```
... t-1 t y_i ...
... x_{k-1} x_k ...
... ...
```

The result is row $i + 1$ with only underlined entries. If we have $y_i = x_k$, the setup is identical to Case R2, but with $z_i$ replaced by an empty box. To perform insertion, we switch $y_i$ to the left until it is directly above and to the right of $x_k$. Then, we switch the last underlined entry with both $y_i$ and $x_k$. Finally, we switch the underlined labels down into row $i + 1$, leaving the underlined entry on the main diagonal. The local picture is:

```
... t-1 t y_i ...
... x_{k-1} y_i ...
```

The result is row $i + 1$ with only underlined entries. Finally, we need to consider the case where adjoining $y_i$ to the end of row $i$ would cause a vertical violation. As before, the setup is identical to that of Case R3, but with $z_i$ replaced by an empty box. The
local picture is:

\[
\begin{array}{|c|c|c|c|c|}
\hline
\cdots & t & \cdots & t & y_i \\
\hline
\cdots & t & x_k & y_i & \cdots \\
\hline
\cdots & x_k & \cdots & t & \cdots \\
\hline
\end{array}
\]

The result is row \( i + 1 \) with only underlined entries. Notice that each of these cases agrees with the result of shifted Hecke insertion and that the switches used are in a valid order.

If the insertion of \( w_n \) never triggers column insertion, this process will terminate after some combination of the above cases, ending with Case R4. When this happens, we will have some row \( i \) with only underlined entries and all rows above that with an underlined entry on the main diagonal. Then, working from right to left, we switch each underlined entry with the one directly below it to get row \( i + 1 \) with only underlined entries, leaving an underlined entry on the main diagonal of row \( i \). Continue this process until the last row of the tableau has only underlined entries and all rows above have an underlined entry on the main diagonal. Notice that the the main diagonal entries will be strictly increasing top to bottom. The tableau is of the form:

\[
\begin{array}{|c|c|c|c|c|}
\hline
\cdots & t_1 & x_{1,1} & x_{1,2} & x_{1,3} & \cdots & x_{1,r_1} \\
\hline
\cdots & t_2 & x_{2,1} & x_{2,2} & x_{2,3} & \cdots & x_{2,r_2} \\
\hline
\vdotswithin{\cdots} & \vdotswithin{\cdots} & \vdotswithin{\cdots} & \vdotswithin{\cdots} & \vdotswithin{\cdots} & \vdotswithin{\cdots} & \vdotswithin{\cdots} \\
\hline
\cdots & t_{k-1} & x_{k-1,1} & x_{k-1,2} & \cdots & x_{k-1,r_{k-1}} \\
\hline
\cdots & t_k & x_{k,1} & \cdots & \cdots & x_{k,r_k} \\
\hline
\end{array}
\]

To complete shifted \( K \)-rectification, we switch these main diagonal entries to the end of their row. We use the switch sequence \((t_k, x_{1,1}), (t_k, x_{k,2}), \ldots, (t_k, x_{k,r_k}), (t_{k-1}, x_{k-1,1}), \ldots, (t_{k-1}, x_{k-1,r_{k-1}}), \ldots, (t_1, x_{1,1}), \ldots, (t_1, x_{1,r_1})\) to switch the main diagonal entries to the end, starting with the bottom row.

Since our tableau is increasing, both of these processes use a valid switch order. So, if column insertion never occurs, shifted \( K \)-rectification will be complete at this point.

**Example 4.25.** Consider the insertion of 5 into the following tableau:

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
1 & 2 & 3 & 4 & 6 \\
\hline
4 & 5 & 6 & 8 \\
\hline
6 & 7 \\
\hline
\end{array}
\]
In shifted Hecke insertion, the 5 will replace the 6 in the first row, bumping it into the second row. There, the 6 does not replace the 8 due to a horizontal violation and the 8 is bumped to the third row. Finally, the 8 does not adjoin to the end of the third row due to a vertical violation. This gives the following resulting tableau:

```
1 2 3 4 5
4 5 6 8
6 7
```

We can demonstrate the same insertion using shifted $K$-jdt. We start with the following setup:

```
1 2 3 4 5 6 5
1 2 3 4 6
4 5 6 8
6 7
```

First, we switch 5 as far as possible to the left using switch sequence $(6, 5)$. This gives us:

```
1 2 3 4 5 5 6
1 2 3 4 6
4 5 6 8
6 7
```

Now we are in Case R1. Using the switch sequence $(5, 4), (4, 3), (3, 2), (2, 1)$ for that case gives us:

```
1 1 2 3 4 5 6
2 3 4 5 6
4 5 6 8
6 7
```

Now we are in Case R2. Since a vertical violation is about to occur, we will only go as far as the second picture before switching up the non-underlined entries in row 3. We perform the switch sequence $(5, 6), (4, 5), (3, 4)$ needed for that case to get:

```
1 1 2 3 4 5 6
2 4 5 6 5
3 4 5 8
6 7
```
Now we are in Case R4, part 3. We perform the switch sequence \((5, 7), (5, 8), (4, 6)\) needed for that case to get:

\[
\begin{array}{cccccccc}
1 & 1 & 2 & 3 & 4 & 5 & 6 \\
2 & 4 & 5 & 6 & 8 \\
3 & 6 & 7 & 5 \\
4 & 5
\end{array}
\]

Finally, we need to switch the underlined entries on the main diagonal to the outside. We work from bottom to top, using switch sequence \((3, 6), (3, 7), (2, 4), (2, 5), (2, 6), (2, 8), (1, 1), (1, 2), (1, 3), (1, 4), (1, 5)\). This gives the tableau

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 & 1 & 6 \\
4 & 5 & 6 & 8 & 2 \\
6 & 7 & 3 & 5 \\
4 & 5
\end{array}
\]

which is the same as the result of shifted Hecke insertion. One can also check that the switch sequence used is valid.

In shifted Hecke insertion, column insertion is only triggered once. We will first describe how column insertion begins and then explain the cases involved in completing insertion.

In the event that \(y_i\) bumps an element \(z_i\) from the main diagonal, we will begin column insertion, described below. The process of changing from row to column insertion proceeds as follows:

Use switch sequence \((t, y_i), (t - 1, y_i), (t - 2, y_i), \ldots (k, y_i)\) to move \(y_i\) to to the left along row \(i\) until it is in the box directly to the left and above \(z_i\) in row \(i + 1\). And then, starting from the right, swap each box in row \(i\) having a underlined label with the one directly below, which has a non-underlined label using switch sequence \((t, x_t), (t - 1, x_{t-1}), \ldots (k + 1, x_{k+1})\). Then the local picture of row \(i\) and \(i + 1\) is:

\[
\begin{array}{cccccccc}
y_i & k & k + 1 & \cdots & z_i & w_{k+1} & \cdots \\
z_i & w_{k+1} & \cdots & k & k + 1 & \cdots
\end{array}
\]

Then swap the box in row \(i + 1\) having the maximum underlined label with the one directly below, which has a non-underlined label. Since each previous row insertion left an underlined entry on the main diagonal, we obtain an initial position like the following, with underlined labels along the diagonal above \(z_i\):
We can shift all entries above and to the left of \( z_i \) one box left using the underlined entries on the diagonal since our tableau is increasing with switch sequence \((t_{k-1}, x_{k-1,1}), (t_{k-2}, x_{k-2,1}), (t_{k-2}, x_{k-2,2}), \ldots, (1, x_{1,1}), (1, x_{1,2}), \ldots, (1, x_{1,i-1})\) This gives a tableau with only underlined entries above \( z_i \), creating a normal column and allowing us to start column insertion.

C1) \( y_i \) is inserted into column \( i \), bumping \( z_j \) into the next column: In this case, the local picture of column \( i \) and \( i + 1 \) is:

\[
\begin{array}{ccc}
1 & z_1 & \cdots \\
2 & z_2 & \\
\vdots & \vdots & \\
t_{k-1} & z_{k-1} & \\
y_i & z_k & \\
\end{array} \quad \begin{array}{ccc}
1 & z_1 & \\
2 & z_2 & \\
\vdots & \vdots & \\
t_{j-1} & z_{j-1} & \\
y_i & z_j & \\
\end{array} \quad \begin{array}{ccc}
z_1 & 1 \\
z_2 & 2 \\
\vdots & \vdots \\
z_{j-1} & t_{j-1} \\
y_i & z_j & \\
t_j & z_{j+1} & \\
z_{j+1} & t_j & \\
\end{array}
\]

Let \( z_j \) be the smallest number greater that \( y_i \) in column \( i + 1 \). Using the switch sequence \((t_{k-1}, y_i), \ldots, (t_j, y_i), (t_{k-1}, z_k), \ldots, (t_j, z_{j+1}), (t_{j-1}, z_{j-1}), \ldots, (1, z_1)\) we shift \( y_i \) up until \( y_i \) is adjacent to \( z_j \) and then shift every other entry in column \( i + 1 \) across. This gives us \( z_j \) the only non-underlined entry in its column and we have the column analogue of a normal row as before. This procedure repeats until there are no entries to the right of the bumped element, leaving it at the bottom of a column.

C2) We get a slight variation in the event of a column violation that is not followed by a row violation. In that case, \( z_{j-1} = y_i \). The local picture of column \( i \) and \( i + 1 \) will look like
Here, we bump $z_j$ to the next column and make no change to column $i+1$ as expected and again return to our initial setup using switch sequence $(t_{k-1}, y_i), \ldots, (t_j, y_i), (t_{k-1}, z_k), \ldots, (t_j, z_j+1), (t_{j-1}, y_i), (t_{j-1}, z_j), (t_{j-2}, z_{j-2}), \ldots, (1, z_1)$.

C3) If we get a row violation following a column violation, then we have the following local picture, where $a < z_j$.

Here, we bump $z_j$ into the next column, as before, and then bump $w_j$ into the next column, using the switch sequence $(t_j, w_{j+1})(t_{j-1}, a)(t_{j-1}, z_j) \ldots (2, w_2)(1, w_1)$. We finish either in the setup for another row violation, in which case we repeat the process, or in the standard initial setup for column insertion, in which case we can perform the switch $(t_{j-1}, w_j)$. Notice that this coincides exactly with the result of column insertions in the shifted Hecke insertion algorithm. Furthermore, notice that any version of column insertion results from a valid switch sequence.

C4) $y_i$ adjoins to the end of column $i$ and insertion terminates:
When this occurs in shifted Hecke insertion, $y_i$ is weakly larger than all elements in column $i$. Let $x_k$ be the lowest element in column $i$. First consider the case where $y_i > x_k$ and no violations occur. Then, as in Case C1, switch $y_i$ until it is directly below and to the left of $x_k$. Then, starting from the bottom, swap every box in column $i$ having an underlined label with the one directly to the right, having a non-underlined label. The local picture is:

\[
\begin{array}{c|c|c}
 & t-1 & x_{k-1} \\
 & t & x_k \\
 y_i & & y_i \\
\end{array} \quad \rightarrow \quad
\begin{array}{c|c|c}
 & x_{k-1} & t-1 \\
 & x_k & t \\
 & & y_i \\
\end{array}
\]

The result is column $i+1$ contains only underlined entries. If we have a vertical violation, we have $y_i = x_k$ and we get the following local picture:

\[
\begin{array}{c|c|c}
 & t-1 & x_{k-1} \\
 & t & y_i \\
 y_i & & y_i \\
\end{array} \quad \rightarrow \quad
\begin{array}{c|c|c}
 & x_{k-1} & t-1 \\
 & y_i & t \\
 & & t \\
\end{array}
\]

We switch the lowest underlined entry with both appearances of $y_i$ and switch all other underlined entries with the non-underlined entries immediately to the right, leaving column $i+1$ containing only fake entries.

Finally, if we get a horizontal violation, it must have followed from another violation and so we must have the following local picture, with $x_k < y_i$:

\[
\begin{array}{c|c|c}
 x_{s-1} & t & x_k \\
 t & y_i & \\
\end{array} \quad \rightarrow \quad
\begin{array}{c|c|c}
 x_{k-1} & t-1 \\
 y_i & t \\
\end{array}
\]

If the insertion of $w_n$ triggers column insertion, this process will terminate after some combination of the cases above, ending with Case C4. When this happens, we will have some column $i$ with only underlined entries. Notice that these entries will be strictly increasing top to bottom. Since we removed the main diagonal underlined entries when column insertion...
began, we only need to move the underlined entries in column $i$ to complete shifted $K$-rectification. The tableau is of the form:

$$
\begin{array}{cccc}
  x_{1,1} & x_{1,2} & x_{1,3} & \cdots & t_1 & x_{1,i+1} & \cdots & x_{1,r_1} \\
  x_{2,1} & x_{2,2} & \cdots & t_2 & x_{2,i+1} & \cdots & x_{2,r_2} \\
  x_{3,1} & \cdots & t_3 & x_{3,i+1} & \cdots & x_{3,r_3} \\
  \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
  t_k & x_{k,i+1} & \cdots & x_{k,r_k} \\
\end{array}
$$

We use the switch sequence $(t_k, x_{k,i+1}), \ldots, (t_1, x_{1,i+1}), \ldots, (t_k, x_{k,r_k}), \ldots, (t_1, x_{1,r_1})$ to switch the column of underlined entries to the end, working from bottom to top. Since our tableau is increasing, this is a valid switch order. So, if column insertion occurs, shifted $K$-rectification will be complete at this point.

**Example 4.26.** Consider the insertion of 4 into the following tableau:

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

In shifted Hecke insertion, the 4 will replace the 5 in the first row, bumping it into the second row. There, the 5 replaces the 6, bumping it into the third row. Then, the 6 replaces the 7, triggering column insertion because 7 is on the main diagonal. The 7 fails to replace the 8 in the fourth column due to a vertical violation, bumping the 8 to the fifth column. The 8 replaces the 9, bumping it into the sixth column. Finally, the 9 adjoins to the end of the sixth column. This gives the following resulting tableau:

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

We can demonstrate the same insertion using shifted $K$-jdt. We start with the following setup:

$$
\begin{array}{cccc}
  1 & 2 & 3 & 4 & 5 & 6 & 4 \\
  1 & 2 & 3 & 5 & 9 \\
  4 & 6 & 7 & 10 \\
  7 & 8 \\
25
\end{array}
$$
The first step is Case R1. Using the switch sequence \((6, 4), (5, 4), (5, 9), (4, 3), (3, 2), (2, 1)\) we get the following tableau:

\[
\begin{array}{ccccccc}
1 & 2 & 3 & 4 & 9 & 6 \\
2 & 3 & 4 & 5 & 5 \\
4 & 6 & 7 & 10 \\
7 & 8 \\
\end{array}
\]

The next step is also Case R1. Using the switch sequence \((4, 5), (5, 10), (4, 7), (3, 4)\) we get the following tableau:

\[
\begin{array}{ccccccc}
1 & 2 & 3 & 4 & 9 & 6 \\
2 & 4 & 5 & 7 & 10 \\
3 & 6 & 4 & 5 \\
7 & 8 \\
\end{array}
\]

Now, we bump 7 and follow the steps to begin column insertion. Using the switch sequence \((3, 6), (3, 7), (4, 8), (2, 4), (2, 5), (1, 1), (1, 2), (1, 3)\) we get the following tableau with the fourth column a normal column:

\[
\begin{array}{ccccccc}
1 & 2 & 3 & 1 & 4 & 9 & 6 \\
4 & 5 & 2 & 7 & 10 \\
6 & 7 & 8 & 5 \\
3 & 4 \\
\end{array}
\]

Next, we apply Case C2. We perform the switch sequence \((2, 7), (2, 8), (1, 4)\) needed for this case to get:

\[
\begin{array}{ccccccc}
1 & 2 & 3 & 4 & 1 & 9 & 6 \\
4 & 5 & 7 & 8 & 10 \\
6 & 8 & 2 & 5 \\
3 & 4 \\
\end{array}
\]

Now, we are in Case C1. We perform the switch sequence \((1, 8), (1, 10)\) to get the following tableau:

\[
\begin{array}{ccccccc}
1 & 2 & 3 & 4 & 8 & 9 & 6 \\
4 & 5 & 7 & 10 & 1 \\
6 & 8 & 2 & 5 \\
3 & 4 \\
\end{array}
\]

We are in Case C4 here, but there are no switches to be done, so shifted \(K\)-rectification terminates here. This tableau is the same as the result of shifted Hecke insertion. One can also check that the switch sequence used is valid.

Since our tableau is increasing, any possible sequence of applications of the cases above will give a valid switch sequence. The result follows.
This result, combined with the following theorem from [3] tells us that if two words have the same insertion tableau, they must be weakly $K$-Knuth equivalent.

**Definition 4.27.** Tableaux $T$ and $T'$ are called *jeu de taquin equivalent* if one can be obtained from another by a sequence of jeu de taquin slides.

**Theorem 4.28.** [3, Theorem 7.8] Let $T$ and $T'$ be increasing shifted tableaux. Then $\text{row}(T) \equiv \text{row}(T')$ if and only if $T$ and $T'$ are jeu de taquin equivalent.

**Corollary 4.29.** If $P_{SK}(u) = P_{SK}(v)$, then $u \equiv v$.

**Example 4.30.** The converse of this statement does not hold. As a counterexample, consider the words 12453 and 124533, which are easily seen to be weakly $K$-Knuth equivalent. However, shifted Hecke insertion gives the following distinct tableaux.

$P_{SK}(12453) = \begin{array}{llll}
1 & 2 & 3 & 5 \\
4 & & & \\
\end{array}$

$P_{SK}(124533) = \begin{array}{llll}
1 & 2 & 3 & 5 \\
4 & 5 & & \\
\end{array}$

**Definition 4.31.** Tableaux $T$ and $T'$ are called equivalent if $\text{row}(T) \equiv \text{row}(T')$.

Using Definition 4.31, we can sort the set of increasing shifted tableaux into equivalence classes, each of which is finite by Lemma 4.1. This will prove useful in the next section.

### 5. Shifted $K$-Poirier-Reutenauer Bialgebra

Jing and Li developed a shifted version of the classical Poirier-Reutenauer bialgebra in [6], and Patrias and Pylyavskyy developed a $K$-theoretic analogue of the Poirier-Reutenauer bialgebra in [10]. In this section, we combine these techniques to introduce a shifted $K$-theoretic analogue. We start by defining the appropriate vector space.

We say a word $h$ is *initial* if the letters appearing in it are exactly the numbers in $[k]$ for some positive integer $k$. For example, the words 54321 and 211345 are initial, but 2344 is not.

Let $[[h]]$ denote the formal sum of all words in the weak $K$-Knuth equivalence class of an initial word $h$:

$$[[h]] = \sum_{\text{w} \equiv h} w.$$  

This is an infinite sum, however, the number of terms in $[[h]]$ of each length is finite.

**Example 5.1.** $[[213]] = 213 + 231 + 123 + 321 + 321 + 3211 + 32111 + \cdots$.

Let $SKPR$ denote the vector space over $\mathbb{R}$ spanned by all sums of the form $[[h]]$ for some initial word $h$. We will define a compatible product and coproduct on $SKPR$, to give a bialgebra structure. We need the following lemma.

**Lemma 5.2.** We have

$$[[h]] = \sum_T \left( \sum_{P_{SK}(w) = T} w \right),$$

where the sum is over all increasing shifted tableaux $T$ whose reading word is in the weak $K$-Knuth equivalence class of $h$.  

27
Proof. Follows from Corollary 4.29.

Note that the set of tableaux we sum over is finite by Lemma 4.1.

5.1. **Product structure.** Let \(\square\) denote the usual shuffle product of words, and denote by \(w[n]\) the word obtained from word \(w\) by increasing each letter by \([n]\). For example, if \(w = 312\), \(w[4] = 756\). Let \(h\) be a word in the alphabet \([n]\) and let \(h'\) be a word in the alphabet \([m]\). We define the product in \(SKPR\) by

\[
[[h]] \cdot [[h']] = \sum_{w \equiv h, w' \equiv h'} w \square w'[n].
\]

**Theorem 5.3.** For any two initial words \(h\) and \(h'\), their product can be written as

\[
[[h]] \cdot [[h']] = \sum_{h''}[[h'']],
\]

where the sum is over some set of initial words \(h''\).

**Proof.** By Lemma 4.13, we know that once a word appears in the right-hand sum, its entire equivalence class must also appear. The result follows from this. \(\square\)

**Example 5.4.** Let \(h = 12\) and \(h' = 1\). Then

\[
[[12]] \cdot [[1]] = [[123]] + [[312]] + [[3123]].
\]

By Lemma 5.2, we can write \([[h]]\) as a sum over tableaux, and we know that these tableaux can be sorted into finite equivalence classes. This suggests that we can give an write the product as an explicit sum over sets of tableaux.

**Theorem 5.5.** Let \(h\) be a word in alphabet \([n]\), and let \(h'\) be a word in alphabet \([m]\). Suppose \(T = \{P_{SK}(h), T_1', T_2', \ldots, T_s'\}\) is the equivalence class containing \(P_{SK}(h)\). Then we have

\[
[[h]] \cdot [[h']] = \sum_{T \in \mathcal{T}(h,h')} \sum_{P_{SK}(w) = T} w,
\]

where \(T(h \sqcup h')\) is the finite set of shifted tableaux \(T\) such that \(T|_n \in \mathcal{T}\) and \(\text{row}(T)|_{n+1,n+m} \equiv h'[n]\).

**Proof.** If \(w\) is a shuffle of some \(w_1 \equiv h\) and \(w_2 \equiv h'[n]\), then by Lemma 4.7, \(P_{SK}(w)|_n = P_{SK}(w_1) \in \mathcal{T}\). Using Lemma 5.2 and Theorem 5.3, we see that the product can be expanded in this way. The fact that the set \(T(h \sqcup h')\) is finite follows immediately from Lemma 4.1 since all of the tableaux in that set are on the finite alphabet \([n + m]\). \(\square\)

**Example 5.6.** Consider \(h = 12\) and \(h' = 21\). The set \(T(12 \sqcup 21)\) consists of the seven tableaux below.

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

For each of these tableaux, restricting to the alphabet \(\{1, 2\}\) gives the tableau \(P_{SK}(12)\). Also, the reading word of each restricted to the alphabet \(\{3, 4\}\) is K-Knuth equivalent to \(h'[2] = 34\).
Corollary 5.7. The vector space $SKPR$ is closed under the product operation. That is, the product of two classes can be expressed as a finite sum of classes.

Proof. By Theorem 5.3, we know that the product of two words can be expressed as a sum of weak $K$-Knuth equivalence classes. From Lemma 5.2, each of these classes can be written as a finite sum of tableau equivalence classes. That is to say, weak $K$-Knuth classes are coarser than insertion tableau classes. Theorem 5.5 tells us that the product of two words can be written as a finite sum of tableau equivalence classes, so the expansion in Theorem 5.3 must be finite. □

5.2. Coproduct structure. For any word $w$, let $\text{std}(w)$ denote the standardization of $w$. That is, if a letter $a$ is the $k$-th smallest letter of $w$, $a$ becomes $k$ in $\text{std}(w)$. For example, $\text{std}(14363) = 13242$. Notice that the standardization of any word will be an initial word.

Let $w = a_1a_2 \ldots a_n$ be an initial word. We define

$$\Delta(w) = \sum_{i=0}^{n} \text{std}(a_1 \ldots a_i) \otimes \text{std}(a_{i+1} \ldots a_n).$$

We define the coproduct of a weak $K$-Knuth class to be the coproduct on words, extended linearly:

$$\Delta([h]) = \sum_{h \equiv w} \Delta(w).$$

Example 5.8. We have

$$\Delta(13242) = \emptyset \otimes 13242 + 1 \otimes 2131 + 12 \otimes 121 + 132 \otimes 21 + 1324 \otimes 1 + 13242 \otimes \emptyset$$

and

$$\Delta([13242]) = \Delta(13242) + \Delta(13424) + \Delta(31424) + \Delta(31242) + \cdots$$

We use $\emptyset$ to denote the empty word, the identity element of the ground field.

Theorem 5.9. For any initial word $h$, its coproduct can be written as

$$\Delta([h]) = \sum_{h',h''} [[h']] \otimes [[h'']]$$

where the sum is over a set of pairs of initial words $h'$ and $h''$.

Proof. For $w = a_1 \ldots a_n$, define $\Delta(w) = \sum_{i=0}^{n} a_1 \ldots a_i \otimes a_{i+1} \ldots a_n$, a nonstandard analogue of $\Delta(w)$. Similarly, define $\Delta([h]) = \sum_{h \equiv w} \Delta(w)$. We now want to show that if the term $u \otimes v$ occurs in $\Delta([h])$, the entire class $[[u]] \otimes [[v]]$ must also appear. Since the weak $K$-Knuth equivalence relations are applied locally, any move made in $u$ or $v$ can be applied in the corresponding portion of the word $h$. Hence, $\Delta([h]) = \sum_{h',h''} [[h']] \otimes [[h'']]$ for some collection of pairs of words (not necessarily initial) $h'$ and $h''$. Now, the weak $K$-Knuth relations commute with standardization, so we can standardize every term on the right to get that $\Delta([h])$ is a sum over pairs of initial words. □
Theorem 5.10. Let $h$ be an initial word. Then
\[
\Delta([[h]]) = \sum_{(T', T'') \in T(h)} \left( \sum_{w \in \text{P}_{SK}(w) = \text{std}(T')} w \right) \otimes \left( \sum_{w \in \text{P}_{SK}(w) = \text{std}(T'')} w \right),
\]
where $T(h)$ is the finite set of pairs of shifted tableaux $T', T''$ such that $\text{row}(T') \equiv \text{row}(T'') \equiv h$.

Proof. From the proof of Theorem 5.9 we have that $\Delta([[h]]) = \sum_{h', h''}[[(h')]] \otimes [[(h'')]]$. Notice that the pairs $h'$ and $h''$ are exactly those pairs of words that, when concatenated, give a word equivalent to $h$, so they correspond to reading words of the pairs of tableaux in $T(h)$. Since this sum is multiplicity-free, when we split each of the $[[h']]$ and $[[h'']]$ into tableau insertion classes we get precisely
\[
\Delta([[h]]) = \sum_{(T', T'') \in T(h)} \left( \sum_{w \in \text{P}_{SK}(w) = \text{std}(T')} w \right) \otimes \left( \sum_{w \in \text{P}_{SK}(w) = \text{std}(T'')} w \right).
\]
Standardizing both sides gives the desired result. The set $T(h)$ is finite by Lemma 4.1. □

Corollary 5.11. The vector space $\text{SKPR}$ is closed under the coproduct operation. That is, the coproduct of a class can be expressed as the finite sum of tensor products of classes.

Proof. By Theorem 5.9, the coproduct of a class can be expressed as a sum of tensor products of weak $K$-Knuth equivalence classes. Theorem 5.10, the coproduct can also be expressed as a finite sum of tensor products of tableau insertion classes. Since weak $K$-Knuth classes are coarser than insertion classes, we have that the sum in Theorem 5.9 must be finite. □

5.3. Compatibility. Recall that a product $\cdot$ and coproduct $\Delta$ are said to be compatible if the coproduct is an algebra morphism
\[
\Delta(X \cdot Y) = \Delta(X) \cdot \Delta(Y).
\]
A vector space with a compatible product and coproduct is called a bialgebra.

Theorem 5.12. The product and coproduct structures on $\text{SKPR}$ as defined above are compatible. Hence, $\text{SKPR}$ has a bialgebra structure.

Proof. It suffices to show that the product and coproduct are compatible on any pair of initial words. Let $u = a_1 \ldots a_n$ and $w = b_1 \ldots b_m$ be two initial words. Then,
\[
\Delta(w \cdot u) = \Delta(w \uplus u[n]) = \sum_{v = c_1 \ldots c_{n+m}} \sum_{i=0}^{n+m} \text{std}(c_1 \ldots c_i) \otimes \text{std}(c_{i+1} \ldots c_{n+m}),
\]
where $v$ ranges over shuffles of $w$ and $u[n]$. On the other hand,
\[
\Delta(w) \cdot \Delta(u) = \left( \sum_{i=0}^{n} \text{std}(a_1 \ldots a_i) \otimes \text{std}(a_{i+1} \ldots a_n) \right) \cdot \left( \sum_{j=0}^{m} \text{std}(b_1 \ldots b_j) \otimes \text{std}(b_{j+1} \ldots b_m) \right)
\]
\[
= \sum_{i,j} \text{std}(a_1 \ldots a_i) \uplus \text{std}(b_1 \ldots b_j)[i] \otimes \text{std}(a_{i+1} \ldots a_n) \uplus \text{std}(b_{j+1} \ldots b_m)[n-j]).
\]
These two expressions are easily seen to be equal. Hence, $\Delta(w \cdot u) = \Delta(w) \cdot \Delta(u)$. □
6. Unique Rectification Targets

As we have seen in Example 4.30, weak $K$-Knuth equivalence classes may have several corresponding insertion tableaux. This is a key difference between weak $K$-Knuth equivalence and the classical Knuth equivalence. Of particular importance in our setting are the classes of tableaux with only one element. The notion of such tableaux will be crucial in our Littlewood-Richardson rules.

**Definition 6.1.** [3, Definition 3.5] We call $T$ a unique rectification target, or a URT, if it is the only tableau in its weak $K$-Knuth equivalence class. That is, $T$ is a URT if for every $w \equiv \text{row}(T)$ we have $P_{SK}(w) = T$. If $P_{SK}(w)$ is a URT, we call the equivalence class of $w$ a unique rectification class.

We refer the reader to [3, 4] for a more detailed discussion of URTs for shifted tableaux and straight shape tableaux.

**Example 6.2.** The tableaux given in Example 4.30 are equivalent to each other, and hence neither is a URT.

In [3], Buch and Samuel describe a way to fill any shifted shape to create a URT, and they call the resulting tableau a minimal increasing shifted tableau. The minimal increasing shifted tableau $M_{\lambda}$ of a shifted shape $\lambda$ is the tableau obtained by filling the boxes of $\lambda$ with the smallest values allowed in an increasing tableau.

**Example 6.3.** The following two tableau are minimal shifted increasing tableaux.

$M_{(4,2)} = \begin{array}{cccc}
1 & 2 & 3 & 4 \\
3 & 4 \\
\end{array}$

$M_{(5,2,1)} = \begin{array}{cccc}
1 & 2 & 3 & 4 & 5 \\
3 & 4 \\
5 \\
\end{array}$

[3, Corollary 7.2] shows that $M_{\lambda}$ is a URT for any shifted shape $\lambda$.

In addition, the superstandard shifted tableaux of shape $\lambda$, $S_{\lambda}$ is a URT by a result of Clifford, Thomas, and Yong [4].

**Example 6.4.** The following two tableaux are superstandard shifted tableaux.

$S_{(4,2)} = \begin{array}{cccc}
1 & 2 & 3 & 4 \\
5 & 6 \\
\end{array}$

$S_{(5,2,1)} = \begin{array}{cccc}
1 & 2 & 3 & 4 & 5 \\
6 & 7 \\
8 \\
\end{array}$

Recall from Theorem 5.3 and Theorem 5.9 that we can define the product and coproduct in $SKPR$ in terms of equivalence classes of tableaux. It follows that classes corresponding to URTs should have particularly simple products and coproducts.

**Theorem 6.5.** Let $T_1$ and $T_2$ be two URTs. Then

$$
\left( \sum_{P_{SK}(u) = T_1} u \right) \cdot \left( \sum_{P_{SK}(v) = T_2} v \right) = \sum_{T \in T(T_1 \cup T_2)} \sum_{P_{SK}(w) = T} w,
$$

where $T(T_1 \cup T_2)$ is the finite set of shifted tableaux $T$ such that $T|_{[n]} = T_1$ and $P_{SK}(\text{row}(T)|_{[n+1,n+m]}) = T_2$. 

31
Proof. Since $T_1$ and $T_2$ are URTs, the left-hand side is $[[\text{row}(T_1)]] \cdot [[\text{row}(T_2)]]$, which equals the right-hand side by Theorem 5.5. □

**Theorem 6.6.** Let $T_0$ be a URT. Then

$$
\Delta \left( \sum_{P_{SK}(w) = T_0} w \right) = \sum_{T \in T(T_0)} \left( \sum_{P_{SK}(u) = \text{std}(T')} u \right) \otimes \left( \sum_{P_{SK}(v) = \text{std}(T'')} v \right),
$$

where $T(T_0)$ is the finite set of pairs of shifted tableaux $(T', T'')$ such that $P_{SK}(\text{row}(T') \cdot \text{row}(T'')) = T_0$.

Proof. Since $T_0$ is a URT, the left-hand side is $[[\text{row}(T_0)]]$ and $T(T_0)$ is $T(\text{row}(T_0))$, giving equality by Theorem 5.10. □

7. Connection to symmetric functions

A composition $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_r)$ of a positive integer $n$ is defined to be an ordered arrangement of positive integers which sum to $n$. The descent set of $\alpha$ is defined to be $D(\alpha) = \{\alpha_1, \alpha_1 + \alpha_2, \ldots, \alpha_1 + \alpha_2 + \ldots + \alpha_{r-1}\} \subseteq [n-1]$. Conversely, for a given set $S = \{s_1, s_2, \ldots, s_r\} \subseteq [n-1]$, we define its composition to be $C(S) = (s_1, s_2 - s_1, \ldots, n - s_r)$.

For example, $\alpha_1 = (1, 2, 4)$ and $\alpha_2 = (2, 3, 2)$ are both compositions of 7. Then $D(\alpha_1) = \{1, 3\} \subseteq [6]$ and $D(\alpha_2) = \{2, 5\} \subseteq [6]$. Conversely, given $S = \{2, 5\} \subseteq [6]$, we can obtain its composition set $C(S) = (2, 3, 2)$.

Given a word $w = w_1w_2 \ldots w_n$, a descent set of $w$, $D(w)$, is defined to be $\{i : w_i > w_{i+1}\}$. For any word $w$, we use $C(w)$ to denote $C(D(w))$. For example, $D(131442) = \{2, 5\}$ and $C(131442) = C(D(131442)) = (2, 3, 1)$.

**Definition 7.1.** The descent set of a standard set-valued shifted tableau $T$, denoted $D(t)$, is defined to be

$$
D(T) = \left\{ i : \begin{array}{l}
\text{both } i \text{ and } (i+1)' \text{ appear} \\
\text{OR} \\
\text{i is strictly above } i + 1 \\
\text{OR} \\
\text{i' is weakly below } (i+1)' \text{ but not in the same box}
\end{array} \right\}.
$$

**Example 7.2.** Consider a word $w = 354211$. We see that $D(w) = \{2, 3, 4\}$, and its recording tableau is shown below.

$Q_{SK}(w) = \begin{array}{ccc}
1 & 2 & 4' 5/6 \\
3 & & \\
\end{array}$

Then $D(Q_{SK}(w)) = \{2, 3, 4\}$ because 2 is strictly above 3, both 3 and 4' appear, and 4' is weakly below 5' but not in the same box.

In the previous example, we see that $D(Q_{SK}(451132)) = D(451132)$. This holds for any word.

**Theorem 7.3.** We have that $D(w) = D(Q_{SK}(w))$ for any word $w$. 32
Proof. Let \( w = w_1 w_2 \ldots w_n \). Let \( t_i \) denote the box where the insertion of \( w_i \) terminates, and let \( Q_{SK}(t_i) \) denote the label of box \( t_i \) in \( Q_{SK}(w) \). Then \( Q_{SK}(t_i) = i \) if insertion of \( w_i \) ends in rows insertion and \( Q_{SK}(t_i) = i' \) if insertion of \( w_i \) ends in column insertion or failed row insertion into an empty row.

First suppose \( i \in \mathcal{D}(w) \). Then \( w_i > w_{i+1} \). It is easy to see that elements bumped during a row insertion caused by the insertion of \( w_{i+1} \) will be weakly to the left of elements bumped during a row insertion caused by the insertion of \( w_i \). It follows that we could never have \( Q_{SK}(t_{i+1}) = i + 1 \) and \( Q_{SK}(t_i) = i' \) as \( w_{i+1} \) will start column insertion or fail to insert into an empty row first. Therefore, we only have three cases:

1. If \( Q_{SK}(t_i) = i \) and \( Q_{SK}(t_{i+1}) = i + 1 \), then both the insertion of \( w_i \) and of \( w_{i+1} \) only use row insertion and never fail to insert into an empty row. Therefore, using the standard doubling argument for shifted tableaux (Section 7 of [3]), \( t_i \) is strictly above \( t_{i+1} \) by the corresponding result for Hecke insertion found in Lemma 2 of [2]. Hence \( i \in \mathcal{D}(Q_{SK}(w)) \).
2. If \( Q_{SK}(t_i) = i \) while \( Q_{SK}(t_{i+1}) = (i+1)', i \in \mathcal{D}(Q_{SK}(w)) \) by definition.
3. Suppose \( Q_{SK}(t_i) = i' \) and \( Q_{SK}(t_{i+1}) = (i+1)' \). Note that \( w_{i+1} \) will bump an element in every column \( w_i \) did, and so \( t_{i+1} \) will be weakly to the right to \( t_i \). If the insertion caused by \( w_{i+1} \) bumps something out of the column where \( t_i \) is and insertion continues, we will have \((i+1)' \) weakly above \( i' \) but not in the same box. If \( t_{i+1} \) is in the same column as \( t_i \), this implies that \( w_{i+1} \) must fail to add a new box in the insertion tableau, as \( t_{i+1} \) must be weakly above \( t_i \). Hence we get an \( i' \) and \((i+1)' \) in the same box. However, reverse inserting from this implies \( w_i \leq w_{i+1} \), which contradicts our assumption that \( w_i > w_{i+1} \).

Hence \( \mathcal{D}(w) \subseteq \mathcal{D}(Q_{SK}(w)) \). Next suppose \( i \notin \mathcal{D}(w) \) and \( w_i < w_{i+1} \). Similarly, We cannot have insertion of \( w_i \) involving only row insertion and \( w_{i+1} \) column inserting, since insertion of \( w_{i+1} \) cause bumping to the right of that caused by \( w_i \). Hence, in the row where \( w_{i+1} \) begins column insertion, \( w_i \) could only bump the diagonal element and would hence start column inserting.

1. If \( Q_{SK}(t_i) = i \) and \( Q_{SK}(t_{i+1}) = i + 1 \), then using the standard doubling argument for shifted tableaux (see section 7 in [3] for more details), this would imply \( t_i \) is weakly below \( t_{i+1} \) by Lemma 2 in [2].
2. If \( Q_{SK}(t_i) = i' \) while \( Q_{SK}(t_{i+1}) = i + 1 \), then \( i \notin \mathcal{D}(Q_{SK}(w)) \) by our definition.
3. If \( Q_{SK}(t_i) = i' \) and \( Q_{SK}(t_{i+1}) = (i+1)' \), this implies that both insertion involve column insertion. Insertion of \( w_{i+1} \) will cause bumping weakly lower than \( w_i \) does and so \( t_{i+1} \) must be weakly to the left of \( t_i \). If \( t_{i+1} \) is strictly left of \( t_i \), we get \( i' \) strictly above \((i+1)' \). In the latter, we either get \( i' \) strictly above \((i+1)' \) or \( i' \) and \((i+1)' \) occur in the same box.

So when \( w_i < w_{i+1} \), we have \( i \notin \mathcal{D}(Q_{SK}(w)) \). The case where \( i \notin \mathcal{D}(w) \) and \( w_i = w_{i+1} \) is similar.

7.1. Weak Set-Valued Tableaux and Fundamental Quasisymmetric Functions. We will define our new symmetric functions as a weighted generating function over certain shifted tableaux, called weak set-valued shifted tableaux, which we define below. Roughly speaking, weak set-valued shifted tableaux are set-valued shifted tableaux (Definition 4.8) where we
allow multisets. So, in particular, set-valued shifted tableaux are weak set-valued shifted tableaux.

Definition 7.4. A **weak set-valued shifted tableau** is a filling of the boxes of a shifted shape with finite, nonempty multisets of primed and unprimed positive integers with ordering $1'<1<2'<2<\cdots$ such that:

1. The smallest number in each box is greater than or equal to the largest number in the box directly to the left of it, if that box exists.
2. The smallest number in each box is greater than or equal to the largest number in the box directly above it, if that box exists.
3. There are no primed entries on the main diagonal.
4. Each unprimed integer appears in at most one box in each column.
5. Each primed integer appears in at most one box in each row.

Given any weak set-valued shifted tableau $T$, we define $x^T$ to be the monomial $\prod_{i\geq 1} x_i^{a_i}$, where $a_i$ is the number of occurrences of $i$ and $i'$ in $T$.

Example 7.5. The weak set-valued tableaux $T_1$ and $T_2$ below have $x^{T_1} = x_1x_2x_3x_4x_5x_6x_7$ and $x^{T_2} = x_1x_2x_3^2x_4x_5x_6^2$.

| 1 2 3' 4 |
| 5 6'7' |

$T_1$

| 1 2 3' 3' |
| 4 5'6' |

$T_2$

Recall that we denote a shifted shape $\lambda$ as $(\lambda_1, \lambda_2, \cdots, \lambda_l)$, where $l$ is the number of rows and $\lambda_i$ is the number of boxes of $i$th row.

Definition 7.6. The **shifted weak stable Grothendieck polynomial**, $K_\lambda$, is defined by

$$K_\lambda = \sum_T x^T,$$

where the sum is over the set of weak set-valued tableaux $T$ of shape $\lambda$.

Example 7.7. We have

$$K_{(2,1)} = x_1^2x_2 + 2x_1x_2x_3 + 3x_1^3x_2^2 + 5x_1^3x_2x_3 + 5x_1^3x_2^3 + \cdots,$$

where the coefficient of $x_1^3x_2^2$ is 3 because of the tableaux shown

| 11 2' |
| 2 |

| 1 12' |
| 2 |

| 1 1 |
| 22 |

The remainder of this section is devoted to expressing the $K_\lambda$ as a sum of fundamental quasisymmetric functions, which we define below. For further reading on quasisymmetric functions, see [14].

Definition 7.8. Given a composition $\alpha$ of $n$, the fundamental quasisymmetric function $L_\alpha$ is defined to be

$$L_\alpha = \sum_{i_1 \leq i_2 \leq \cdots \leq i_\nu} x_{i_1}x_{i_2}\cdots x_{i_\nu},$$

where $i_j < i_{j+1}$ if $j \in D(\alpha)$. 

34
Example 7.9. The fundamental quasisymmetric function indexed by \((2,3)\) is
\[
L_{(2,3)} = x_1^2 x_2^3 + x_2^2 x_3^3 + x_1 x_2 x_3^3 + x_2^2 x_3 x_4 x_5 + x_1 x_2 x_3 x_4 x_5 + \ldots .
\]
It is an infinite sum of monomials of degree 5, where each term has \(i_2 < i_3\) as \(D(2,3) = \{2\}\). Therefore, \(x_1^5\) will never appear in the sum.

**Definition 7.10.** We say that a monomial \(\sigma = x_{s_1} x_{s_2} \ldots x_{s_r}\), where \(s_1 \leq s_2 \leq \cdots \leq s_r\), agrees with a descent set \(D\) if for each \(k \in D\), \(s_k < s_{k+1}\).

Given a monomial \(\sigma = x_{s_1} x_{s_2} \ldots x_{s_r}\), where \(s_1 \leq s_2 \leq \cdots \leq s_r\), and a set-valued shifted tableau \(T\), we *relabel* \(T\) by \(\sigma\) by replacing the \(i\)th smallest letter in \(T\) with \(s_i\), primed if that letter had a prime to begin with.

**Example 7.11.** Given \(T\) below and \(\sigma = x_1 x_3 x_5^2 x_6 x_7\), the relabeling of \(T\) with \(\sigma\) is the tableau on the right.

\[
\begin{array}{ccc}
12 & 3' & 6 \\
5 & & \\
\end{array} \quad \begin{array}{ccc}
13 & 5' & 7 \\
6 & & \\
\end{array}
\]

**Lemma 7.12.** Let \(T\) be a standard set-valued shifted tableau and let \(\sigma = x_{s_1} x_{s_2} \ldots x_{s_r}\) be a monomial that agrees with \(D(T)\). Then the result of relabeling \(T\) by \(\sigma\) is a weak set-valued shifted tableau.

**Proof.** Let \(T'\) be the result of relabeling \(T\) by \(\sigma\). We verify that the five conditions to be a weak set-valued shifted tableau hold for \(T'\).

First, observe that we had a strictly increasing tableau before and we replaced the alphabet with one that is weakly increasing. Thus both the rows and columns will be weakly increasing, verifying conditions (1) and (2).

Condition (3) follows from the fact that there are no primed entries on the main diagonal of \(T\) by definition, so there will be no primed entries on the main diagonal of \(T'\).

We will show conditions (4) and (5) by contradiction. Suppose that we have two boxes in the same column of \(T'\), each containing a letter \(a\). Since \(T'\) is increasing, we can assume that these two boxes are adjacent.

Allowing for multiple occurrences of \(a\) in each box, we therefore have the following picture:

\[
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
a^m \\
\end{array} \\
\begin{array}{c}
a^n \\
\end{array}
\end{array}
\end{array}
\]

where \(m\) and \(n\) are at least 1 and represent the multiplicity of \(a\) in each box, and where we ignore entries in these two boxes not equal to \(a\). Considering the \(m\)th \(a\) in the upper box and the first \(a\) in the lower box, this picture must have resulted from the relabeling of

\[
\begin{array}{ccc}
i \\
\end{array} \quad \begin{array}{ccc}
i + k \\
35
\end{array}
\]


where $\sigma$ has $s_i = s_{i+k} = a$. If $k = 1$, then $i \in D(T)$ and so we would have $s_i < s_{i+1}$, a contradiction. If $k \neq 1$, then $i + 1$ must be either strictly to the right and weakly above $i$ or strictly to the left and weakly below $i + k$, as shown below:

![Diagram](image)

If $i + 1$ or $(i + 1)'$ is in the lower left, then $i \in D(T)$, a contradiction. If $(i + 1)'$ is in the upper right, then $i \in D(T)$, a contradiction. If $i + 1$ is in the upper right, then consider $i + 2$. If $(i + 2)'$ occurs, then $i + 1 \in D(T)$ and so we would have $s_i \leq s_{i+1} < s_{i+2} \leq \cdots \leq s_{i+k}$, a contradiction. If $i + 2$ occurs strictly below $i + 1$, then $i + 1 \in D(T)$, a contradiction. Otherwise, $i + 2$ occurs weakly above $i + 1$. By the same reasoning, $i + l$ must occur weakly above $i + l - 1$ for all $1 \leq l < k$, otherwise $i + l - 1$ would be an element of $D(T)$. Hence we must $i + k - 1$ somewhere in the upper right. In that case, we have $i + k - 1$ strictly above $i + k$, thus $i + k - 1 \in D(T)$, which gives a contradiction. We conclude that $T'$ satisfies condition (4).

For the final condition, suppose that we have two boxes in the same row of $T'$, each containing a letter $a'$. As before, we may assume without loss of generality that we have

$$
(a')^m \quad (a')^n
$$

where $m$ and $n$ represent the multiplicities of $a'$ in each box and we ignore entries in these boxes not equal to $a$. As before, this resulted from the relabeling of

$$
\begin{array}{c|c}
  i' & (i + k)' \\
\end{array}
$$

where $\sigma$ has $s_i = s_{i+k} = a$. If $k = 1$, then $i \in D(T)$, a contradiction. If $k \neq 1$, then because $T$ is increasing, $i + 1$ must be either strictly below and weakly to the left of $i'$ or strictly above and weakly to the right of $(i + k)'$ as shown below.
If \((i+1)′\) is in the upper right, then \(i \in \mathcal{D}(T)\), a contradiction. If \(i + 1\) is in the upper right or lower left, look for \(i + 2\). If \((i + 2)′\) occurs anywhere, then \(i + 1 \in \mathcal{D}(T)\), a contradiction. If \(i + 2\) appear and its placement does not give \(i + 1 \in \mathcal{D}(T)\), consider \(i + 3\). Eventually we must have \(i + \ell \in \mathcal{D}(T)\) for some \(1 \leq \ell < k - 1\), giving a contradiction, or we see that \(i + k - 1\) appears unprimed. In the latter case, since we have \((i+k)′\), we get \(i+k-1 \in \mathcal{D}(T)\), which also gives a contradiction.

Finally, if we have \((i+1)′\) in the lower left, consider the position of \(i + 2\). If it is unprimed, we can proceed as in the previous case to obtain a contradiction. Hence \((i + 2)′\) appears, and it must be strictly below \((i+1)′\) because otherwise \(i + 1 \in \mathcal{D}(T)\). We then consider \(i + 3\). As before, it must be primed and strictly below \((i+2)′\). Similarly, \(T′\) must contain \((i+l)′\) strictly below \((i+l-1)′\) for all \(1 \leq \ell \leq k - 1\). Then, we have \((i+k)′\) weakly above \((i+k-1)′\) and \(i + k - 1 \in \mathcal{D}(T)\), a contradiction. Therefore \(T′\) satisfies condition (5).

We define the standardization of a weak set-valued tableau \(T\), denoted \(\text{st}(T)\), to be the refinement of the order of entries of \(T\) given by reading the \(k′\)'s in \(T\) from left to right and the \(k\)'s in \(T\) from top to bottom, using the total order \((1′ < 1 < 2′ < 2 < \cdots)\). Notice that \(\text{st}(T)\) will be a standard set-valued shifted tableau.

**Example 7.13.** Below is an example of a weak set-valued shifted tableau and its standardization.

\[
T = \begin{array}{cccccccc}
1 & 2 & 3′ & 4 & 5 & 6′ & 7′ & 9 & 10 \\
4 & 4 & 6 & 7 & 8′ & & & & \\
& & & & & 8 & & & \\
& & & & & & 4 & 5 & 9 & 11 & 13′ & 14 & 15 & 16 \\
\end{array}
\]

\[
\text{st}(T) = \begin{array}{cccccccc}
1 & 2 & 3′ & 6 & 7 & 8′ & 10′ & 12 & 15 & 16 \\
4 & 5 & 9 & 11 & 13′ & & & & \\
& & & & & 14 & & & & \\
\end{array}
\]

We can now write \(K_\lambda\) as a sum of fundamental quasisymmetric functions.

37
Theorem 7.14. For any fixed increasing shifted tableau $T$ of shape $\lambda$, we have

$$K_\lambda = \sum_{P_{SK}(h) \cong T} L_C(h).$$

Proof. We establish a bijection between weak set-valued shifted tableau of shape $\lambda$ and the set of pairs $(h, \sigma)$ where $h = h_1h_2\ldots h_n$ and $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ is a sequence of weakly increasing positive integers where $\sigma_j < \sigma_{j+1}$ if $j \in D(h)$.

Given an arbitrary weak set-valued shifted tableau $W$ of shape $\lambda$ where $W$ is filled in with $\sigma_1, \sigma_2, \ldots, \sigma_n$, some of which may be primed, we consider the standardization $st(W)$. To obtain $h$, we let $st(W)$ be the recording tableau. By Theorem 4.10, $(T, st(W))$ will uniquely determine $h$. We will show that $x^W$ agrees with $D(h) = D(Q_{SK}(h)) = D(st(W))$.

Suppose $j$ is strictly above $j + 1$ after the standardization. It suffices to show that $\sigma_j \neq \sigma_{j+1}$. If $\sigma_j = \sigma_{j+1}$, then by the weakly increasing property, we can conclude that they have to be in the same column, which violates the definition of weak set-valued tableaux.

Suppose $j$ and $(j + 1)'$ both appear in $st(W)$. By the definition of standardization, $\sigma_j$ and $\sigma_{j+1}$ cannot be the same number, as otherwise the primed entry will get standardized first, in which case we have $j'$ and $(j + 1)'$. Therefore, $\sigma_j < \sigma_{j+1}$.

Suppose $(j + 1)'$ is weakly above $j'$ and not in the same box. It suffices to show that $\sigma_j \neq \sigma_{j+1}$. If $\sigma_j = \sigma_{j+1}$, they cannot appear on the same row otherwise it will violate our definition of weak set-valued shifted tableaux. If $(j + 1)'$ is strictly above $j'$, then the higher one would have to be strictly smaller after standardization and we get a contradiction.

Lemma 7.12 gives us the other direction. □

8. A Bialgebra Morphism

In this section, we construct a bialgebra morphism that takes a weak $K$-Knuth equivalence class $[[h]]$ to a sum of fundamental quasisymmetric functions. In Section 9, we will use this map to prove a Littlewood-Richardson rule for the $K_\lambda$.

Theorem 8.1. The linear map $\phi : SKPR \to QSym$ defined by

$$[[h]] = \sum_{w \in h} L_C(w)$$

is a bialgebra morphism.

Proof. We first show that the map preserves the product. Then [7, Proposition 5.9] gives that

$$L_C(w') \cdot L_C(w'') = \sum_{w \in Sh(w', w''[n])} L_C(w),$$
where the sum is over all shuffles of $w', w''[n]$. It follows that
\[
\phi([[h_1]] \cdot [[h_2]]) = \phi \left( \sum_{w' \equiv h_1, w'' \equiv h_2} w' \sqcup w'' \right) = \sum_{w' \equiv h_1} \sum_{w'' \equiv h_2} L_C(w) = \left( \sum_{w' \equiv h_1} L_C(w') \right) \left( \sum_{w'' \equiv h_2} L_C(w'') \right) = \phi([[h_1]]) \phi([[h_2]]).
\]

Next we consider the coproduct. Proposition 5.10 in [7] showed that $\Delta(L_C(w)) = \sum L_\beta \otimes L_\gamma$, where we sum over all $\beta = (\beta_1, \beta_2, \ldots, \beta_k), \gamma = (\gamma_1, \ldots, \gamma_n)$ satisfying $(\beta_1, \ldots, \beta_k, \gamma_1, \ldots, \gamma_n) = C(w)$ or $(\beta_1, \ldots, \beta_k - 1, \beta_k + \gamma_1, \gamma_2, \ldots, \gamma_n) = C(w)$. Then for any $w \equiv h$,
\[
\Delta(\phi(w)) = \Delta(L_C(w)) = \sum L_\beta \otimes L_\gamma = (\phi \otimes \phi)(\Delta(w)),
\]
where the last equality holds since the terms in $\Delta(w) = \sum_{i=0}^n w_1 w_2 \ldots w_i \otimes w_{i+1} \ldots w_n$ with $i \in D(w)$ give the terms in $L_\beta \otimes L_\gamma$ with $(\beta_1, \ldots, \beta_k, \gamma_1, \ldots, \gamma_n) = C(w)$ and those with $i \notin D(w)$ give the terms with $(\beta_1, \ldots, \beta_k - 1, \beta_k + \gamma_1, \gamma_2, \ldots, \gamma_n) = C(w)$. Hence $\Delta(\phi([[h]])) = (\phi \otimes \phi)(\Delta([[h]]))$. \hfill \Box

**Theorem 8.2.** Letting $\lambda(T)$ denote the shape of $T$, we have
\[
\phi([[h]]) = \sum_{\text{row}(T) \equiv h} K_{\lambda(T)}.
\]

**Proof.** We have, by Theorem 7.14, that
\[
\phi([[h]]) = \sum_{w \equiv h} L_C(w) = \sum_{T \equiv P_{SK}(h)} \sum_{P_{SK}(w) = T} L_C(w) = \sum_{T \equiv P_{SK}(h)} K_{\lambda(T)} = \sum_{\text{row}(T) \equiv h} K_{\lambda(T)}.
\]
\hfill \Box

**9. A Littlewood-Richardson Rule**

With the bialgebra morphism $\phi$ defined above, we can show the Littlewood-Richardson rule for $K_\lambda$ by using Theorem 8.2.

**Theorem 9.1.** Let $T$ be a URT of shape $\mu$. Then we have
\[
K_{\lambda} K_\mu = \sum_{\nu} c^\nu_{\lambda, \mu} K_\nu,
\]
where $c^\nu_{\lambda, \mu}$ is given by the number of increasing shifted skew tableaux $R$ of shape $\nu/\lambda$ such that $P_{SK(\text{row}(R))} = T$. \hfill 39
Proof. In addition to \(T\), fix a URT \(T'\) of shape \(\lambda\). Then by Theorems 6.5 and 8.2, we have

\[
K_\lambda K_\mu = \phi([[\text{row}(T')]]) \phi([[\text{row}(T)]])
\]

\[
= \phi([[\text{row}(T')]] \cdot [[\text{row}(T)]]))
\]

\[
= \sum_{R \in T(T' \cup T') \cup T(T)} \sum_{P_{SK}(w) = R} w
\]

\[
= \sum_{R \in T(T' \cup T')} K_{\lambda(R)},
\]

where \(T(T' \cup T)\) is the set of shifted tableaux \(R\) such that \(R_{[|\lambda|]} = T'\) and \(P_{SK}(\text{row}(R_{[|\lambda|+1,|\lambda|+\mu]})) = T\), giving our result.

Theorem 9.2. Let \(T_0\) be a URT of shape \(\nu\). Then

\[
\Delta(K_\nu) = \sum_{\lambda, \mu} d_{\lambda, \mu}^\nu K_\lambda \otimes K_\mu
\]

where \(d_{\lambda, \mu}^\nu\) is the number of ordered pairs of increasing shifted tableaux \(T', T''\), of shapes \(\lambda, \mu\) respectively, such that \(P_{SK}(\text{row}(T') \text{row}(T'')) = T_0\).

Proof. We have

\[
\Delta(K_\nu) = \Delta(\phi([[\text{row}(T_0)]]))
\]

\[
= (\phi \otimes \phi)(\Delta([[\text{row}(T_0)]]))
\]

\[
= (\phi \otimes \phi) \left( \sum_{\langle T', T'' \rangle \in T(T_0)} \sum_{P_{SK}(w) = \text{std}(T')} w \otimes \sum_{P_{SK}(w) = \text{std}(T'')} w \right)
\]

\[
= (\phi \otimes \phi) \left( \sum_{\langle T', T'' \rangle \in T(T_0)} [[\text{row}(\text{std}(T'))]] \otimes [[\text{row}(\text{std}(T''))]] \right)
\]

\[
= \sum_{\langle T', T'' \rangle \in T(T_0)} K_{\lambda(T')} \otimes K_{\lambda(T'')}
\]

where \(T(T_0)\) is the finite set of pairs of shifted tableaux such that \(P_{SK}(\text{row}(T') \text{row}(T'')) = T_0\). Our result follows.

This gives something resembling the Littlewood-Richardson given in [4]. Indeed we get the same structure constants, suggesting that our \(K_{\lambda}\) functions are somehow representative of the Schubert structure sheaves \([O_{\lambda}]\) discussed in [3] and [4].

During the REU, we conjectured that the \(K_{\lambda}\) were symmetric functions. For the purposes of this REU report, we leave this as a conjecture. The result with proof will appear in the arXiv preprint thanks to Zach Hamaker.

Conjecture 9.3. For any \(\lambda\), \(K_{\lambda}\) is symmetric.

10. Acknowledgments

This research was part of the 2015 REU program at the University of Minnesota, Twin Cities, and was supported by RTG grant NSF/DMS-1148634. We would like to thank
Rebecca Patrias, Zachary Hamaker, Pasha Pylyavskyy, and Vic Reiner for their mentorship and support.

References