A ROW ANALOGUE OF HECKE COLUMN INSERTION

Daoji Huang*[1], Mark Shimozono[2], and Tianyi Yu[3]

[1]*School of Mathematics, University of Minnesota, Minneapolis, MN, U.S.A.*
huan0664@umn.edu
[2]*Department of Mathematics, Virginia Tech, Blacksburg, VA, U.S.A.*
mshimo@vt.edu
[3]*Department of Mathematics, University of California, San Diego, San Diego, CA, U.S.A.*
tiy059@ucsd.edu

Submitted: Mar 12, 2023; Accepted: May 21, 2024; Published: Sep 30, 2024

Abstract. We introduce a new row insertion algorithm on decreasing tableaux and increasing tableaux, generalizing Edelman–Greene (EG) row insertion. Our row insertion algorithm is a nontrivial variation of Hecke column insertion which generalizes EG column insertion. Similar to Hecke column insertion, our row insertion is bijective and respects Hecke equivalence, and therefore recovers the expansions of stable Grothendieck functions into Grassmannian stable Grothendieck functions.

Keywords. Hecke insertion, Grothendieck polynomials

Mathematics Subject Classifications. 05E05

1. Introduction

Edelman–Greene (EG) insertion was introduced to give combinatorial expansions of Stanley symmetric functions into Schur functions [EG87]. EG insertion achieves this expansion because it respects Coxeter–Knuth (nilplactic) equivalence (an equivalence relation on reduced words for permutations) and satisfies a Pieri rule which guarantees that the recording tableau is semistandard. EG insertion comes in four flavors depending on the use of increasing versus decreasing tableaux and row versus column insertion. The four flavors are essentially the same: the row and column versions are related by the naive transpose and the increasing and decreasing versions are related by reversing the total order on entries. For particular applications one might require a specific flavor. For the Stanley-to-Schur expansions both EG column insertion and EG row insertion can be used. But there is a subtlety here. To get a semistandard recording tableau for EG column insertion, one must use a certain kind of biword as input. Using the same biword for EG row insertion results in a recording tableau which is the *transpose* of semistandard. One

*Supported by NSF-DMS2202900.

must use a different kind of input biword. The transformation between the two kinds of input biwords involves reversing the reduced word.

Hecke column insertion was introduced in [BKS+08] to give combinatorial expansions of stable Grothendieck functions G_w into stable Grassmannian Grothendieck functions G_λ. Hecke column insertion realizes these expansions because it respects Hecke equivalence and satisfies a Pieri property which guarantees that the recording tableaux be set-valued. The increasing and decreasing versions of Hecke column insertion generalize the two flavors of EG column insertion.

Let us consider the problem of generalizing the two kinds (increasing/decreasing tableaux) of EG row insertion while respecting Hecke equivalence. The naive transpose of Hecke column insertion respects Hecke equivalence and directly generalizes EG row insertion but satisfies a Pieri property which implies that the recording tableau is the *transpose* of a set-valued tableau. This also cannot be fixed by transforming the input biword; operations such as reversal will not even recognizably transform the shape of the output tableau.

Our new insertion generalizes EG row insertion, respects Hecke equivalence and also satisfies the Pieri property which produces set-valued tableaux (as opposed to the transpose of set-valued tableaux). This novel insertion has the very unusual property that some values may be moved which are not part of the bumping path. One application of this insertion is the expansion of G_w times a Lascoux polynomial into Lascoux polynomials, which is not achievable by Hecke column insertion [OY23].

1.1. Various functions

The main application of these various insertion algorithms, is to expand the Stanley symmetric functions F_w and stable Grothendieck functions G_w. We define them combinatorially using words. A pair of words $(a_1 \cdots a_n, i_1 \cdots i_n)$ is called *compatible*[1] if $i_j \geqslant i_{j+1}$ and $i_j = i_{j+1}$ implies $a_j < a_{j+1}$ for all $1 \leqslant j < n$. Each word a has an associated permutation $[a]_H$ (see §2.2). We say a is a *Hecke word* for w if $[a]_H = w$. Let CP_w be the set of compatible pairs (a, i) such that a is a Hecke word for w. Let $\mathsf{CP}_w^{\mathrm{Red}}$ consist of $(a, i) \in \mathsf{CP}_w$ such that a is reduced (i.e. $\mathrm{len}(a) = \ell(w)$ where $\ell(w)$ is the Coxeter length). By [BJS93] and [FK94],

$$F_w = \sum_{(a,i)\in \mathsf{CP}_w^{\mathrm{Red}}} x^{\mathrm{wt}(i)}$$
$$G_w = \sum_{(a,i)\in \mathsf{CP}_w} (-1)^{|\mathrm{wt}(i)|-\mathrm{len}(w)} x^{\mathrm{wt}(i)} .$$

We define the Schur function s_λ and the stable Grassmannian Grothendieck function G_λ using tableaux. Let $\mathbb{Y}$ be the set of partitions. For $\lambda = (\lambda_1 \geqslant \lambda_2 \geqslant \cdots) \in \mathbb{Y}$, let $D(\lambda) = \{(i, j) \in \mathbb{Z}^2_{>0} \mid 1 \leqslant j \leqslant \lambda_i\}$ be its diagram under the English convention with matrix-style indexing. A set-valued tableau T of shape $\lambda \in \mathbb{Y}$ is a function which assigns to each $s \in D(\lambda)$ a nonempty finite subset of $\mathbb{Z}_{>0}$, such that if s' is immediately to the right (resp. below) s in the same row (resp. column) then $\max(T(s)) \leqslant \min(T(s'))$

[1] These compatible sequences are the reverse words of those defined in [BJS93].

(resp. $\max(T(s)) < \min(T(s'))$). We denote by SVT (resp. RSVT) the set of set-valued tableaux (resp. reverse set-valued tableaux, meaning all inequalities are reversed). Let SSYT (resp. RSSYT) denote the set of semistandard (resp. reverse semistandard) Young tableaux, meaning set-valued (resp. reverse set-valued) tableaux in which each set is a singleton. The following is a reverse-set-valued tableau of shape $(3,2)$.

5	$5,4$	3,2,1
$3,2$	2	

Then s_λ and G_λ each have two equivalent formulas ([Buc02] for G_λ):

$$s_\lambda = \sum_{\substack{Q\in\mathsf{SSYT}\\ \mathrm{shape}(Q)=\lambda}} x^{\mathrm{wt}(Q)} = \sum_{\substack{Q\in\mathsf{RSSYT}\\ \mathrm{shape}(Q)=\lambda}} x^{\mathrm{wt}(Q)}$$

$$G_\lambda = \sum_{\substack{Q\in\mathsf{SVT}\\ \mathrm{shape}(Q)=\lambda}} (-1)^{|\mathrm{wt}(Q)|-|\lambda|} x^{\mathrm{wt}(Q)} = \sum_{\substack{Q\in\mathsf{RSVT}\\ \mathrm{shape}(Q)=\lambda}} (-1)^{|\mathrm{wt}(Q)|-|\lambda|} x^{\mathrm{wt}(Q)}\,.$$

1.2. Expansions

The F_w (resp. G_w) can be expanded into s_λ (resp. G_λ). The expansion coefficients have geometric meaning; they contain all cohomological (resp. K-theoretic) equioriented type A quiver constants as special cases [BKS+08].

There are two ways to write down either of the two expansions, using either increasing tableaux or decreasing tableaux. For a permutation w, let Inc_w (resp. Dec_w) be the set of increasing (resp. decreasing) tableaux P whose row word $\mathrm{row}(P)$ (resp. reverse row word $\mathrm{rev}(\mathrm{row}(P))$; see §2.2) is a Hecke word for w. Let $\mathsf{Inc}_w^{\mathrm{Red}}$ (resp. $\mathsf{Dec}_w^{\mathrm{Red}}$) consists of tableaux in Inc_w (resp. Dec_w) whose row word is reduced. Then we have ([EG87] for F_w and [BKS+08] for G_w)

$$F_w = \sum_{P\in\mathsf{Inc}_w^{\mathrm{Red}}} s_{\mathrm{shape}(P)} \tag{1.1}$$

$$= \sum_{P\in\mathsf{Dec}_w^{\mathrm{Red}}} s_{\mathrm{shape}(P)}, \tag{1.2}$$

$$G_w = \sum_{P\in\mathsf{Inc}_w} (-1)^{\ell(w)-|\mathrm{shape}(P)|} G_{\mathrm{shape}(P)} \tag{1.3}$$

$$= \sum_{P\in\mathsf{Dec}_w} (-1)^{\ell(w)-|\mathrm{shape}(P)|} G_{\mathrm{shape}(P)}. \tag{1.4}$$

1.3. Insertion algorithms: General requirements

Let A and B be sets of tableaux of partition shape. We use the notation

$$A \times_{\mathbb{Y}} B = \{(P,Q) \in A \times B \mid \mathrm{shape}(P) = \mathrm{shape}(Q)\} \tag{1.5}$$

for the fiber product over the maps $A \to \mathbb{Y}$ and $B \to \mathbb{Y}$ given by taking the shape of a tableau.

To give a combinatorial proof of (1.3) it suffices to produce a bijection

$$\Phi_{IS} : \mathsf{CP}_w \to \mathsf{Inc}_w \times_{\mathbb{Y}} \mathsf{SVT} \quad \text{or} \quad \Phi_{IR} : \mathsf{CP}_w \to \mathsf{Inc}_w \times_{\mathbb{Y}} \mathsf{RSVT}$$

which is weight-preserving:

$$(a, i) \mapsto (P, Q) \tag{1.6}$$
$$\mathrm{wt}(i) = \mathrm{wt}(Q). \tag{1.7}$$

Similarly to prove (1.4) it suffices to supply a weight-preserving bijection

$$\Phi_{DS} : \mathsf{CP}_w \to \mathsf{Dec}_w \times_{\mathbb{Y}} \mathsf{SVT} \quad \text{or} \quad \Phi_{DR} : \mathsf{CP}_w \to \mathsf{Dec}_w \times_{\mathbb{Y}} \mathsf{RSVT}.$$

1.4. Edelman–Greene insertion: solution for reduced case

Historically first to be discovered were "reduced" restrictions of the above bijections. The expansions (1.1) and (1.2) are obtained via four weight-preserving bijections. These bijections are given by four variations of the Edelman–Greene insertion (EG insertion) [EG87]:

- $\Phi_{IS}^{\mathrm{Red}} : \mathsf{CP}_w^{\mathrm{Red}} \to \mathsf{Inc}_w^{\mathrm{Red}} \times_{\mathbb{Y}} \mathsf{SSYT}$: EG column insertion into increasing tableaux, starting from the right end of the compatible pairs.
- $\Phi_{IR}^{\mathrm{Red}} : \mathsf{CP}_w^{\mathrm{Red}} \to \mathsf{Inc}_w^{\mathrm{Red}} \times_{\mathbb{Y}} \mathsf{RSSYT}$: EG row insertion into increasing tableaux, starting from the left end of the compatible pairs.
- $\Phi_{DS}^{\mathrm{Red}} : \mathsf{CP}_w^{\mathrm{Red}} \to \mathsf{Dec}_w^{\mathrm{Red}} \times_{\mathbb{Y}} \mathsf{SSYT}$: EG row insertion into decreasing tableaux, starting from the right end of the compatible pairs.
- $\Phi_{DR}^{\mathrm{Red}} : \mathsf{CP}_w^{\mathrm{Red}} \to \mathsf{Dec}_w^{\mathrm{Red}} \times_{\mathbb{Y}} \mathsf{RSSYT}$: EG column insertion into decreasing tableaux, starting from the left end of the compatible pairs.

The four "reduced" bijections are essentially equivalent: EG row insertion and EG column insertion are merely transposes of each other. The relationships are summarized in the commutative diagram in Figure 1.1.

Example 1.1. Consider the following element $(a, i) \in \mathsf{CP}^{\mathrm{Red}}$.

i	3	3	3	2	2	2	1	1
a	1	2	4	1	3	5	2	4

We have

$$\Phi_{DR}(a,i) = \left(\begin{array}{|c|c|c|c|}\hline 5&4&2&1\\\hline 4&3&1\\\cline{1-3} 2\\\cline{1-1}\end{array} \;,\; \begin{array}{|c|c|c|c|}\hline 3&3&3&2\\\hline 2&2&1\\\cline{1-3} 1\\\cline{1-1}\end{array} \right) \quad \Phi_{DS}(a,i) = \left(\begin{array}{|c|c|c|c|}\hline 5&4&2&1\\\hline 4&3&1\\\cline{1-3} 2\\\cline{1-1}\end{array} \;,\; \begin{array}{|c|c|c|c|}\hline 1&1&2&3\\\hline 2&2&3\\\cline{1-3} 3\\\cline{1-1}\end{array} \right)$$

$$\Phi_{IR}(a,i) = \left(\begin{array}{|c|c|c|c|}\hline 1&2&3&4\\\hline 2&3&5\\\cline{1-3} 4\\\cline{1-1}\end{array} \;,\; \begin{array}{|c|c|c|c|}\hline 3&3&3&2\\\hline 2&2&1\\\cline{1-3} 1\\\cline{1-1}\end{array} \right) \quad \Phi_{IS}(a,i) = \left(\begin{array}{|c|c|c|c|}\hline 1&2&3&4\\\hline 2&3&5\\\cline{1-3} 4\\\cline{1-1}\end{array} \;,\; \begin{array}{|c|c|c|c|}\hline 1&1&2&3\\\hline 2&2&3\\\cline{1-3} 3\\\cline{1-1}\end{array} \right)$$

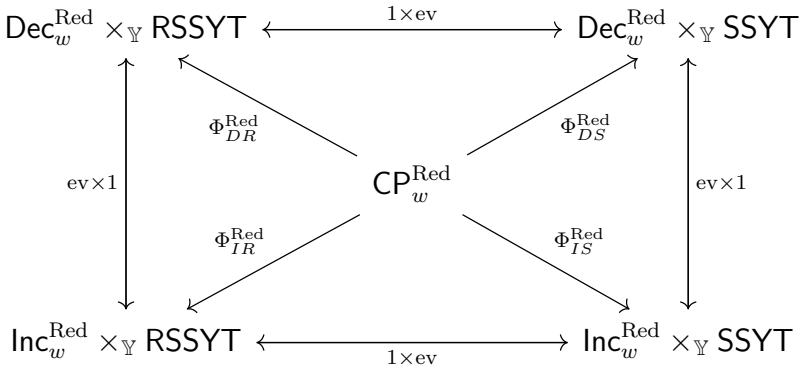
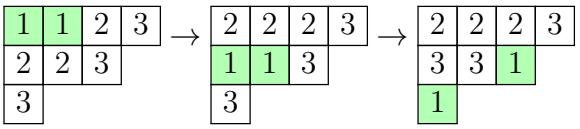
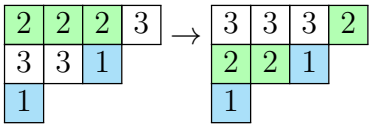


Figure 1.1: The four reduced bijections.

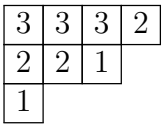
There are two kinds of evacuation maps in Figure 1.1. The map $\mathrm{ev} : \mathsf{SSYT} \to \mathsf{RSSYT}$ is defined as follows. For a $T \in \mathsf{SSYT}$, there is a unique $T' \in \mathsf{RSSYT}$ such that $\mathrm{row}(T)$ and $\mathrm{rev}(\mathrm{row}(T'))$ are Knuth equivalent, where $\mathrm{rev}(\cdot)$ is the operator that reverses a word. Then $\mathrm{ev}(T) := T'$. The computation $\mathrm{ev} : \mathsf{SSYT} \to \mathsf{RSSYT}$ can be done by jeu-de-taquin as follows. Sliding out the 1's using the usual jeu-de-taquin we obtain



Then the 2's are slid out but not past the 1s.



The 3's need no moving. The result is



The following Proposition asserts that the lower triangle in Figure 1.1 commutes.

Proposition 1.2 ([EG87, Corollary 7.22])**.** *Let* $(a, i) \in \mathsf{CP}_w^{\mathrm{Red}}$ *and* $\Phi_{IS}^{\mathrm{Red}}(a, i) = (P, Q)$ *and* $\Phi_{IR}^{\mathrm{Red}}(a, i) = (P', Q')$*. Then* $P = P'$ *and* $Q' = \mathrm{ev}(Q)$ *where* $\mathrm{ev} : \mathsf{SSYT} \to \mathsf{RSSYT}$ *is Schützenberger's evacuation involution (usual evacuation but without relabeling).*

The upper triangle also commutes: it is the same statement but with the total order on values reversed.

The other kind of evacuation map $\mathrm{ev} : \mathsf{Inc}_w^{\mathrm{Red}} \to \mathsf{Dec}_w^{\mathrm{Red}}$ can be defined similarly. For a $T \in \mathsf{Inc}_w^{\mathrm{Red}}$, there is a unique $T' \in \mathsf{Dec}_w^{\mathrm{Red}}$ such that $\mathrm{row}(T)$ and $\mathrm{rev}(\mathrm{row}(T'))$ are Coxeter–Knuth equivalent [EG87]. Then $\mathrm{ev}(T) := T'$. The following result says that the triangle on the right of Figure 1.1 is commutative.

Proposition 1.3. *Let* $(a, i) \in \mathsf{CP}_w^{\mathrm{Red}}$, $\Phi_{IS}^{\mathrm{Red}}(a, i) = (P, Q)$ *and* $\Phi_{DS}^{\mathrm{Red}}(a, i) = (P', Q')$. *Then* $Q = Q'$ *and* $P' = \mathrm{ev}(P)$.

Proof. The statement for Q tableaux is proved in [EG87, Corollary 7.21]. By [EG87, Theorem 6.24], $\mathrm{row}(P)$ and a are Coxeter–Knuth equivalent. On the other hand, $\mathrm{row}(P')$ and $\mathrm{rev}(a)$ are Coxeter–Knuth equivalent. Thus, $\mathrm{rev}(\mathrm{row}(P'))$ and $\mathrm{row}(P)$ are Coxeter–Knuth equivalent, so $P' = \mathrm{ev}(P)$. □

Similarly, the triangle on the left also commutes. Thus, Figure 1.1 commutes. In particular, for any fixed $(a, i) \in \mathsf{CP}_w^{\mathrm{Red}}$, upon applying any of the four EG bijections, the tableau pair has the same shape.

1.5. Solutions for general case

Hecke column insertion [BKS+08] defines a bijection $\Phi_{IS} : \mathsf{CP}_w \to \mathsf{Inc}_w \times_{\mathbb{Y}} \mathsf{SVT}_w$ whose restriction to $\mathsf{CP}_w^{\mathrm{Red}}$ is EG column insertion; the insertion starts at the left end of the word. By merely reversing the total order on entries in tableaux and inserting starting from the right end of the input word, the resulting variant of Hecke column insertion gives a bijection $\Phi_{DR} : \mathsf{CP}_w \to \mathsf{Dec}_w \times_{\mathbb{Y}} \mathsf{RSVT}_w$.

However, there are no known easy variations of the Hecke insertion which achieve the bijections Φ_{IR} or Φ_{DS}. The Hecke row insertion is the variant of the Hecke column insertion in which the roles of rows and columns are exchanged. A slightly restricted version of the row Hecke insertion was considered by Patrias and Pylyavskyy [PP18], where they studied the growth diagrams of Hecke row insertion of a (a, i) where i has distinct entries. In this case, the recording tableaux are valid SVTs. Unfortunately, when the input is an arbitrary compatible pair, the row Hecke insertion does not always produce a valid SVT as the recording tableau. We illustrate this failure with an example.

Example 1.4 (Pathology of Hecke row insertion)**.** Consider the compatible pair

i	3	3	2	2	2	1	1
a	1	3	1	2	3	1	3

The procedure for Hecke column insertion $\Phi_{IS} : \mathsf{CP}_w \to \mathsf{Inc}_w \times_{\mathbb{Y}} \mathsf{SVT}$ is

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

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

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

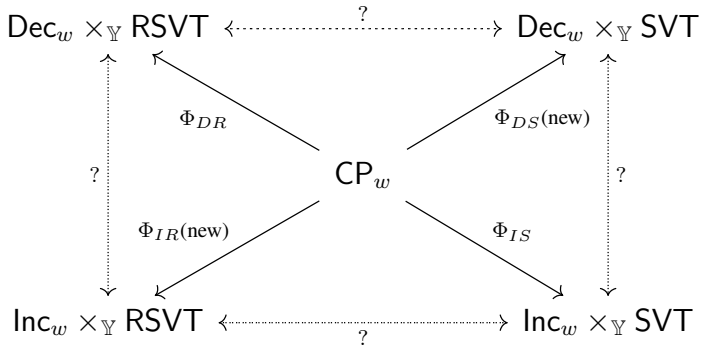


Figure 1.2: Non-reduced analogue of Figure 1.1.

and similarly we may compute the Hecke column insertion $\Phi_{DR} : \mathsf{CP}_w \to \mathsf{Dec}_w \times_{\mathbb{Y}} \mathsf{RSVT}$ and get the result

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

However, the row versions of these algorithms, which we denote Φ_{IR}^{Hecke} and Φ_{DS}^{Hecke}, do not work: the recording tableaux can fail to be set-valued (or reverse set-valued). Explicitly, if we apply Φ_{IR}^{Hecke} on the (a, i) above, the recording tableau fails to be set-valued after four insertions:

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

For Φ_{DS}^{Hecke}, we suffer from a similar pathology:

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

This paper introduces a new insertion algorithm Φ which gives an explicit weight-preserving bijection $\Phi_{DS} : \mathsf{CP}_w \to \mathsf{Dec}_w \times_{\mathbb{Y}} \mathsf{SVT}_w$. Our algorithm is a row insertion which, like Hecke insertion, respects the Hecke equivalence relation $\equiv_H$. Our insertion possesses a different Pieri property than the one satisfied by Hecke row insertion; this is necessary to achieve set-valued recording tableaux. Moreover, when restricted to $\mathsf{CP}_w^{\text{Red}}$, our algorithm recovers EG row insertion. A simple variation of our algorithm (reversing the total order on entries) gives a bijection $\Phi_{IR} : \mathsf{CP}_w \to \mathsf{Inc}_w \times_{\mathbb{Y}} \mathsf{RSVT}_w$. Together with the variants of Hecke insertion, our insertion completes the picture in §1.3: we have produced the generalization of the four diagonal maps in Figure 1.1. Now the picture looks like Figure 1.2.

Our insertion generally gives a tableau of a different shape than row Hecke insertion.

Example 1.5. Both our insertion and (the decreasing tableau version of) row Hecke insertion of the word 2421 produce the same tableau T. Let P and P' denote the tableau obtained when

the number 3 is inserted, for our insertion and row Hecke insertion respectively. P and P' have different shapes. See Example 5.7 for our insertion of 3 into T.

$$T = \begin{array}{|c|c|c|}\hline 4&2&1\\\hline 2\\\cline{1-1}\end{array} \qquad P = \begin{array}{|c|c|c|}\hline 4&3&1\\\hline 2&1\\\cline{1-2}\end{array} \neq P' = \begin{array}{|c|c|c|}\hline 4&3&1\\\hline 2\\\cline{1-1}\end{array}.$$

Remark 1.6. None of the coherence properties of Propositions 1.2 or 1.3 generalize to any of the bijections in the nonreduced setting. In general the four bijections produce 4 different groupings of compatible pairs for the various expansions of G_w.

In addition, we are not aware of any map that can be one of the four maps labeled by the question marks. Unlike the reduced case, these four maps have to change both the insertion tableaux and recording tableaux. Even worse, as shown in the following examples, these four maps are not shape-preserving. Therefore, doing an analogue of $\mathrm{ev}(\cdot)$ using the K-Bender–Knuth moves in [MPS21] would not work since the moves are shape-preserving.

Example 1.7. Consider the following element $(a, i) \in \mathsf{CP}_{31524}$.

i	3	3	2	2	1
a	2	4	1	3	1

We first compute $\Phi_{DS}(a, i)$. Starting with the empty tableau pair we insert the word a starting with the rightmost entry using our row insertion. We obtain the following sequence of tableaux, where the ending box is green.

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

The tableau pair $\Phi_{DS}(a, i)$ is

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

Under successive Hecke column insertions for increasing tableaux, we obtain

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

so $\Phi_{IS}(a, i)$ is the tableau pair

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

This implies that the map on the right in Figure 1.2 cannot be shape-preserving. By simply reversing both words in (a, i) and replacing every number j by $5 - j$, we obtain an example that implies the map on the left in Figure 1.2 cannot be shape-preserving.

Example 1.8. Consider the following element $(a, i) \in \mathsf{CP}_{24153}$.

i	5	4	4	4	2
a	3	1	2	4	2

We first compute $\Phi_{DS}(a, i)$. Starting with the empty tableau pair we insert the word a starting with the rightmost entry using our row insertion. We obtain the following sequence of tableaux, where the ending box is green.

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

The tableau pair $\Phi_{DS}(a, i)$ is

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

Under successive Hecke column insertions for decreasing tableaux starting with the left end, we obtain

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

so $\Phi_{DR}(a, i)$ is the tableau pair

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

This implies that the map on the top in Figure 1.2 cannot be shape-preserving. By simply reversing both words in (a, i) and replacing every number j by $6 - j$, we obtain an example that implies the map on the bottom in Figure 1.2 cannot be shape-preserving.

1.6. Restriction to bounded compatible pairs

A compatible pair $(a_1 \cdots a_n, i_1 \cdots i_n)$ is *bounded* if $a_j \geqslant i_j$ for all $j \in [n]$. Let CP_w^b be the set of bounded compatible pairs in CP_w. Fomin and Kirillov [FK94] showed that the generating function of CP_w^b is the Grothendieck polynomial, which can be viewed as the non-symmetric refinement of G_w.

Shimozono and Yu characterized the image of CP_w^b under Φ_{DR} in [SY23, Definition 4.1, Theorem 4.2]. Shimozono and Yu used this description to expand Grothendieck polynomials into Lascoux polynomials positively. This expansion was first conjectured by Reiner and Yong [RY21].

In a follow-up work, Orelowitz and Yu [OY23] characterized the image of CP_w^b under the map Φ_{DS} introduced in this paper. Their description also leads to the Grothendieck-to-Lascoux expansion, as shown in [OY23, Corollary 6.15]. Furthermore, they use the restriction of Φ_{DS} on CP_w^b to expand the product of a Lascoux polynomial and G_w into Lascoux polynomials. See [OY23, Remark 3.4] for a discussion on why their arguments cannot work using Φ_{DR}.

1.7. Related and future works

We mention a few works in the literature where the usual Hecke row insertion algorithm of (Hecke) words were applied and studied. It would be interesting to investigate how our insertion algorithm behave in these contexts. We thank the anonymous referee for pointing out these relevant works.

In [TY11], Thomas and Yong studied the row insertion of a Hecke word through the lens of sampling algorithms for probability measures and proved a symmetry property of the insertion tableaux. In [GP20], Guo and Poznanović proved that the number of 0-1 filling of a stack polyomino subject to certain restrictions only depends on the set of row lengths, using the Hecke insertion algorithm as a main technical tool. Their work was extended by Bloom and Saracino [BS24]. These works exploit the properties of the longest increasing/decreasing subsequences of a word being encoded in the insertion tableau. We leave the investigation of similar properties for our algorithm for future work.

2. New reverse row insertion

2.1. Ejectable values in decreasing tableaux

To define the new reverse insertion algorithm on decreasing tableaux, we require the notion of an *ejectable* value in a decreasing tableau. This is defined recursively.

In this article English notation is used for partitions and tableaux. A tableau is *decreasing* if its entries strictly decrease from left to right along each row and strictly decrease from top to bottom in each column. For a decreasing tableau P let $P_{>r}$ denote the decreasing tableau obtained by removing the first r rows of P. Let $P_{\geqslant r} = P_{>(r-1)}$.

Definition 2.1. Let P be a decreasing tableau. A value x is *P-ejectable* if x occurs in the first row of P and either $x-1$ is not in the first row of P, or $x-1$ is in the first row of P and $x-1$ is $P_{>1}$-ejectable.

Example 2.2. The value 3 is P-ejectable for the tableau P depicted below.

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

Since 3 and 2 both occur in the first row, 3 is P-ejectable if and only if 2 is $P_{>1}$-ejectable.

$$P_{>1} = \begin{array}{|c|c|c|}\hline 5 & 2 & 1 \\ \cline{1-3} 3 & 1 \\ \cline{1-2}\end{array}$$

Since 2 and 1 occur in the first row of $P_{>1}$, 2 is $P_{>1}$-ejectable if and only if 1 is $P_{>2}$-ejectable.

$$P_{>2} = \begin{array}{|c|c|}\hline 3 & 1 \\ \hline\end{array}$$

Since the first row of $P_{>2}$ has a 1 but no 0, 1 is $P_{>2}$-ejectable. Hence 3 is P-ejectable.

The value 7 is not P-ejectable because there is a 6 in the first row but not in the second.

2.2. Ejectable values and Hecke equivalence

The *0-Hecke monoid* is the quotient of the free monoid of words on the alphabet $\mathbb{Z}_{>0}$ by the relations

$$\begin{aligned} ii &\equiv_H i \\ i(i+1)i &\equiv_H (i+1)i(i+1) \\ ij &\equiv_H ji \qquad \text{for } |i-j| \geqslant 2. \end{aligned}$$

The minimum-length elements of each $\equiv_H$ class are the reduced words of some permutation w, giving a canonical bijection between the $\equiv_H$ classes and permutations of $\mathbb{Z}_{>0}$ moving finitely many elements. We denote by $[a]_H$ the permutation associated with the $\equiv_H$ class of the word a.

The row-reading word $\mathrm{row}(P)$ of a tableau P is the word $\cdots u^{(2)}u^{(1)}$ where $u^{(i)}$ is the word given by reading the i-th row of P from left to right.

Lemma 2.3. *Let P be a decreasing tableau. If x is an ejectable entry of P then $\mathrm{row}(P) \equiv_H \mathrm{row}(P)x$.*

Proof. This is proved by induction on the number of rows in P. Let w be the decreasing word given by the first row of P and let R be the set of letters in w. By definition $\mathrm{row}(P) = \mathrm{row}(P_{>1})w$. It suffices to show that

$$\mathrm{row}(P_{>1})w \equiv_H \mathrm{row}(P_{>1})wx. \tag{2.1}$$

If $x \in R$ and $x-1 \notin R$ then $w \equiv_H wx$ and hence (2.1) holds. Otherwise $x, x-1 \in R$ and the $x-1$ is ejectable in $P_{>1}$. By the inductive hypothesis, $\mathrm{row}(P_{>1}) \equiv_H \mathrm{row}(P_{>1})(x-1)$. In this case $(x-1)w \equiv_H wx$ and

$$\mathrm{row}(P_{>1})wx \equiv_H \mathrm{row}(P_{>1})(x-1)w \equiv_H \mathrm{row}(P_{>1})w$$

and again (2.1) holds as required. □

2.3. Bumping paths

Let $D(\lambda) = \{(i,j) \in \mathbb{Z}^2_{>0} \mid i \geqslant 1, j \leqslant \lambda_i\}$ be the diagram of the partition λ. The elements of $D(\lambda)$ are called the *cells* of λ and have a matrix-style indexing: the cell (i,j) is depicted as a box in the i-th row and j-th column. For a partition λ, a λ-removable cell is one that is at the end of its row and bottom of its column. For a tableau P, a P-removable cell is a λ-removable cell where λ is the shape of P.

Definition 2.4. Let (r,c) be a removable cell for the decreasing tableau P. The (reverse) *bumping path* of (r,c) in P is the following sequence of numbers $m_r < m_{r-1} < \cdots < m_1$ together with their positions in P. Let m_r be the value of P in (r,c). With the entry m_{i+1} in row $i+1$ defined, let m_i be the smallest number in row i such that $m_{i+1} < m_i$.

Example 2.5. A decreasing tableau and the bumping path for its removable cell $(3, 2)$ are pictured below.

8	7	6
5	4	2
3	2	
1		

In the example above, notice that the column index is weakly increasing, as you go up in bumping path. This is true in general.

Lemma 2.6. *Let $m_r < m_{r-1} < \cdots < m_1$ be a bumping path in P. For $r \geqslant j > i \geqslant 1$, the m_i in row i of P is weakly right of the m_j in row j of P.*

Proof. We only need to prove this claim for $j = i + 1$. Let y be the number immediately above the m_{i+1} in row $i + 1$ of P, We have $y > m_{i+1}$. Thus, $y \geqslant m_i$, The m_i in row i is weakly right of the y in this row, which implies our claim. □

The element in the first row of any bumping path is ejectable.

Lemma 2.7. *Let $m_r < \cdots < m_1$ be the bumping path of a removable cell of P. Then m_1 is ejectable in P.*

Proof. The proof proceeds by induction on the number of rows in P. Let R be first row of P. If $m_1 - 1 \notin R$ then m_1 is ejectable in P. Otherwise $m_1, m_1 - 1 \in R$. Since m_1 is the smallest in R such that $m_1 > m_2$, it follows that $m_2 = m_1 - 1$. It suffices to show that $m_1 - 1 = m_2$ is ejectable in $P_{>1}$. This follows from the inductive hypothesis since $m_r < \cdots < m_2$ is the bumping path of a removable cell in $P_{>1}$. □

2.4. New reverse insertion

The reverse insertion algorithm is a map Ψ

$$(P, s, \alpha) \mapsto (P', m)$$

where the input triple consists of a decreasing tableau P, a P-removable cell $s = (r, c)$, and $\alpha \in \{0, 1\}$. The output pair consists of a decreasing tableau P' and $m \in \mathbb{Z}_{>0}$ such that

$$\operatorname{shape}(P') = \begin{cases} \operatorname{shape}(P) & \text{if } \alpha = 0 \\ \operatorname{shape}(P) - \{s\} & \text{if } \alpha = 1. \end{cases} \tag{2.2}$$

For conceptual clarity we precompute the bumping path in P starting at (r, c). For $1 \leqslant i \leqslant r$ let m_i denote the entry in the i-th row of the bumping path. The output value m is by definition the value m_1 in the first row of the bumping path.

The output tableau P' will only differ from P along the bumping path. It is only necessary to specify whether each m_i on the bumping path gets replaced, and if so, by what value. Unlike most insertion algorithms, the replacement value might not come from the bumping path,

but does come from the row below. The behavior on each row is determined iteratively by decreasing i based on the values m_i and m_{i+1}, the i-th row of P, the subtableau $P'_{>i}$, and a status indicator $\alpha_{i+1} \in \{0, 1\}$. The i-th iteration updates the i-th row of P (which becomes the i-th row of P') and produces $\alpha_i \in \{0, 1\}$.

Let P' be a working tableau which is initialized to P. In the initialization step, if $\alpha = 1$, remove from P' the removable cell in row r and its contents m_r and set $\alpha_r = 1$ and $i = r - 1$. If $\alpha = 0$ set $m_{r+1} = 0$, $\alpha_{r+1} = 0$ and $i = r$.

The algorithm does the following for $i = r, r-1, \cdots, 2, 1$. Let R be the set consisting of numbers in row i of the current tableau P' (or equivalently P, since P and P' only differ under row i). By definition, $m_i \in R$.

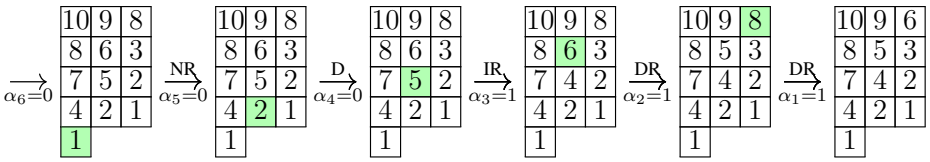
There are several cases. We give each a nickname and mnemonic.

- **Dummy (D)**: If $m_i - 1 \in R$ (which implies $m_{i+1} = m_i - 1$) do not change the i-th row and set $\alpha_i = \alpha_{i+1}$.
- **Direct Replacement (DR)**: Otherwise if $\alpha_{i+1} = 1$ and $m_{i+1} \notin R$, replace m_i by m_{i+1} in row i of P' and set $\alpha_i = 1$.

Suppose neither of the two above cases hold. Find the smallest ejectable entry x in $P'_{>i}$ such that $m_i > x > m_{i+1}$.

- **Indirect Replacement (IR)**: Suppose x exists. Replace m_i by x in row i of P' and set $\alpha_i = 1$.
- **No Replacement (NR)**: Suppose x does not exist. Do not change the i-th row and set $\alpha_i = 0$.

Example 2.8. In the following example, the input parameters are $s = (5, 1)$ and $\alpha = 0$. To initialize, set $(m_6, m_5, m_4, m_3, m_2, m_1) = (0, 1, 2, 5, 6, 8)$, $i = 5$, and $\alpha_6 = 0$. The shaded box in the i-th row indicates the value m_i. The label on the arrow leaving this tableau is the mnemonic for the case of Ψ.



3. Properties of the reverse insertion

In this section the reverse insertion map Ψ is shown to be well-defined and some of its properties are established.

Lemma 3.1. *For $r \geqslant i \geqslant 1$, $\alpha_i = 0$ if and only if m_i is ejectable in $P'_{\geqslant i}$.*

Proof. Note that after the i-th row is processed, the subtableau $P'_{\geqslant i}$ remains the same thereafter: only bumping path entries in rows above may be changed.

The proof proceeds by descending induction on i. For the initial step, if $\alpha = 1$, the algorithm sets $\alpha_r = 1$ and m_r gets removed and is therefore absent from the r-th row of P'. Thus, m_r is not ejectable in $P_{\geqslant r}$. If $\alpha = 0$, during the first iteration, m_r is replaced in the r-th row of P' if and only if $\alpha_r = 1$. Hence our claim holds for $i = r$.

Now suppose the claim holds for row $i+1$. In the Dummy case, $m_{i+1} = m_i - 1$ and m_i is not replaced. Thus m_i is ejectable in $P'_{\geqslant i}$ if and only if the entry $m_i - 1 = m_{i+1}$ is ejectable in $P'_{>i}$ (by definition of ejectable) if and only if $\alpha_{i+1} = 0$ (by induction) if and only if $\alpha_i = 0$ (since in the Dummy case $\alpha_i = \alpha_{i+1}$). Otherwise suppose the Dummy case does not hold. Then m_i and $m_i - 1$ cannot both live in row i of P'. Thus m_i is ejectable in $P'_{\geqslant i}$ if and only if it is not removed from row i. This happens only in the No Replacement case and $\alpha_i = 0$ only in that case. Thus our claim holds for row i as required. □

Theorem 3.2. *The reverse insertion is a well-defined map.*

Proof. It must be shown that the output tableau P' is a decreasing tableau. We show P' is a decreasing tableau after each iteration of the algorithm.

During initialization, if $\alpha = 0$, the iteration for $i = r$ either leaves P' unchanged or replaces m_r in the removable cell (r, c) by a smaller number. If $\alpha = 1$, after the initialization step P' is a decreasing tableau, since it is obtained from a decreasing tableau by removing a corner entry. In either case P' is a decreasing tableau before the $i = r - 1$ iteration.

Suppose P' is a decreasing tableau after the iteration for row $i + 1$. It is enough to check that P' is still a decreasing tableau after the iteration for row i. In the Dummy or No Replacement cases there is nothing to check. In the two remaining cases, the number m_i is replaced by a smaller number. We need to make sure this smaller number is larger than all numbers on its right and under it.

Consider P' before this iteration. By the definition of a bumping path, numbers on the right of the m_i in row i are at most m_{i+1}. By Lemma 2.6, numbers below this m_i are also at most m_{i+1}. Next, we consider the two cases.

- Direct Replacement: m_i is replaced by m_{i+1}. We need to make sure m_{i+1} is not on the right or under this m_i in P' before this iteration. First, m_{i+1} cannot be in row i by the condition of this case. Now assume toward contradiction that m_{i+1} is immediately below m_i. This part of P' looks like

m_i	a
m_{i+1}	b

  Since $\alpha_{i+1} = 1$, m_{i+1} is not ejectable in $P'_{\geqslant i+1}$. Thus, $b = m_{i+1} - 1$. Then $a > m_{i+1} - 1$ and $m_{i+1} \geqslant a$, so $a = m_{i+1}$. Since m_{i+1} is not in row i a contradiction is reached. Thus, after replacing the m_i by m_{i+1}, P' is still a decreasing tableau.

- Indirect Replacement: m_i is replaced by x. We know $x > m_{i+1}$. After replacing m_i by x, P' is still a decreasing tableau, since numbers on its right and under it are at most m_{i+1}. □

The reverse insertion respects Hecke equivalence.

Lemma 3.3. *Let* $\Psi(P,(r,c),\alpha)=(P',m)$. *Then* $\mathrm{row}(P)\equiv_H \mathrm{row}(P')m$.

Proof. Let w be the decreasing word given by the first row of P and let R be the set of letters in w. Define R' and w' similarly for P'. Notice that $\mathrm{row}(P)=\mathrm{row}(P_{>1})w$ and $\mathrm{row}(P')=\mathrm{row}(P'_{>1})w'$. It suffices to show that

$$\mathrm{row}(P_{>1})w\equiv_H \mathrm{row}(P'_{>1})w'm. \tag{3.1}$$

The proof proceeds by induction on r, the row index of the entry in the input of Ψ. The base case is $r=1$. In this case $\mathrm{row}(P_{>1})=\mathrm{row}(P'_{>1})$. If $\alpha=1$, then $w=w'm$. Otherwise, in the first iteration, the algorithm searches for the smallest ejectable $x<m$ in $P_{>1}$. If x does not exist then $w\equiv_H wm=w'm$. Otherwise w' is obtained by changing m in w into x. We see that (3.1) holds:

$$\mathrm{row}(P_{>1})w\equiv_H \mathrm{row}(P_{>1})xw\equiv_H \mathrm{row}(P_{>1})w'm.$$

For the inductive step let $r>1$. Before the last iteration the algorithm behaves as if doing Ψ on $(P_{>1},(r-1,c),\alpha)$. By the definition of Ψ the result is $(P'_{>1},m_2)$. By the inductive hypothesis, $\mathrm{row}(P_{>1})\equiv_H \mathrm{row}(P'_{>1})m_2$. It is enough to check

$$\mathrm{row}(P'_{>1})m_2w\equiv_H \mathrm{row}(P'_{>1})w'm.$$

Consider the first two cases of the last iteration.

- Dummy: In this case, $m,m-1\in R$ and $m_2=m-1$. We have $m_2w\equiv_H wm=w'm$.
- Direct Replacement: In this case, $m-1,m_2\notin R$. We know w is obtained from w by changing m into m_2. We have $m_2w\equiv_H w'm$.

We may assume the above two cases do not hold. Then either $m_2\in R$ or $\alpha_2=0$ (m_2 is ejectable in $P'_{>1}$). In either case we claim $\mathrm{row}(P'_{>1})m_2w\equiv_H \mathrm{row}(P'_{>1})w$: If $m_2\in R$, then $m_2w\equiv_H w$ since m_2+1 is not in w and m_2 is in w. If the m_2 is ejectable in $P'_{>1}$ then $\mathrm{row}(P'_{>1})m_2\equiv_H \mathrm{row}(P'_{>1})$ by Lemma 2.3.

With this claim, it must be shown that

$$\mathrm{row}(P'_{>1})w\equiv_H \mathrm{row}(P'_{>1})w'm.$$

It must be verified that this holds in the remaining two cases:

- Indirect Replacement: Since x is ejectable in $P'_{>1}$, $\mathrm{row}(P'_{>1})\equiv_H \mathrm{row}(P'_{>1})x$. w' is obtained by changing m to x in w. Thus $xw\equiv_H w'm$.
- No Replacement: We have $w\equiv_H wm=w'm$. □

Our reverse row insertion satisfies the following Pieri condition, which is not satisfied by Hecke reverse row insertion.

Lemma 3.4. *Let $\alpha, \alpha' \in \{0,1\}$, P a decreasing tableau with removable corner (r_1, c_1),*

$$\Psi(P,(r_1,c_1),\alpha) = (P', m) \text{ and } \Psi(P',(r_2,c_2),\alpha') = (P'', m')$$

with (r_2, c_2) a removable corner of P' with $c_2 < c_1$. Then $m' > m$.

Proof. Let $m_{r_1} < \cdots < m_1$ be the bumping path for Ψ on $(P,(r_1,c_1),\alpha)$ and $n_{r_2} < \cdots < n_1$ the bumping path for Ψ on $(P',(r_2,c_2),\alpha')$. By definition $m_1 = m$ and $n_1 = m'$ so it is enough to show that $n_1 > m_1$.

When m_i is ejected from $P'_{\geqslant i}$, n_i is a number in the top row of $P'_{\geqslant i}$. Moreover, since n_i is the last number in a bumping path in $P'_{\geqslant i}$, n_i is ejectable in $P'_{\geqslant i}$. We check $n_i > m_i$ for all $1 \leqslant i \leqslant r_1$ by descending induction on i; in the case $\alpha = 0$ the initial index is $i = r_1 + 1$.

For the base case consider the value of α. If $\alpha = 1$, m_{r_1} is removed from row r_1 and ejected. Clearly $n_{r_1} > m_{r_1}$. If $\alpha = 0$, $m_{r_1+1} = 0 < n_{r_1+1}$.

By induction we assume that $m_{i+1} < n_{i+1}$. We consider the cases of the two reverse insertions when they process row i.

- (Dummy case): In this case, $m_i = m_{i+1} + 1$. We have $n_i > n_{i+1} \geqslant m_i$.
- (Direct Replacement case): In this case, we replace m_i by m_{i+1} in row i. Then n_i is a number in row i of P', and since $n_i > n_{i+1} > m_{i+1}$ it must be to the left of m_{i+1}. Thus n_i is to the left of m_i in row i of P. We conclude that $n_i > m_i$.
- (Indirect Replacement case): In this case, we replace m_i by x on row i. Since $n_{i+1} > m_{i+1}$ and n_{i+1} is ejectable in $P'_{>i}$, $n_{i+1} \geqslant x$ by the choice of x. Thus n_i is a number in row i of P', and since $n_i > n_{i+1} \geqslant x$, it must be to the left of x. Similar to the previous case, $n_i > m_i$.
- (No Replacement case): In this case, there is no x that is ejectable in $P'_{\geqslant i+1}$ and $m_{i+1} < x < m_i$. Since n_{i+1} is ejectable in $P'_{\geqslant i+1}$, $n_{i+1} \geqslant m_i$. Thus $n_i > m_i$. □

Example 3.5. Let P be the following decreasing tableau:

4	3	1
2	1	

.

Invoke the reverse insertion with input $(r_1, c_1) = (2, 2)$ and $\alpha_1 = 1$. We obtain (P', m) where $m_1 = 3$ and P' is the following decreasing tableau

4	2	1
2		

.

If we invoke the reverse insertion on P' with input $(r_2, c_2) = (2, 1)$, the output number will be $m' = 4$. This aligns with Lemma 3.4 since $c_2 < c_1$ and $m' > m$.

The reverse insertion algorithm is a generalization of EG reverse insertion.

Lemma 3.6. *Let P be a decreasing tableau such that $\mathrm{row}(P)$ is reduced. Let $\Psi(P,(r,c),1) = (P',m)$. Then we also get (P',m) if we apply EG reverse row insertion at (r,c) in P.*

Proof. Since the Dummy and Direct Replacement cases agree with EG reverse insertion, it is enough to show that during each iteration, one of these cases must apply.

For $\alpha = 1$ the initial step agrees with reverse EG insertion. By induction we assume $\alpha_{i+1} = 1$. We will assume the iteration for row i is not in the Dummy nor the Direct Replacement cases and reach a contradiction. Let R be the i-th row of P. We assume $m_i - 1 \notin R$ and $m_{i+1} \in R$. By the minimality of m_i, $m_{i+1} + 1 \notin R$. Let w be the row word of the first i rows of P. We have $m_{i+1}w \equiv_H w$. Then notice that

$$\mathrm{row}(P) = \mathrm{row}(P_{>i})w \equiv_H \mathrm{row}(P'_{>i})m_{i+1}w \equiv_H \mathrm{row}(P'_{>i})w.$$

Then $\mathrm{row}(P)$ is not reduced and we obtain the required contradiction. □

4. The insertion

This section gives a direct description of the inverse of Ψ, an insertion algorithm Φ which "inserts m into P":

$$(P,m) \mapsto (P',s,\alpha)$$

where the input pair consists of a decreasing tableau P and $m \in \mathbb{Z}_{>0}$, and the output triple consists of a decreasing tableau P', a removable cell $s = (r,c)$ of P', and $\alpha \in \{0,1\}$ such that the following holds:

$$\mathrm{shape}(P') = \begin{cases} \mathrm{shape}(P) & \text{if } \alpha = 0 \\ \mathrm{shape}(P) \cup \{s\} & \text{if } \alpha = 1. \end{cases} \tag{4.1}$$

The working tableau P' has initial value P. The i-th iteration consists of an insertion of a number $N \in \mathbb{Z}_{>0}$ into $P'_{\geqslant i}$. At this point $P'_{\geqslant i} = P_{\geqslant i}$; only values in rows before the i-th have been changed. Let R be the set consisting of numbers in row i of P. Find the largest $n_1 \in R$ such that $n_1 \leqslant N$.

- **Terminating case 1 (T1)**: If n_1 does not exist, put N at the end of row i in P' and terminate the algorithm. The output P' is the current tableau. The output (r,c) is the coordinate of this newly added N. Set $\alpha = 1$.

Otherwise n_1 exists. Change the n_1 in row i of P' into N.

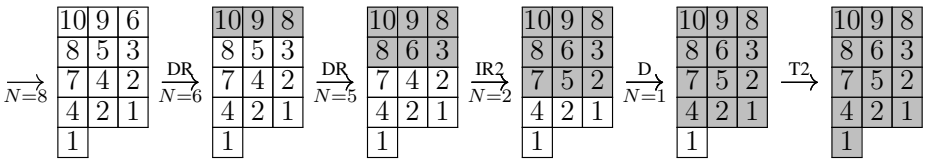
- **Dummy case (D)**: If $n_1 = N$ and $N - 1 \in R$: insert $N - 1$ into $P'_{>i}$.
- **Direct Replacement case (DR)**: If $n_1 < N$ and n_1 is not ejectable in $P_{>i}$: insert n_1 into $P'_{>i}$.

Otherwise assume none of the above cases hold. Let n_2 be the number to the right of n_1 in row i of P, or $n_2 = 0$ if n_1 is the rightmost number in this row. Find the largest ejectable y in $P_{>i}$ such that $n_1 > y > n_2$.

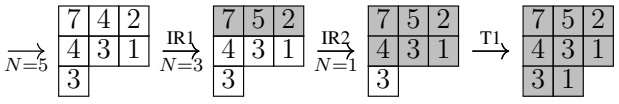
- **Indirect Replacement case 1 (IR1)**: If y exists: insert y into $P'_{>i}$.
- **Indirect Replacement case 2 (IR2)**: If y does not exist and $n_2 > 0$: insert n_2 into $P'_{>i}$.
- **Terminating case 2 (T2)**: If y does not exist and $n_2 = 0$: terminate the algorithm. The output P' is the current tableau. The output (r, c) is the coordinate of this N in row i of P'.

  Set $\alpha = 0$.

Example 4.1. In the following example, we let P be the leftmost tableau and insert $m = 8$ into P. The output is the rightmost tableau P', $s = (5, 1)$, and $\alpha = 0$. The unshaded part of each tableau is the part being considered by the insertion in each step.



Example 4.2. In the following example, we insert $m = 5$ into the leftmost tableau P. The output is the rightmost tableau P', $s = (3, 2)$, and $\alpha = 1$.



5. Properties of the insertion

In this section the well-definedness of the insertion algorithm Φ is established and some of its properties are studied.

Lemma 5.1. *Consider an iteration of Φ in which N is being inserted into $P'_{\geqslant i}$ in Indirect Replacement case 2. Consider the value n_1 in row i of P. If there is a number below this n_1, it must be at most n_2.*

Proof. Let t_1 be the number below this n_1. This part of P looks like

n_1	n_2
t_1	t_2

Now assume toward contradiction that $t_1 > n_2$. The number t_2 either does not exist or we have $t_2 < n_2 \leqslant t_1 - 1$. In either case, t_1 is ejectable in $P'_{>i}$. By $n_1 > t_1 > n_2$, we should go to Indirect Replacement case 1. Contradiction. □

Lemma 5.2. *The insertion algorithm is well-defined.*

Proof. The algorithm initializes the working tableau to equal P which is decreasing. To show the output tableau is decreasing it suffices to assume that before any particular iteration the working tableau is decreasing and show that after that iteration, the resulting tableau is decreasing.

Let P' be the working tableau at the beginning of the current iteration, in which N is being inserted into $P'_{\geqslant i}$. Let P'' be the working tableau after this iteration. During the iteration, in row i the number n_1 is replaced by N or N is appended at the end; let (i, j') be the position of this N. After this iteration, the row will clearly be strictly decreasing. We may assume $i > 1$ and must show that there is a number M in position $(i-1, j')$ of P'' and it satisfies $M > N$.

If $n_1 = N$, then we are done since this iteration does not change the working tableau at all. We assume $n_1 < N$, so the previous iteration is not in the Indirect Replacement case 1. Consequently, N is in row $i-1$ of P, say at $(i-1, j)$. We have $j' \leqslant j$ by the choice of n_1. In particular there is a number M in position $(i-1, j')$ of P''. It remains to show that $M > N$.

If $j' < j$ then we obtain the required inequality $M > N$ since the $(i-1)$-th row was strictly decreasing before the previous iteration. So we may assume $j' = j$.

We consider the cases of the previous iteration:

- Dummy case. $N+1$ and N are in row $i-1$ of P. Below this $N+1$, we have a number at most N, so $j' \leqslant j-1$, contradiction.
- Direct Replacement case. During the previous iteration, the N in cell $(i-1, j)$ is replaced by a larger number. Thus, there is an $M > N$ at $(i-1, j)$ of P''.
- Indirect Replacement case 2. By Lemma 5.1, the n_1 is in the first $j-1$ columns, a contradiction.

Since the row $i-1$ iteration was not terminal and we ruled out Indirect Replacement case 1, all cases are covered. □

Theorem 5.3. Φ *and* Ψ *are mutually inverse functions.*

We prove Theorem 5.3 directly by splitting the proof into two lemmas.

Lemma 5.4. *Let* $\Psi(P, (r, c), \alpha) = (P', m)$. *Then* $\Phi(P', m) = (P, (r, c), \alpha)$.

Proof. Let R (resp. R') consist of the numbers in row 1 of P (resp. P'). The proof proceeds by induction on r.

For the base case, assume $r = 1$. If $\alpha = 1$, $m = \min(R)$ and $R' = R - \{m\}$. When we insert m into P', the first iteration is in Terminating case 1. We will just append m at the end of row 1 and terminate at this cell. If $\alpha = 0$, we study the cases of the only iteration in the reverse insertion:

- Indirect Replacement case: In this case, $R' = R - \{m\} \sqcup \{x\}$ where $x < m$ and x is the smallest number in row 2 of P. When we insert m into P', it sets $n_1 = x$. Since n_1 is ejectable in $P'_{>1}$, it does not go to the first 3 cases. Then we have $n_2 = 0$. There are no ejectable numbers in $P'_{>1}$ between n_2 and n_1. Thus, it goes to the Terminating case 2. It replaces x by m and ends at this cell with $\alpha = 0$.

- No Replacement case: In this case, $R = R'$ and there are no ejectable numbers in $P'_{>1}$ that are less than m. When we insert m into P', it sets $n_1 = m$ and $n_2 = 0$. Thus, it goes to Terminating case 2. It replaces m by m and ends at this cell with $\alpha = 0$.

Now assume $r > 1$. Consider the reverse insertion. Before the last iteration, a number $m_2 > 0$ is ejected from $P'_{>1}$. During the last iteration, it changes at most one number in R and get R'. Then it ejects m. By induction it suffices to show that when m is inserted into P', the first iteration of insertion changes R' back to R and inserts m_2 into $P'_{>1}$. Let us do a case study on the last iteration of the reverse insertion.

- Dummy case: $m_2 = m - 1$ and $m_2, m \in R$. The algorithm fixes row 1 of P so $R = R'$. The first iteration of insertion goes to the Dummy case. It fixes row 1 and inserts $m - 1 = m_2$ into $P'_{>1}$.

- Direct Replacement case: m_2 was ejected with $\alpha_2 = 1$. Thus m_2 is not ejectable in $P'_{>1}$. The algorithm replaces m by m_2. The first iteration of insertion sets $n_1 = m_2$. It goes to the Direct Replacement case: m_2 is replaced by m and m_2 is inserted.

- Indirect Replacement case: m is replaced by x which is ejectable in $P'_{>1}$. By the choice of x there are no ejectable numbers in $P'_{>1}$ between m_2 and x. The first iteration of insertion sets $n_1 = x$ and replaces it by m. It will not go to the first 3 cases. Since m is the smallest number in R that is larger than m_2, $m_2 \geqslant n_2$. If $m_2 > n_2$ then $\alpha_2 = 0$. Thus m_2 is ejectable in $P'_{>1}$. During the subsequent insertion, the algorithm sets $y = m_2$ and inserts m_2. Now assume $m_2 = n_2$. Then the first iteration of the insertion cannot find such a y. It inserts $n_2 = m_2$.

- No Replacement case: $R = R'$. There are no ejectable numbers in $P'_{>1}$ between m_2 and m. During the insertion, the algorithm sets $n_1 = m$ and will not go to the first three cases. The proof proceeds as in the Indirect Replacement case. □

Lemma 5.5. *Let* $\Phi(P, m) = (P', (r, c), \alpha)$. *Then* $\Psi(P', (r, c), \alpha) = (P, m)$.

Proof. Let R (resp. R') consist of the numbers in row 1 of P (resp. P'). The proof proceeds by induction on r.

In the base case $r = 1$, the insertion has only one iteration. If $\alpha = 1$, this iteration is in Terminating case 1. It appends m at the end of row 1. During the subsequent reverse insertion, m will be removed from row 1 and ejected. Now assume $\alpha = 0$. If $n_1 = m$, then the insertion leaves row 1 unchanged. There are no ejectable numbers in $P_{>1}$ that are less than m. During the reverse insertion, the sole iteration goes to the No Replacement case: row 1 is unchanged and m is ejected. If $n_1 < m$, then the insertion replaces n_1 by m. Since it is in the Terminating case 2, n_1 is smallest ejectable number in $P_{>1}$. During the reverse insertion, the only iteration goes to the Indirect Replacement case: the m is changed to n_1 and m is ejected.

Now assume $r > 1$. Consider the insertion. During the first iteration, it changes n_1 into m in row 1. Then it inserts a number into $P'_{>1}$. Let z be that number. Now consider the reverse insertion. By our inductive hypothesis, before the last iteration, z is ejected under row 1 of the tableau. Moreover, currently the tableau below row 1 is identical to $P_{>1}$. We need to make sure

the last iteration changes R' back to R and ejects m. Let us do a case study on the first iteration of the insertion.

- Dummy case: $m, m-1 \in R$, $R = R'$, and $z = m-1$. The last iteration of the reverse insertion goes to the Dummy case: it fixes the first row and ejects m.
- Direct replacement case: n_1 is changed to m and $z = n_1 < m$. Moreover z is not ejectable in $P_{>1}$. When z is ejected from $P_{>1}$, $\alpha_2 = 1$. The last iteration of the reverse insertion goes to the Direct Replacement case: It changes m into n_1 and ejects m.
- Indirect Replacement case 1: n_1 is changed to m and $z = y$. y is the largest ejectable number in $P_{>1}$ less than n_1. Moreover $y > n_2$. Consider the last iteration of the reverse insertion. Before this iteration, by induction and Lemma 3.1 y is ejected from $P_{>1}$ with $\alpha_2 = 0$. Then it sets $m_1 = m$. It looks for x, which is the smallest ejectable number in $P_{>1}$ between y and m. If $n_1 = m$, then it goes to the No Replacement case: Row 1 is fixed and m is ejected. If $n_1 < m$ then n_1 must be ejectable in $P_{>1}$ and $x = n_1$. It goes to the Indirect Replacement case: m is replaced by n_1 and m is ejected.
- Indirect Replacement case 2: n_1 is replaced by m and $z = n_2 > 0$. There are no ejectable numbers between n_2 and n_1 in $P_{>1}$. Consider the last iteration of the reverse insertion. It sets $m_1 = m$. Since n_2 is already in row 1, it must go to the last two cases. If $n_1 = m$, then it goes to the No Replacement case: The first row is fixed and m is ejected. If $n_1 < m$, then n_1 must be ejectable in $P_{>1}$. It goes to the Indirect Replacement case: m is replaced by n_1 and m is ejected. □

Our insertion satisfies a Pieri property.

Lemma 5.6. *Let* $\Phi(P, m) = (P', (r_1, c_1), \alpha)$ *and* $\Phi(P', m') = (P'', (r_2, c_2), \alpha')$. *If* $m' < m$, *then* $c_1 < c_2$.

Proof. Let $m_{r_1} < \cdots < m_1$ be the bumping path of (r_1, c_1) in P'. By the definition of Ψ on $(P', (r_1, c_1), \alpha)$, the output value is m_1. Since Ψ is inverse to Φ, $\Psi(P', (r_1, c_1), \alpha) = (P, m)$. Thus $m_1 = m > m'$. If Φ on (P', m') ends in the first iteration, we are done. Otherwise, after this iteration, another number is inserted into $P'_{>1}$. It is enough to ensure that this number is smaller than m_2. During this iteration, Φ finds a number n_1 in row 1 of P'. We have $n_1 \leqslant m' < m_1$. Since $m_{r_1} < \cdots < m_1$ is a bumping path, $m_2 \geqslant n_1$. Now consider the case of the first iteration.

- Dummy case: The number $m'-1$ is inserted into $P'_{>1}$. We have $m_2 \geqslant n_1 = m' > m'-1$.
- Direct Replacement case: The number n_1 is inserted into $P'_{>1}$. Notice that m_2 is ejectable in $P'_{>1}$ since it is the end of a bumping path in $P'_{>1}$. However, n_1 is not ejectable in $P'_{>1}$ by the condition of this case. Thus $m_2 \neq n_1$. Since $m_2 \geqslant n_1$ we deduce that $m_2 > n_1$.
- Indirect Replacement case 1: The number y is inserted into $P'_{>1}$. We have $m_2 \geqslant n_1 > y$.
- Indirect Replacement case 2: The number n_2 is inserted into $P'_{>1}$. We have $m_2 \geqslant n_1 > n_2$. □

Example 5.7. Let P be the following decreasing tableau:

$$\begin{array}{|c|c|c|}\hline 4 & 2 & 1 \\ \hline \end{array}.$$

After inserting the number $m = 4$ to P, we obtain $(P', (r_1, c_1), \alpha)$ where $(r_1, c_1) = (2, 1)$, $\alpha = 1$ and P' is

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

Next, insert $m' = 3$ to P' and get $(P'', (r_2, c_2), \alpha')$ where $(r_2, c_2) = (2, 2)$, $\alpha = 1$ and P'' is

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

This aligns with Lemma 5.6 since $c_1 < c_2$ and $m' < m$.

To summarize, our new reverse insertion satisfies the following Pieri property.

Theorem 5.8. *Let P be a decreasing tableau. Apply successive reverse insertions*

$$\Psi(P, (r_1, c_1), \alpha) = (P', m)$$
$$\Psi(P', (r_2, c_2), \alpha') = (P'', m')$$

Then $c_2 < c_1$ if and only if $m' > m$.

Proof. Follows from Lemma 3.4, Lemma 5.6, and Theorem 5.3. □

The following is an equivalent restatement for insertion.

Corollary 5.9. *Let P be a decreasing tableau and $m, m' \in \mathbb{Z}_{>0}$. Applying successive insertions $\Phi(P, m) = (P', (r_1, c_1), \alpha)$ and $\Phi(P', m') = (P'', (r_2, c_2), \alpha')$, we have $m > m'$ if and only if $c_1 < c_2$.*

Given a compatible pair (a, i) and starting with the empty tableau pair, use Φ to insert a_1, then a_2, and so on, recording the insertion of a_k by i_k, producing a tableau pair (P, Q) where P is decreasing. Denote this map by $\widetilde{\Phi}(a, i) = (P, Q)$.

Corollary 5.10. *$\widetilde{\Phi}$ is a weight-preserving bijection* $\mathsf{CP}_w \to \mathsf{Dec}_w \times_{\mathbb{Y}} \mathsf{SVT}$.

Proof. Corollary 5.9 implies that Q is set-valued. Moreover it also implies that the process can be inverted: if $(P, Q) \in \mathsf{Dec}_w \times_{\mathbb{Y}} \mathsf{SVT}$, then using Ψ at the sequence of removable boxes given by the entries of Q, one recovers $(a, i) \in \mathsf{CP}_w$. □

Remark 5.11. Reversing comparison of values, one obtains a weight-preserving bijection $\widetilde{\Phi}^{\mathsf{Inc}} : \mathsf{CP}_w \cong \mathsf{Inc}_w \times_{\mathbb{Y}} \mathsf{RSVT}$.

References

[BJS93] Sara C Billey, William Jockusch, and Richard P Stanley. Some combinatorial properties of Schubert polynomials. *Journal of Algebraic Combinatorics*, 2(4):345–374, 1993. doi:10.1023/A:1022419800503.

[BKS+08] Anders Skovsted Buch, Andrew Kresch, Mark Shimozono, Harry Tamvakis, and Alexander Yong. Stable Grothendieck polynomials and K-theoretic factor sequences. *Mathematische Annalen*, 340:359–382, 2008. doi:10.1007/s00208-007-0155-6.

[BS24] Jonathan S. Bloom and Dan Saracino. Strictly increasing and decreasing sequences in subintervals of words and a conjecture of Guo and Poznanović. *European Journal of Combinatorics*, 118:103898, 2024. doi:10.1016/j.ejc.2023.103898.

[Buc02] Anders Skovsted Buch. A Littlewood-Richardson rule for the K-theory of Grassmannians. *Acta Mathematica*, 189(1):37–78, 2002. doi:10.1007/bf02392644.

[EG87] Paul Edelman and Curtis Greene. Balanced tableaux. *Advances in Mathematics*, 63(1):42–99, 1987. doi:10.1016/0001-8708(87)90063-6.

[FK94] Sergey Fomin and Anatol N Kirillov. Grothendieck polynomials and the Yang-Baxter equation. In *Proceedings of Formal Power Series and Algebraic Combinatorics*, pages 183–190, 1994.

[GP20] Ting Guo and Svetlana Poznanović. Hecke insertion and maximal increasing and decreasing sequences in fillings of stack polyominoes. *Journal of Combinatorial Theory, Series A*, 176:105304, 2020. doi:10.1016/j.jcta.2020.105304.

[MPS21] Cara Monical, Oliver Pechenik, and Travis Scrimshaw. Crystal structures for symmetric Grothendieck polynomials. *Transformation Groups*, 26:1025–1075, 2021. doi:10.1007/s00031-020-09623-y.

[OY23] Gidon Orelowitz and Tianyi Yu. Lascoux expansion of the product of a Lascoux and a stable Grothendieck. 2023. arXiv:2312.01647.

[PP18] Rebecca Patrias and Pavlo Pylyavskyy. Dual filtered graphs. *Algebraic Combinatorics*, 1(4):441–500, 2018. doi:10.5802/alco.21.

[RY21] Victor Reiner and Alexander Yong. The "Grothendieck to Lascoux" conjecture. 2021. arXiv:2102.12399.

[SY23] Mark Shimozono and Tianyi Yu. Grothendieck-to-Lascoux expansions. *Trans. Amer. Math. Soc.*, 376(7):5181–5220, 2023. doi:10.1090/tran/8912.

[TY11] Hugh Thomas and Alexander Yong. Longest increasing subsequences, Plancherel-type measure and the Hecke insertion algorithm. *Advances in Applied Mathematics*, 46(1-4):610–642, 2011. Special issue in honor of Dennis Stanton. doi:10.1016/j.aam.2009.07.005.