Ann. Comb. 26 (2022) 261–301

Published online January 30, 2022
https://doi.org/10.1007/s00026-022-00567-6

Annals of Combinatorics

Uncrowding Algorithm for Hook-Valued Tableaux

Jianping Pan, Joseph Pappe, Wencin Poh and Anne Schilling

Abstract. Whereas set-valued tableaux are the combinatorial objects associated to stable Grothendieck polynomials, hook-valued tableaux are associated with stable canonical Grothendieck polynomials. In this paper, we define a novel uncrowding algorithm for hook-valued tableaux. The algorithm "uncrowds" the entries in the arm of the hooks, and yields a set-valued tableau and a column-flagged increasing tableau. We prove that our uncrowding algorithm intertwines with crystal operators. An alternative uncrowding algorithm that "uncrowds" the entries in the leg instead of the arm of the hooks is also given. As an application of uncrowding, we obtain various expansions of the canonical Grothendieck polynomials.

Mathematics Subject Classification. Primary 05E05, 05E10; Secondary 14N10, 14N15, 20G42.

Keywords. Stable (canonical) Grothendieck polynomials, Hook-valued tableaux, crystal bases, uncrowding algorithm.

1. Introduction

Set-valued tableaux play an important role in the K-theory of the Grassmannian. They form a generalization of semistandard Young tableaux, where boxes may contain sets of integers rather than just integers [3]. In particular, the stable symmetric Grothendieck polynomial indexed by the partition λ is the generating function of set-valued tableaux

$$G_\lambda(x;\beta) = \sum_{T \in \mathsf{SVT}(\lambda)} \beta^{|T|-|\lambda|} x^{\mathsf{weight}(T)}, \tag{1.1}$$

where $\mathsf{SVT}(\lambda)$ is the set of set-valued tableaux of shape λ and $\mathsf{weight}(T)$ is the vector with ith entry being the number of i in T. Here, $|T|$ is the number of entries in T and $|\lambda|$ is the size of λ. Stable symmetric Grothendieck polynomials G_λ can be viewed as a K-theory analog of the Schur functions s_λ (while the

Grothendieck polynomial is an analog of the Schubert polynomial [11]). Buch [3] also described the structure coefficients $c^{\nu}_{\lambda\mu}$, which is the coefficient of G_ν in the expansion of $G_\lambda G_\mu$ in terms of set-valued tableaux, generalizing the Littlewood–Richardson rule for Schur functions.

The Grassmannian $\mathrm{Gr}(k, \mathbb{C}^n)$ of k-planes in $\mathbb{C}^n$ has a fundamental duality isomorphism

$$\mathrm{Gr}(k, \mathbb{C}^n) \cong \mathrm{Gr}(n-k, \mathbb{C}^n).$$

This implies that the structure constants have the symmetry $c^{\nu}_{\lambda\mu} = c^{\nu'}_{\lambda'\mu'}$, where λ' denotes the conjugate of the partition λ (see for example [5, Example 9.20]). Hence, one expects a ring homomorphism on the completion of the ring of symmetric function defined on the basis of stable symmetric Grothendieck polynomials $\tau(G_\lambda) = G_{\lambda'}$. The standard involutive ring automorphism ω defined on the Schur basis by $\omega(s_\lambda) = s_{\lambda'}$ does not have this property [10]

$$\omega(G_\lambda) = J_\lambda \neq G_{\lambda'},$$

where J_λ is the weak symmetric Grothendieck polynomial.

Yeliussizov [16] introduced a new family of canonical stable Grothendieck polynomials $G_\lambda(x; \alpha, \beta)$, such that

$$\omega(G_\lambda(x; \alpha, \beta)) = G_{\lambda'}(x; \beta, \alpha).$$

Combinatorially, the canonical stable Grothendieck polynomials can be expressed as generating functions of hook-valued tableaux. In a hook-valued tableau, each box contains a semistandard Young tableau of hook shape, which is weakly increasing in rows and strictly increasing in columns. More precisely

$$G_\lambda(x; \alpha, \beta) = \sum_{T \in \mathsf{HVT}(\lambda)} \alpha^{a(T)} \beta^{\ell(T)} x^{\mathsf{weight}(T)},$$

where $\mathsf{HVT}(\lambda)$ is the set of hook-valued tableaux of shape λ, $a(T)$ is the sum of all arm lengths, and $\ell(T)$ is the sum of all leg lengths of the hook tableaux in T.

A hook-valued tableau T is a set-valued tableau when all hook tableaux entries are single columns or equivalently $a(T) = 0$. Hence, $G_\lambda(x; \alpha, \beta)$ specializes to $G_\lambda(x; \beta)$ for $\alpha = 0$. Similarly, a hook-valued tableau T is a multiset-valued tableau when all hook tableaux entries are single rows or equivalently $\ell(T) = 0$. Hence, $G_\lambda(x; \alpha, \beta)$ specializes to $J_\lambda(x; \alpha)$ for $\beta = 0$.

In this paper, we describe a novel uncrowding algorithm on hook-valued tableaux (see Definitions 3.2, 3.4, and 3.5). The uncrowding algorithm on set-valued tableaux was originally developed by Buch [3, Theorem 6.11] to give a bijective proof of Lenart's Schur expansion of symmetric stable Grothendieck polynomials [9]. This uncrowding algorithm takes as input a set-valued tableau and produces a semistandard Young tableau (using the RSK bumping algorithm to uncrowd cells that contain more than one integer) and a flagged increasing tableau [9] (also known as an elegant filling [1,10,14]), which serves as a recording tableau.

Chan and Pflueger [4] provide an expansion of stable Grothendieck polynomials indexed by skew partitions in terms of skew Schur functions. Their proof uses a generalization of the uncrowding algorithm of Lenart [9], Buch [3], and Reiner et al. [15] to skew shapes. Their analysis is motivated geometrically by identifying Euler characteristics of Brill–Noether varieties up to sign as counts of set-valued standard tableaux. The uncrowding algorithm was also used in the analysis of K-theoretic analogs of the Hopf algebras of symmetric functions, quasisymmetric functions, noncommutative symmetric functions, and of the Malvenuto–Reutenauer Hopf algebra of permutations [1,10,14]. In [6], a vertex model for canonical Grothendieck polynomials and their duals was studied, which was used to derive Cauchy identities.

An important property of the uncrowding algorithm on set-valued tableaux is that it intertwines with crystal operators [13] (see also [12]). The crystal structure on a combinatorial set is the combinatorial shadow of a (quantum) group representation (see, for example, [2,7]). A crystal structure on hook-valued tableaux was recently introduced by Hawkes and Scrimshaw [8]. Our novel uncrowding map on hook-valued tableaux yields a set-valued tableau and a recording tableau. We prove that it intertwines with crystal operators (see Proposition 3.12 and Theorem 3.14). This was stated as an open problem in [8].

The paper is organized as follows. In Sect. 2, we review the definition of semistandard hook-valued tableaux of [16] and the crystal structure on them [8]. In Sect. 3, we define the new uncrowding map on hook-valued tableaux and prove that it intertwines with the crystal operators and other properties. We also give a variant of the uncrowding algorithm on hook-valued tableaux. In Sect. 4, we consider applications of the uncrowding algorithm, in particular expansions of the canonical Grothendieck polynomials using techniques developed in [1].

2. Hook-Valued Tableaux

In Sect. 2.1, we define hook-valued tableaux [16], and in Sect. 2.2, we review the crystal structure on hook-valued tableaux as introduced in [8].

2.1. Hook-Valued Tableaux

A semistandard Young tableau U of hook shape is a tableau of the form

$$U = \begin{array}{|c|cccc} \hline \ell_p & & & & \\ \hline \vdots & & & & \\ \hline \ell_1 & & & & \\ \hline x & \multicolumn{1}{|c|}{a_1} & \multicolumn{1}{c|}{\dots} & \multicolumn{1}{c|}{a_q} \\ \hline \end{array},$$

where the integer entries weakly increase from left to right and strictly increase from bottom to top. Note that we use French notation for Young diagrams and tableaux throughout the paper. In this case, $\mathsf{H}(U) = x$ is called the hook entry of U, $\mathsf{L}(U) = (\ell_1, \ell_2, \dots, \ell_p)$ is the leg of U, and $\mathsf{A}(U) = (a_1, a_2, \dots, a_q)$ is the arm of U. Both the arm and the leg of U are allowed to be empty. Additionally, the extended leg of U is defined as $\mathsf{L}^+(U) = (x, \ell_1, \ell_2, \ldots, \ell_p)$. We denote by $\max(U)$ (resp. $\min(U)$) the maximal (resp. minimal) entry in U.

Definition 2.1 [16]. Fix a partition λ. A semistandard hook-valued tableau (or hook-valued tableau for short) T of shape λ is a filling of the Young diagram for λ with (nonempty) semistandard Young tableaux of hook shape, such that

(i) $\max(A) \leq \min(B)$ whenever the cell containing A is in the same row, but left of the cell containing B;
(ii) $\max(A) < \min(C)$ whenever the cell containing A is in the same column, but below the cell containing C.

The set of all hook-valued tableaux of shape λ (respectively, with entries at most m) is denoted by $\mathsf{HVT}(\lambda)$ (respectively, $\mathsf{HVT}^m(\lambda)$).

Given a hook-valued tableau T, its arm excess is the total number of integers in the arms of all cells of T, while its leg excess is the total number of integers in the legs of all cells of T.

Remark 2.2. In the special case when a hook-valued tableau has arm excess 0, it is also called a set-valued tableau. Similarly, a multiset-valued tableau is a hook-valued tableau with leg excess 0. We use the notation $\mathsf{SVT}(\lambda)$ (resp. $\mathsf{SVT}^m(\lambda)$) and $\mathsf{MVT}(\lambda)$ (resp. $\mathsf{MVT}^m(\lambda)$) for the set of all set-valued tableaux of shape λ (resp. with entries at most m) and the set of all multiset-valued tableaux of shape λ (resp. with entries at most m), respectively.

2.2. Crystal Structure on Hook-Valued Tableaux

Hawkes and Scrimshaw [8] defined a crystal structure on hook-valued tableaux. We review their definition here.

Definition 2.3 ([8], Definition 4.1). Let C be a hook-valued tableau of column shape. The column reading word $R(C)$ is obtained by reading the extended leg in each cell from top to bottom, followed by reading all of the remaining entries, arranged in a weakly increasing order.

For a hook-valued tableau T, its column reading word is formed by concatenating the column reading words of all of its columns, read from left to right, that is

$$R(T) = R(C_1)R(C_2)\ldots R(C_\ell),$$

where ℓ is the number of columns of T and C_i is the ith column of T.

Example 2.4. Let T be the hook-valued tableau

$$T = \begin{array}{|l|l|l}\cline{1-2} 4 & & \\ 33 & 5 & \\ \hline 2 & 4 & \multicolumn{1}{l|}{} \\ 11 & 334 & \multicolumn{1}{l|}{4445} \\ \hline \end{array}.$$

The column reading words for the columns of T are, respectively, 432113, 54334, and 4445, so that

$$R(T) = 432113543344445.$$

Definition 2.5 [8, Definition 4.3]. Let $T \in \mathsf{HVT}^m(\lambda)$. For any $1 \le i < m$, we employ the following pairing rules. Assign $-$ to every i in $R(T)$ and assign $+$ to every $i+1$ in $R(T)$. Then, successively pair each $+$ that is adjacent and to the left of a $-$, removing all paired signs until nothing can be paired.

The operator f_i acts on T according to the following rules in the given order. If there is no unpaired $-$, then f_i annihilates T. Otherwise, locate the cell c with entry the hook-valued tableau $B = T(c)$ containing the i corresponding to the rightmost unpaired $-$.

(M) If there is an $i+1$ in the cell above c with entry $B^{\uparrow}$, then f_i removes an i from $\mathsf{A}(B)$ and adds $i+1$ to $\mathsf{A}(B^{\uparrow})$.
(S) Otherwise, if there is a cell to the right of c with entry $B^{\rightarrow}$, such that it contains an i in $\mathsf{L}^+(B^{\rightarrow})$, then f_i removes the i from $\mathsf{L}^+(B^{\rightarrow})$ and adds $i+1$ to $\mathsf{L}(B)$.
(N) Else, f_i changes the i in B into an $i+1$.

Similarly, the operator e_i acts on T according to the following rules in the given order. If there is no unpaired $+$, then e_i annihilates T. Otherwise, locate the cell c with entry the hook-valued tableau $B = T(c)$ containing the entry $i+1$ corresponding to the leftmost unpaired $+$.

(M) If there is an i in the cell below c with entry $B^{\downarrow}$, then e_i removes the $i+1$ from $\mathsf{A}(B)$ and adds i to $\mathsf{A}(B^{\downarrow})$.
(S) Otherwise, if there is a cell to the left of c with entry $B^{\leftarrow}$, such that it contains an $i+1$ in $\mathsf{L}(B^{\leftarrow})$, then e_i removes the $i+1$ from $\mathsf{L}(B^{\leftarrow})$ and adds i to $\mathsf{L}^+(B)$.
(N) Else, e_i changes the $i+1$ in B into an i.

Based on the pairing procedure above, $\varphi_i(T)$ is the number of unpaired $-$, whereas $\varepsilon_i(T)$ is the number of unpaired $+$.

We remark that the definition of crystal operators on HVT specializes to the definition on SVT in [13] or the one on MVT in [8] when the arm excess or leg excess of the tableaux is set to 0, respectively.

Example 2.6. Consider the following hook-valued tableau T:

$$T = \begin{array}{|l|l|}\hline \begin{array}{l}4\\34\end{array} & \begin{array}{l}5\\4\end{array} \\\hline \begin{array}{l}2\\11\end{array} & \begin{array}{l}3\\233\end{array} \\\hline\end{array}.$$

Then, e_3 annihilates T, whereas

$$e_1(T) = \begin{array}{|l|l|}\hline \begin{array}{l}4\\34\end{array} & \begin{array}{l}5\\4\end{array} \\\hline \begin{array}{l}\\ \\11\end{array} & \begin{array}{l}3\\2\\133\end{array} \\\hline\end{array}, \quad f_1(T) = \begin{array}{|l|l|}\hline \begin{array}{l}4\\34\end{array} & \begin{array}{l}5\\4\end{array} \\\hline \begin{array}{l}2\\12\end{array} & \begin{array}{l}3\\233\end{array} \\\hline\end{array}, \quad f_3(T) = \begin{array}{|l|l|}\hline \begin{array}{l}4\\34\end{array} & \begin{array}{l}5\\44\end{array} \\\hline \begin{array}{l}2\\11\end{array} & \begin{array}{l}3\\23\end{array} \\\hline\end{array}.$$

For a given cell (r,c) in row r and column c in a hook-valued tableau T, let $L_T(r,c)$ be the leg of $T(r,c)$, let $\mathsf{A}_T(r,c)$ be arm of $T(r,c)$, let $H_T(r,c)$ be the hook entry of $T(r,c)$, and let $L_T^+(r,c)$ be the extended leg of $T(r,c)$.

3. Uncrowding Map on Hook-Valued Tableaux

In Sect. 3.1, we first review the uncrowding map on set-valued tableaux. In Sect. 3.2, we give a new uncrowding map on hook-valued tableaux and prove some of its properties in Sect. 3.3. The relation to the uncrowding map on multiset-valued tableaux is given in Sect. 3.4. In Sect. 3.5, we give the inverse of the uncrowding map on hook-valued tableaux, called the crowding map. In Sect. 3.6, an alternative definition of the uncrowding map on hook-valued tableaux is provided.

3.1. Uncrowding Map on Set-Valued Tableaux

For set-valued tableaux, there exists an uncrowding operator, which maps a set-valued tableau to a pair of tableaux, one being a semistandard Young tableau and the other a flagged increasing tableau (see, for example, [1,3,9, 15]). In this setting, the uncrowding operator intertwines with the crystal operators on set-valued tableaux and semistandard Young tableaux, respectively [13].

Consider partitions λ, μ with $\lambda \subseteq \mu$ and $\lambda_1 = \mu_1$. A flagged increasing tableau (introduced in [9] and called (strict) elegant fillings by various authors [1,10,14]) is a row and column strict filling of the skew shape μ/λ, such that the positive integer entries in the ith row of the tableau are at most $i-1$ for all $1 \le i \le \ell(\mu)$, where $\ell(\mu)$ is the length of partition μ. In particular, the bottom row is empty. The set of all flagged increasing tableaux is denoted by $\mathcal{F}$. The set of all flagged increasing tableaux of shape μ/λ with $\lambda_1 = \mu_1$ is denoted by $\mathcal{F}(\mu/\lambda)$.

We now review the uncrowding operation on set-valued tableaux. We call a cell in a set-valued tableau a multicell if it contains more than one letter.

Definition 3.1. Define the uncrowding operation on $T \in \mathsf{SVT}(\lambda)$ as follows. First identify the topmost row r in T with a multicell. Let x be the largest letter in row r that lies in a multicell; remove x from the cell and perform RSK row bumping with x into the rows above. The resulting tableau, whose shape differs from λ by the addition of one cell, is the output of this operation.

The uncrowding map on set-valued tableaux

$$\mathcal{U}_{\mathsf{SVT}} : \mathsf{SVT}(\lambda) \longrightarrow \bigsqcup_{\mu \supseteq \lambda} \mathsf{SSYT}(\mu) \times \mathcal{F}(\mu/\lambda) \tag{3.1}$$

is defined as follows. Let $T \in \mathsf{SVT}(\lambda)$ with leg excess ℓ.

(1) Initialize $P_0 = T$ and $Q_0 = F_0$, where F_0 is the unique flagged increasing tableau of shape λ/λ.
(2) For each $1 \le i \le \ell$, P_i is obtained from P_{i-1} by applying the uncrowding operation. Let C be the cell in $\mathsf{shape}(P_i)/\mathsf{shape}(P_{i-1})$. If C is in row r', then F_i is obtained from F_{i-1} by adding cell C with entry $r' - r$.
(3) Set $\mathcal{U}_{\mathsf{SVT}}(T) = (P, F) := (P_\ell, F_\ell)$.

It was proved in [3, Section 6] that $\mathcal{U}_{\mathsf{SVT}}$ in (3.1) is a bijection. Monical et al. [13] proved that $\mathcal{U}_{\mathsf{SVT}}$ intertwines with the crystal operators on set-valued tableaux (see also [12]). A similar uncrowding algorithm for multiset-valued tableaux was given in [8, Section 3.2].

3.2. Uncrowding Map on Hook-Valued Tableaux

In [8], the authors ask for an uncrowding map for hook-valued tableaux which intertwines with the crystal operators. Here, we provide such an uncrowding map by uncrowding the arm excess in a hook-valued tableaux to obtain a set-valued tableaux. An alternative obtained by uncrowding the leg excess first is given in Sect. 3.4.

Definition 3.2. The uncrowding bumping $\mathcal{V}_b \colon \mathsf{HVT} \to \mathsf{HVT}$ is defined by the following algorithm:

(1) Initialize T as the input.
(2) If the arm excess of T equals zero, return T.
(3) Else, find the rightmost column that contains a cell with nonzero arm excess. Within this column, find the cell with the largest value in its arm. (In French notation, this is the topmost cell with nonzero arm excess in the specified column.) Denote the row index and column index of this cell by r and c, respectively. Denote the cell as (r, c), its rightmost arm entry by a, and its largest leg entry by ℓ.
(4) Look at the column to the right of (r, c) (i.e., column $c+1$) and find the smallest number that is greater than or equal to a.
    - If no such number exists, attach an empty cell to the top of column $c+1$ and label the cell as $(\tilde{r}, c+1)$, where $\tilde{r}$ is its row index. Let k be the empty character.
    - If such a number exists, label the value as k and the cell containing k as $(\tilde{r}, c+1)$ where $\tilde{r}$ is the cell's row index.

    We now break into cases:
    (a) If $\tilde{r} \neq r$, then remove a from $\mathsf{A}_T(r, c)$, replace k with a, and attach k to the arm of $\mathsf{A}_T(\tilde{r}, c+1)$.
    (b) If $\tilde{r} = r$ then remove $(a, \ell] \cap \mathsf{L}_T(r, c)$ from $\mathsf{L}_T(r, c)$ where $(a, \ell] = \{a+1, a+2, \ldots, \ell\}$, remove a from $\mathsf{A}_T(r, c)$, insert $(a, \ell] \cap \mathsf{L}_T(r, c)$ into $\mathsf{L}_T(\tilde{r}, c+1)$, replace the hook entry of $(\tilde{r}, c+1)$ with a, and attach k to $\mathsf{A}_T(\tilde{r}, c+1)$.
(5) Output the resulting tableau.

See Figs. 1 and 2 for illustration.

Lemma 3.3. *The map $\mathcal{V}_b$ is well defined. More precisely, for $T \in \mathsf{HVT}$, we have $\mathcal{V}_b(T) \in \mathsf{HVT}$.*

Proof. It suffices to check that $\mathcal{V}_b$ preserves the semistandardness condition of both the entire hook-valued tableau and the filling within each cell. We break into two cases depending on whether Step a or b in Definition 3.2 is applied.

Case 1 Assume Step a is applied. To verify semistandardness within each cell, it suffices to check cells (r, c) and $(\tilde{r}, c+1)$. The semistandardness within cell (r, c) is clearly preserved as the only change to the hook-shaped tableau in cell (r, c) is that an entry was removed from $\mathsf{A}_T(r, c)$.

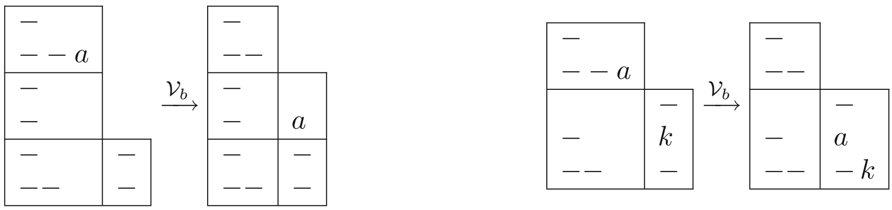


FIGURE 1. When $\tilde{r} \neq r$. Left: $(\tilde{r}, c+1)$ is a new cell; right: $(\tilde{r}, c+1)$ is an existing cell

FIGURE 2. When $\tilde{r} = r$. Left: $(r, c+1)$ is a new cell; right: $(r, c+1)$ is an existing cell

We now check the semistandardness condition within cell $(\tilde{r}, c+1)$. We have that $\mathcal{V}_b$ either created the cell $(\tilde{r}, c+1)$ and inserted the number a in it or $\mathcal{V}_b$ replaced k with a and appended k to the arm of cell $(\tilde{r}, c+1)$. In both cases, the tableau in cell $(\tilde{r}, c+1)$ is a semistandard hook-shaped tableau. In the second case, this is true, since k is weakly greater than $\mathsf{H}_T(\tilde{r}, c+1)$ and k is the smallest number weakly greater than a in column $c+1$.

We now check the semistandardness of the entire tableau. Note that it suffices to check the semistandardness in row $\tilde{r}$ and column $c+1$. Since $\tilde{r} < r$, the semistandardness in row $\tilde{r}$ is preserved as a is larger than every number in $(\tilde{r}, c)$ and k remains in the same cell. Also, the semistandardness in column $c+1$ is preserved as k is chosen to be the smallest number in column $c+1$ that is weakly greater than a.

Case 2 Assume Step b is applied. The semistandardness within cell (r, c) is clearly preserved as the only change to (r, c) is that entries from $\mathsf{L}_T(r, c)$ and $\mathsf{A}_T(r, c)$ are removed. We now check the semistandardness condition within cell $(r, c+1)$. If $(a, \ell] \cap \mathsf{L}_T(r, c) = \emptyset$, then a is weakly larger than all elements of (r, c). In this case, the semistandardness within cell $(r, c+1)$ follows from the argument in Case 1. If $(a, \ell] \cap \mathsf{L}_T(r, c) \neq \emptyset$, then a is not weakly larger than all elements of (r, c). After applying $\mathcal{V}_b$, the semistandardness condition in the leg of $(r, c+1)$ will still hold as $a < x < z$ for all $x \in (a, \ell] \cap \mathsf{L}_T(r, c)$, where z is the smallest value in $\mathsf{L}_T(r, c+1)$. Similarly, the semistandardness condition in the arm of $(r, c+1)$ holds as $a < k$ or k is the empty character. Thus, the semistandardness condition in each cell is preserved. The semistandardness of row r is preserved as all numbers strictly greater than a in (r, c) are moved to $(r, c+1)$ along with a. The semistandardness condition within column $c+1$ is preserved as every number in $(r+1, c+1)$ is strictly greater than ℓ and every number in $(r-1, c+1)$ is strictly less than a.

$\square$

Definition 3.4. The uncrowding insertion $\mathcal{V}\colon \mathsf{HVT} \to \mathsf{HVT}$ is defined as $\mathcal{V}(T) = \mathcal{V}_b^d(T)$, where the integer $d \geq 1$ is minimal, such that $\mathsf{shape}(\mathcal{V}_b^d(T))/\mathsf{shape}(\mathcal{V}_b^{d-1}(T)) \neq \emptyset$ or $\mathcal{V}_b^d(T) = \mathcal{V}_b^{d-1}(T)$.

A column-flagged increasing tableau is a tableau whose transpose is a flagged increasing tableau. Let $\hat{\mathcal{F}}$ denote the set of all column-flagged increasing tableaux. Let $\hat{\mathcal{F}}(\mu/\lambda)$ denote the set of all column-flagged increasing tableaux of shape μ/λ.

Definition 3.5. Let $T \in \mathsf{HVT}(\lambda)$ with arm excess α. The uncrowding map

$$\mathcal{U}\colon \mathsf{HVT}(\lambda) \to \bigsqcup_{\mu \supseteq \lambda} \mathsf{SVT}(\mu) \times \hat{\mathcal{F}}(\mu/\lambda)$$

is defined by the following algorithm:

(1) Let $P_0 = T$ and let Q_0 be the column-flagged increasing tableau of shape λ/λ.

(2) For $1 \leq i \leq \alpha$, let $P_{i+1} = \mathcal{V}(P_i)$. Let c be the index of the rightmost column of P_i containing a cell with nonzero arm excess and let $\tilde{c}$ be the column index of the cell $\mathsf{shape}(P_{i+1})/\mathsf{shape}(P_i)$. Then, Q_{i+1} is obtained from Q_i by appending the cell $\mathsf{shape}(P_{i+1})/\mathsf{shape}(P_i)$ to Q_i and filling this cell with $\tilde{c} - c$.

Define $\mathcal{U}(T) = (P(T), Q(T)) := (P_\alpha, Q_\alpha)$.

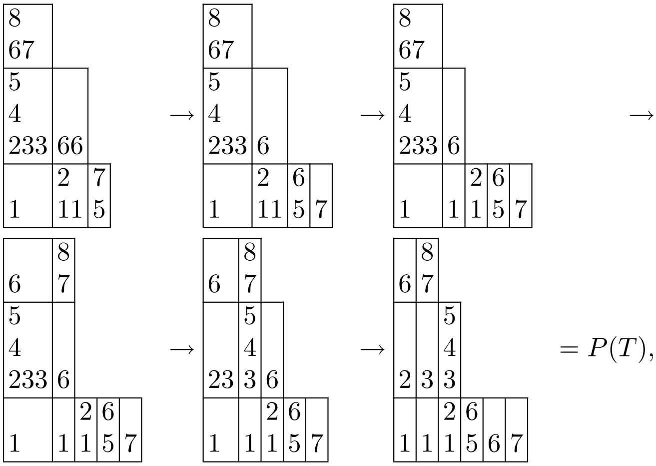
Example 3.6. Let T be the hook-valued tableau

$$\begin{array}{lll}\cline{1-1} \multicolumn{1}{|l|}{\begin{matrix}8\\67\end{matrix}} & & \\ \cline{1-2} \multicolumn{1}{|l|}{\begin{matrix}5\\4\\233\end{matrix}} & \multicolumn{1}{l|}{66} & \\ \hline \multicolumn{1}{|l|}{1} & \multicolumn{1}{l|}{\begin{matrix}2\\11\end{matrix}} & \multicolumn{1}{l|}{\begin{matrix}7\\5\end{matrix}} \\ \hline \end{array}$$

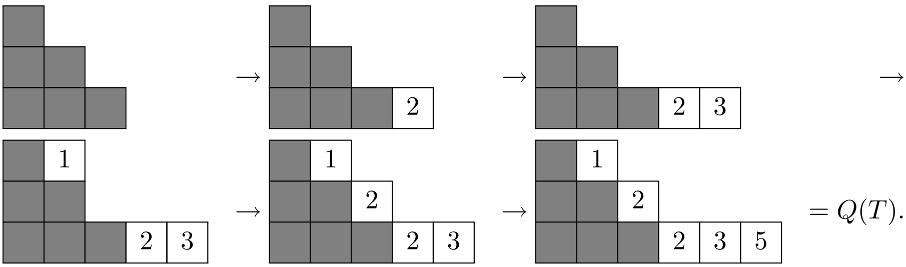
Then, we obtain the following sequence of tableaux $\mathcal{V}_b^i(T)$ for $0 \leq i \leq 2 = d$ when computing the first uncrowding insertion:

$$\begin{array}{lll}\cline{1-1} \multicolumn{1}{|l|}{\begin{matrix}8\\67\end{matrix}} & & \\ \cline{1-2} \multicolumn{1}{|l|}{\begin{matrix}5\\4\\233\end{matrix}} & \multicolumn{1}{l|}{66} & \\ \hline \multicolumn{1}{|l|}{1} & \multicolumn{1}{l|}{\begin{matrix}2\\11\end{matrix}} & \multicolumn{1}{l|}{\begin{matrix}7\\5\end{matrix}} \\ \hline \end{array} \to \begin{array}{lll}\cline{1-1} \multicolumn{1}{|l|}{\begin{matrix}8\\67\end{matrix}} & & \\ \cline{1-2} \multicolumn{1}{|l|}{\begin{matrix}5\\4\\233\end{matrix}} & \multicolumn{1}{l|}{6} & \\ \hline \multicolumn{1}{|l|}{1} & \multicolumn{1}{l|}{\begin{matrix}2\\11\end{matrix}} & \multicolumn{1}{l|}{\begin{matrix}6\\57\end{matrix}} \\ \hline \end{array} \to \begin{array}{llll}\cline{1-1} \multicolumn{1}{|l|}{\begin{matrix}8\\67\end{matrix}} & & & \\ \cline{1-2} \multicolumn{1}{|l|}{\begin{matrix}5\\4\\233\end{matrix}} & \multicolumn{1}{l|}{6} & & \\ \hline \multicolumn{1}{|l|}{1} & \multicolumn{1}{l|}{\begin{matrix}2\\11\end{matrix}} & \multicolumn{1}{l|}{\begin{matrix}6\\5\end{matrix}} & \multicolumn{1}{l|}{7} \\ \hline \end{array} = \mathcal{V}(T).$$

Continuing with the remaining uncrowding insertions, we obtain the following sequences of tableaux for the uncrowding map:



$= P(T),$



$= Q(T).$

Corollary 3.7. *Let $T \in \mathsf{HVT}$. Then, $P(T)$ is a set-valued tableau.*

Proof. By Lemma 3.3 and Definition 3.4, we have that $\mathcal{V}(T)$ is a hook-valued tableau. Note that if the arm excess of T is nonzero, then the arm excess of $\mathcal{V}(T)$ is one less than that of T. Since $P(T) = \mathcal{V}^{\alpha}(T)$, where α is the arm excess of T, we have that the arm excess of $P(T)$ is zero. Thus, $P(T)$ is a set-valued tableau. $\square$

Definition 3.8. Let $T \in \mathsf{HVT}$ and let d be minimal, such that $\mathcal{V}(T) = \mathcal{V}_b^d(T)$. The insertion path p of $T \to \mathcal{V}(T)$ is defined as follows:

- If $d = 0$, set $p = \emptyset$.
- Otherwise, let (r_0, c_0) be the rightmost and topmost cell of T containing a cell with nonzero arm excess. For all $1 \leq j \leq d$, let $c_j = c_0 + j$ and let $r_j = \tilde{r}$ be $\tilde{r}$ in Definition 3.2 when $\mathcal{V}_b$ is applied to $\mathcal{V}_b^{j-1}(T)$. Set $p = ((r_0, c_0), (r_1, c_1), \ldots, (r_d, c_d))$.

Lemma 3.9. *Let $T \in \mathsf{HVT}$. Then, $Q(T)$ is a column-flagged increasing tableau.*

Proof. By construction, the positive integer entries in column i of $Q(T)$ are at most $i-1$. Let m be the smallest nonnegative integer, such that $\mathcal{V}^m(T) = P(T)$.

Let $p^i = ((r_0^i, c_0^i), (r_1^i, c_1^i), \ldots, (r_{d_i}^i, c_{d_i}^i))$ for $0 \leq i < m$ be the insertion path of $\mathcal{V}^i(T) \to \mathcal{V}^{i+1}(T)$. Since $c_0^{i+1} \leq c_0^i$ for all $0 \leq i < m$, the entries in each row of $Q(T)$ are strictly increasing. To check that the entries in each column of $Q(T)$ are strictly increasing, it suffices to show that if $c_0^{i+1} = c_0^i$ then p^{i+1} lies weakly below p^i. In other words, it suffices to check that $c_0^{i+1} = c_0^i$ implies that $r_j^{i+1} \leq r_j^i$ for all $0 \leq j \leq d_i$. We prove this by induction on j. Note that $r_0^{i+1} \leq r_0^i$ by the definition of $\mathcal{U}$. Assume by induction that $r_j^{i+1} \leq r_j^i$. This implies that the a when applying $\mathcal{V}_b$ to $\mathcal{V}_b^j(\mathcal{V}^i(T))$ is weakly smaller than the a when applying $\mathcal{V}_b$ to $\mathcal{V}_b^j(\mathcal{V}^{i-1}(T))$. Thus, we must have $r_{j+1}^{i+1} \leq r_{j+1}^i$. □

3.3. Properties of the Uncrowding Map

Let T be a hook-valued tableau. Define $R_i(T)$ as the induced subword of $R(T)$ consisting only of the letters i and $i+1$. In the next lemma, we use the same notation as in Definition 3.2. Furthermore, two words are Knuth equivalent if one can be transformed to the other by a sequence of Knuth equivalences on three consecutive letters

$$xzy \equiv zxy \quad \text{for} x \leq y < z, \qquad yxz \equiv yzx \quad \text{for } x < y \leq z.$$

Lemma 3.10. *For $T \in \mathsf{HVT}$, $R_i(T) = R_i(\mathcal{V}_b(T))$ unless T satisfies one of the following three conditions:*

(a) *$a = i$ or $a = i+1$ and column $c+1$ contains both an i and an $i+1$,*
(b) *$\tilde{r} = r$, $i \in (a, \ell] \cap \mathsf{L}_T(r, c)$, $k = i$, and column $c+1$ contains an $i+1$,*
(c) *$\tilde{r} = r$, $a = i$, $i+1 \in (a, \ell] \cap \mathsf{L}_T(r, c)$, and (r, c) contains another i besides a.*

Moreover, $R_i(T)$ is Knuth equivalent to $R_i(\mathcal{V}_b(T))$.

Proof. Let $R_i(T) = r_1 r_2 \ldots r_m$. We break into cases based on the value of a.
Case 1: Assume $a \neq i, i+1$.

Assume Step a is applied by $\mathcal{V}_b$. If $k \neq i, i+1$, then $R_i(T) = R_i(\mathcal{V}_b(T))$ as the position of all letters i and $i+1$ remains the same. Let $k = i$. We have that k is the only i in column $c+1$. Hence, when k gets bumped from $\mathsf{L}_T(\tilde{r}, c+1)$ and appended to $\mathsf{A}_T(\tilde{r}, c+1)$, the relative position of k to the other letters i and $i+1$ in $R_i(T)$ does not change. Thus, $R_i(T) = R_i(\mathcal{V}_b(T))$. Let $k = i+1$. Note that column $c+1$ cannot have a cell containing an i as k is the smallest number weakly greater than a. Hence, moving k from $\mathsf{L}_T(\tilde{r}, c+1)$ to $\mathsf{A}_T(\tilde{r}, c+1)$ will not change $R_i(T)$. Therefore, we once again have that $R_i(T) = R_i(\mathcal{V}_b(T))$.

Assume Step b is applied by $\mathcal{V}_b$. Consider the subcase when $(a, \ell] \cap \mathsf{L}_T(r, c) = \emptyset$. By a similar argument to the previous paragraph, we have that $R_i(T) = R_i(\mathcal{V}_b(T))$. Next, consider the subcase when $i+1 \in (a, \ell] \cap \mathsf{L}_T(r, c)$. This implies that $a < i$ and the only time $i+1$ occurs in column c is in $\mathsf{L}_T(r, c)$. Note that if an i exists in column c, it must be contained in $\mathsf{L}_T(r, c)$. We also have that $k \geq i+1$ or k is the empty character and no cell in column $c+1$ contains an i. Thus, removing $(a, \ell] \cap \mathsf{L}_T(r, c)$ from $\mathsf{L}_T(r, c)$, replacing k with $(a, \ell] \cap \mathsf{L}_T(r, c)$ in $\mathsf{L}_T(r, c+1)$, and appending k to $\mathsf{A}_T(r, c+1)$ do not change $R_i(T)$. Therefore, $R_i(T) = R_i(\mathcal{V}_b(T))$. Let $i \in (a, \ell] \cap \mathsf{L}_T(r, c)$

and $i+1 \notin (a,\ell] \cap \mathsf{L}_T(r,c)$. Note that the only place $i+1$ can occur in column c is as $\mathsf{H}_T(r+1,c)$ and the only place i can occur in column c is in $\mathsf{L}_T(r,c)$. This implies that removing $(a,\ell] \cap \mathsf{L}_T(r,c)$ from $\mathsf{L}_T(r,c)$, replacing k with $(a,\ell] \cap \mathsf{L}_T(r,c)$ in $\mathsf{L}_T(r,c+1)$, and appending k to $\mathsf{A}_T(r,c+1)$ will not change $R_i(T)$ unless both $i+1$ and i show up in column $c+1$. This can only occur when $k=i$ which implies that $R_i(T) = r_1 \dots i\ i+1\ k \dots r_m$ and $R_i(\mathcal{V}_b(T)) = r_1 \dots i{+}1\ i\ k \dots r_m$. We see that $R_i(T)$ and $R_i(\mathcal{V}_b(T))$ only differ by a Knuth relation implying that they are Knuth equivalent. Assume that $i, i+1 \notin (a,\ell] \cap \mathsf{L}_T(r,c) \neq \emptyset$. If $a > i+1$ the positions of all letters i and $i+1$ remain the same after $\mathcal{V}_b$ is applied. If $a < i$, then the positions of all letters i and $i+1$ also remain the same unless $k=i$ or $k=i+1$. In both of these special subcases, it can be checked that still $R_i(T) = R_i(\mathcal{V}_b(T))$.

Case 2: Assume $a = i$.

Assume Step a is applied by $\mathcal{V}_b$. If column $c+1$ does not contain both an i and an $i+1$, then we have $R_i(T) = R_i(\mathcal{V}_b(T))$. However, if both an i and an $i+1$ are in column $c+1$, then $R_i(T) = r_1 \dots i\ i+1\ i \dots r_m$ and $R_i(\mathcal{V}_b(T)) = r_1 \dots i+1\ i\ i \dots r_m$ which are Knuth equivalent.

Assume Step b is applied by $\mathcal{V}_b$. Consider the subcase when $(a,\ell] \cap \mathsf{L}_T(r,c) = \emptyset$. By a similar argument to the previous paragraph, we have that $R_i(T) = R_i(\mathcal{V}_b(T))$ unless both an i and an $i+1$ are in column $c+1$ in which case $R_i(T)$ and $R_i(\mathcal{V}_b(T))$ are only Knuth equivalent. Consider the subcase given by $i+1 \in (a,\ell] \cap \mathsf{L}_T(r,c)$. Note that no cell in column $c+1$ can contain an i, the only cell that could contain an $i+1$ in column $c+1$ is $(r,c+1)$, and the only cell containing letters i or $i+1$ in column c is (r,c). This implies that it suffices to look at the changes to (r,c) and $(r,c+1)$. We see that $R_i(T) = r_1 \dots i+1\ \underbrace{i \dots i\, a}_{\gamma} \dots r_m$ and $R_i(\mathcal{V}_b(T)) = r_1 \dots \underbrace{i \dots i}_{\gamma-1}\ i+1\ a$ where $\gamma \geq 1$ is the number of letters i in cell (r,c) including a. We see that $R_i(T)$ and $R_i(\mathcal{V}_b(T))$ are Knuth equivalent. Consider the subcase when $i+1 \notin (a,\ell] \cap \mathsf{L}_T(r,c) \neq \emptyset$. We have that both i and $i+1$ cannot be in a cell in column $c+1$ and an $i+1$ cannot be in column c. Thus applying $\mathcal{V}_b$ does not change $R_i(T)$ giving us that $R_i(T) = R_i(\mathcal{V}_b(T))$.

Case 3: Assume $a = i+1$.

Assume Step a is applied by $\mathcal{V}_b$. If column $c+1$ does not contain both i and $i+1$, then we have that $R_i(T) = R_i(\mathcal{V}_b(T))$. However, if both i and $i+1$ occur in column $c+1$, then $R_i(T) = r_1 \dots i+1\ i+1\ i \dots r_m$ and $R_i(\mathcal{V}_b(T)) = r_1 \dots i+1\ i\ i+1 \dots r_m$ which are Knuth equivalent.

Assume Step b is applied by $\mathcal{V}_b$. If $(a,\ell] \cap \mathsf{L}_T(r,c) = \emptyset$, then $R_i(T) = R_i(\mathcal{V}_b(T))$ unless both i and $i+1$ occur in column $c+1$. In this exceptional case, we have that $R_i(T)$ and $R_i(\mathcal{V}_b(T))$ are only Knuth equivalent by a similar argument to the previous paragraph. If $(a,\ell] \cap \mathsf{L}_T(r,c) \neq \emptyset$, then $k > i+1$ or k is the empty character and no cell in column $c+1$ contains an $i+1$. Thus, applying $\mathcal{V}_b$ does not change $R_i(T)$ giving us that $R_i(T) = R_i(\mathcal{V}_b(T))$. $\square$

Remark 3.11. In general, the full reading words are not Knuth equivalent under the uncrowding map. For example, take the following hook-valued tableau

T, which uncrowds to a set-valued tableau S:

$$T = \begin{array}{|l|l|}\hline 4 & \\ 3 & \\ 2 & 5 \\ 12 & 4 \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline & 4 & \\ 2 & 3 & 5 \\ 1 & 2 & 4 \\ \hline\end{array} = S.$$

The reading word changed from 4321254 to 2143254, which are not Knuth equivalent.

Proposition 3.12. *Let $T \in \mathsf{HVT}$.*

(1) *If $f_i(T) = 0$, then $f_i(P(T)) = 0$.*
(2) *If $e_i(T) = 0$, then $e_i(P(T)) = 0$.*

Proof. Since $P(T) = \mathcal{V}_b^s(T)$ for some $s \in \mathbb{N}$ and Knuth equivalence is transitive, we have that $R_i(T)$ is Knuth equivalent to $R_i(P(T))$ by the previous lemma. As $f_i(T) = 0$, we have that every i in $R_i(T)$ is i-paired with an $i+1$ to its left. This property is preserved under Knuth equivalence giving us that $f_i(P(T)) = 0$. The same reasoning implies (2). □

Lemma 3.13. *Let $T \in \mathsf{HVT}$.*

(1) *If $f_i(T) \neq 0$, then $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T)) \neq 0$.*
(2) *If $e_i(T) \neq 0$, then $e_i(\mathcal{V}_b(T)) = \mathcal{V}_b(e_i(T)) \neq 0$.*

Proof. We are going to prove (1). Part (2) follows, since e_i and f_i are partial inverses.

Let a, ℓ, k, r, c, and $\tilde{r}$ be defined as in Definition 3.2 when $\mathcal{V}_b$ is applied to T. Similarly, define a', ℓ', k', r', c', and $\tilde{r}'$ for when $\mathcal{V}_b$ is applied to $f_i(T)$. Let $R_i(T) = r_1 r_2 \dots r_m$ and $R_i(\mathcal{V}_b(T)) = r_1' r_2' \dots r_m'$ be the corresponding reading words. Let $(\hat{r}, \hat{c})$ denote the cell containing the rightmost unpaired i in T, where $\hat{r}$ and $\hat{c}$ are its row and column index, respectively. We break into cases based on the position of $(\hat{r}, \hat{c})$ to (r, c).

Case 1 Assume $(\hat{r}, \hat{c}) = (r, c)$. We break into subcases based on how f_i acts on T.

- Assume that $(r+1, c)$ contains an $i+1$.
  As every entry in (r, c) must be strictly smaller than the values in $(r+1, c)$ and (r, c) must contain an i, we have that $\ell = i$ or $a = i$. If $\ell = i$, then ℓ is i-paired with the $i+1$ in $(r+1, c)$. Hence, a is always equal to i and a must correspond to the rightmost unpaired i of T. Thus, f_i acts on T by removing a from (r, c) and appending an $i+1$ to $\mathsf{A}_T(r+1, c)$. Note that $(a, \ell] \cap \mathsf{L}_T(r, c) = \emptyset$ implying $\mathcal{V}_b$ acts on T by removing a from $\mathsf{A}_T(r, c)$, replacing k in $(\tilde{r}, c+1)$ with a, and appending k to $\mathsf{A}_T(\tilde{r}, c+1)$ where $\tilde{r} \leq r$. We break into subcases based on where the values of i and $i+1$ are in column $c+1$ utilizing the fact that column $c+1$ cannot contain an i without an $i+1$ (since the arm excess of cell $(r+1, c)$ is zero and cell (r, c) contains the rightmost unpaired i).

Assume that column $c+1$ does not contain an i. Since a corresponds to the rightmost unpaired i in T and column $c+1$ does not contain an i, we have that the rightmost unpaired i in $\mathcal{V}_b(T)$ is precisely a in the cell $(\tilde{r}, c+1)$. Note that $(\tilde{r}+1, c+1)$ does not contain an $i+1$ in $\mathcal{V}_b(T)$ as $k \geq i+1$ or k is the empty character. Similarly, we have that $(\tilde{r}, c+2)$ does not contain an i. Thus, f_i acts on $\mathcal{V}_b(T)$ by changing a to an $i+1$ in $(\tilde{r}, c+1)$. We now consider $\mathcal{V}_b(f_i(T))$. When applying $\mathcal{V}_b$ to $f_i(T)$, a' is precisely the $i+1$ appended to $\mathsf{A}_T(r+1, c)$ and k' is the same as k. Since $\tilde{r}' = \tilde{r} < r+1$, we have that $\mathcal{V}_b$ acts on $f_i(T)$ by removing $i+1$ from $\mathsf{A}_{f_i(T)}(r+1, c)$, replacing k with an $i+1$ in $(\tilde{r}, c+1)$, and appending k to $\mathsf{A}_{f_i(T)}(\tilde{r}, c+1)$. We see that $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$.

Assume that column $c+1$ contains both an i and an $i+1$ in the same cell. Note that this implies that $k = i$. Since a is the rightmost unpaired i in T and the only cell in column $c+1$ that contained an $i+1$ or an i is $(\tilde{r}, c+1)$, we have that the rightmost unpaired i in $\mathcal{V}_b(T)$ is the i appended to $\mathsf{A}_T(\tilde{r}, c+1)$. Since $(\tilde{r}, c+1)$ contains an $i+1$, we have that $(\tilde{r}+1, c+1)$ cannot contain an $i+1$ and $(\tilde{r}, c+2)$ cannot contain an i. Thus, f_i acts on $\mathcal{V}_b(T)$ by changing the i in $\mathsf{A}_{\mathcal{V}_b(T)}(\tilde{r}, c+1)$ to an $i+1$. We now consider $\mathcal{V}_b(f_i(T))$. When applying $\mathcal{V}_b$ to $f_i(T)$, a' is precisely the $i+1$ appended to $\mathsf{A}_T(r+1, c)$ and k' is the $i+1$ in $(\tilde{r}, c+1)$. Since $\tilde{r}' = \tilde{r} < r+1$, we have that $\mathcal{V}_b$ acts on $f_i(T)$ by removing $i+1$ from $\mathsf{A}_{f_i(T)}(r+1, c)$, replacing $i+1$ in $(\tilde{r}, c+1)$ with the $i+1$ from $\mathsf{A}_{f_i(T)}(r+1, c)$, and appending an $i+1$ to $\mathsf{A}_{f_i(T)}(\tilde{r}, c+1)$. We see that $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$.

Assume that column $c+1$ contains both an i and an $i+1$ in different cells. Note that this implies that $k = i$. Since a corresponds to the rightmost unpaired i in $R_i(T)$ and the only $i+1$ and i in column $c+1$ are in cells $(\tilde{r}+1, c+1)$ and $(\tilde{r}, c+1)$, respectively, we have that the rightmost unpaired i in $R_i(\mathcal{V}_b(T))$ corresponds to the i appended to $\mathsf{A}_T(\tilde{r}, c+1)$. By assumption, we have that $(\tilde{r}+1, c+1)$ contains an $i+1$. Thus, f_i acts on $\mathcal{V}_b(T)$ by removing the i from $\mathsf{A}_{\mathcal{V}_b(T)}(\tilde{r}, c+1)$ and appending an $i+1$ to $\mathsf{A}_{\mathcal{V}_b(T)}(\tilde{r}+1, c+1)$. We now consider $\mathcal{V}_b(f_i(T))$. When applying $\mathcal{V}_b$ to $f_i(T)$, a' is precisely the $i+1$ appended to $\mathsf{A}_T(r+1, c)$ and k' is the $i+1$ in cell $(\tilde{r}+1, c+1)$. If $\tilde{r}' = r+1$, then $i+1$ is weakly larger than every value in $(r+1, c)$. Thus, either $(a', \ell'] \cap \mathsf{L}_{f_i(T)}(r+1, c) = \emptyset$ or $\tilde{r}' < r+1$. This implies that $\mathcal{V}_b$ acts on $f_i(T)$ by removing $i+1$ from $\mathsf{A}_{f_i(T)}(r+1, c)$, replacing the $i+1$ in $\mathsf{H}_{f_i(T)}(\tilde{r}+1, c+1)$ with the $i+1$ removed from $\mathsf{A}_{f_i(T)}(r+1, c)$, and appending an $i+1$ to $\mathsf{A}_{f_i(T)}(\tilde{r}+1, c+1)$. We see that $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$.

- Assume that $(r+1, c)$ does not contain an $i+1$ and $(r, c+1)$ contains an i.

Under these assumptions, we have that no cell in column c can contain an $i+1$. This implies that column $c+1$ must contain an $i+1$. The cell $(r+1, c+1)$ cannot have an $i+1$ as this would force $(r+1, c)$ to also have an $i+1$. Thus, $(r, c+1)$ must contain an $i+1$ in its leg. By our assumption, we have that f_i acts on T by removing the i from $(r, c+1)$ and appending an $i+1$ to $\mathsf{L}_T(r, c)$. We break into subcases according to where the rightmost unpaired i sits inside the cell (r, c). If the rightmost unpaired i is in $\mathsf{H}_T(r, c)$, then $a \geq i$ which would either contradict the hook entry being the rightmost unpaired i or cell $(r, c+1)$ containing an i. Thus, we only need to consider the subcases where the rightmost unpaired i is either in the leg or arm of (r, c).

Assume that the rightmost unpaired i is in $\mathsf{L}_T(r, c)$ for this entire paragraph. This implies that $\ell = i$. Since $(r, c+1)$ contains an i, we have that $a < i$. If $\tilde{r} < r$, then $\mathcal{V}_b$ acts on T by removing a from (r, c), replacing k with a in $(\tilde{r}, c+1)$, and appending k to $\mathsf{A}_T(\tilde{r}, c+1)$. Since $a, k < i$, we have that $\mathcal{V}_b$ does not change position of the rightmost unpaired i. Note that $(r+1, c)$ still does not contain an $i+1$, while $(r, c+1)$ still contains an i. Thus, f_i acts on $\mathcal{V}_b(T)$ by removing the i from $(r, c+1)$ and appending an $i+1$ to $\mathsf{L}_{\mathcal{V}_b(T)}(r, c)$. We now consider $\mathcal{V}_b(f_i(T))$. Note that (r', c'), a', and k' are the same as (r, c), a, and k, respectively. Thus, $\mathcal{V}_b$ acts in the same way as before. This gives us that $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$. If $\tilde{r} = r$, then k is precisely the i in cell $(r, c+1)$. We see that $\mathcal{V}_b$ acts on T by removing $(a, i] \cap \mathsf{L}_T(r, c)$ from $\mathsf{L}_T(r, c)$ and a from $\mathsf{A}_T(r, c)$, replacing k with $((a, i] \cap \mathsf{L}_T(r, c)) \cup \{a\}$, and appending k to $\mathsf{A}_T(r+1, c)$. Since there is an $i+1$ in $\mathsf{L}_{\mathcal{V}_b(T)}(r, c+1)$, we see that the rightmost unpaired i in $\mathcal{V}_b(T)$ is precisely k in $\mathsf{A}_{\mathcal{V}_b(T)}(r, c+1)$. Note that $(r+1, c+1)$ does not contain an $i+1$ and $(r, c+2)$ does not contain an i, because $(r, c+1)$ contains an $i+1$. Thus, f_i acts on $\mathcal{V}_b(T)$ by changing the i in $\mathsf{A}_{\mathcal{V}_b(T)}(r, c+1)$ to an $i+1$. We now consider $\mathcal{V}_b(f_i(T))$. We have that a' is the same as a and k' are the $i+1$ in $(r, c+1)$. We have $(a', \ell'] \cap \mathsf{L}_{f_i(T)}(r', c') = \{i+1\} \cup ((a, i] \cap \mathsf{L}_T(r, c))$. This implies that $\mathcal{V}_b$ acts on $f_i(T)$ by removing $\{i+1\} \cup ((a, i] \cap \mathsf{L}_T(r, c))$ from $\mathsf{L}_{f_i(T)}(r, c)$ and a from $\mathsf{A}_{f_i(T)}(r, c)$, replacing $i+1$ with $\{i+1\} \cup ((a, i] \cap \mathsf{L}_T(r, c)) \cup \{a\}$ in $(r, c+1)$, and appending an $i+1$ to $\mathsf{A}_{f_i(T)}(r, c+1)$. We see that $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$.

Assume that the rightmost unpaired i is in $\mathsf{A}_T(r, c)$. This implies that $a = i$ and forces a to correspond to the rightmost unpaired i. We also have that k is the i in $(r, c+1)$. Since i is weakly greater than all values in (r, c), we have that $(a, \ell] \cap \mathsf{L}_T(r, c) = \emptyset$. Thus, $\mathcal{V}_b$ acts on T by removing a from (r, c), replacing k with a in $(r, c+1)$, and appending k to $\mathsf{A}_T(r, c+1)$. Since a was the rightmost unpaired

i in T and cell $(r, c+1)$ contains an $i+1$ in its leg, we have that the rightmost unpaired i in $\mathcal{V}_b(T)$ is k in $\mathsf{A}_{\mathcal{V}_b(T)}(r, c+1)$. As $i+1$ is in $(r, c+1)$, we have that $(r+1, c+1)$ cannot contain an $i+1$ and $(r, c+2)$ cannot contain an i. This implies that f_i acts on $\mathcal{V}_b(T)$ by changing the i in $\mathsf{A}_{\mathcal{V}_b(T)}(r, c+1)$ to an $i+1$. We now consider $\mathcal{V}_b(f_i(T))$. We have that a' is the same as a and k' is equal to the $i+1$ in $(r, c+1)$. Note that $(a', \ell'] \cap \mathsf{L}_T(r, c) = \{i+1\}$. This implies that $\mathcal{V}_b$ acts on $f_i(T)$ by removing $i+1$ from $\mathsf{L}_{f_i(T)}(r, c)$ and a from $\mathsf{A}_{f_i(T)}(r, c)$, replacing the $i+1$ in $(r, c+1)$ with $\{i+1, a\}$, and appending an $i+1$ to $\mathsf{A}_{f_i(T)}(r, c+1)$. We see that $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$.

- Assume that $(r+1, c)$ does not contain an $i+1$ and $(r, c+1)$ does not contain an i. We break into subcases based on where the rightmost unpaired i sits inside (r, c).

  Assume that the rightmost unpaired i is in the hook entry of (r, c) for the remainder of this paragraph. Note that this implies that $a > i$ and the rightmost unpaired i in $\mathcal{V}_b(T)$ is still the hook entry of (r, c). We see that $\mathcal{V}_b$ does not insert an $i+1$ into $(r+1, c)$ nor an i into $(r, c+1)$. This implies that f_i acts on T and $\mathcal{V}_b(T)$ in the same way by changing the hook entry of (r, c) into an $i+1$. Next, we note that (r', c'), a', k', and $(a', \ell'] \cap \mathsf{L}_{f_i(T)}(r', c')$ are the same as (r, c), a, k, and $(a, \ell] \cap \mathsf{L}_T(r, c)$, respectively. Thus, $\mathcal{V}_b$ acts on T and $f_i(T)$ in the same manner without affecting the hook entry of (r, c). Therefore, we have that the actions of f_i and $\mathcal{V}_b$ on T are independent and $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$.

  Assume that the rightmost unpaired i is in the leg of (r, c) for the remainder of this paragraph. This implies that $a \neq i$. First, we assume that $a > i$ or $\tilde{r} < r$. Under this extra assumption, we observe that the action of $\mathcal{V}_b$ does not change the position of the rightmost unpaired i. Also, $\mathcal{V}_b$ does not insert an $i+1$ into $(r+1, c)$ nor an i into $(r, c+1)$. We see that f_i acts on T and $\mathcal{V}_b(T)$ in the same way by changing the i in the leg of (r, c) into an $i+1$. Next, we note that (r', c'), a', and k' are the same as (r, c), a, and k, respectively. If $a > i$, we have that $a \geq i+1$, implying that $(a', \ell'] \cap \mathsf{L}_{f_i(T)}(r', c') = (a, \ell] \cap \mathsf{L}_T(r, c)$. Thus, either $(a', \ell'] \cap \mathsf{L}_{f_i(T)}(r', c') = (a, \ell] \cap \mathsf{L}_T(r, c)$ or $\tilde{r} < r$. This implies that $\mathcal{V}_b$ acts on T and $f_i(T)$ in the same manner and does not affect the i or $i+1$ in the leg of (r, c). Therefore, we have that the actions of f_i and $\mathcal{V}_b$ on T are independent and $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$. Next, assume that $\tilde{r} = r$ and $a < i$. This implies that $(a, \ell] \cap \mathsf{L}_T(r, c) \neq \emptyset$ as $i \in (a, \ell] \cap \mathsf{L}_T(r, c)$. We have that $\mathcal{V}_b$ acts on T by removing $(a, \ell] \cap \mathsf{L}_T(r, c)$ from $\mathsf{L}_T(r, c)$ and a from $\mathsf{A}_T(r, c)$, replacing k with $((a, l] \cap \mathsf{L}_T(r, c)) \cup \{a\}$ in $(r, c+1)$, and appending k to $\mathsf{A}_T(r, c+1)$. By assumption, there was no i in $(r, c+1)$ to begin with. Thus, we have that the rightmost unpaired i of $\mathcal{V}_b(T)$ is the i in $(r, c+1)$ that replaced k. Since $k \geq i+1$ or k is the empty character, we have that the cell $(r+1, c+1)$

does not contain an $i+1$ and the cell $(r, c+2)$ does not contain an i. Hence, f_i acts on $\mathcal{V}_b(T)$ by replacing the i in $\mathsf{L}_{\mathcal{V}_b(T)}(r, c+1)$ with an $i+1$. We now consider $\mathcal{V}_b(f_i(T))$. We have that f_i acts on T by changing the i in $\mathsf{L}_T(r, c)$ to an $i+1$. We see that a' and k' are the same as a and k, respectively. Since $i > a$, we have that $i+1 > a$ or in other words $i+1 \in (a', \ell'] \cap \mathsf{L}_T(r, c)$. This implies that $(a', \ell'] \cap \mathsf{L}_{f_i(T)}(r', c') = (((a', \ell'] \cap \mathsf{L}_T(r, c)) \cup \{i+1\}) - \{i\}$. We have $\mathcal{V}_b$ acts on $f_i(T)$ by removing $(a', \ell'] \cap \mathsf{L}_{f_i(T)}(r, c)$ from $\mathsf{L}_{f_i(T)}(r, c)$ and a from $\mathsf{A}_{f_i(T)}(r, c)$, replacing k with $(a', \ell'] \cap \mathsf{L}_{f_i(T)}(r, c)$ in $(r, c+1)$, and appending k to $\mathsf{A}_{f_i(T)}(r, c+1)$. We see that $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$.

Assume that the rightmost unpaired i is in $\mathsf{A}_T(r, c)$ and $\tilde{r} < r$ or $(a, \ell] \cap \mathsf{L}_T(r, c) = \emptyset$ for this entire paragraph. Under this assumption, f_i acts on T by changing the rightmost i in the arm of (r, c) to an $i+1$. Also, $\mathcal{V}_b$ acts on T by removing a from $\mathsf{A}_T(r, c)$, replacing k in $(\tilde{r}, c+1)$ with a, and appending k to $\mathsf{A}_T(\tilde{r}, c+1)$. First, we make the additional assumption that $i < a$. Since we assume the rightmost unpaired i is in the arm of (r, c) and $i < a$, we have the rightmost unpaired i in $\mathcal{V}_b(T)$ is in the same position as in T. Note that the cell $(r+1, c)$ still does not contain an $i+1$ and the cell $(r, c+1)$ still does not contain an i. Thus, we have that f_i acts on $\mathcal{V}_b(T)$ by changing the rightmost i in $\mathsf{A}_{\mathcal{V}_b}(r, c)$ into an $i+1$. We now consider $\mathcal{V}_b(f_i(T))$. We see that a' and k' are the same as a and k, respectively. This implies that $\mathcal{V}_b$ acts on $f_i(T)$ by removing a from (r, c), replacing k with a in $(\tilde{r}, c)$, and appending k to $\mathsf{A}_{f_i(T)}(\tilde{r}, c+1)$. We see that $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$. Next, we make the assumption that $a = i$ and column $c+1$ does not contain both an i and an $i+1$. We have that the rightmost unpaired i in $\mathcal{V}_b(T)$ is precisely the i that replaced k in $(\tilde{r}, c+1)$. We also have that $k \geq i+1$ or k is the empty character, implying that the cell $(\tilde{r}+1, c+1)$ does not contain an $i+1$ and the cell $(\tilde{r}, c+2)$ does not contain an i. This implies that f_i acts on $\mathcal{V}_b(T)$ by changing the i in $\mathsf{L}^+_{\mathcal{V}_b(T)}(\tilde{r}, c+1)$ to an $i+1$. We now consider $\mathcal{V}_b(f_i(T))$. We see that a' is the $i+1$ in (r, c) created by applying f_i and k' is the same as k. Thus, $\mathcal{V}_b$ acts on $f_i(T)$ by removing the $i+1$ from (r, c), replacing k with an $i+1$ in $(\tilde{r}, c)$, and appending k to $\mathsf{A}_{f_i(T)}(\tilde{r}, c+1)$. We see that $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$. Next, we assume that $a = i$ and column $c+1$ contains both an i and an $i+1$ in the same cell. Note that this implies that $k = i$. Since a corresponded to the rightmost unpaired i in T and the only cell in column $c+1$ that contains an $i+1$ or an i is $(\tilde{r}, c+1)$, we have that the rightmost unpaired i in $\mathcal{V}_b(T)$ corresponds to the i appended to $\mathsf{A}_T(\tilde{r}, c+1)$. Since $(\tilde{r}, c+1)$ contains an $i+1$ in $\mathcal{V}_b(T)$, we have that $(\tilde{r}+1, c+1)$ cannot contain an $i+1$ and $(\tilde{r}, c+2)$ cannot contain an i. Thus, f_i acts on $\mathcal{V}_b(T)$ by changing the i in $\mathsf{A}_{\mathcal{V}_b(T)}(\tilde{r}, c+1)$ to an $i+1$. We now consider $\mathcal{V}_b(f_i(T))$. We see that a' is the $i+1$ in (r, c) obtained after applying f_i and k' is the $i+1$

in cell $(\tilde{r}, c+1)$. This implies that $\mathcal{V}_b$ acts on $f_i(T)$ by removing the $i+1$ from (r,c), replacing k' with an $i+1$ in $(\tilde{r}, c+1)$, and appending k' to $\mathsf{A}_{f_i(T)}(\tilde{r}, c+1)$. We see that $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$. Finally, we make the assumption that $a = i$ and column $c+1$ contains both an i and an $i+1$ but in different cells. We once again have that $k = i$, but now we have that $(\tilde{r}+1, c+1)$ contains an $i+1$. We have that the rightmost unpaired i in $\mathcal{V}_b(T)$ is the i that was appended to $\mathsf{A}_T(\tilde{r}, c+1)$. Since $(\tilde{r}+1, c+1)$ contains an $i+1$, we have that f_i acts on $\mathcal{V}_b(T)$ by removing the i from $\mathsf{A}_{\mathcal{V}_b(T)}(\tilde{r}, c+1)$ and appending an $i+1$ to $\mathsf{A}_{\mathcal{V}_b(T)}(\tilde{r}+1, c+1)$. We now consider $\mathcal{V}_b(f_i(T))$. We see that a' is the $i+1$ in (r,c) obtained after applying f_i and k' the $i+1$ in cell $(\tilde{r}+1, c+1)$. This implies that $\mathcal{V}_b$ acts on $f_i(T)$ by removing the $i+1$ from (r,c), replacing k' with an $i+1$ in $(\tilde{r}+1, c+1)$, and appending k' to $\mathsf{A}_{f_i(T)}(\tilde{r}+1, c+1)$. We see that $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$.

Assume that the rightmost unpaired i is in the arm of (r,c), $\tilde{r} = r$, and $(a, \ell] \cap \mathsf{L}_T(r,c) \neq \emptyset$ for this entire paragraph. First, we make the additional assumption that $i < a$. This gives us that $\mathcal{V}_b(T)$ is attained from T by removing $(a,\ell] \cap \mathsf{L}_T(r,c)$ from $\mathsf{L}_T(r,c)$ and a from $\mathsf{A}_T(r,c)$, replacing k in cell $(r, c+1)$ with $((a,\ell] \cap \mathsf{L}_T(r,c)) \cup \{a\}$, and appending k to $\mathsf{A}_T(r, c+1)$. Since $k, a > i$, we have that the rightmost unpaired i in $\mathcal{V}_b(T)$ remains the same as in T. We also have that the cell $(r+1, c)$ does not contain an $i+1$ and the cell $(r, c+1)$ does not contain an i. Thus, f_i acts on $\mathcal{V}_b(T)$ by changing the rightmost i in $\mathsf{A}_{\mathcal{V}_b(T)}(r,c)$ to an $i+1$. We now consider $\mathcal{V}_b(f_i(T))$. We have that f_i acts on T by changing the rightmost i in $\mathsf{A}_T(r,c)$ to an $i+1$. We see that a', k', and $(a', l'] \cap \mathsf{L}_{f_i(T)}(r', c')$ are the same as a, k, and $(a,\ell] \cap \mathsf{L}_T(r,c)$, respectively. This implies that $\mathcal{V}_b$ acts on $f_i(T)$ by removing $(a,\ell] \cap \mathsf{L}_T(r,c)$ from $\mathsf{L}_{f_i(T)}(r,c)$ and a from $\mathsf{A}_{f_i(T)}(r,c)$, replacing k in cell $(r, c+1)$ with $((a, l] \cap \mathsf{L}_T(r,c)) \cup \{a\}$, and appending k to $\mathsf{A}_{f_i(T)}(r, c+1)$. We see that $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$. Next, we assume that $a = i$ and (r,c) contains an $i+1$. Since $a = i$, the $i+1$ in (r,c) must be in its leg. Also as a is the rightmost unpaired i of T, we must have that (r,c) contains another i besides a. This gives us that $\mathcal{V}_b(T)$ is attained from T by removing $(a,\ell] \cap \mathsf{L}_T(r,c)$ from $\mathsf{L}_T(r,c)$ and a from $\mathsf{A}_T(r,c)$, replacing k in cell $(r, c+1)$ with $((a,\ell] \cap \mathsf{L}_T(r,c)) \cup \{a\}$, and appending k to $\mathsf{A}_T(r, c+1)$. Note that the i inserted into $(r, c+1)$ becomes i-paired, while an i in (r,c) becomes unpaired. This implies that the rightmost unpaired i in $\mathcal{V}_b(T)$ still sits in the cell (r,c). We see that the cell $(r+1, c)$ still does not contain an $i+1$; however, the cell $(r, c+1)$ now contains an i. This implies that f_i acts on $\mathcal{V}_b(T)$ by removing the i from the cell $(r, c+1)$ and appending an $i+1$ to $\mathsf{L}_{\mathcal{V}_b(T)}(r,c)$. We now consider $\mathcal{V}_b(f_i(T))$. We have that f_i acts on T by changing a into an $i+1$. We have that a' is the $i+1$ obtained from applying f_i and k' is the same as k. We see that $(a', \ell'] \cap \mathsf{L}_{f_i(T)}(r', c')$ is the same as

$(a, \ell] \cap \mathsf{L}_T(r, c)$ excluding the $i+1$. We have that $\mathcal{V}_b$ acts on $f_i(T)$ by removing $(a', \ell'] \cap \mathsf{L}_{f_i(T)}(r', c')$ from $\mathsf{L}_{f_i(T)}(r, c)$ and $i+1$ from $\mathsf{A}_{f_i(T)}(r, c)$, leaving the $i+1$ in $\mathsf{L}_{f_i(T)}(r, c)$, replacing k in $(r, c+1)$ with $((a', \ell'] \cap \mathsf{L}_{f_i(T)}(r', c')) \cup \{a'\}$, and appending k to $\mathsf{A}_{f_i(T)}(r, c+1)$. We see that $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$. Finally, we assume that $a = i$ and $i+1$ is not in the cell (r, c). This gives us that $\mathcal{V}_b(T)$ is attained from T by removing $(a, \ell] \cap \mathsf{L}_T(r, c)$ from $\mathsf{L}_T(r, c)$ and a from $\mathsf{A}_T(r, c)$, replacing k in cell $(r, c+1)$ with $((a, \ell] \cap \mathsf{L}_T(r, c)) \cup \{a\}$, and appending k to $\mathsf{A}_T(r, c+1)$. Since $k \geq j > i+1$ for all $j \in (a, \ell] \cap \mathsf{L}_T(r, c)$, we have that the i inserted into the cell $(r, c+1)$ is the rightmost unpaired i in $\mathcal{V}_b(T)$. Note that the cell $(r+1, c+1)$ does not contain an $i+1$ and the cell $(r, c+2)$ does not contain an i. Thus, f_i acts on $\mathcal{V}_b(T)$ by changing the i in $(r, c+1)$ to an $i+1$. We now consider $\mathcal{V}_b(f_i(T))$. We have that f_i acts on T by changing a into an $i+1$. We have that a' is the $i+1$ obtained from applying f_i and k' is the same as k. We see that $(a', \ell'] \cap \mathsf{L}_{f_i(T)}(r', c') = (a, \ell] \cap \mathsf{L}_T(r, c)$. We have that $\mathcal{V}_b$ acts on $f_i(T)$ by removing $(a, \ell] \cap \mathsf{L}_T(r, c)$ from $\mathsf{L}_{f_i(T)}(r, c)$ and $i+1$ from $\mathsf{A}_{f_i(T)}(r, c)$, replacing k in $(r, c+1)$ with $((a, \ell] \cap \mathsf{L}_T(r, c)) \cup \{a'\}$, and appending k to $\mathsf{A}_{f_i(T)}(r, c+1)$. We see that $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$.

Case 2 Assume that $\hat{r} < r$ and $\hat{c} = c$.
Note that $a > i$. By Lemma 3.10, we have that $R_i(T) = R_i(\mathcal{V}_b(T))$ unless $a = i+1$ and column $c+1$ contains both an i and an $i+1$. However, even in this special case, we see that the rightmost unpaired i of $\mathcal{V}_b(T)$ is in the same position as the rightmost unpaired i of T. We also see that $\mathcal{V}_b(T)$ does not change whether or not cell $(\hat{r}+1, c)$ contains an $i+1$ and whether or not cell $(\hat{r}, c+1)$ contains an i. Thus, f_i acts on the same i and in the same way for both T and $\mathcal{V}_b(T)$. Since $a > i$, we have that k' is the same as k. Note that the only way for f_i to affect the cell (r, c) in T is if $\hat{r} = r-1$ and (r, c) contains an $i+1$. However, even in this special case, we see that (r', c'), a', l', and $(a', \ell'] \cap \mathsf{L}_{f_i(T)}(r', c')$ are the same as (r, c), a, ℓ, and $(a, \ell] \cap \mathsf{L}_T(r, c)$. Thus, $\mathcal{V}_b$ acts on T and $f_i(T)$ in the same way. Therefore, we have that the actions of f_i and $\mathcal{V}_b$ on T are independent and $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$.

Case 3 Assume that $\hat{c} < c$. Let $\tilde{i}$ denote the rightmost unpaired i of T. From the proof of Lemma 3.10, we have that $\mathcal{V}_b$ does not change whether or not the i's to the right of $\tilde{i}$ in $R_i(T)$ are i-paired. Thus, the rightmost unpaired i in $R_i(T)$ and $R_i(\mathcal{V}_b(T))$ are in the same position. As $\mathcal{V}_b$ does not affect any column to the left of column c, we have that the rightmost unpaired i for $\mathcal{V}_b(T)$ is in the same position as the rightmost unpaired i for T. Note that $\mathcal{V}_b$ also does not affect whether or not cell $(\hat{r}+1, \hat{c})$ contains an $i+1$ and whether or not cell $(\hat{r}, \hat{c}+1)$ contains an i. Thus, f_i acts on the rightmost unpaired i in T and $\mathcal{V}_b(T)$ in exactly the same way. Next, we note that (r', c'), a', k', and $(a', \ell'] \cap \mathsf{L}_{f_i(T)}(r', c')$ are the same as (r, c), a, k, and $(a, \ell] \cap \mathsf{L}_T(r, c)$, respectively. Thus,

$\mathcal{V}_b$ acts on T and $f_i(T)$ in the same way. Therefore, we have that the actions of f_i and $\mathcal{V}_b$ on T are independent and $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$.

Case 4 Assume that $\hat{r} \leq r$ and $\hat{c} = c + 1$.

Under this assumption, we have that column $c + 1$ does not contain an $i + 1$ and $a \neq i + 1$, since the cells in column $c + 1$ do not contain any arms. We break into subcases.

- Assume that $k \neq i$. This implies that the rightmost unpaired i in $\mathcal{V}_b(T)$ is in the same position as the rightmost unpaired i in T. We see that $\mathcal{V}_b$ does not change whether or not cell $(\hat{r}+1, c+1)$ contains an $i+1$ and whether or not cell $(\hat{r}, c+2)$ contains an i. Thus, f_i acts on the rightmost unpaired i in T and $\mathcal{V}_b(T)$ in exactly the same way. We also observe that (r', c'), a', ℓ', k', and $(a', \ell'] \cap \mathsf{L}_{f_i(T)}(r', c')$ are the same as a, ℓ, k, and $(a, \ell] \cap \mathsf{L}_{f_i(T)}(r, c)$, respectively. Thus, $\mathcal{V}_b$ acts on T and $f_i(T)$ in the same way. Therefore, we have that the actions of f_i and $\mathcal{V}_b$ on T are independent and $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$.
- Assume that $k = i$. We see that the rightmost unpaired i in $\mathcal{V}_b(T)$ is the i that was appended to $\mathsf{A}_T(\hat{r}, c+1)$. Note that $\mathcal{V}_b$ does not change whether or not cell $(\hat{r} + 1, c + 1)$ contains an $i + 1$ and whether or not cell $(\hat{r}, c+2)$ contains an i. We first make the extra assumption that $(\hat{r}, c+2)$ in T contains an i. This implies that f_i acts on $\mathcal{V}_b(T)$ and T in the same way by removing the i from the hook entry of $(\hat{r}, c + 2)$ and appending an $i + 1$ to the leg of $(\hat{r}, c + 1)$. We also have that (r', c'), a', ℓ', k', and $(a', \ell'] \cap \mathsf{L}_{f_i(T)}(r', c')$ are equal to (r, c), a, ℓ, k, and $(a, \ell] \cap \mathsf{L}_{f_i(T)}(r, c)$, respectively. Thus, $\mathcal{V}_b$ acts on T and $f_i(T)$ in the same way. Therefore, we have that the actions of f_i and $\mathcal{V}_b$ on T are independent and $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$. We now assume that $(\hat{r}, c + 2)$ does not contain an i. This implies that f_i acts on $\mathcal{V}_b(T)$ by changing the i in $\mathsf{A}_{\mathcal{V}_b(T)}(\hat{r}, c + 1)$ to an $i + 1$ and acts on T similarly by changing the i in $\mathsf{L}_{\mathcal{V}_b(T)}(\hat{r}, c + 1)$ to an $i + 1$. Note that (r', c'), a', ℓ', and $(a', \ell'] \cap \mathsf{L}_{f_i(T)}(r', c')$ are equal to (r, c), a, ℓ, and $(a, \ell] \cap \mathsf{L}_{f_i(T)}(r, c)$, respectively, while k' is the $i + 1$ in $\mathsf{L}_{f_i(T)}(\hat{r}, c + 1)$. Thus, besides the value of the number that is bumped from the leg of $(\hat{r}, c + 1)$ to its arm, we have $\mathcal{V}_b$ acts on T and $f_i(T)$ in the same way. Looking at $f_i(\mathcal{V}_b(T))$ and $\mathcal{V}_b(f_i(T))$, we see that $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$.

Case 5 Assume that $\hat{r} > r$ and $\hat{c} = c$ or $c+1$. Under this assumption, we have that $\mathcal{V}_b$ does not change the cells $(\hat{r}, \hat{c})$, $(\hat{r}+1, \hat{c})$, and $(\hat{r}, \hat{c}+1)$. We also have that $R_i(T) = R_i(\mathcal{V}_b(T))$, implying that the rightmost unpaired i in $\mathcal{V}_b(T)$ is in the same position as the rightmost unpaired i in T. Thus, f_i acts on the rightmost unpaired i in $\mathcal{V}_b(T)$ and T in the same way. Note that $i + 1$ cannot be in column $\hat{c}$, implying that f_i can only make changes to the legs and hook entries of $(\hat{r}, \hat{c})$ and $(\hat{r}, \hat{c}+1)$. Since these changes only affect the legs and hook entries of cells outside of the possible cells that $\mathcal{V}_b$ can change, we have that $\mathcal{V}_b$ acts on T and $f_i(T)$ in the same way. Therefore, we have that the actions of f_i and $\mathcal{V}_b$ on T are independent and $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$.

Case 6 Assume that $\hat{c} \geq c+2$. Let $\tilde{i}$ denote the rightmost unpaired i of T. From the proof of Lemma 3.10, we have that $\mathcal{V}_b$ does not change whether or not the $i+1$'s to the left of $\tilde{i}$ are i-paired. Thus, the rightmost unpaired i in $R_i(T)$ and $R_i(\mathcal{V}_b(T))$ are in the same position. As $\mathcal{V}_b$ does not affect any column to the right of column $c+1$, we have that the rightmost unpaired i for $\mathcal{V}_b(T)$ is in the same position as the rightmost unpaired i for T. Note that $\mathcal{V}_b$ also does not affect whether or not cell $(\hat{r}+1, \hat{c})$ contains an $i+1$ and whether or not cell $(\hat{r}, \hat{c}+1)$ contains an i. Since the cells that f_i and $\mathcal{V}_b$ could change are different and the rightmost unpaired i does not change, we have that the actions of f_i and $\mathcal{V}_b$ on T are independent and $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$.

□

Theorem 3.14. *Let $T \in \mathsf{HVT}$.*

(1) *If $f_i(T) \neq 0$, we have $f_i(P(T)) = P(f_i(T))$ and $Q(T) = Q(f_i(T))$.*
(2) *If $e_i(T) \neq 0$, we have $e_i(P(T)) = P(e_i(T))$ and $Q(T) = Q(e_i(T))$.*

Proof. Part (2) follows from part (1), since e_i and f_i are partial inverse. We prove part (1) here.

Let $T \in \mathsf{HVT}$ with arm excess α, such that $f_i(T) \neq 0$ for some i. Then, $f_i(P(T)) = P(f_i(T))$ follows from Lemma 3.13, as $P(T)$ is obtained by successive applications of $\mathcal{V}$ on T and each application of $\mathcal{V}$ is a string of applications of $\mathcal{V}_b$.

Since crystal operators do not change arm excess, we may employ the notation in Definition 3.5 and denote the pair of insertion and recording tableaux produced at the jth step for $0 \leq j \leq \alpha$ of the uncrowding map $\mathcal{U}$ for T and $f_i(T)$ as $(P_j(T), Q_j(T))$ and $(P_j(f_i(T)), Q_j(f_i(T)))$, respectively. As crystal operators do not change the shape of T, we have $\mathsf{shape}(P_j(f_iT)) = \mathsf{shape}(f_i(P_j(T))) = \mathsf{shape}(P_j(T))$ for all $0 \leq j \leq \alpha$. Hence

$$\mathsf{shape}(P_{j+1}(T))/\mathsf{shape}(P_j(T)) = \mathsf{shape}(P_{j+1}(f_i(T)))/\mathsf{shape}(P_j(f_i(T))) \quad \text{for all } 0 \leq j \leq \alpha - 1. \tag{3.2}$$

Next, we show $Q_j(T) = Q_j(f_i(T))$ for all $0 \leq j \leq \alpha$ by induction. When $j = 0$, $Q_0(T) = Q_0(f_i(T))$, since $\mathsf{shape}(P_0(T)) = \mathsf{shape}(P_0(f_i(T))) = \mathsf{shape}(T)$.

Suppose $Q_j(T) = Q_j(f_i(T))$ for a given $j \geq 0$. It suffices to show that the cells

$$\mathsf{shape}(Q_{j+1}(T))/\mathsf{shape}(Q_j(T)) = \mathsf{shape}(P_{j+1}(T))/\mathsf{shape}(P_j(T)) \quad \text{and}$$
$$\mathsf{shape}(Q_{j+1}(f_i(T)))/\mathsf{shape}(Q_j(f_i(T))) = \mathsf{shape}(P_{j+1}(f_i(T)))/\mathsf{shape}(P_j(f_i(T)))$$

in $Q_{j+1}(T)$ and $Q_{j+1}(f_i(T))$ are at the same position with the same entry. By (3.2), the cells are in the same position, say in column $\tilde{c}$. By Definition 2.5, f_i does not move elements in the arm to a different column, so the columns in which we start the uncrowding insertion $\mathcal{V}$ on $P_j(T)$ and $P_j(f_i(T))$ are the same, say c, by Definition 3.5. Hence, the cells $\mathsf{shape}(Q_{j+1}(T))/\mathsf{shape}(Q_j(T))$ and $\mathsf{shape}(Q_{j+1}(f_i(T)))/\mathsf{shape}(Q_j(f_i(T)))$ are at the same position with entry $\tilde{c} - c$. The theorem follows. □

Hawkes and Scrimshaw [8, Theorem 4.6] proved that $\mathsf{HVT}^m(\lambda)$ is a Stembridge crystal by checking the Stembridge axioms. This also follows directly from our analysis above.

Corollary 3.15. *The crystal* $\mathsf{HVT}^m(\lambda)$ *of Definition* 2.5 *is a Stembridge crystal of type* A_{m-1}.

Proof. According to [13], $\mathsf{SVT}^m(\mu)$ is a Stembridge crystal of type A_{m-1}. By Theorem 3.14, the map

$$\mathcal{U}\colon \mathsf{HVT}^m(\lambda) \to \bigsqcup_{\mu\supseteq\lambda} \mathsf{SVT}^m(\mu) \times \hat{\mathcal{F}}(\mu/\lambda),$$

is a strict crystal morphism (see, for example, [2, Chapter 2]). The statement follows. $\square$

3.4. Uncrowding Map on Multiset-Valued Tableaux

The uncrowding map on hook-valued tableaux described above turns out to be a generalization of the uncrowding map on multiset-valued tableaux by Hawkes and Scrimshaw [8, Section 3.2]. We will prove that this is indeed the case in this section. Let us recall the definition of the uncrowding map in [8, Section 3.2].

Definition 3.16. Let $T \in \mathsf{MVT}(\lambda)$. The uncrowding map

$$\Upsilon : \mathsf{MVT}(\lambda) \to \bigsqcup_{\mu\supseteq\lambda} \mathsf{SSYT}(\mu) \times \hat{\mathcal{F}}(\mu/\lambda)$$

sends T to a pair of tableaux using the following algorithm:

(1) Set $U_{\lambda_1+1} = \emptyset$ and F_{λ_1+1} be the unique column-flagged increasing tableau of shape $\emptyset/\emptyset$.

(2) Let $1 \le k \le \lambda_1$ and assume that the pair (U_{k+1}, F_{k+1}) is defined. The pair (U_k, F_k) is defined recursively from (U_{k+1}, F_{k+1}) using the following two steps:

(a) Define U_k as the RSK row insertion tableau from the word

$$R(C_k)R(C_{k+1})\cdots R(C_{\lambda_1}),$$

where C_j is the jth column of T for every $1 \le j \le \lambda_1$. In other words, if we denote by $T_{\ge k}$ the tableau formed by the columns weakly to the right of the kth column of T, U_k is obtained by performing the RSK row insertion using the column reading word of $T_{\ge k}$.

(b) Form the tableau F_k of shape $\mathsf{shape}(U_k)/\mathsf{shape}(T_{\ge k})$ as follows. Shift F_{k+1} by one column to the right and fill the boxes in the same positions into F_k; for every unfilled box in the shape $\mathsf{shape}(U_k)/\mathsf{shape}(U_{k+1})$, label each box in column i with entry $i-1$.

Define $\Upsilon(T) = (U, F) := (U_1, F_1)$.

Example 3.17. Let T be the multiset-valued tableau

$$T = \begin{array}{lll} 45 & & \\ 233 & 345 & \\ 1 & 11 & 4 \end{array}.$$

Then, we obtain the following pairs of tableaux for the uncrowding map Υ:

$(U_4, F_4) = (\emptyset, \emptyset)$

$$(U_3, F_3) = \left(\begin{array}{l} 4 \end{array}, \begin{array}{l} \blacksquare \end{array} \right)$$

$$(U_2, F_2) = \left(\begin{array}{llll} 3 & 5 & & \\ 1 & 1 & 4 & 4 \end{array}, \begin{array}{llll} \blacksquare & 1 & & \\ \blacksquare & \blacksquare & 2 & 3 \end{array} \right)$$

$$(U_1, F_1) = \left(\begin{array}{llllll} 4 & 5 & & & & \\ 2 & 3 & 3 & 5 & & \\ 1 & 1 & 1 & 3 & 4 & 4 \end{array}, \begin{array}{llllll} \blacksquare & 1 & & & & \\ \blacksquare & \blacksquare & 1 & 3 & & \\ \blacksquare & \blacksquare & \blacksquare & 2 & 3 & 5 \end{array} \right) = (U, F) = \Upsilon(T).$$

Proposition 3.18. *Let $T \in \mathsf{MVT}(\lambda)$. Then, $\mathcal{U}(T) = \Upsilon(T)$. In other words, the uncrowding map as defined in Definition 3.5 is equivalent to the uncrowding map of Definition 3.16 in [8, Section 3.2].*

Proof. Recall from Definition 3.5 that the pair of uncrowding and recording tableaux for $\mathcal{U}(T)$ is denoted by $(P(T), Q(T)) = \mathcal{U}(T)$. Similarly, let us denote $(U(T), F(T)) := \Upsilon(T)$.

Assume that $S \in \mathsf{MVT}(\lambda)$ is highest weight, that is, $e_i(S) = 0$ for $i \geq 1$. By [8, Proposition 3.10], row i of S only contains the letter i. Thus, its weight is some partition $\mu = (\mu_1, \mu_2, \ldots, \mu_\ell)$ if $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_\ell)$. By Proposition 3.12 and Theorem 3.14, $P(S) \in \mathsf{SSYT}$ is highest weight. As weights of tableaux are preserved under uncrowding, the weight of $P(S)$ is equal to μ. By a similar argument using [8, Theorem 3.17], $U(S) \in \mathsf{SSYT}$ is also highest weight with weight μ. Since highest weight semistandard Young tableaux are uniquely determined by their weights, we have $P(S) = U(S)$.

Recall that as long as $f_i T \neq 0$ for $T \in \mathsf{MVT}(\lambda)$, we have $U(f_i T) = f_i U(T)$ by [8, Theorem 3.17] and $P(f_i T) = f_i P(T)$ by Theorem 3.14. Now, let $T \in \mathsf{MVT}(\lambda)$ be arbitrary. Then, $T = f_{i_1} \cdots f_{i_k}(S)$ for some sequence of $i_1, \ldots, i_k$ and S highest weight. Hence

$$\begin{aligned} P(T) &= P(f_{i_1} \cdots f_{i_k} S) = f_{i_1} \cdots f_{i_k} P(S) = f_{i_1} \cdots f_{i_k} \\ U(S) &= U(f_{i_1} \cdots f_{i_k} S) = U(T). \end{aligned}$$

It remains to show that $Q(T) = F(T)$ for all $T \in \mathsf{MVT}(\lambda)$. To do this, we show that the newly created boxes of the uncrowding map up to a specified column in Definition 3.16 are in the same positions as those for the uncrowding insertion in Definition 3.5. For every $Y \in \mathsf{MVT}(\mu)$ and for every $1 \leq j \leq \mu_1$, denote by $Y_{\geq j}$ the tableau formed by the rightmost j columns of Y; here, $Y_{\geq \mu_1+1}$ is the empty tableau.

Let $T \in \mathsf{MVT}(\lambda)$ be arbitrary. For $1 \leq k \leq \lambda_1 + 1$, let $P^{(k)}$ be the tableau obtained by performing the uncrowding map $\mathcal{U}$ on T on the columns from right to left up to and including the kth column of T; here, $P^{(\lambda_1+1)} = T$. In other words, $P^{(k)} = \mathcal{V}^{\alpha_k}(T)$ as in Definition 3.4, where α_k is the arm excess of $T_{\geq k}$. As the entries to the left of column k of T are untouched by the uncrowding insertion in Definition 3.4, for every $1 \leq k \leq \lambda_1 + 1$, we have $(P^{(k)})_{\geq k} = P(T_{\geq k}) = U(T_{\geq k})$. It follows that for every $1 \leq k \leq \lambda_1$, up to horizontal shifts, the newly formed boxes in $\mathsf{shape}(P^{(k)})/\mathsf{shape}(P^{(k+1)}) = \mathsf{shape}[(P^{(k)})_{\geq k+1}]/\mathsf{shape}[(P^{(k+1)})_{\geq k+1}]$ and $\mathsf{shape}([U(T_{\geq k})]_{\geq k+1})/\mathsf{shape}([U(T_{\geq k+1})]_{\geq k+1})$ are in the same positions. Since the entries in these boxes both record the difference in column indices relative to the kth column for each $1 \leq k \leq \lambda_1$ and since the recording tableaux for both maps are formed from the union of these boxes, we conclude that $Q(T) = F(T)$, completing the proof. $\square$

3.5. Crowding Map

In this section, we give a description of the "inverse" of the uncrowding map.

We begin by introducing some notation. Let $F \in \hat{\mathcal{F}}$ with e entries. For each cell (r, c) in F with entry $F(r, c)$, define the corresponding destination column to be $d(r, c) = c - F(r, c)$. Define the crowding order on F by ordering all the cells in F with a filling, first determined by their destination column (smallest to largest) and then by column index (largest to smallest). Denote the order by $(r_1, c_1), (r_2, c_2), \ldots, (r_e, c_e)$. Set $\alpha(F) = (\alpha_1, \alpha_2, \ldots, \alpha_e)$, where $\alpha_i = F(r_i, c_i)$. Let the arm excess for a column of a hook-valued tableau be the sum of arm excesses of all its cells.

Definition 3.19. Let $h \in \mathsf{HVT}$ and let (r, c) be a cell in h with $c > 1$ and with at most one element in $\mathsf{A}_h(r, c)$. If $\mathsf{A}_h(r, c)$ is empty, we also require that the cell (r, c) is a corner cell in h. Then, we define the crowding bumping $\mathcal{C}_b$ on the pair $[h, (r, c)]$ by the following algorithm:

(1) If $\mathsf{A}_h(r, c)$ is nonempty, set m to be the only element in $\mathsf{A}_h(r, c)$ and $b = \max\{x \in \mathsf{L}_h^+(r, c) \mid x \leq m\}$. Otherwise, set $m = \mathsf{H}_h(r, c)$ and $b = \max(\mathsf{L}_h^+(r, c))$.

(2) Find the largest r', such that $\mathsf{H}_h(r', c-1) \leq b$. If $r' = r$, set $q = \mathsf{H}_h(r, c)$. Otherwise, set $q = b$. In either case, append q to $\mathsf{A}_h(r', c-1)$.

(3) (a) If r' from Step 2 equals r, perform either of the following:
  - (i) If $\mathsf{A}_h(r, c)$ is nonempty, move the set $\{x \in \mathsf{L}_h(r, c) \mid q < x \leq m\}$ from $\mathsf{L}_h(r, c)$ to $\mathsf{L}_h(r', c-1)$ and keep it strictly increasing. Remove m from $\mathsf{A}_h(r, c)$ and set $H_h(r, c) = m$.
  - (ii) Otherwise, $\mathsf{A}_h(r, c)$ is empty, so move $\mathsf{L}_h(r, c)$ into $\mathsf{L}_h(r', c-1)$ and keep it to be strictly increasing. Remove cell (r, c) from h.

(b) Otherwise, $r' \neq r$ and perform either of the following:
  - (i) Suppose that $\mathsf{A}_h(r, c)$ is nonempty. Replace q in $\mathsf{L}_h^+(r, c)$ with m. Remove m from $\mathsf{A}_h(r, c)$.
  - (ii) If instead $\mathsf{A}_h(r, c)$ is empty, then remove cell (r, c) from h.

Denote the resulting (not necessarily semistandard) hook-valued tableau by h'. We write $\mathcal{C}_b([h, (r, c)]) = [h', (r', c-1)]$. We also define the projections p_1

FIGURE 3. When $r' = r$. Left: (i) $\mathsf{A}_h(r,c) \neq \emptyset$. Right: (ii) $\mathsf{A}_h(r,c) = \emptyset$

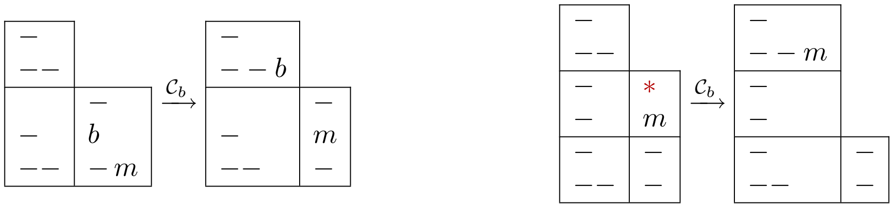


FIGURE 4. When $r' \neq r$. Left: $\mathsf{A}_h(r,c) \neq \emptyset$. Right: $\mathsf{A}_h(r,c) = \emptyset$

and p_2 by $p_1 \circ \mathcal{C}_b([h,(r,c)]) = h'$ and $p_2 \circ \mathcal{C}_b([h,(r,c)]) = (r', c-1)$. See Figs. 3 and 4 for illustration.

Example 3.20. We compute $\mathcal{C}_b$ in two examples

$$T = \begin{array}{|c|c|}\hline 5 & \\ \hline \begin{matrix} \\ \\ \\ 1\,1\end{matrix} & \begin{matrix}5\\4\\3\\2\,4\end{matrix} \\ \hline\end{array}, \quad \mathcal{C}_b([T,(1,2)]) = [\begin{array}{|c|c|}\hline 5 & \\ \hline \begin{matrix}4\\3\\1\,1\,2\end{matrix} & \begin{matrix}5\\4\end{matrix} \\ \hline\end{array}, (1,1)] = [\,T', (1,1)\,].$$

$$S = \begin{array}{|c|c|}\hline 3 & \\ \hline 2 & 3 \\ \hline 1 & 2 \\ \hline\end{array}, \quad \mathcal{C}_b([S,(1,2)]) = [\begin{array}{|c|}\hline 33 \\ \hline \begin{matrix}2\\1\end{matrix} \\ \hline\end{array}, (2,1)] = [\,S', (2,1)\,].$$

Remark 3.21. In Definition 3.19,

- if $r' = r$, then h' is always semistandard and has the same weight as h;
- if $r' \neq r$ and $\mathsf{A}_h(r,c)$ is empty, then h' might have fewer letters than h. In Example 3.20, S contains 5 letters, while S' only contains 4. This happens precisely when $\mathsf{L}_h(r,c)$ is nonempty.

In principle, the arm in cell $(r', c-1)$ could be greater than the q that is to be inserted. However, we only consider the cases as defined in the order described by the next paragraph. We refer to Proposition 3.27 which states that all tableaux we deal with in this section are indeed semistandard hook-valued tableaux.

Let $(S, F) \in \mathsf{SVT}(\mu) \times \hat{\mathcal{F}}(\mu/\lambda)$ with crowding order $(r_1, c_1), (r_2, c_2), \dots, (r_e, c_e)$ and $\alpha(F) = (\alpha_1, \alpha_2, \dots, \alpha_e)$. For all $0 \le j \le e - 1$ and for all $0 \le s \le \alpha_{j+1}$, define $T_j^{(s)}$ recursively by setting $T_0^{(0)} := S$ and

$$T_j^{(s)} := \begin{cases} p_1 \circ \mathcal{C}_b([T_j^{(s-1)}, (r_{j+1}, c_{j+1})]) & \text{when } s > 0, \\ T_{j-1}^{(\alpha_j)} & \text{when } s = 0 \text{ and } j > 0. \end{cases}$$

Additionally, define $T_e^{(0)} := T_{e-1}^{(\alpha_e)}$.

Thus, we obtain the following sequence:

$$S = T_0^{(0)} \xrightarrow[(r_1,c_1)]{p_1 \circ \mathcal{C}_b^{\alpha_1}} T_1^{(0)} \xrightarrow[(r_2,c_2)]{p_1 \circ \mathcal{C}_b^{\alpha_2}} T_2^{(0)} \xrightarrow[(r_3,c_3)]{p_1 \circ \mathcal{C}_b^{\alpha_3}} \dots \xrightarrow[(r_e,c_e)]{p_1 \circ \mathcal{C}_b^{\alpha_e}} T_e^{(0)}.$$

Remark 3.22. The tableaux $T_j^{(s)}$ are well defined. We check the conditions in Definition 3.19. Let $h = T_j^{(s)}$ for some $0 \le j \le e-1$ and for some $0 \le s < \alpha_{j+1}$, with cell (r, c).

- Since $F \in \hat{\mathcal{F}}$, we always have $c > 1$.
- The case that $\mathsf{A}_h(r, c)$ is empty can only occur in $T_{j-1}^{(0)}$ for some $j > 0$. In this case, $(r, c) = (r_j, c_j)$, which is a corner cell.
- Consider the α_j steps in $T_{j-1}^{(0)} \xrightarrow[(r_j,c_j)]{p_1 \circ \mathcal{C}_b^{\alpha_j}} T_j^{(0)}$. We first delete cell (r_j, c_j), which has no arm. Then, at every step after that, we move leftward one column at a time. Before we reach column $d(r_j, c_j)$, there is exactly one column with arm excess being 1 and the rest has zero arm excess among columns to the right of $d(r_j, c_j)$, since recall that the cells (r_j, c_j) are ordered from smallest to largest destination column. Once we reach column $d(r_j, c_j)$, the cell there may contain more than one arm element, but we then go to (r_{j+1}, c_{j+1}), which is a corner cell instead. Thus, there is at most one element in $\mathsf{A}_h(r, c)$.

Definition 3.23. With the same notation as above, define the insertion path of $T_{j-1}^{(0)} \to T_j^{(0)}$ for $1 \le j \le e$ to be

$$\mathsf{path}_j := \left((r_j^{(0)}, c_j^{(0)}), (r_j^{(1)}, c_j^{(1)}), \dots, (r_j^{(\alpha_j)}, c_j^{(\alpha_j)}) \right),$$

where $(r_j^{(s)}, c_j^{(s)}) := p_2 \circ \mathcal{C}_b^s([T_{j-1}^{(0)}, (r_j, c_j)])$ for $0 \le s \le \alpha_j$.

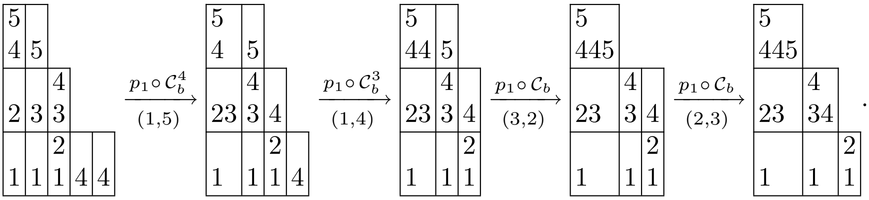
Example 3.24. Consider the following pair of tableaux $(S, F) \in \mathsf{HVT}((5, 3, 2)) \times \hat{\mathcal{F}}((5, 3, 2)/((3, 2, 1)))$:

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

The crowding order is $(1,5),(1,4),(3,2),(2,3)$. The insertion path and destination column for each of them are

$$\begin{aligned}\mathsf{path}_1 &= ((1,5),(1,4),(2,3),(2,2),(2,1)),\ d(1,5)=1,\\ \mathsf{path}_2 &= ((1,4),(2,3),(2,2),(3,1)),\ d(1,4)=1,\\ \mathsf{path}_3 &= ((3,2),(3,1)),\ d(3,2)=1,\\ \mathsf{path}_4 &= ((2,3),(2,2)),\ d(2,3)=2.\end{aligned}$$

We obtain the sequence from the algorithm



Lemma 3.25. *If $d(r_j,c_j)=d(r_{j+1},c_{j+1})$, then path_{j+1} is weakly above path_j.*

Proof. By the definition of crowding order, $d(r_j,c_j)=d(r_{j+1},c_{j+1})$ implies $c_j>c_{j+1}$. Set $z_j:=c_j-c_{j+1}$. Then, we have $c_j^{(s+z_j)}=c_j-z_j-s=c_{j+1}-s=c_{j+1}^{(s)}$ for $0\le s\le\alpha_{j+1}$. We need to show that $r_{j+1}^{(s)}\ge r_j^{(s+z_j)}$ for $0\le s\le\alpha_{j+1}$. Computing $T_{j-1}^{(s)}$ from $T_{j-1}^{(s-1)}$ for $1\le s\le\alpha_j$, we denote b and q in Step (1) and Step (2) of Definition 3.19 by $b_j^{(s)}$ and $q_j^{(s)}$.

Since (r_{j+1},c_{j+1}) is a corner cell in $T_{j-1}^{(z_j)}$, we have $r_{j+1}^{(0)}\ge r_j^{(z_j)}$. We prove that, for $1\le s\le\alpha_{j+1}$, we have that $q_{j+1}^{(s)}\ge q_j^{(s+z_j)}$, which implies $b_{j+1}^{(s)}\ge b_j^{(s+z_j)}$ and thus $r_{j+1}^{(s)}\ge r_j^{(s+z_j)}$.

We prove $q_{j+1}^{(s)}\ge q_j^{(s+z_j)}$ by induction on s. First, we check the case $k=1$. If $r_{j+1}^{(0)}>r_j^{(z_j)}$, then it is obvious that $q_{j+1}^{(1)}>q_j^{(z_j+1)}$. Otherwise if $r_{j+1}^{(0)}=r_j^{(z_j)}$, we consider the following cases. $q_j^{(z_j)}$ is the only element in $\mathsf{A}_{T_{j-1}^{(z_j)}}(r_{j+1},c_{j+1})$. Let $x=\mathsf{H}_{T_{j-1}^{(z_j)}}(r_{j+1},c_{j+1})$, $y=\max(\mathsf{L}_{T_{j-1}^{(z_j)}}(r_{j+1},c_{j+1}))$ and $y'=\max\{z\in\mathsf{L}^{+}_{T_{j-1}^{(z_j)}}(r_{j+1},c_{j+1})\mid z\le q_j^{(z_j)}\}$. See Fig. 5 for illustration.

Case (1): If $r_j^{(z_j+1)}=r_j^{(z_j)}$, then $q_j^{(z_j+1)}=x$. If $r_{j+1}^{(1)}=r_{j+1}^{(0)}$, then $q_{j+1}^{(1)}=q_j^{(z_j)}$. If $r_{j+1}^{(1)}\ne r_{j+1}^{(0)}$, then $q_{j+1}^{(1)}$ equals y when $y>y'$ and $q_j^{(z_j)}$ when $y=y'$. In both cases, $q_{j+1}^{(1)}\ge x=q_j^{(z_j+1)}$.

Case (2): If $r_j^{(z_j+1)}\ne r_j^{(z_j)}$, then $q_j^{(z_j+1)}=y'$. In this case, we have $\mathsf{H}_{T_{j-1}^{(z_j)}}(r_{j+1}+1,c_{j+1}-1)\le y'\le y$. Since $\mathsf{H}_{T_j^{(0)}}(r_{j+1}+1,c_{j+1}-1)$ is smaller or equal to y', we have that $r_{j+1}^{(1)}\ne r_{j+1}^{(0)}$. Therefore, $q_{j+1}^{(1)}$ equals y when $y>y'$ and $q_j^{(z_j)}$ when $y=y'$. In this case, $q_{j+1}^{(1)}\ge y'=q_j^{(z_j+1)}$.

$$\boxed{\begin{array}{ll} y & \\ - & \\ y' & \\ * & \\ x & q_j^{(z_j)} \end{array}} \qquad \boxed{\begin{array}{l} y \\ - \\ q_j^{(z_j)} \end{array}} \qquad \boxed{\begin{array}{l} y \\ - \\ q_j^{(z_j)} \\ * \\ x \end{array}}$$

FIGURE 5. Cell $(r_{j+1}^{(0)}, c_{j+1}^{(0)}) = (r_j^{(z_j)}, c_j^{(z_j)})$ in $T_{j-1}^{(z_j)}$ (left); in $T_j^{(0)}$, case(1) (middle), case(2) (right)

$$\boxed{\begin{array}{ll} y & \\ - & \\ y' & \\ * & \\ x & q_j^{(s+z_j)} \end{array}} \qquad \boxed{\begin{array}{ll} y & \\ - & \\ q_j^{(s+z_j)} & q_{j+1}^{(s)} \end{array}} \qquad \boxed{\begin{array}{ll} y & \\ - & \\ q_j^{(s+z_j)} & \\ * & \\ x & q_{j+1}^{(s)} \end{array}}$$

FIGURE 6. Cell $(r_{j+1}^{(s)}, c_{j+1}^{(s)}) = (r_j^{(s+z_j)}, c_j^{(s+z_j)})$ in $T_{j-1}^{(s+z_j)}$ (left); in $T_j^{(s)}$, case(1) (middle), case(2) (right)

Now, we have proved the base case $s = 1$. Next, suppose it holds for some $s \geq 1$ that $q_{j+1}^{(s)} \geq q_j^{(s+z_j)}$ and $r_{j+1}^{(s)} \geq r_j^{(s+z_j)}$. The statement is similar to the argument of the base case. If $r_{j+1}^{(s)} > r_j^{(z_j+s)}$, it is obvious that $q_{j+1}^{(s+1)} > q_j^{(s+1+z_j)}$ and thus $r_{j+1}^{(s+1)} \geq r_j^{(s+1+z_j)}$. If $r_{j+1}^{(s)} = r_j^{(z_j+s)}$, we discuss the following cases. $q_j^{(s+z_j)}$ is the only element in $\mathsf{A}_{T_{j-1}^{(s+z_j)}}(r_j^{(s+z_j)}, c_j^{(s+z_j)})$. Let $x = \mathsf{H}_{T_{j-1}^{(s+z_j)}}(r_j^{(s+z_j)}, c_j^{(s+z_j)})$, $y = \max(\mathsf{L}_{T_{j-1}^{(s+z_j)}}(r_j^{(s+z_j)}, c_j^{(s+z_j)}))$ and $y' = \max\{z \in \mathsf{L}^+_{T_{j-1}^{(s+z_j)}}(r_j^{(s+z_j)}, c_j^{(s+z_j)}) \mid z \leq q_j^{(s+z_j)}\}$. See Fig. 6 for illustration.

Case (1): If $r_j^{(s+1+z_j)} = r_j^{(s+z_j)}$, then $q_j^{(s+1+z_j)} = x$. If $r_{j+1}^{(s+1)} = r_{j+1}^{(s)}$, then $q_{j+1}^{(s+1)} = q_j^{(s+z_j)} \geq x$. If $r_{j+1}^{(s+1)} \neq r_{j+1}^{(s)}$, then $q_{j+1}^{(s+1)} = \max\{z \in \mathsf{L}^+_{T_j^{(s)}}(r_{j+1}^{(s)}, c_{j+1}^{(s)}) \mid z \leq q_{j+1}^{(s)}\} \geq q_j^{(s+z_j)} \geq x$. Therefore, in either case, we have $q_{j+1}^{(s+1)} \geq q_j^{(s+1+z_j)}$.
Case (2): If $r_j^{(s+1+z_j)} \neq r_j^{(s+z_j)}$, then $q_j^{(s+1+z_j)} = y'$. In this case, we have $\mathsf{H}_{T_{j-1}^{(s+z_j)}}(r_j^{(s+z_j)} + 1, c_j^{(s+z_j)} - 1) \leq y' \leq q_j^{(s+z_j)}$. Since $\mathsf{H}_{T_j^{(s)}}(r_{j+1}^{(s)} + 1, c_{j+1}^{(s)} - 1)$ is smaller or equal to $q_j^{(s+z_j)}$, we have that $r_{j+1}^{(s+1)} \neq r_{j+1}^{(s)}$. Therefore, $q_{j+1}^{(s+1)} = \max\{z \in \mathsf{L}^+_{T_j^{(s)}}(r_{j+1}^{(s)}, c_{j+1}^{(s)}) \mid z \leq q_{j+1}^{(s)}\}$. By induction, we have $q_j^{(s+z_j)} \leq q_{j+1}^{(s)}$; thus, $q_{j+1}^{(s+1)} \geq q_j^{(s+z_j)} \geq y' = q_j^{(s+1+z_j)}$. This completes the proof. □

Lemma 3.26. *With the notations as above, let* $0 \leq j \leq e-1$, $0 \leq s < \alpha_{j+1}$ *and* $\mathcal{C}_b([T_j^{(s)}, (r, c)]) = [T_j^{(s+1)}, (r', c-1)]$ *for some* r, c, r'. *Then, in* $T_j^{(s+1)}$, *column $c-1$ is the rightmost column with nonzero arm excess and $(r', c-1)$ is the topmost cell in column $c-1$ with nonzero arm excess.*

Proof. In any path_j, consider the arm excess of its columns. Those with column index c such that $d(r_j, c_j) < c < c_j$ started with arm excess 0, then changed to arm excess 1 when the insertion path passed through that column, and immediately decreased to 0.

Thus, the $q_j^{(s)}$ that is being moved to cell $(r', c-1)$ is always at the rightmost column containing nonzero arm excess. When $c - 1 > d(r_j, c_j)$, the arm excess of the column $c-1$ is exactly 1, $(r', c-1)$ is also the topmost cell containing an arm. For $c - 1 = d(r_j, c_j)$, the path path_j has reached its destination. At that point, any column to the right of $d(r_j, c_j)$ has 0 arm excess. It follows from Lemma 3.25 that the cell $(r_j^{(\alpha_j)}, c_j^{(\alpha_j)})$ is also the topmost cell containing an arm (Figs. 7, 8, 9, 10). $\square$

Proposition 3.27. *The tableau $T_j^{(s+1)}$ is a semistandard hook-valued tableau for all $0 \leq j \leq e-1$ and for all $0 \leq s < \alpha_{j+1}$.*

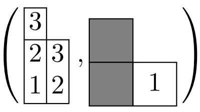
Proof. We only need to check that the q in Step 2 of Definition 3.19 is greater or equal to the hook entry and arm of the cell q is to be inserted into. When q is the only arm element, it is obvious that q is greater or equal to the hook entry.

The case when q is not the only arm element can only happen when we reach the destination column of the path. By the proof of Lemma 3.25, we have that for $q_{j+1}^{(s)} \geq q_j^{(s+z_j)}$ for $s \geq 1$ and for j, such that $d(r_j, c_j) = d(r_{j+1}, c_{j+1})$. Hence, the statement follows by setting $k = \alpha_{j+1}$. $\square$

Before we define the "inverse" of the uncrowding map $\mathcal{U} : \mathsf{HVT}(\lambda) \to \sqcup_{\mu \supseteq \lambda} \mathsf{SVT}(\mu) \times \hat{\mathcal{F}}(\mu/\lambda)$, we need to restrict our domain to a subset K_λ of $\sqcup_{\mu \supseteq \lambda} \mathsf{SVT}(\mu) \times \hat{\mathcal{F}}(\mu/\lambda)$, as the image of $\mathcal{U}$ is not all of $\sqcup_{\mu \supseteq \lambda} \mathsf{SVT}(\mu) \times \hat{\mathcal{F}}(\mu/\lambda)$. We define

$$
\begin{aligned}
\mathsf{K}_\lambda(\mu) :=& \{(S, F) \in \mathsf{SVT}(\mu) \times \hat{\mathcal{F}}(\mu/\lambda) \mid \mathsf{weight}(T_j^{(s)}) = \mathsf{weight}(S), \\
& \forall\, 0 \leq j \leq e-1, \forall\, 0 \leq s \leq \alpha_{j+1}\}, \\
\mathsf{K}_\lambda :=& \bigsqcup_{\mu \supseteq \lambda} \mathsf{K}_\lambda(\mu).
\end{aligned}
$$

Remark 3.28. From the perspective of the uncrowding map, the set-valued tableau S in Example 3.20 cannot be obtained from a shape $(1,1)$ hook-valued tableau via the uncrowding map as explained in Remark 3.21. We say the cell $(1,2)$ in S practices social distancing. In this case

 $\notin \mathsf{K}_{(1,1)}.$

The (S, F) in Example 3.24 is in $\mathsf{K}_{(3,2,1)}(5,3,2)$.

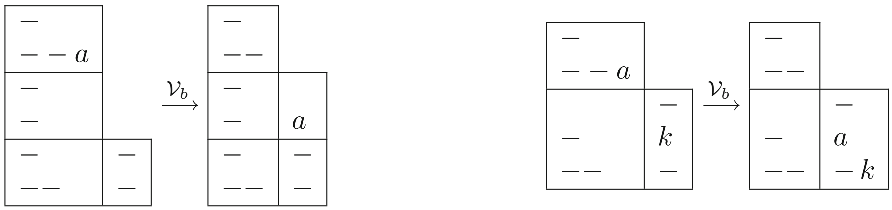


FIGURE 7. Left: case (1A): $(\tilde{r}, c+1)$ is not in h. Right: case (1B): $(\tilde{r}, c+1)$ is in h

Definition 3.29. We can now define the crowding map $\mathcal{C}$ for any partition λ as follows:

$$\mathcal{C}\colon \mathsf{K}_\lambda \longrightarrow \mathsf{HVT}(\lambda)$$
$$(S, F) \mapsto T_e^{(0)}.$$

Proposition 3.30. *The image of the uncrowding map* $\mathcal{U} : \mathsf{HVT}(\lambda) \to \sqcup_{\mu \supseteq \lambda} \mathsf{SVT}(\mu) \times \hat{\mathcal{F}}(\mu/\lambda)$ *is a subset of* K_λ*. Moreover, we have* $\mathcal{C} \circ \mathcal{U} = \mathbf{1}_{\mathsf{HVT}(\lambda)}$.

Proof. We show that if $\tilde{h} = \mathcal{V}_b(h)$, where $h \in \mathsf{HVT}$, $\mathcal{V}_b$ is as defined in Definition 3.2 and $\tilde{h}$ is obtained by moving some letter(s) from the cell (r, c) to $(\tilde{r}, c+1)$ (potentially adding a box), then $\mathcal{C}_b([\tilde{h}, (\tilde{r}, c+1)]) = [h', (r', c)]$ satisfies $[h', (r', c)] = [h, (r, c)]$.

We follow the notation used in Definitions 3.2 and 3.19. Thus $a = \max(\mathsf{A}_h(r, c))$. We have that $\mathsf{H}_h(\tilde{r}, c) \leq a$. If cell $(r+1, c)$ is in h, then $\mathsf{H}_h(r+1, c) > a$.
Case (1): $\tilde{r} \neq r$.
Case (1A): If cell $(\tilde{r}, c+1)$ is not in h, then h' is obtained by adding cell $(\tilde{r}, c+1)$ and moving a from $\mathsf{A}_h(r, c)$ to $\mathsf{H}_h(\tilde{r}, c+1)$. Under the action of $\mathcal{C}_b$, by Step (1), $b = a$ and $r' = r$. $\mathcal{C}_b$ appends a to $\mathsf{A}_{\tilde{h}}(r, c)$ and removes cell $(\tilde{r}, c+1)$, which recovers h.
Case (1B): If cell $(\tilde{r}, c+1)$ is in h, then $k \in \mathsf{L}_h^+(\tilde{r}, c+1)$ is the smallest number that is greater than or equal to a in column $c+1$. h' is obtained by removing a from $\mathsf{A}_h(r, c)$, replacing k with a, and attaching k to $\mathsf{A}_h(\tilde{r}, c+1)$. Under the action of $\mathcal{C}_b$, by Step (1), we can see that $m = k$, $b = a$, and $r' = r$. By Step (1), $q = b = a$, and a is appended to $\mathsf{A}_{\tilde{h}}(r, c)$ and $q = a$ in $\mathsf{L}_{\tilde{h}}(\tilde{r}, c+1)$ is replaced with $m = k$. In the end, m is removed from $\mathsf{A}_{\tilde{h}}(\tilde{r}, c+1)$. We recover h.
Case (2): $\tilde{r} = r$. Let $\ell = \max(\mathsf{L}_h^+(r, c))$.
Case (2A): If cell $(r, c+1)$ is not in h, $\mathcal{V}_b$ adds cell $(r, c+1)$, removes the part of $\mathsf{L}_h(r, c)$ that is greater than a to $\mathsf{L}_h(r, c+1)$, and moves a from $\mathsf{A}_h(r, c)$ to $\mathsf{H}_h(r, c+1)$. Under the action of $\mathcal{C}_b$, by Step 1, $m = a$ and $b = \ell$. Thus, $r' = r$. By Step 3(a)ii, we move $\mathsf{L}_{\tilde{h}}(r, c+1)$ into $\mathsf{L}_{\tilde{h}}(r, c)$ and we recover h.
Case (2B): If cell $(r, c+1)$ is in h, $\tilde{h}$ is obtained by moving the part of $\mathsf{L}_h(r, c)$ that is greater than a to $\mathsf{L}_h(r, c+1)$, moving a from $\mathsf{A}_h(r, c)$ to $\mathsf{H}_h(r, c+1)$, and appending k to $\mathsf{A}_h(r, c+1)$. Under the action of $\mathcal{C}_b$, by Step 1, $m = k$ and $b = \ell$. Then, $r' = r$ and $q = a$. By Step 3(a)i, we move the set $\{x \in \mathsf{L}_{\tilde{h}}(r, c) \mid$

FIGURE 8. Left: Case (1A): $(r, c+1)$ is not in h. Right: Case (1B): $(r, c+1)$ is in h

$a < x \leq k\}$ from $\mathsf{L}_{\tilde{h}}(r, c+1)$ into $\mathsf{L}_{\tilde{h}}(r, c)$, which is the set that was moved from cell (r, c) by $\mathcal{V}_b$. Removing k from $\mathsf{A}_{\tilde{h}}(r, c+1)$ and setting $\mathsf{H}_{\tilde{h}}(r, c+1) = k$, we recover h.

Now, we have proven $\mathcal{C}_b([\tilde{h}, (\tilde{r}, c+1)]) = [h', (r', c)] = [h, (r, c)]$. It follows that for any $(S, F) = \mathcal{U}(h)$, we have that $T_j^{(s)}$ is semistandard and has the same weight as S for all $0 \leq j \leq e-1$, for all $0 \leq s \leq \alpha_{j+1}$. Thus, $\mathsf{image}(\mathcal{U}) \subset \mathsf{K}_\lambda$ and $\mathcal{C} \circ \mathcal{U} = \mathbf{1}_{\mathsf{HVT}(\lambda)}$. □

Proposition 3.31. *K_λ is a subset of the image of $\mathcal{U} : \mathsf{HVT}(\lambda) \to \sqcup_{\mu \supseteq \lambda} \mathsf{SVT}(\mu) \times \hat{\mathcal{F}}(\mu/\lambda)$. Moreover, $\mathcal{U} \circ \mathcal{C} = \mathbf{1}_{\mathsf{K}_\lambda}$.*

Proof. Let $(S, F) \in \mathsf{K}_\lambda$, then for all $0 \leq j < e$ and for all $0 \leq s < \alpha_{j+1}$, $\mathcal{C}_b([T_j^{(s)}, (r, c)]) = [T_j^{(s+1)}, (r', c-1)]$ for some r, c, r'. We show that $\mathcal{V}_b(T_j^{(s+1)}) = T_j^{(s)}$ for all $0 \leq j < e$ and for all $0 \leq s < \alpha_{j+1}$. Following the notation in Definition 3.2, we first locate the rightmost column that contains nonzero arm excess, and then determine the topmost cell in row $\tilde{r}$ in that column with nonzero arm excess. We denote by a the largest arm element in that cell.

By Lemma 3.26, in $T_j^{(s+1)}$, column $c-1$ is the rightmost column with nonzero arm excess and $(r', c-1)$ is the topmost cell in column $c-1$ with nonzero arm excess.

Case (1): $r' = r$. In this case, either cell $(r+1, c-1)$ does not exist in $T_j^{(s)}$, or $\mathsf{H}_{T_j^{(s)}}(r+1, c-1) > b$.

Case (1A): $\mathsf{A}_{T_j^{(s)}}(r, c) = \emptyset$. $m = \mathsf{H}_{T_j^{(s)}}(r, c)$ and $b = \max(\mathsf{L}^+_{T_j^{(s)}}(r, c))$. Since $r' = r$, $q = m$, $T_j^{(s+1)}$ is obtained by appending m to $\mathsf{A}_{T_j^{(s)}}(r, c-1)$, moving $\mathsf{L}_{T_j^{(s)}}(r, c)$ into $\mathsf{L}_{T_j^{(s)}}(r, c-1)$, and removing cell (r, c) from $T_j^{(s)}$. Note that everything in $\mathsf{L}_{T_j^{(s)}}(r, c)$ is greater than m and everything in $\mathsf{L}_{T_j^{(s)}}(r, c-1)$ is smaller or equal to m.

For the $\mathcal{V}_b$ action, we have $a = m$ and b is the greatest letter in $\mathsf{L}_{T_j^{(s+1)}}(r, c-1)$. Since every letter in $T_j^{(s+1)}(r'', c)$ is smaller than m for $r'' < r$, we have $\tilde{r} = r$. $\mathcal{V}_b$ acts on $T_j^{(s+1)}$ by adding the cell (r, c), setting the hook entry to be m, and moving $(m, b] \cap \mathsf{L}_{T_j^{(s+1)}}(r, c-1)$ to $\mathsf{L}_{T_j^{(s+1)}}(r, c)$. Then, we recover $T_j^{(s)}$.

FIGURE 9. Left: Case (1A): $\mathsf{A}_{T_j^{(s)}}(r,c)=\emptyset$. Right: Case (1B): $\mathsf{A}_{T_j^{(s)}}(r,c)\neq\emptyset$

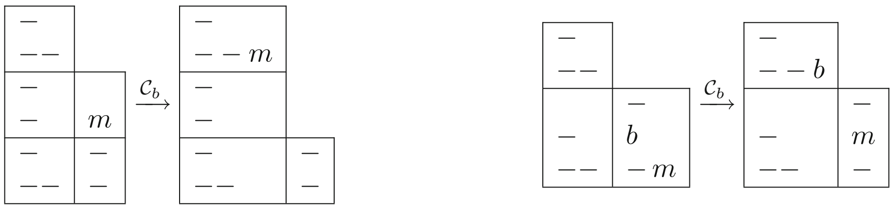


FIGURE 10. Left: case (2A): $\mathsf{A}_{T_j^{(s)}}(r,c)=\emptyset$. Right: case (2B): $\mathsf{A}_{T_j^{(s)}}(r,c)\neq\emptyset$

Case (1B): $\mathsf{A}_{T_j^{(s)}}(r,c)\neq\emptyset$. m is the only element in $\mathsf{A}_{T_j^{(s)}}(r,c)$, $q=\mathsf{H}_{T_j^{(s)}}(r,c)$ and $b=\max\{x\in \mathsf{L}^+_{T_j^{(s)}} \mid x\leq m\}$. $T_j^{(s+1)}$ is obtained by appending q to $\mathsf{A}_{T_j^{(s)}}(r,c-1)$, setting $\mathsf{H}_{T_j^{(s)}}(r,c)$ to be m, deleting $\mathsf{A}_{T_j^{(s)}}$, and moving $\{x\in \mathsf{L}_{T_j^{(s)}(r,c)} \mid q<x\leq m\}$ to $\mathsf{L}_{T_j^{(s)}}(r,c-1)$.

For the $\mathcal{V}_b$ action, $a=q$ and b is the greatest letter in $\mathsf{L}_{T_j^{(s+1)}}(r,c-1)$. Since every letter in $T_j^{(s+1)}(r'',c)$ is smaller than q for $r''<r$ and $m\geq q$, $\tilde{r}=r$. $\mathcal{V}_b$ acts on $T_j^{(s+1)}$ by setting $\mathsf{H}_{T_j^{(s+1)}}(r,c)=q$, $\mathsf{A}_{T_j^{(s+1)}}(r,c)=m$, and moving $(q,b]\cap \mathsf{L}_{T_j^{(s+1)}}(r,c-1)$ to $\mathsf{L}_{T_j^{(s+1)}}(r,c)$. We recover $T_j^{(s)}$.

Case (2): $r'\neq r$.

Case (2A): $\mathsf{A}_{T_j^{(s)}}(r,c)=\emptyset$. Note that in this case, $\mathcal{C}_b$ will move m somewhere else and remove the cell (r,c). Since $\mathsf{weight}(T_j^{(s+1)})=\mathsf{weight}(T_j^{(s)})$, we must have that $\mathsf{L}_{T_j^{(s)}}(r,c)=\emptyset$. So $b=q=m$. $T_j^{(s+1)}$ is obtained from $T_j^{(s)}$ by appending m to $\mathsf{A}_{T_j^{(s)}}(r',c-1)$ and removing the cell (r,c).

For the $\mathcal{V}_b$ action, $a=m$. Since every letter in $T_j^{(s+1)}(r'',c)$ is smaller than m for $r''<r$, a new cell (r,c) is added, $\tilde{r}=r$. $\mathcal{V}_b$ acts on $T_j^{(s+1)}$ by moving m to $\mathsf{H}_{T_j^{(s+1)}}(r,c)$. We recover $T_j^{(s)}$.

Case (2B): $\mathsf{A}_{T_j^{(s)}}(r,c) \neq \emptyset$. m is the only element in $\mathsf{A}_{T_j^{(s)}}(r,c)$, $q = b = \max\{x \in \mathsf{L}^+_{T_j^{(s)}}(r,c) \mid x \leq m\}$. $T_j^{(s+1)}$ is obtained by appending b to $\mathsf{A}_{T_j^{(s)}}(r', c-1)$, replacing b in $\mathsf{L}_{T_j^{(s)}}(r,c)$ with m, and removing m from $\mathsf{A}_{T_j^{(s)}}(r,c)$.

For the $\mathcal{V}_b$ action, $a = b$. Since every letter in $T_j^{(s+1)}(r'', c)$ is smaller than b for $r'' < r$, m is the smallest letter that is greater or equal to b in column c. Hence, $\tilde{r} = r$. $\mathcal{V}_b$ acts on $T_j^{(s+1)}$ by removing b from $\mathsf{A}_{T_j^{(s+1)}}(r', c-1)$, replacing m in $\mathsf{L}_{T_j^{(s+1)}}(r,c)$ with b, and attaching m to $\mathsf{A}_{T_j^{(s+1)}}(r,c)$. We recover $T_j^{(s)}$.

Therefore, we have $\mathcal{V}_b(T_j^{(s+1)}) = T_j^{(s)}$ for all $0 \leq j \leq e-1$, for all $0 \leq s < \alpha_j$, and $\mathcal{V}(T_{j+1}^{(0)}) = T_j^{(0)}$. It follows that we also recover the recording tableau F. Thus, $\mathcal{U}(T_e^{(0)}) = (S, F)$. □

Corollary 3.32. *The uncrowding map $\mathcal{U}$ is a bijection between* $\mathsf{HVT}(\lambda)$ *and* K_λ *with inverse $\mathcal{C}$.*

3.6. Alternative Uncrowding on Hook-Valued Tableaux

In Sect. 3.2, we defined an uncrowding map sending hook-valued tableaux to pairs of tableaux with one being set-valued and the other being column-flagged increasing. As hook-valued tableaux were introduced as a generalization of both set-valued tableaux and multiset-valued tableaux, it is natural to ask if there is an uncrowding map taking hook-valued tableaux to pairs of tableaux with one being multiset-valued. In this section, we provide such a map.

Definition 3.33. The multiset uncrowding bumping $\tilde{\mathcal{V}}_b \colon \mathsf{HVT} \to \mathsf{HVT}$ is defined by the following algorithm:

(1) Initialize T as the input.
(2) If the leg excess of T equals zero, return T.
(3) Find the topmost row that contains a cell with nonzero leg excess. Within this column, find the cell with the largest value in its leg. (This is the rightmost cell with nonzero leg excess in the specified row.) Denote the row index and column index of this cell by r and c, respectively. Denote the cell as (r,c), its largest leg entry by ℓ, and its rightmost arm entry by a.
(4) Look at the row above (r,c) (i.e. row $r+1$) and find the leftmost number that is strictly greater than ℓ.
   - If no such number exists, attach an empty cell to the end of row $r+1$ and label the cell as $(r+1, \tilde{c})$, where $\tilde{c}$ is its column index. Let k be the empty character.
   - If such a number exists, label the value as k and the cell containing k as $(r+1, \tilde{c})$ where $\tilde{c}$ is the cell's column index.

We now break into cases:

(a) If $\tilde{c} \neq c$, then remove ℓ from $\mathsf{L}_T(r,c)$, replace k with ℓ, and attach k to the leg of $\mathsf{L}_T(r+1, \tilde{c})$.
(b) If $\tilde{c} = c$, then remove $[\ell, a] \cap \mathsf{A}_T(r,c)$ from $\mathsf{A}_T(r,c)$ where $[\ell, a] \cap \mathsf{A}_T(r,c)$ is the multiset $\{z \in \mathsf{A}_T(r,c) \mid \ell \leq z \leq a\}$. Remove ℓ from $\mathsf{L}_T(r,c)$, insert

$[\ell, a] \cap \mathsf{A}_T(r, c)$ into $\mathsf{A}_T(r+1, \tilde{c})$, replace the hook entry of $(r+1, \tilde{c})$ with ℓ, and attach k to $\mathsf{L}_T(r+1, \tilde{c})$.

(5) Output the resulting tableau.

Definition 3.34. The multiset uncrowding insertion $\tilde{\mathcal{V}} \colon \mathsf{HVT} \to \mathsf{HVT}$ is defined as $\tilde{\mathcal{V}}(T) = \tilde{\mathcal{V}}_b^d(T)$, where the integer $d \geq 1$ is minimal, such that $\mathsf{shape}(\tilde{\mathcal{V}}_b^d(T))/\mathsf{shape}(\tilde{\mathcal{V}}_b^{d-1}(T)) \neq \emptyset$ or $\tilde{\mathcal{V}}_b^d(T) = \tilde{\mathcal{V}}_b^{d-1}(T)$.

Definition 3.35. Let $T \in \mathsf{HVT}(\lambda)$ with leg excess α. The multiset uncrowding map

$$\tilde{\mathcal{U}} \colon \mathsf{HVT}(\lambda) \to \bigsqcup_{\mu \supseteq \lambda} \mathsf{MVT}(\mu) \times \mathcal{F}(\mu/\lambda)$$

is defined by the following algorithm:

(1) Let $\tilde{P}_0 = T$ and let $\tilde{Q}_0$ be the flagged increasing tableau of shape λ/λ.

(2) For $1 \leq i \leq \alpha$, let $\tilde{P}_{i+1} = \tilde{\mathcal{V}}(\tilde{P}_i)$. Let r be the index of the topmost row of $\tilde{P}_i$ containing a cell with nonzero leg excess and let $\tilde{r}$ be the row index of the cell $\mathsf{shape}(\tilde{P}_{i+1})/\mathsf{shape}(\tilde{P}_i)$. Then, $\tilde{Q}_{i+1}$ is obtained from $\tilde{Q}_i$ by appending the cell $\mathsf{shape}(\tilde{P}_{i+1})/\mathsf{shape}(\tilde{P}_i)$ to $\tilde{Q}_i$ and filling this cell with $\tilde{r} - r$.

Define $\tilde{\mathcal{U}}(T) = (\tilde{P}(T), \tilde{Q}(T)) := (\tilde{P}_\alpha, \tilde{Q}_\alpha)$.

Example 3.36. Let T be the hook-valued tableau

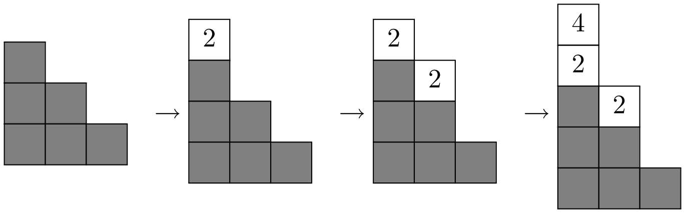
$$T = \begin{array}{|l|l|l|} 79 & & \\ 233 & \begin{smallmatrix}8\\78\end{smallmatrix} & \\ 1 & \begin{smallmatrix}3\\223\end{smallmatrix} & \begin{smallmatrix}7\\4\end{smallmatrix} \end{array}.$$

Then, we obtain the following sequence of tableaux $\tilde{\mathcal{V}}_b^i(T)$ for $0 \leq i \leq 2 = d$ when computing the first multiset uncrowding insertion:

$$\begin{array}{|l|l|l|} 79 & & \\ 233 & \begin{smallmatrix}8\\78\end{smallmatrix} & \\ 1 & \begin{smallmatrix}3\\223\end{smallmatrix} & \begin{smallmatrix}7\\4\end{smallmatrix} \end{array} \to \begin{array}{|l|l|l|} 9 & & \\ 78 & & \\ 233 & 78 & \\ 1 & \begin{smallmatrix}3\\223\end{smallmatrix} & \begin{smallmatrix}7\\4\end{smallmatrix} \end{array} \to \begin{array}{|l|l|l|} 9 & & \\ 78 & & \\ 233 & 78 & \\ 1 & \begin{smallmatrix}3\\223\end{smallmatrix} & \begin{smallmatrix}7\\4\end{smallmatrix} \end{array} = \tilde{\mathcal{V}}(T).$$

Continuing with the remaining multiset uncrowding insertions, we obtain the following sequences of tableaux for the multiset uncrowding map:

$$\begin{array}{|l|l|l|} 79 & & \\ 233 & \begin{smallmatrix}8\\78\end{smallmatrix} & \\ 1 & \begin{smallmatrix}3\\223\end{smallmatrix} & \begin{smallmatrix}7\\4\end{smallmatrix} \end{array} \to \begin{array}{|l|l|l|} 9 & & \\ 78 & & \\ 233 & 78 & \\ 1 & \begin{smallmatrix}3\\223\end{smallmatrix} & \begin{smallmatrix}7\\4\end{smallmatrix} \end{array} \to \begin{array}{|l|l|l|} 9 & & \\ 78 & 8 & \\ 233 & 77 & \\ 1 & \begin{smallmatrix}3\\223\end{smallmatrix} & 4 \end{array} \to \begin{array}{|l|l|l|} 9 & & \\ 8 & & \\ 77 & 8 & \\ 233 & 337 & \\ 1 & 22 & 4 \end{array} = \tilde{P}(T),$$



$= \tilde{Q}(T)$.

Proposition 3.37. *Let $T \in \mathsf{HVT}$. Then, $\tilde{\mathcal{U}}(T)$ is well defined.*

Proof. The statement follows from a similar argument to the proofs found in Corollary 3.7 and Lemma 3.9. $\square$

Similar to the uncrowding map $\mathcal{U}$, the multiset uncrowding map $\tilde{\mathcal{U}}$ intertwines with the corresponding crystal operators.

Theorem 3.38. *Let $T \in \mathsf{HVT}$.*
1. *If $f_i(T) = 0$, then $f_i(\tilde{P}(T)) = 0$.*
2. *If $e_i(T) = 0$, then $e_i(\tilde{P}(T)) = 0$.*
3. *If $f_i(T) \neq 0$, we have $f_i(\tilde{P}(T)) = \tilde{P}(f_i(T))$ and $\tilde{Q}(T) = \tilde{Q}(f_i(T))$.*
4. *If $e_i(T) \neq 0$, we have $e_i(\tilde{P}(T)) = \tilde{P}(e_i(T))$ and $\tilde{Q}(T) = \tilde{Q}(e_i(T))$.*

Proof. The proof follows similarly to those found in Proposition 3.12, Lemma 3.13, and Theorem 3.14. $\square$

4. Applications

In this section, we provide the expansion of the canonicalGrothendieck polynomials $G_\lambda(x; \alpha, \beta)$ in terms of the stable symmetric Grothendieck polynomials $G_\mu(x; \beta = -1)$ and in terms of the dual stable symmetric Grothendieck polynomials $g_\mu(x; \beta = 1)$ using techniques developed in [1]. We first review the basic definitions and Schur expansions of the two polynomials.

Recall from (1.1) that the stable symmetric Grothendieck polynomial is the generating function of set-valued tableaux

$$G_\mu(x; -1) = \sum_{S \in \mathsf{SVT}(\mu)} (-1)^{|S|-|\mu|} x^{\mathsf{weight}(S)}.$$

Its Schur expansion can be obtained from the crystal structure on set-valued tableaux [13]

$$G_\mu(x; -1) = \sum_{\substack{S \in \mathsf{SVT}(\mu) \\ e_i(S)=0 \ \forall i}} (-1)^{|S|-|\mu|} \, s_{\mathsf{weight}(S)}.$$

Definition 4.1. The reading word $\mathsf{word}(S) = w_1 w_2 \cdots w_n$ of a set-valued tableau $S \in \mathsf{SVT}(\mu)$ is obtained by reading the elements in the rows of S from the top row to the bottom row in the following way. In each row, first ignore the smallest element of each cell and read all remaining elements in descending order. Then, read the smallest elements of each cell in ascending order.

Example 4.2. The reading word of $P(T)$ in Example 3.6 is $\mathsf{word}(P(T)) = 8675423362111567$.

Example 4.3. The highest weight set-valued tableaux of shape (2) are

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

which gives the Schur expansion

$$G_{(2)}(x;-1) = s_2 - s_{21} + s_{211} - s_{2111} \pm \cdots .$$

The dual stable symmetric Grothendieck polynomials $g_\mu(x;1)$ are dual to $G_\mu(x;-1)$ under the Hall inner product on the ring of symmetric functions.

Definition 4.4. A reverse plane partition of shape μ is a filling of the cells in the Ferrers diagram of μ with positive integers, such that the entries are weakly increasing in rows and columns. We denote the collection of all reverse plane partitions of shape μ by $\mathsf{RPP}(\mu)$ and the set of all reverse plane partitions by RPP.

The evaluation $\mathsf{ev}(R)$ of a reverse plane partition $R \in \mathsf{RPP}$ is a composition $\alpha = (\alpha_i)_{i\geq 1}$, where α_i is the total number of columns in which i appears. The reading word $\mathsf{word}(R)$ is obtained by first circling the bottommost occurrence of each letter in each column, and then reading the circled letters row-by-row from top to bottom and left to right within each row.

Example 4.5. Consider the reverse plane partition

$$R = \begin{array}{|c|c|c}\hline 1 & 2 & \\ \hline 1 & 1 & \multicolumn{1}{c|}{3} \\ \hline\end{array} \in \mathsf{RPP}((3,2)).$$

By circling the bottommost occurrence of each letter in each column, we obtain

$$R = \begin{array}{|c|c|c}\hline 1 & 2 & \\ \hline 1 & 1 & \multicolumn{1}{c|}{3} \\ \hline\end{array}\,, \ \mathsf{ev}(R) = (2,1,1),\ \mathsf{word}(R) = 2113.$$

Lam and Pylyavskyy [10] showed that the dual stable symmetric Grothendieck polynomials $g_\mu(x;1)$ are generating functions of reverse plane partitions of shape μ

$$g_\mu(x;1) = \sum_{R\in\mathsf{RPP}(\mu)} x^{\mathsf{ev}(R)}.$$

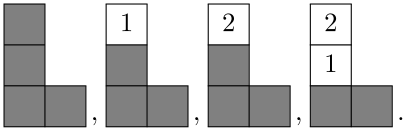
They also provided the Schur expansion of the dual stable symmetric Grothendieck polynomials [10, Theorem 9.8]

$$g_\mu(x;1) = \sum_F s_{\mathsf{innershape}(F)},$$

where the sum is over all flagged increasing tableaux whose outer shape is μ.

Example 4.6. When $\mu = (\mu_1)$ is a partition with only one row, we have $g_{(\mu_1)}(x;1) = s_{(\mu_1)}$.

The flagged increasing tableaux of outer shape $(2,1,1)$ are



Hence, $g_{211}(x;1) = s_{211} + 2s_{21} + s_2$.

According to [1], a symmetric function f_α over the ring R is said to have a tableaux Schur expansion if there is a set of tableaux $\mathbb{T}(\alpha)$ and a weight function $\mathsf{wt}_\alpha \colon \mathbb{T}(\alpha) \to R$, so that

$$f_\alpha = \sum_{T\in\mathbb{T}(\alpha)} \mathsf{wt}_\alpha(T) s_{\mathsf{shape}(T)}.$$

Furthermore, any symmetric function with such a property has the following expansion in terms of $G_\mu(x;-1)$ and $g_\mu(x;1)$.

Theorem 4.7. [1, Theorem 3.5] *Let f_α be a symmetric function with a tableaux Schur expansion $f_\alpha = \sum_{T\in\mathbb{T}(\alpha)} \mathsf{wt}_\alpha(T) s_{\mathsf{shape}(T)}$ for some $\mathbb{T}(\alpha)$. Let $\mathbb{S}(\alpha)$ and $\mathbb{R}(\alpha)$ be defined as sets of set-valued tableaux and reverse plane partitions, respectively, by*

$$\begin{aligned} S \in \mathbb{S}(\alpha) &\textit{ if and only if } P(\mathsf{word}(S)) \in \mathbb{T}(\alpha), \textit{ and}\\ R \in \mathbb{R}(\alpha) &\textit{ if and only if } P(\mathsf{word}(R)) \in \mathbb{T}(\alpha),\end{aligned}$$

where $P(w)$ is the RSK insertion tableau of the word w. We also extend wt_α to $\mathbb{S}(\alpha)$ and $\mathbb{R}(\alpha)$ by setting $\mathsf{wt}_\alpha(X) := \mathsf{wt}_\alpha(P(\mathsf{word}(X)))$ for any $X \in \mathbb{S}(\alpha)$ or $\mathbb{R}(\alpha)$. Then, we have

$$\begin{aligned} f_\alpha &= \sum_{R\in\mathbb{R}(\alpha)} \mathsf{wt}_\alpha(R) G_{\mathsf{shape}(R)}(x;-1), \textit{ and}\\ f_\alpha &= \sum_{S\in\mathbb{S}(\alpha)} \mathsf{wt}_\alpha(S)(-1)^{|S|-|\mathsf{shape}(S)|} g_{\mathsf{shape}(S)}(x;1).\end{aligned}$$

Proposition 4.8. *The canonical Grothendieck polynomials have a tableaux Schur expansion.*

Proof. Recall the uncrowding map on set-valued tableaux of Definition 3.1

$$\mathcal{U}_{\mathsf{SVT}} : \mathsf{SVT}(\mu) \longrightarrow \bigsqcup_{\nu\supseteq\mu} \mathsf{SSYT}(\nu) \times \mathcal{F}(\nu/\mu).$$

By Corollary 3.32, we have a bijection

$$\mathcal{U} : \mathsf{HVT}(\lambda) \to \mathsf{K}_\lambda = \bigsqcup_{\mu\supseteq\lambda} \mathsf{K}_\lambda(\mu).$$

Note that $\mathsf{K}_\lambda \subseteq \bigsqcup_{\mu\supseteq\lambda} \mathsf{SVT}(\mu) \times \hat{\mathcal{F}}(\mu/\lambda)$. Denote

$$\phi_\lambda(S) = |\{F \in \hat{\mathcal{F}} \mid (S,F) \in \mathsf{K}_\lambda\}|.$$

Note that sometimes $\phi_\lambda(S) = 0$.

Given $H \in \mathsf{HVT}(\lambda)$, we have $\mathcal{U}(H) = (S, F) \in \mathsf{SVT}(\mu) \times \hat{\mathcal{F}}(\mu/\lambda)$ for some $\mu \supseteq \lambda$ and $|\mu| = |\lambda| + a(H)$. We can also obtain $\mathcal{U}_{\mathsf{SVT}}(S) = (T, Q) \in \mathsf{SSYT}(\nu) \times \mathcal{F}(\nu/\mu)$ for some $\nu \supseteq \mu$ and $|\nu| = |H|$. The weights of H, S and T are the same. When H is highest weight, that is $e_i(H) = 0$ for all i, then S and T are also of highest weight and $\mathsf{weight}(H) = \mathsf{shape}(T)$. Denote by $\mathsf{HVT}_h(\lambda), \mathsf{SVT}_h(\lambda), \mathsf{SSYT}_h(\lambda)$ the subset of highest weight elements in $\mathsf{HVT}(\lambda), \mathsf{SVT}(\lambda), \mathsf{SSYT}(\lambda)$, respectively.

Applying [8, Theorem 4.6] and the above correspondence, we obtain

$$\begin{aligned}
G_\lambda(x; \alpha, \beta) &= \sum_{H \in \mathsf{HVT}_h(\lambda)} \alpha^{a(H)} \beta^{\ell(H)} s_{\mathsf{weight}(H)} \\
&= \sum_{\mu \supseteq \lambda} \sum_{(S,F) \in \mathsf{K}_\lambda(\mu)} \alpha^{|\mu|-|\lambda|} \beta^{|S|-|\mu|} s_{\mathsf{weight}(S)} \\
&= \sum_{\mu \supseteq \lambda} \sum_{S \in \mathsf{SVT}_h(\mu)} \phi_\lambda(S) \alpha^{|\mu|-|\lambda|} \beta^{|S|-|\mu|} s_{\mathsf{weight}(S)} \\
&= \sum_{\mu \supseteq \lambda} \sum_{\nu \supseteq \mu} \sum_{T \in \mathsf{SSYT}_h(\nu)} \sum_{Q \in \mathcal{F}(\nu/\mu)} \phi_\lambda(\mathcal{U}_{\mathsf{SVT}}^{-1}(T, Q)) \alpha^{|\mu|-|\lambda|} \beta^{|\nu|-|\mu|} s_{\mathsf{weight}(T)} \\
&= \sum_{\mu \supseteq \lambda} \sum_{\nu \supseteq \mu} \sum_{T \in \mathsf{SSYT}_h(\nu)} \alpha^{|\mu|-|\lambda|} \beta^{|\nu|-|\mu|} \sum_{Q \in \mathcal{F}(\nu/\mu)} \phi_\lambda(\mathcal{U}_{\mathsf{SVT}}^{-1}(T, Q)) s_{\mathsf{shape}(T)} \\
&= \sum_{T \in \mathbb{T}(\lambda)} \mathsf{wt}_\lambda(T) s_{\mathsf{shape}(T)},
\end{aligned}$$

where $\mathbb{T}(\lambda) = \{T \in \mathsf{SSYT}_h(\nu) \mid \nu \supseteq \lambda\}$ and

$$\mathsf{wt}_\lambda(T) = \sum_{\mu : \lambda \subseteq \mu \subseteq \mathsf{shape}(T)} \alpha^{|\mu|-|\lambda|} \beta^{|\mathsf{shape}(T)|-|\mu|} \sum_{Q \in \mathcal{F}(\mathsf{shape}(T)/\mu)} \phi_\lambda(\mathcal{U}_{\mathsf{SVT}}^{-1}(T, Q)).$$

□

Note that Proposition 4.8 in particular implies that the canonical Grothendieck polynomials are Schur positive. This was known from [8], but here an explicit tableaux formula is given.

Corollary 4.9. *The canonical Grothendieck polynomials have $G_\mu(x; -1)$ and $g_\mu(x; 1)$ expansions*

$$\begin{aligned}
G_\lambda(x; \alpha, \beta) &= \sum_{R \in \mathbb{R}(\lambda)} \mathsf{wt}_\lambda(R) G_{\mathsf{shape}(R)}(x; -1), \\
G_\lambda(x; \alpha, \beta) &= \sum_{S \in \mathbb{S}(\lambda)} \mathsf{wt}_\lambda(S) (-1)^{|S|-|\mathsf{shape}(S)|} g_{\mathsf{shape}(S)}(x; 1).
\end{aligned}$$

Example 4.10. We compute the first two terms in $G_{(2)}(x; \alpha, \beta) = s_2 + \beta s_{21} + 2\alpha s_3 + 2\alpha\beta s_{31} + \cdots$. The semistandard Young tableaux involved are

$$\mathbb{T}((2)) = \left\{ \begin{array}{|c|c|} \hline 1 & 1 \\ \hline \end{array}\,, \begin{array}{|c|c} \hline 2 & \\ \hline 1 & \multicolumn{1}{c|}{1} \\ \hline \end{array}\,, \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline \end{array}\,, \begin{array}{|c|cc} \hline 2 & & \\ \hline 1 & \multicolumn{1}{c|}{1} & \multicolumn{1}{c|}{1} \\ \hline \end{array}\,, \dots \right\}.$$

Labelling the tableaux $T_1, T_2, T_3, T_4, \ldots$, we have $\mathsf{wt}_{(2)}(T_1) = 1, \mathsf{wt}_{(2)}(T_2) = \beta, \mathsf{wt}_{(2)}(T_3) = 2\alpha, \mathsf{wt}_{(2)}(T_4) = 2\alpha\beta$. Next, we compute the elements in $\mathbb{R}((2))$ and $\mathbb{S}((2))$ that correspond to T_1 and T_2

$$\{R \in \mathbb{R}((2)) \mid P(\mathsf{word}(R)) = T_1\} = \left\{ \begin{array}{|c|c|}\hline 1 & 1 \\\hline\end{array}, \begin{array}{|c|c}\hline 1 & \\\hline 1 & \multicolumn{1}{c|}{1} \\\hline\end{array}, \begin{array}{|c|c|}\hline 1 & 1 \\\hline 1 & 1 \\\hline\end{array}, \begin{array}{|c|c}\hline 1 & \\\hline 1 & \\\hline 1 & \multicolumn{1}{c|}{1} \\\hline\end{array}, \ldots \right\}$$

$$\{R \in \mathbb{R}((2)) \mid P(\mathsf{word}(R)) = T_2\} = \left\{ \begin{array}{|c|c}\hline 2 & \\\hline 1 & \multicolumn{1}{c|}{1} \\\hline\end{array}, \begin{array}{|c|c|}\hline 1 & 2 \\\hline 1 & 1 \\\hline\end{array}, \begin{array}{|c|c}\hline 2 & \\\hline 1 & \\\hline 1 & \multicolumn{1}{c|}{1} \\\hline\end{array}, \begin{array}{|c|c}\hline 2 & \\\hline 2 & \\\hline 1 & \multicolumn{1}{c|}{1} \\\hline\end{array}, \ldots \right\}$$

$$\{S \in \mathbb{S}((2)) \mid P(\mathsf{word}(S)) = T_1\} = \left\{ \begin{array}{|c|c|}\hline 1 & 1 \\\hline\end{array} \right\}$$

$$\{S \in \mathbb{S}((2)) \mid P(\mathsf{word}(S)) = T_2\} = \left\{ \begin{array}{|c|c}\hline 2 & \\\hline 1 & \multicolumn{1}{c|}{1} \\\hline\end{array}, \begin{array}{|c|c|}\hline 1 & \begin{smallmatrix}2\\1\end{smallmatrix} \\\hline\end{array} \right\}.$$

Applying the expansion formulas, we obtain

$$\begin{aligned} G_{(2)}(x;\alpha,\beta) =& (G_{(2)}(x;-1) + G_{(21)}(x;-1) + G_{(22)}(x;-1) + G_{(211)}(x;-1) + \cdots) \\ &+ \beta(G_{(21)}(x;-1) + G_{(22)}(x;-1) + 2G_{(211)}(x;-1) + \cdots) + \cdots \\ G_{(2)}(x;\alpha,\beta) =& g_{(2)}(x;1) + \beta(g_{(21)}(x;1) - g_{(2)}(x;1)) + \cdots . \end{aligned}$$

Acknowledgements

We are grateful to Graham Hawkes and Travis Scrimshaw for discussions. This work was partially supported by NSF grant DMS–1764153. JiP was partially supported by NSF grant DMS–1700814. AS was partially supported by NSF grant DMS–1760329.

References

[1] Jason Bandlow and Jennifer Morse. Combinatorial expansions in K-theoretic bases. Electron. J. Combin., 19(4):Paper 39, 27, 2012.

[2] Daniel Bump and Anne Schilling. Crystal bases. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2017. Representations and combinatorics.

[3] Anders Skovsted Buch. A Littlewood-Richardson rule for the K-theory of Grassmannians. Acta Math., 189(1):37–78, 2002.

[4] Melody Chan and Nathan Pflueger. Combinatorial relations on skew Schur and skew stable Grothendieck polynomials. Algebr. Comb., 4(1):175–188, 2021.

[5] William Fulton. Young tableaux, volume 35 of London Mathematical Society Student Texts. Cambridge University Press, Cambridge, 1997. With applications to representation theory and geometry.

[6] Ajeeth Gunna and Paul Zinn-Justin. Vertex models for canonical Grothendieck polynomials and their duals. arXiv preprint arXiv:2009.13172, 2020.

[7] Jin Hong and Seok-Jin Kang. Introduction to quantum groups and crystal bases, volume 42 of Graduate Studies in Mathematics. American Mathematical Society, Providence, RI, 2002.

[8] Graham Hawkes and Travis Scrimshaw. Crystal structures for canonical Grothendieck functions. Algebraic Combinatorics, 3(3):727–755, 2020.

[9] Cristian Lenart. Combinatorial aspects of the K-theory of Grassmannians. Ann. Comb., 4(1):67–82, 2000.

[10] Thomas Lam and Pavlo Pylyavskyy. Combinatorial Hopf algebras and K-homology of Grassmannians. Int. Math. Res. Not. IMRN, (24):Art. ID rnm125, 48, 2007.

[11] Alain Lascoux and Marcel-Paul Schützenberger. Symmetry and flag manifolds. In Invariant theory (Montecatini, 1982), volume 996 of Lecture Notes in Math., pages 118–144. Springer, Berlin, 1983.

[12] Jennifer Morse, Jianping Pan, Wencin Poh, and Anne Schilling. A crystal on decreasing factorizations in the 0-Hecke monoid. Electron. J. Combin., 27(2):Paper 2, 29, 2020.

[13] Cara Monical, Oliver Pechenik, and Travis Scrimshaw. Crystal structures for symmetric Grothendieck polynomials. Transform. Groups, 26(3):1025–1075, 2021.

[14] Rebecca Patrias. Antipode formulas for some combinatorial Hopf algebras. Electron. J. Combin., 23(4):Paper 4, 30, 2016.

[15] Vic Reiner, Bridget E. Tenner, and Alexander Yong. Poset edge densities, nearly reduced words, and barely set-valued tableaux. J. Combin. Theory, Ser. A, 158:66–125, 2018.

[16] Damir Yeliussizov. Duality and deformations of stable Grothendieck polynomials. J. Algebraic Combin., 45(1):295–344, 2017.

Jianping Pan, Joseph Pappe, Wencin Poh and Anne Schilling
Department of Mathematics
UC Davis
One Shields Ave.,
Davis CA 95616-8633
USA
e-mail: anne@math.ucdavis.edu

Joseph Pappe
e-mail: jhpappe@ucdavis.edu

Wencin Poh
e-mail: wpoh@ucdavis.edu

Present Address
Jianping Pan
Present address:
Department of Mathematics
NC State University
Raleigh NC 27695-8205
USA
e-mail: jpan9@ncsu.edu

Communicated by Alexander Yong
Received: 4 January 2021.
Accepted: 10 January 2022.