Bumping algorithm for set-valued shifted tableaux
Takeshi Ikeda, Hiroshi Naruse, Yasuhide Numata

To cite this version:
Bumping algorithm for set-valued shifted tableaux

Takeshi IKEDA$^1$, Hiroshi NARUSE$^2$ and Yasuhide NUMATA$^{34}$

1Department of Applied Mathematics, Okayama University of Science, Ridai-cho, Kita-ku, Okayama, Japan
2Graduate School of Education, Okayama University, Tsushima-naka, Kita-ku, Okayama, Japan
3Department of Mathematical Informatics, The University of Tokyo, Hongo, Bunkyo-ku, Tokyo, Japan
4Japan Science and Technology (JST) CREST

Abstract. We present an insertion algorithm of Robinson–Schensted type that applies to set-valued shifted Young tableaux. Our algorithm is a generalization of both set-valued non-shifted tableaux by Buch and non set-valued shifted tableaux by Worley and Sagan. As an application, we obtain a Pieri rule for a $K$-theoretic analogue of the Schur $Q$-functions.

Résumé Nous présentons un algorithme d’insertion de Robinson–Schensted qui s’applique aux tableaux décalés à valeurs sur des ensembles. Notre algorithme est une généralisation de l’algorithme de Buch pour les tableaux à valeurs sur des ensembles et de l’algorithme de Worley et Sagan pour les tableaux décalés. Comme application, nous obtenons une formule de Pieri pour un analogue en $K$-théorie des $Q$-fonctions de Schur.

Keywords: set-valued shifted tableaux, insertion, Robinson–Schensted, Pieri rule, $K$-theory, Schur $Q$-functions

1 Introduction

This article is an extended abstract of the paper [INN] of the same title. Most details of the proofs are omitted.

In [IN], we introduced a non-homogeneous ($K$-theoretic) analogue of Schur $Q$-functions. These functions are labeled by strict partitions (or shifted Young diagrams), as are the original $Q$-functions. For a strict partition $\lambda$, the corresponding $K$-theoretic Schur $Q$-function $G_Q(\lambda)(x)$ can be expressed as a weighted generating function of shifted set-valued semistandard tableaux of shape $\lambda$, which are the central concern of this article.

The main result of the paper is a Robinson–Schensted type insertion algorithm for the shifted set-valued tableaux (Thm 3.4). Our algorithm is a generalization of both set-valued non-shifted tableaux by Buch [Bu] and non set-valued shifted tableaux by Worley [Wo] and Sagan [Sa]. As an immediate consequence of our algorithm, we have a Pieri rule for $G_Q(\lambda)(x)$ (Cor. 3.5).

The original purpose for introducing functions $G_Q(\lambda)(x)$ was to apply them to Schubert calculus. In [IN] we introduced function $G_Q(\lambda|x)b$ (resp. $GP_Q(\lambda|x)b$) with the equivariant parameter $b = (b_1, b_2, \ldots)$.
which represents the structure sheaf of the Schubert variety indexed by \( \lambda \) in the \( K \)-ring of \( T \)-equivariant coherent sheaves on Lagrangian (resp. orthogonal) Grassmannian, where \( T \) is the maximal torus acting on the Grassmannians. Thus our Pieri rule gives an explicit description of \( K \)-theoretic Schubert structure constant for an arbitrary Schubert class times a special (one row type) Schubert class in the \( K \)-ring of Lagrangian Grassmannian.

Recently, a \( K \)-theoretic Littlewood-Richardson rule in terms of the jeu de taquin for odd orthogonal Grassmannians of maximal isotropic subspaces has been obtained by Clifford, Thomas and Yong \cite{CTY}. Their method starts from a Pieri rule for the \( K \)-theory by Buch and Ravikumar \cite{BR}, which applies to cominuscule Grassmannians. Our approach differs from them substantially. We proceeded independently a different approach of tableaux insertion to result in the same formula as \cite{BR}, i.e. the counting of KLG-tableaux. But our method is only applicable to the case of Lagrangian Grassmannians, although there is a set valued tableaux description for \( GP_{\lambda}(x) \).

Organization of the paper is as follows. In Section 2, we give the definition of shifted set-valued tableaux, and \( K \)-theoretic Schur \( Q \)-functions \( GP_{\lambda}(x) \). In Section 3, we present our main result, an existence of a Robinson–Schensted type bijection for set-valued shifted tableaux. As a corollary, we have a Pieri rule for \( GP_{\lambda}(x) \). Precise description of the bijection is given by a bumping algorithm which is given in Section 4. In Section 5, we discuss a variant of the bijection, which is analogous to the results by Sagan and Worley. In Section 6, we give an outline of the proof of the main theorem.
Let $X$ denote the set of non-empty finite subsets of $B$. We extend the relations $\leq_r, \leq_c$ on $X$ by $A \leq_r B \iff \max A \leq_r \min B$ and $A \leq_c B \iff \max A \leq_c \min B$ for $A, B \in X$.

**Definition 2.1 (Shifted set-valued semistandard tableaux)** Let $\lambda$ be a shifted Young diagram. A set-valued semistandard tableau of shape $\lambda$ is a map $T$ from the set of boxes in $\lambda$ to $X$ satisfying the following “semistandardness”:

1. $T(\alpha) \leq_r T(\beta)$ if $\beta \in \lambda$ is directly right of $\alpha \in \lambda$.
2. $T(\alpha) \leq_c T(\beta)$ if $\beta \in \lambda$ is directly below $\alpha \in \lambda$.

**Example 2.2** An example of a set-valued tableau is given by the following:

$$T = \begin{array}{cccc}
1' & 2 & 3' & 4 \\
2' & 4' & 6 \\
6 & 
\end{array}$$

We denote by $T(\lambda)$ the set of all set-valued tableaux of shape $\lambda$.

### 2.3 $K$-theoretic $Q$-Schur functions

Let $x = (x_1, x_2, \ldots)$ be a sequence of variables. Let $\lambda \in S$ and $T \in T(\lambda)$. We define the corresponding monomial $x_T = \prod_{i=1}^{\infty} x_i^{e_i(T)}$ where $e_i(T)$ denotes the total number of $i$ and $i'$ appearing in $T$. The weight of $T \in T(\lambda)$ is defined to be $\beta^{\{T\}-|\lambda|}x^T$, where $\beta$ is a formal parameter and $|T|$ is the total number of letters in $T$. The $K$-theoretic $Q$-Schur function $GQ_{\lambda}(x)$ is defined as the following formal sum of the weights of the elements in $T(\lambda)$:

$$GQ_{\lambda}(x) = \sum_{T \in T(\lambda)} \beta^{\{T\}-|\lambda|}x^T.$$  

When $\beta = 0$ this becomes the Schur $Q$-function $Q_{\lambda}(x)$, and when $\beta = -1$ this represents $K$-theory Schubert class corresponding to $\lambda$ for Lagrangian Grassmannians. See [IN] for other expressions of $GQ_{\lambda}(x)$ and geometric background.

### 3 Statements of main results

#### 3.1 Admissible strips

Let $\theta = \lambda/\mu$ be a broken border strip. We consider a decomposition $\theta = C \sqcup C'$, with $C, C'$ skew diagrams, i.e. there is a diagram $\nu$ satisfying $\mu \subset \nu \subset \lambda$ and $C = \lambda/\nu$ and $C' = \nu/\mu$. Such a decomposition of $\theta$ is called admissible if the following conditions are satisfied:

1. in each of the diagrams $C$ and $C'$, there is no pair of boxes in the same row or column.
2. there is no diagonal box in $C'$.

A non-empty broken border strip $\theta$ is called a 1-admissible strip if there exists an admissible decomposition of $\theta$. For a 1-admissible strip $\theta$, we denote by $C(\theta)$ the set of all admissible decompositions of $\theta$. Later we define the notion of $m$-admissible decomposition of a broken border strip.
Example 3.1 The following is an example of a 1-admissible strip and its 1-admissible decomposition,

\[
\begin{array}{ccccccc}
 & & & & & & 1' \\
 & & & & & 1' & 1 \\
 & & & & 1 & & \\
 & & & 1 & & & \\
\end{array}
\]

where the boxes with entry 1’s form \( C \) and 1’’s form \( C' \).

The next result shows the role of 1-admissible strip. The detailed construction of the map is given in Section 4. We define the weight of a 1-admissible strip \( \theta \) to be \( \beta|\theta|-1 \).

**Proposition 3.2** There is a weight preserving bijection:

\[
\phi : \mathcal{T}(\lambda) \times \mathcal{X} \rightarrow \bigsqcup_{\mu} \mathcal{T}(\mu) \times \mathcal{C}(\mu/\lambda)
\]

where \( \mu \in \mathcal{S} \) runs for those \( \mu \) such that \( \mu/\lambda \) is a 1-admissible strip.

3.2 Composable admissible strips

Let \( \lambda, \mu, \nu \in \mathcal{S} \) be such that \( \mu \subset \nu \subset \lambda \). Suppose \( \theta_1 = \nu/\mu, \theta_2 = \lambda/\nu \) are 1-admissible strips. Let \((C'_i, C_i) \in \mathcal{C}(\theta_i) (i = 1, 2)\). We say that \((C'_1, C_1)\) precedes \((C'_2, C_2)\) and denote \((C'_1, C_1) < (C'_2, C_2)\), if the following conditions are satisfied:

1. \( C'_1 \cup C'_2 \) is a vertical strip.
2. \( C_1 \cup C_2 \) is a horizontal strip.
3. Each box in \( C'_2 \) is strictly below any box in \( C'_1 \).
4. Each box in \( C_2 \) is strictly right of any box in \( C_1 \).
5. If \( C_1 \neq \emptyset \), then \( C'_2 = \emptyset \).

3.3 Main results

Let \( \theta = \mu/\lambda \) be a broken border strip, and \( m \) be a positive integer. Suppose there is a nested sequence of shifted diagrams

\[
\lambda = \nu^{(0)} \subset \nu^{(1)} \subset \nu^{(2)} \subset \cdots \subset \nu^{(m)} = \mu
\]

such that \( \theta^{(i)} := \nu^{(i)}/\nu^{(i-1)} (1 \leq i \leq m) \) are 1-admissible strips. If, moreover, there is a sequence of 1-admissible decompositions \((C'_i, C_i) \in \mathcal{C}(\theta^{(i)}) (1 \leq i \leq m)\) such that

\[
(C'_i, C_i) < (C'_{i+1}, C_{i+1}), \quad (1 \leq i \leq m-1).
\]

then we say \( \theta \) is an \( m \)-admissible strip. For an \( m \)-admissible strip \( \theta \), let \( \mathcal{C}_m(\theta) \) denote the set of pairs \((\{\nu^{(i)}\}_{i=1}^m, \{(C'_i, C_i)\}_{i=1}^m)\) satisfying the above conditions, which we call \( m \)-admissible decompositions of \( \theta \). Note \( \mathcal{C}_1(\theta) = \mathcal{C}(\theta) \) since condition (2) is vacant for \( m = 1 \).
Example 3.3 The following is a 4-admissible strip

```
  1' 4
  3
  1' 2 3
  2' 2
```

where the boxes with entry $i$ are $C_i$, and $i'$ are $C_i'$.

We denote by $(m)$ the shifted diagram consisting of one row with $m$ boxes. We simply denote $T((m))$ for $T((0))$. Recall that we define the weight of $T \in T(\lambda)$ as $\beta[|T|-|\lambda|]x^T$. Define the weight of $U \in \mathcal{C}_m(\theta)$ to be $\beta[|\theta|-m]$.

Theorem 3.4 By algorithm 4.4 we have a weight preserving bijection:

$$
\phi_m : T(\lambda) \times T((m)) \longrightarrow \bigsqcup_{\mu} T(\mu) \times \mathcal{C}_m(\mu/\lambda),
$$

where $\mu$ runs for shifted diagrams $\mu$ such that $\mu \supset \lambda$ and $\mu/\lambda$ are $m$-admissible strips.

As an immediate consequence, we have the following.

Corollary 3.5 (Pieri rule) We have

$$
GQ_\lambda(x) \cdot GQ_m(x) = \sum_{\mu \supset \lambda} \beta[|\mu|-|\lambda|-m] \#\mathcal{C}_m(\mu/\lambda) \times GQ_\mu(x),
$$

where $\mu$ runs for shifted diagrams $\mu$ such that $\mu \supset \lambda$ and $\mu/\lambda$ are $m$-admissible strips.

For example we have

$$
GQ_{2,1} \cdot GQ_2 = 2GQ_{4,1} + 2GQ_{3,2} + 3\beta GQ_{4,2} + \beta GQ_{5,1} + \beta GQ_{3,2,1} + \beta^2 GQ_{5,2} + \beta^2 GQ_{4,2,1}.
$$

In order to give the coefficient of $GQ_{4,2}$, we count the elements in $\mathcal{C}_2(\mu/\lambda)$ with $\mu = (4,2), \lambda = (2,1)$:

```
  1' 2
  2' 1
```

N.B. The elements in $\mathcal{C}_m(\mu/\lambda)$ are exactly the $KLG$-tableaux of shape $\mu/\lambda$ with content $\{1, 2, ..., m\}$ in [BR].

4 Bumping algorithm

The aim of this section is to describe the bijection of Prop 3.2.

The input of our algorithm is a pair $(T, w)$ with $T \in T(\lambda)$ for some $\lambda \in \mathcal{S}$ and $w \in \mathcal{X}$. Basic output is a tableau $T'$ of some shape $\mu \in \mathcal{S}$ such that $\mu \supset \lambda$. The skew diagram $\theta = \mu/\lambda$, the set of “new boxes”, turns out to be a 1-admissible strip. We also have some “recording data” on $\theta$ which gives an element of $\mathcal{C}(\theta)$.
4.1 Parts of “L” shape of a tableau

Let $\lambda \in \mathcal{S}$. Let $\ell(\lambda)$ be the number of rows of $\lambda$. For $1 \leq t \leq \lambda_1$ we define a subset of $\lambda$ by

$$L_t(\lambda) = \{(i, j) \in \lambda \mid i = t \text{ or } j = t\}.$$ 

For example, $L_1(\lambda)$ consists of the boxes in the first row. For $k \geq \ell(\lambda)$, $L_k(\lambda)$ is just the $k$-th column. In general, this is a subset of shape “L” including the diagonal box $(t, t)$. Let $T \in \mathcal{T}(\lambda)$. By restriction we have a map $L_t(T): L_t(\lambda) \rightarrow \mathcal{X}$, which we call the $t$-th part of $T$.

Our algorithm starts from inserting $w = w^{(0)} \in \mathcal{X}$ into $L_1 = L_1(T)$, the first row of $T$, resulting a row $L_1'$ with possibly a new box at the right end, and a set $w^{(1)} \in \mathcal{X}$ “bumped out” from the procedure. Then we modify the original tableau $T = T^{(0)}$ by replacing $L_1$ with $L_1'$ to obtain $T^{(1)}$. Next we insert $w^{(1)}$ into the second part of the modified tableau $T^{(2)}$. We repeat this procedure until no boxes are bumped out.

4.2 Insertion into a part of “L” shape (a rough idea)

We define a procedure to insert some sets $w \in \mathcal{X}$ into an L part $X$ of a tableaux.

Here we present a rough idea of constructing the procedure. First, we look at the minimum letters of each boxes in order to decide the box into which a letter in $w$ to be inserted, in the same manner as the classical bumping procedure (some letters go into empty box at the end). If we might simply insert these letters into $X$, some letters in $w$ may violate the semistandardness, while some letters are not. So we eject some element in $X$ before inserting $w$. Let $\hat{w}$ be the set of letters in $w$ which do not conflict any original letters in $X$, and let $\hat{w} := w - \hat{w}$ be the complement. If $\hat{w} \neq \emptyset$, let $\hat{u}$ be the set of elements in $X$ that conflict some element in $\hat{w}$. To ensure the semistandardness, we first eject the elements in $\hat{u}$ from the tableau. Furthermore, if a letter in $\hat{w}$ is inserted into a non-empty box, we eject all the remaining (original) entries of the box. Thus any letter inserted into a non-empty box “does some work” (bumps out at least one letter). This feature is important for constructing the inverse algorithm.

There is a flaw in this idea. For example, we consider a tableau $T = \begin{bmatrix} 1 \\ \end{bmatrix}$ and $w = w^{(1)} = \{1'\}$. According to the naive algorithm above, the resulting tableau is $T^{(1)} = \begin{bmatrix} 1' \\ \end{bmatrix}$ and the ejected set is $w^{(2)} = \{1'\}$. Since the second part is empty, the final result is $\begin{bmatrix} 1' & 1' \end{bmatrix}$, which is not semistandard. This is a reason why we need the “unmark” process introduced in the next section. In fact, we should care for the case of inserting elements into the diagonal boxes.

4.3 Insertion into a diagonal box

Let $X \in \mathcal{X}$, and $u$ be a subset of $X$. We insert $w \in \mathcal{X}$ into $X$, where we consider $X$ to be a diagonal box.

Algorithm 4.1 (Bumping for a diagonal box)

input $X, w, u \in \mathcal{X}$ satisfying $u \subset X$ and $\max w \leq \min X$.

output $Y, v$.

procedure

1. If $X \neq u$, then let $Y = (X - u) \cup w$ and $v = u$; and return $Y, v$.
2. If $i' = \max(w) \in A'$ and $i \in X$, $i' \notin X$, then let $Y = \{ i \} \cup (w - \{ i' \})$ and $v = X$; and return $Y, v$. 


3. If \( i' = \max(w) \in A' \) and \( i' \in X, i \notin X \), then let \( Y = w \) and \( v = \{ i \} \cup (X - \{ i' \}) \); and return \( Y, v \).

4. If \( i' = \max(w) \in A' \) and \( i, i' \in X \), then let \( Y = \{ i \} \cup w \) and \( v = X - \{ i' \} \); and return \( Y, v \).

5. Otherwise, let \( Y = w \) and \( v = X \); and return \( Y, v \).

For example, if \( u = X = \begin{bmatrix} 3 \end{bmatrix} \) and \( w = 13' \), then we apply (2) to obtain \( Y = \begin{bmatrix} 4 \end{bmatrix} \) rather than \( \begin{bmatrix} 3 \end{bmatrix} \), and \( u = 34 \). Thus letter 3' is unprimed to be 3 in \( u \). If \( u = X = \begin{bmatrix} 3 \end{bmatrix} \) and \( w = 13' \), then we apply (3) to obtain \( Y = \begin{bmatrix} 4 \end{bmatrix} \) and \( u = 34 \), rather than \( u = 3'4 \). In this case, two 3' are involved, and one may think of this process as unpriming "bigger" 3'. Case (4) is a bit strange. If \( u = X = \begin{bmatrix} 3 \end{bmatrix} \) and \( w = 3' \), then we have \( Y = \begin{bmatrix} 4 \end{bmatrix} \) and \( u = 3 \). This case we are unpriming "bigger" 3' also, and let it remain in the box.

4.4 Insertion into a part of “L” shape (definition)

Let \( T \) be a tableau of shape \( \lambda \), and \( t \) be a positive integer such that \( t \leq \lambda_1 \). Let \( X = L_t(T) \) be the \( t \)-th part of \( T \). If \( t = 1 \), then \( X \) is a row: \( X = (X_{(1,1)} \leq_r X_{(1,2)} \leq_r \cdots \leq_r X_{(1,\lambda_1)}) \). If \( t > \ell(\lambda) \) then \( X \) is a column: \( X = (X_{(1,t)} \leq_c \cdots \leq_c X_{(k,t)}) \) for some \( k < t \). We say that \( X \) is a pure column in this case (note that \( X \) does not contain diagonal box). If \( 1 < t \leq \ell(\lambda) \) then \( X = L_t(T) \) is a sequence of elements in \( \mathcal{X}^{+} \):

\[
X = (X_{(t,1)} \leq_c \cdots \leq_c X_{(t-1,t)} \leq_c X_{(t,t)} \leq_r X_{(t,t+1)} \leq_r \cdots \leq_r X_{(t+t_2-1,1)}).
\]

The following algorithm takes as an input a sequence of elements in \( \mathcal{X}^{+} \) satisfying

\[
X = (X_{-k} \leq_c \cdots \leq_c X_{-1} \leq_c X_0 \leq_r X_1 \leq_r \cdots \leq_r X_l),
\]

for some \( k, l \geq 0 \), and \( w \in \mathcal{X} \). If \( k = 0 \), we consider \( X \) as a row. Output is a triple \((Y, Y_+, v)\), where \( Y \) is a sequence \( Y = (Y_{i})_{i=-k}^{l} \) satisfying the same condition as \( X \), and \( Y_+, v \in \mathcal{X} \cup \emptyset \). If \( Y_+ \neq \emptyset \) we will make a new box with entry \( Y_+ \) at the right end of \( Y \).

**Algorithm 4.2 (Bumping rule for an L part)**

**input** \( X = (X_{i})_{i=-k}^{l} \) : tableau of \( L \) shape, i.e.

\[
X = (X_{-k} \leq_c \cdots \leq_c X_{-1} \leq_c X_0 \leq_r X_1 \leq_r \cdots \leq_r X_l),
\]

and \( w \in \mathcal{X}^{+} \).

**output** \( Y \) tableau of \( L \) shape of the same length of \( X \), and \( Y_+, v \in \mathcal{X} \cup \emptyset \).

**procedure**

1. Define the subsets \( w_{-k}, \ldots, w_{l+1} \) of \( w \) by

\[
w_t = \begin{cases}
\{ x \in w \mid x \leq_r \min X_{-k} \} & (t = -k) \\
\{ x \in w \mid \min X_{t-1} \leq_c x \leq_r \min X_{t} \} & (t = -k, \ldots, -1) \\
\{ x \in w \mid \min X_{-1} \leq_c x \leq_r \min X_{0} \} & (t = 0) \\
\{ x \in w \mid \min X_{t-1} \leq_r x \leq_c \min X_{t} \} & (t = 1, \ldots, l) \\
\{ x \in w \mid \min X_{l} \leq_r x \} & (t = l + 1)
\end{cases}
\]
2. Decompose \( w_t \) into the subsets \( \hat{\mathcal{w}}_t \) and \( \check{\mathcal{w}}_t \) defined by

\[
\check{\mathcal{w}}_t = \begin{cases}
\{ x \in w_t | \max X_{t-1} \leq c \} & (t = -k) \\
\{ x \in w_t | \max X_{t-1} \leq r \} & (t = 1, \ldots, l+1)
\end{cases},
\]

\[
\hat{\mathcal{w}}_t = w_t - \check{\mathcal{w}}_t, \quad \text{for } t = -k, \ldots, l+1.
\]

3. Define \( \check{u}_t, \hat{u}_k, \) and \( u_k \) \((t = -k, \ldots, l)\) by:

\[
\check{u}_t = \begin{cases}
\emptyset & \text{(if } \hat{\mathcal{w}}_{t+1} = \emptyset) \\
\{ y \in X_t | y \not\leq \min \hat{\mathcal{w}}_{t+1} \} & \text{(if } t = -k, \ldots, -1 \text{ and } \hat{\mathcal{w}}_{t+1} \neq \emptyset) \\
\emptyset & \text{(if } t = 0, \ldots, l \text{ and } \hat{\mathcal{w}}_{t+1} \neq \emptyset) \\
X_t - \check{u}_t & \text{(if } \hat{u}_t \neq \emptyset)
\end{cases}
\]

\[
\hat{u}_t = \begin{cases}
\emptyset & \text{(if } \hat{u}_t = \emptyset) \\
\emptyset & \text{(if } \hat{u}_t \neq \emptyset)
\end{cases}
\]

\[
u_t = \check{u}_t \cup \hat{u}_t \subset X_t.
\]

4. Define \( Y_t = (X_t - u_t) \cup w_t \) and \( v_t = u_t \) for \( t \neq 0 \).

5. Let \((Y_0, v_0)\) be the pair obtained from the triple \((X_0, w_0, u_0)\) by Algorithm 4.1 if \( l \geq 0 \).

6. Let \( Y = (Y_{-k}, \ldots, Y_l), Y_+ = w_{l+1}, \) and \( v = \bigcup_{t=-k}^l v_t \); and return \( Y, Y_+, v \).

**Example 4.3** Let \( X = (X_{-2}, X_{-1}, X_0; X_1, X_2, X_3) \) be

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

Let us insert \( w = 25'6'79'9 \in X' \) into \( X \). Since the minimums in \( X \) is

\[
\begin{array}{cccccc}
1 & 4 & 7 & 8 & 9
\end{array}
\]

we have \((w_{-2}, \ldots, w_4) = (\emptyset, 2, 5', 6', 9', 9)\). Since the maximums of \( X \) is

\[
\begin{array}{cccccc}
3 & 4 & 5 & 6 & 8 & 9
\end{array}
\]

we have

<table>
<thead>
<tr>
<th>( t )</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{\mathcal{w}}_t )</td>
<td>\emptyset</td>
<td>\emptyset</td>
<td>5'</td>
<td>\emptyset</td>
<td>7</td>
<td>9'</td>
<td>9</td>
</tr>
<tr>
<td>( \check{\mathcal{w}}_t )</td>
<td>\emptyset</td>
<td>2</td>
<td>\emptyset</td>
<td>\emptyset</td>
<td>6'</td>
<td>\emptyset</td>
<td>\emptyset</td>
</tr>
<tr>
<td>( \check{u}_t )</td>
<td>\emptyset</td>
<td>\emptyset</td>
<td>5</td>
<td>\emptyset</td>
<td>8</td>
<td>9</td>
<td>-</td>
</tr>
<tr>
<td>( \hat{u}_t )</td>
<td>3</td>
<td>\emptyset</td>
<td>6</td>
<td>\emptyset</td>
<td>\emptyset</td>
<td>\emptyset</td>
<td>-</td>
</tr>
</tbody>
</table>

Finally we get

\[
Y = \begin{array}{cccccc}
1 & 2 & 4 & 5 & 6 & 9
\end{array}, \quad Y_+ = \{ 9 \}, \quad u = \{ 3, 5, 6, 8, 9 \}.
\]

We need to define the bumping algorithm applicable also when \( X = L_t(T) \) is a pure column case, i.e. \( t > \ell(\lambda) \). However, extension of the algorithm to the column case is straightforward, so we omit detailed description here.
4.5 Insertion of $w$ into arbitrary tableau

We define a procedure to insert an element $w \in \mathcal{X}$ into an arbitrary tableau $T$. In the procedure, we insert $w$ into the first L part of the tableaux. When some letters are bumped out, we insert them into the second L part of the tableau. Then, while some letters are bumped out, we try to insert them into the next L part of the tableau until no letters are bumped out.

Algorithm 4.4

input $T \in \mathcal{T}(\lambda)$ and $w \in \mathcal{X}$.

output $U$, $S'$, $S$.

procedure

1. Let $u = w$, $U = T$, $S = \emptyset$ and $S' = \emptyset$.
2. While $u \neq \emptyset$, do the following:
   (a) Let $X$ be the $t$-th L part of $U$,
   (b) Let $(Y, Y_+, u)$ be the triple obtained from $(X, u)$ by Algorithm 4.2
   (c) Let $U$ be the tableaux obtained from $U$ by replacing the $t$-th L part by $Y$.
   (d) If $Y_+ \neq \emptyset$, then do the following:
      i. Add a new box to the end of $t$-th L part of $U$, and insert $Y_+$ into the box.
      ii. If $X$ is a pure column, then add the new box to $S$, else add the new box to $S'$.

Example 4.5 Let $T$ be the leftmost tableau below. We insert $w = \{1', 1, 2', 3\}$ into $T$ as follows.

\[
\begin{array}{ccc}
\begin{array}{ccc}
1' & 12' & 23' \\
2' & 3 & 3
\end{array} & \rightarrow & \\
\begin{array}{ccc}
1' & 1 & 2'23' & 3 \\
2' & 3 & 3
\end{array} & \rightarrow & \\
\begin{array}{ccc}
1' & 1 & 12'23' & 3 \\
2' & 3 & 3
\end{array} & \rightarrow & \\
\begin{array}{ccc}
1' & 1 & 1 & 3 \\
2' & 23 & 3
\end{array}
\end{array}
\]

$u = 1'2'3$ $u = 12'$ $u = 123$ $u = 12'23'$

\[
\begin{array}{ccc}
\begin{array}{ccc}
1' & 1 & 1 & 2'23' & 3 \\
2' & 23 & 3
\end{array} & \rightarrow & \\
\begin{array}{ccc}
1' & 1 & 1 & 2'23' & 3 \\
2' & 23 & 3
\end{array}
\end{array}
\]

$u = 3$ $u = 0$

For each step, the relevant part of modification is enclosed.

Sets $S'$ and $S$ are as follows:

\[
\begin{array}{ccc}
\begin{array}{cc}
1' & 1
\end{array} & \rightarrow & \\
\begin{array}{cc}
1' & 1
\end{array}
\end{array}
\]

where the box with entry 1' (resp. 1) is $S'$ (resp. $S$).
4.6 Definition of the map \( \phi \)

In order to complete the description of the map \( \phi \), we need one more combinatorial idea. Let \( \theta \) be a 1-admissible strip. We define an involution \( \varrho : C(\theta) \to C(\theta) \). A box \( \alpha \in \theta \) is said to be isolated if \( \alpha \) is not a diagonal box and there is no other box than \( \alpha \) in the row and column where \( \alpha \) presents. For each isolated box, apply its entry the obvious involution \( 1 \mapsto 1' \), \( 1' \mapsto 1 \), while the non-isolated boxes are untouched. The resulting decomposition of \( \theta \) is obviously admissible. For example, we have

\[
\begin{array}{cccc}
1 & 1' & 1 & 1' \\
1' & 1 & 1 & 1 \\
1 & 1 & 1 & 1
\end{array}
\xrightarrow{\varrho}
\begin{array}{cccc}
1 & 1 & 1' & 1' \\
1' & 1 & 1' & 1 \\
1 & 1 & 1 & 1
\end{array}
\]

It is obvious that \( \varrho \) is an involution.

**Proposition 4.6** Let \( \lambda \in \mathbb{S} \), \( T \in T(\lambda) \), and \( w \in X = T(1) \). We have by Algorithm 4.4 a tableau \( U = (T \leftarrow w) \in T(\mu) \) for some \( \mu \in \mathbb{S} \) such that \( \mu \supset \lambda \) and a decomposition \((S', S)\) of \( \theta = \mu/\lambda \). We have \((S', S) \in C(\theta)\), and therefore \( \theta \) is a 1-admissible strip.

Let \( T \in T(\lambda) \) and \( w \in X \) as in the above proposition. We define \( \phi(T, w) \) to be \((U, \varrho(S', S)) \in T(\mu) \times C(\mu/\lambda)\).

4.7 Proof of Prop. 3.2

To show that \( \phi \) is a bijection, we construct its inverse map. See [INN] for details.

5 Robinson–Schensted type correspondence

5.1 Quasi-standard tableaux

We will define a notion of “recording” tableaux in our setting. The resulting object is an analogue of a standard tableau, which we will call a quasi-standard tableau.

For \( T \in T(\lambda) \) and \( w \in X \) we denote by \( T \leftarrow w \) the tableau given in Prop. 3.2. Let \( T \in T(\lambda) \) and \((w_1, \ldots, w_m) \in X^m\). By the consecutive insertions

\[
T^{(i)} = (\cdots ((T \leftarrow w_1) \leftarrow w_2) \cdots \leftarrow w_i)
\]

we have a tableaux \( T^{(i)} \in T(\nu^{(i)}) \) for some shifted diagram \( \nu^{(i)} \) and an element of \( C(\nu^{(i)}/\nu^{(i-1)}) \) given by Proposition 3.2. Thus we have a nested sequence of shifted diagrams

\[
\lambda = \nu^{(0)} \subset \nu^{(1)} \subset \nu^{(2)} \subset \cdots \subset \nu^{(m)} = \mu,
\]

and also 1-admissible decompositions \((C'_i, C_i)\) of \( \theta^{(i)} = \nu^{(i)}/\nu^{(i-1)} \). These objects are expressed as a tableau like

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

\[
\begin{array}{cccccccc}
& & & & & & & \\
& & & & & & 1' & 2' 4 \\
& & 1' & 2' & 3' 4' 4 \\
1' & 1' & 2' & 3' & 4' & 4
\end{array}
\]
where the boxes filled with $i$ (resp. $i'$) are $C_i$ (resp. $C_i'$).

We call such a tableau a quasi-standard tableau of degree $m$. The precise definition is the following.

**Definition 5.1** A map $U : \mu/\lambda \longrightarrow B_m := \{1', 1, \ldots, m', m\}$ is a quasi-standard tableau of degree $m$, if $U$ is semistandard in the sense of Def. 2.1 and for any $1 \leq i \leq m$, $U^{-1}(\{i, i'\})$ is a 1-admissible strip with admissible decomposition given by $(U^{-1}(i'), U^{-1}(i))$.

Let $S_m(\mu/\lambda)$ denote the set of quasi-standard tableaux of degree $m$ on $\mu/\lambda$.

**Remark.** By the construction, $S_1(\mu/\lambda)$ is non-empty if and only if $\theta = \mu/\lambda$ is a 1-admissible strip. Then we have $S_1(\theta) = C(\theta) = C_1(\theta)$. For an $m$-admissible strip $\theta$, the set $C_m(\theta)$ is a subset of $S_m(\theta)$.

### 5.2 Robinson–Schensted correspondence

The following result is an immediate consequence of Prop. 3.2.

**Proposition 5.2** Let $T \in \mathcal{T}(\lambda)$ and $(w_1, \ldots, w_m) \in X^m$. By consecutive insertions

$$T' = ((T \leftarrow w_1) \leftarrow w_2) \cdots \leftarrow w_m$$

we have a tableaux $T' \in \mathcal{T}(\mu)$ for some shifted diagram $\mu \supset \lambda$ and the recording tableau $U$. Then we have $U \in S_m(\mu/\lambda)$. By this correspondence we have a weight preserving bijection

$$\phi_m : \mathcal{T}(\lambda) \times X^m \longrightarrow \bigsqcup_{\mu} \mathcal{T}(\mu) \times S_m(\mu/\lambda),$$

where the sum runs for shifted diagrams $\mu$ such that $S_m(\mu/\lambda) \neq \emptyset$.

Then we have immediately the following:

**Corollary 5.3** We have

$$GQ_\lambda(x) \cdot GQ_1(x)^m = \sum_{\mu} \beta_{|\mu/\lambda|-m} S_m(\mu/\lambda) \times GQ_\mu(x),$$

where the sum runs for shifted diagrams $\mu$ such that $S_m(\mu/\lambda) \neq \emptyset$.

As a special case of $\lambda = \emptyset$, we have the following.

**Corollary 5.4 (Robinson–Schensted correspondence)** There is a weight preserving bijection

$$X^m \longrightarrow \bigsqcup_{\lambda} \mathcal{T}(\lambda) \times S_m(\lambda).$$

This bijection is a set-valued extension of the results in [Sa] and [Wo].

**Example 5.5** Let $(w_1, w_2, w_3) = (2', 12', 23')$. By the correspondence in Cor. 5.3, we have pair of tableaux

$$\begin{pmatrix} 1 & 1 & 2 & 2 & 3 \end{pmatrix}, \quad \begin{pmatrix} 1 & 2' & 2 & 3 & 3 \end{pmatrix},$$

as a result of bumping process:
6 Outline of proof of Thm 3.4

Now we have the bijection $\phi_m$ in Prop. 5.2. Since a tableau in $T(m)$ is a sequence in $X$ such that

$$X_1 \leq r \cdots \leq r X_m,$$

we can think of $T(m)$ as a subset of $X^m$. Thus we only need to determine the image of $T(\lambda) \times T(m)$ under the map $\phi_m$. The case $m = 1$ is obvious since $T(1) = X$. The case $m = 2$ is crucial.

Lemma 6.1 Let $T \in T(\lambda)$ and $w = (w_1, w_2) \in X^2$, and

$$\phi_2(T, w) = (T', (C'_1, C_1), (C'_2, C_2)).$$

Then the following are equivalent:

1. $w_1 \leq w_2$.
2. $(C'_1, C_1) \triangleleft (C'_2, C_2)$.

It is easy to see that the lemma leads to a proof of Thm 3.4. We show this lemma by an argument using “bumping routes”. Details are given in [INN].

References


[CTY] E. Clifford, H. Thomas, A. Yong, $K$-theoretic Schubert calculus for $OG(n, 2n + 1)$ and jeu de taquin for shifted increasing tableaux, arXiv:1002.1664v2


