Crystal Combinatorics and Grothendieck Polynomials

By

JIANPING PAN

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

MATHEMATICS

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Professor Anne Schilling, Chair

Professor Fu Liu

Professor Monica Vazirani

Committee in Charge

2021

© Jianping Pan, 2021. All rights reserved.

To my grandpa.

Contents

Crystal Combinatorics and Grothendieck Polynomials

Abstract

Crystals are models for representations of symmetrizable Kac-Moody Lie algebras. They have close connections to algebra and geometry via symmetric functions. In this dissertation, combinatorics related to two kinds of symmetric functions arising from Schubert calculus is discussed. The first one is the stable Grothendieck polynomial. We introduce a type A crystal structure for the combinatorial objects underlying the stable Grothendieck polynomials which we call $\star$-crystal. This crystal is a K-theoretic generalization of the Morse-Schilling crystal on decreasing factorizations in the symmetric group. We prove that under the residue map the $\star$-crystal intertwines with the crystal on set-valued tableaux introduced by Monical, Pechenik and Scrimshaw. We also define a new insertion from decreasing factorizations to pairs of semistandard Young tableaux and prove several properties, such as its relation to the Hecke insertion and the uncrowding algorithm. The new insertion also intertwines with the crystal operators.

The second one is the stable canonical Grothendieck polynomial. Whereas set-valued tableaux are the combinatorial objects associated to stable Grothendieck polynomials, hook-valued tableaux are associated to stable canonical Grothendieck polynomials. 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.

Acknowledgments

Time flies like an arrow; fruit flies like a banana. After a lot of time and a lot of bananas, I am leaving for my next journey, and I owe a lot of thanks to a lot of people.

First, I would like to thank my advisor Prof. Anne Schilling. Being her student is one of the best things that has happened to me. Her passion and curiosity are contagious. Her devotion and perseverance are impressive. In research, it takes many ideas to form a conjecture, and many failed conjectures to form a theorem (or not). Along the long winding road, I am thankful for her being so patient, encouraging, advocating, and caring.

I would like to thank the informal discrete group. I had the opportunity to learn a variety of topics and practice giving presentations with them. I am especially especially thankful to Prof. Monica Vazirani, for giving me feedback after my presentations and writing me recommendation letters, and to Prof. Eugene Gorskiy for giving me advice on job search and funding.

I would like to thank Prof. Fu Liu and Prof. Monica Vazirani for serving on my dissertation committee, and Prof. Eugene Gorskiy and Prof. Jaroslav Trnka for serving on my qualifying exam committee. I would like to thank my collaborator, Prof. Jennifer Morse for the good work we did and for writing me recommendation letters. I would like to thank Prof. Elena Fuchs for lots of inspiring conversations over coffee and for writing me recommendation letters. I would like to thank Prof. Greg Kuperberg, for telling me many stories, and teaching me some English, including the first sentence of this page.

I am grateful for my academic brothers Wencin Poh and Joseph Pappe. They are the ones that I have been reading papers with, staring at figures with, debugging with, arguing with, staying up with, and also writing papers with. I am thankful for them putting up with me.

I appreciate the insightful discussions and shared learning experiences with many friends, especially Aram Dermenjian, Emily Gunawan, Colin Hagemeyer, Yonggyu Lee, Beibei Liu, Yanwen Luo, Jingyang Shu, Austin Trans, Zhenyang Zhang, Yue Zhao, and Yunshen Zhou.

I would like to thank the office staff, especially Tina Denena and Victoria Whistler, for always being there for students.

Last, I would like to thank the National Science Foundation, for partially supporting this dissertation with grants DMS–1764153 and DMS–1700814.

CHAPTER 1

Introduction

In this chapter, we provide the overview of the dissertation, and some preliminaries on crystals, tableaux and Grothendieck polynomials.

1.1. Overview

A complete flag $F_\bullet = \{\{0\} = F_0 \subset F_1 \subset \cdots \subset F_n = \mathbb{C}^n\}$ in $\mathbb{C}^n$ is a nested sequence of vector spaces such that $\mathsf{dim}(F_i) = i$ for $0 \leqslant i \leqslant n$. The flag manifold $\mathcal{F}l_n$ is all complete flags in $\mathbb{C}^n$. The Schubert varieties of $\mathcal{F}l_n$ are subsets of $\mathcal{F}l_n$ indexed by permutations in $\mathbb{S}_n$, which will be defined in Definition 1.2.6. An important problem in Schubert calculus is how to compute the intersection numbers of the Schubert varieties. One concrete solution is to compute the expansion of the product of their polynomial representatives. Lascoux and Schützenberger [**16**, **17**] introduced Grothendieck polynomials in 1982, which are representatives for the Schubert classes in the K-theory of the flag manifold. The stabilizations of Grothendieck polynomials are symmetric functions and were studied by Fomin and Kirillov [**8**]. They gave a combinatorial definition of the stable Grothendieck polynomials, labeled by permutations $w \in \mathbb{S}_n$, as

$$G_w(x_1, \ldots, x_m; \beta) = \sum_{(\mathbf{k},\mathbf{h})} \beta^{\ell(\mathbf{h})-\ell(w)} x^{\mathbf{k}}, \tag{1.1}$$

where the sum is over decreasing factorizations $[\mathbf{k}, \mathbf{h}]^t$ of w in the 0-Hecke monoid, which will be defined in Definition 2.1.2. When $\beta = 0$, G_w specializes to the Stanley symmetric function F_w [**29**].

A robust combinatorial picture has been developed for the special case of Grothendieck polynomials indexed by Grassmannian permutations. Buch [**3**] showed that the *stable Grassmannian (or symmetric) Grothendieck polynomials* can be realized as the generating functions of semistandard set-valued tableaux:

$$G_\lambda(x_1,\ldots,x_m;\beta) = \sum_{T\in \mathsf{SVT}^m(\lambda)} \beta^{\mathsf{ex}(T)} \boldsymbol{x}^{\mathrm{wt}(T)}, \tag{1.2}$$

where semistandard set-valued tableaux $\mathsf{SVT}^m(\lambda), \mathrm{wt}(T)$ and $\mathsf{ex}(T)$ will be defined in Definition 1.2.4. The set-valued tableaux play an important role in K-Theory. They form a generalization of semistandard Young tableaux, where boxes may contain sets of intergers instead of just integers. Stable symmetric Grothendieck polynomials G_λ can be viewed as a K-theory analogue of the Schur functions s_λ (while the Grothendieck polynomial is an analog of the Schubert polynomial), which will be defined in (1.3). Buch [**3**] also described the structure coefficients $c^\nu_{\lambda\mu}$ of $G_\lambda G_\mu$ in terms of set-valued tableaux, generalizing the Littlewood–Richardson rule for Schur functions.

The crystal structure on a combinatorial set is the combinatorial shadow of a (quantum) group representation (see for example [**5**, **12**, **13**, **14**, **20**, **21**]). The characters of connected components of crystal graphs are the Schur polynomials. Beyond the representation theory, crystals can be used to prove certain polynomial expansions. Recently, Monical, Pechenik and Scrimshaw [**22**] provided a type A_{m-1}-crystal structure on $\mathsf{SVT}^m(\lambda)$ which, in particular, implies that

$$G_\lambda(x_1,\ldots,x_m;\beta) = \sum_\mu \beta^{|\mu|-|\lambda|} M^\mu_\lambda\, s_\mu(x_1,\ldots,x_m),$$

where M^μ_λ is the number of highest-weight set-valued tableaux of weight μ in the crystal $\mathsf{SVT}^m(\lambda)$ and s_μ is the Schur function. Their approach recovers a Schur expansion formula for Grassmannian Grothendieck polynomials given by Lenart [**18**, Theorem 2.2] in terms of flagged increasing tableaux. Blasiak, Morse and Pun proved Schur and key expansions of a certain Catalan function by relating it to the DARK crystal, see [**2**].

In Chapter 2, we define a type A crystal structure on decreasing factorizations of w in the 0-Hecke monoid of (1.1), when w is fully-commutative [**30**] (or equivalently 321-avoiding). A permutation w is fully-commutative if its reduced expressions do not contain any braids. The number of fully-commutative elements of $\mathbb{S}_n$ is the n-th Catalan number. The residue map (see Section 2.1.4) shows that fully-commutative permutations correspond to skew shapes. We call our crystal $\star$-crystal. It is local in the sense that the crystal operators $f^\star_i$ and $e^\star_i$ only act on the

i-th and $(i+1)$-th factors of the decreasing factorization. It generalizes the crystal of Morse and Schilling [**24**] for Stanley symmetric functions (or equivalently reduced decreasing factorizations of w) in the fully-commutative case. We show that the $\star$-crystal and the crystal on set-valued tableaux intertwine under the residue map (see Theorem 2.1.2). We also show that the residue map and the Hecke insertion [**4**] are related (see Theorem 2.2.1), thereby resolving [**22**, Open Problem 5.8] in the fully-commutative case. In addition, we provide a new insertion algorithm, which we call $\star$-insertion, from decreasing factorizations on fully-commutative elements in the 0-Hecke monoid to pairs of (transposes of) semistandard Young tableaux of the same shape (see Definition 2.2.2 and Theorem 2.2.2), which intertwines with crystal operators (see Theorem 2.3.1). This recovers the Schur expansion of G_w of Fomin and Greene [**7**] when w is fully-commutative, stating that

$$G_w = \sum_{\mu} \beta^{|\mu|-\ell(w)} g_w^{\mu} s_{\mu},$$

where

$$g_w^{\mu} = |\{T \in \mathsf{SSYT}^n(\mu') \mid w_C(T) \equiv w\}|,$$

and $w_C(T)$ is the column reading word of T (see Remark 2.3.3). We also show that the composition of the residue map with the $\star$-insertion is related to the uncrowding algorithm [**3**] (see Theorem 2.3.2). Other insertion algorithms have recently been studied in [**6**].

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 $c_{\lambda\mu}^{\nu} = c_{\lambda'\mu'}^{\nu'}$, where λ' denotes the conjugate of the partition λ. 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 [**15**]

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

Yeliussizov [**32**] 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 Chapter 3, we describe a novel *uncrowding algorithm* on hook-valued tableaux (see Definitions 3.2.1, 3.2.2 and 3.2.3). 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 [**18**]. 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 (also known as an elegant filling), which serves as a recording tableau.

Chan and Pflueger [**6**] 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 [**18**], Buch [**3**], and Reiner, Tenner and Yong [**28**] 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 analogues of the Hopf algebras of symmetric functions, quasisymmetric functions, noncommutative symmetric functions, and of the Malvenuto–Reutenauer Hopf algebra of permutations [**1**, **15**, **26**]. In [**10**], 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 [**22**] (see also [**23**]). A crystal structure on hook-valued tableaux was recently introduced by Hawkes and Scrimshaw [**11**]. 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 3.2.1 and Theorem 3.2.1). This was stated as an open problem in [**11**]. As a consequence it provides another proof for that the crystal structure on hook-valued tableaux is Stembridge.

1.2. Preliminaries

1.2.1. Kawhiwara crystals. Let $\mathfrak{g}$ be a symmetrizable Kac-Moody algebra with associated weight lattice Λ. Let I be the index set of the Dynkin diagram and denote the simple roots and simple coroots α_i and $\alpha_i^\vee$.

DEFINITION 1.2.1. *An abstract $U_q(\mathfrak{g})$-Kashiwara crystal (or crystal for short) is a nonempty set $\mathcal{B}$ together with maps*

$$e_i, f_i : \mathcal{B} \to \mathcal{B} \sqcup \{0\}, \qquad \text{(they are called the crystal operators)}$$

$$\varepsilon_i, \varphi_i : \mathcal{B} \to \mathbb{Z} \sqcup \{-\infty\}, \qquad \text{(they are called the string lengths)}$$

$$\mathrm{wt} : \mathcal{B} \to \Lambda, \qquad \text{(this is called the weight map)}$$

where $i \in I$ and $0 \notin \mathcal{B}$ is an auxiliary element, satisfying the following conditions:

A1 f_i and e_i are partial inverses of each other. That is, if $x, y \in \mathcal{B}$ then $e_i(x) = y$ if and only if $f_i(y) = x$. In this case, it is assumed that

$$\mathrm{wt}(y) = \mathrm{wt}(x) + \alpha, \quad \varepsilon_i(y) = \varepsilon_i(x) - 1, \quad \varphi_i(y) = \varphi_i(x) + 1.$$

A2 We require that

$$\varphi_i(x) = \langle \mathrm{wt}(x), \alpha_i^\vee \rangle + \varepsilon_i(x)$$

for all $x \in \mathcal{B}$ and $i \in I$. In particular, if $\varphi_i(x) = -\infty$, then $\varepsilon_i(x) = -\infty$. If $\varphi_i(x) = -\infty$, then we require that $e_i(x) = f_i(x) = 0$.

If $\varphi_i(x) = \max\{k \in \mathbb{Z}_{\geqslant 0} | \, f_i^k(x) \neq 0\}$, and $\varepsilon_i(x) = \max\{k \in \mathbb{Z}_{\geqslant 0} | \, e_i^k(x) \neq 0\}$, for all $i \in I$, then $\mathcal{B}$ is also called seminormal.

The above Definition 1.2.1 defines an edge-labeled directed graph where the vertices are $\mathcal{B}$ and the edges are formed by *crystal operators* f_i's. In general, they may not correspond to representations. Stembridge [**31**] gave a set of local axioms that uniquely characterize the crystals correspond to representations of algebras of simply-laced types. These crystals are called *Stembridge crystals*.

Example 1.2.1. *When $\mathfrak{g}$ is of type A_n, the weight lattice is $\mathbb{Z}^{n+1}$ generated by $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{n+1}$ and the simple roots are $\alpha_1 = \boldsymbol{e}_1 - \boldsymbol{e}_2, \ldots, \alpha_n = \boldsymbol{e}_n - \boldsymbol{e}_{n+1}$. The standard seminormal crystal has the following crystal graph*

$$\boxed{1} \xrightarrow{1} \boxed{2} \xrightarrow{2} \ldots \xrightarrow{n} \boxed{n+1}.$$

The weight map is $\mathrm{wt}(\boxed{i}) = \boldsymbol{e}_i$. We denote this crystal by $\mathbb{B}$.

Definition 1.2.2. *Let $\mathcal{B}$ and $\mathcal{C}$ be two abstract $U_q(\mathfrak{g})$-crystals with index set I. A* crystal morphism *is a map $\phi : \mathcal{B} \to \mathcal{C} \sqcup \{0\}$ such that*

(1) if $b \in \mathcal{B}$ and $\phi(b) \in \mathcal{C}$, then
  - *(a) $\mathrm{wt}(\phi(b)) = \mathrm{wt}(b)$,*
  - *(b) $\varepsilon_i(\phi(b)) = \varepsilon_i(b)$ for all $i \in I$, and*
  - *(c) $\varphi_i(\phi(b)) = \varphi_i(b)$ for all $i \in I$;*

(2) if $b, e_i b \in \mathcal{B}$ such that $\phi(b), \phi(e_i b) \in \mathcal{C}$, then we have $\phi(e_i b) = e_i \phi(b)$;

(3) if $b, f_i b \in \mathcal{B}$ such that $\phi(b), \phi(f_i b) \in \mathcal{C}$, then we have $\phi(f_i b) = f_i \phi(b)$.

A morphism ϕ is called *strict* if ϕ commutes with e_i and f_i for all $i \in I$. Moreover, a crystal morphism $\phi : \mathcal{B} \to \mathcal{C} \sqcup \{0\}$ is called a *crystal isomorphism* if the induced map $\phi : \mathcal{B} \sqcup \{0\} \to \mathcal{C} \sqcup \{0\}$ with $\phi(0) = 0$ is a bijection. In this case, we write $\mathcal{B} \cong \mathcal{C}$.

Let $\mathcal{B}$ and $\mathcal{C}$ be two abstract $U_q(\mathfrak{g})$-crystals, we can also take their *tensor product* $\mathcal{B} \otimes \mathcal{C}$. As a set, it is the Cartesian product, denote by $x \otimes y$ where $x \in \mathcal{B}$ and $y \in \mathcal{C}$. The weight map is defined to be $\mathrm{wt}(x \otimes y) = \mathrm{wt}(x) + \mathrm{wt}(y)$, and the crystal operators are defined as follows:

$$f_i(x \otimes y) = \begin{cases} f_i(x) \otimes y & \text{if } \varphi_i(y) \leqslant \varepsilon_i(x), \\ x \otimes f_i(y) & \text{if } \varphi_i(y) > \varepsilon_i(x). \end{cases}$$

and

$$e_i(x \otimes y) = \begin{cases} e_i(x) \otimes y & \text{if } \varphi_i(y) < \varepsilon_i(x), \\ x \otimes e_i(y) & \text{if } \varphi_i(y) \geqslant \varepsilon_i(x). \end{cases}$$

Note that $x \otimes 0 = 0 \otimes x = 0$. The string maps are defined as follows:

$$\varphi_i(x \otimes y) = \max(\varphi_i(x), \varphi_i(y) + \langle \mathrm{wt}(x), \alpha_i^\vee \rangle), \quad \varepsilon_i(x \otimes y) = \max(\varepsilon_i(y), \varepsilon_i(x) - \langle \mathrm{wt}(y), \alpha_i^\vee \rangle).$$

1.2.2. Tableaux combinatorics. The *Young diagram* of a partition $\lambda = (\lambda_1 \geqslant \lambda_2 \cdots \geqslant \lambda_\ell)$ is a finite collection of boxes, aligned at the left, in which the i-th row has λ_i boxes. We use the French convention, where the botton row is the first row. A *Young tableau* is a way of filling each box of Young diagram with symbols from an alphabet, with restrictions depending on the context. They are important models for crystals. A *semistandard Young tableau* of shape λ is a filling of boxes of λ by positive integers so that each box contains a single number, the entries of each row weakly increase from left to right, and the entries of each column strictly increase from bottom to top. Denote the set of all semistandard Young tableaux of shape λ by $\mathsf{SSYT}^m(\lambda)$ where m is the maximal integer allowed in the filling and m is allowed to be ∞. The generating function for $\mathsf{SSYT}^m(\lambda)$ is the *Schur* polynomials of partition λ:

$$s_\lambda(x_1, \ldots, x_m) = \sum_{T \in \mathsf{SSYT}^m(\lambda)} \boldsymbol{x}^{\mathrm{wt}(T)}. \tag{1.3}$$

When setting $m = \infty$, we obtain the Schur function in infinite number of variables.

We can define a crystal structure on $\mathsf{SSYT}^m(\lambda)$ by embedding them into $\mathbb{B}^{\otimes k}$ via row reading or column reading. The weight of $T \in \mathsf{SSYT}^m(\lambda)$ is $(c_1, c_2, \ldots)$ where c_i is the number of i's in the filling of T. We give the simplified version of the crystal operators via the *signature rule*.

Definition 1.2.3. *Let $T \in \mathsf{SSYT}^m(\lambda)$. We employ the following pairing rule for letters i and $i+1$. Assign $-$ to every column of T containing an i but not an $i+1$. Similarly, assign $+$ to every* *column of T containing an $i+1$ but not an i. Then, successively pair each $+$ that is to the left of and adjacent to a $-$, removing all paired signs until nothing can be paired.*

The operator f_i changes the i in the rightmost column with an unpaired $-$ (if this exists) to $i+1$. If no unpaired $-$ exists, then f_i annihilates T. Similarly, the operator e_i changes the $i+1$ in the leftmost column with an unpaired $+$ (if this exists) to i. If no unpaired $+$ exists, then e_i annihilates T.

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

Monical, Pechenik and Scrimshaw in [**22**] defined a semistandard set-valued filling (generalizing semistandard Young tableaux SSYT by Alfred Young) on Young diagrams of partition shape λ, called *semistandard set-valued tableaux* SVT. They also defined a *Stembridge crystal* structure on them. A slightly generalized version of the crystal structure on SVT will be given in Section 2.1.3.

Definition 1.2.4 ([**3**]). *A semistandard set-valued tableau T is the filling of a skew shape λ/μ with nonempty subsets of positive integers such that:*

- *for all adjacent cells A, B in the same row with A to the left of B, we have* $\max(A) \leqslant \min(B)$,
- *for all adjacent cells A, C in the same column with A below C, we have* $\max(A) < \min(C)$.

The weight of T, denoted by $\mathrm{wt}(T)$, is the integer vector whose i-th component counts the number of i's that occur in T. The excess of T is defined as $\mathsf{ex}(T) = |\,\mathrm{wt}(T)| - |\lambda|$. We denote the set of all semistandard set-valued tableaux of shape λ/μ by $\mathsf{SVT}(\lambda/\mu)$. Similarly, if the maximum entry is restricted to m, the set is denoted by $\mathsf{SVT}^m(\lambda/\mu)$.

The generating function for $\mathsf{SVT}^m(\lambda/\mu)$ is the *Grothendieck polynomials for skew shapes*,

$$G_{\lambda/\mu}(x_1, \ldots, x_m; \beta) = \sum_{S \in \mathsf{SVT}^m(\lambda/\mu)} \beta^{\mathsf{ex}(S)} \boldsymbol{x}^{\mathrm{wt}(T)}. \tag{1.4}$$

For set-valued tableaux, there exists an uncrowding operator, which maps a set-valued tableau to a pair of tableaux of identical shape, one being a semistandard Young tableau and the other a flagged increasing tableau (see for example [**1**, **3**, **18**, **28**]). In this setting, the uncrowding operator intertwines with the crystal operators on set-valued tableaux and semistandard Young tableaux, respectively [**22**].

Consider partitions λ, μ with $\lambda \subseteq \mu$ and $\lambda_1 = \mu_1$. A *flagged increasing tableau* (introduced in [**18**] and called *elegant fillings* by various authors [**1**, **15**, **26**]) is a row and column strict filling of the skew shape μ/λ such that the positive integer entries in the i-th row of the tableau are at most $i-1$ for all $1 \leqslant i \leqslant \ell(\mu)$, 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 call a cell in a set-valued tableau a *multicell* if it contains more than one letter.

Definition 1.2.5. *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*

$$\mathcal{U}_{\mathsf{SVT}} : \mathsf{SVT}(\lambda) \longrightarrow \bigsqcup_{\mu \supseteq \lambda} \mathsf{SSYT}(\mu) \times \mathcal{F}(\mu/\lambda) \tag{1.5}$$

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 $1 \leqslant i \leqslant \ell$, 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 by Buch in [**3**, Section 6] that $\mathcal{U}_{\mathsf{SVT}}$ in (1.5) is a bijection. Monical, Pechenik and Scrimshaw [**22**] proved that $\mathcal{U}_{\mathsf{SVT}}$ intertwines with the crystal operators on set-valued tableaux (see also [**23**]). A similar uncrowding algorithm for multiset-valued tableaux was given in [**11**, Section 3.2].

Example 1.2.2. *Let T be the semistandard set-valued tableau. Perform an uncrowding operation on T to obtain T':*

$$T = \begin{ytableau} 5 \\ 4 & 4 & 5 \\ 2 & 23 & 3 \\ 1 & 1 & 1 & 12 & 234 & 5 \end{ytableau} \to T' = \begin{ytableau} 5 \\ 4 \\ 3 & 4 & 5 \\ 2 & 2 & 3 \\ 1 & 1 & 1 & 12 & 234 & 5 \end{ytableau}.$$

Proceeding with uncrowding the remaining multicells and recording the changes, we have $\mathsf{uncrowd}(T) = (P, Q)$, *where*

$$P \begin{ytableau} 5 & 5 \\ 4 & 4 \\ 3 & 3 & 4 \\ 2 & 2 & 2 & 3 \\ 1 & 1 & 1 & 1 & 2 & 5 \end{ytableau} \quad \textit{and} \quad Q = \begin{ytableau} 3 & 4 \\ *(gray) & 3 \\ *(gray) & *(gray) & *(gray) \\ *(gray) & *(gray) & *(gray) & 1 \\ *(gray) & *(gray) & *(gray) & *(gray) & *(gray) & *(gray) \end{ytableau}.$$

1.2.3. Various definitions of the Grothendieck polynomials. We now introduce two equivalent definitions of the Grothendieck polynomials $\mathfrak{G}_w^{(\beta)}$. The stable Grothendieck polynomial is defined to be its stable limit $G_w^{(\beta)} := \lim_{m\to\infty} \mathfrak{G}_{1^m \times w}^{(\beta)}$.

Definition 1.2.6. *The symmetric group $\mathbb{S}_n$ for $n \geqslant 1$ is generated by the simple transpositions $s_1, s_2, \ldots, s_{n-1}$ subject to the relations*

$$\begin{aligned} s_i s_j &= s_j s_i, && \textit{if } |i-j| > 1,\\ s_i s_{i+1} s_i &= s_{i+1} s_i s_{i+1}, && \textit{for } 1 \leqslant i < n-1,\\ s_i^2 &= 1, && \textit{for } 1 \leqslant i \leqslant n-1. \end{aligned}$$

A reduced expression for an element $w \in \mathbb{S}_n$ is a word $a_1 a_2 \ldots a_\ell$ with $a_i \in [n-1] := \{1, 2, \ldots, n-1\}$ such that

$$w = s_{a_1} \cdots s_{a_\ell} \tag{1.6}$$

and ℓ is minimal among all words satisfying (1.6)*. In this case, ℓ is called the length of w.*

Definition 1.2.7 (Recursive definition, [**16**,**17**]). *Let w_0 be the permutation in $\mathbb{S}_n$ with maximal length. The Grothendieck polynomials are defined recursively by*

(1) $\mathfrak{G}^{(\beta)}_{w_0} = x_1^{n-1}x_2^{n-2}\dots x_{n-1}$*;*

(2) $\mathfrak{G}^{(\beta)}_{w} = \pi_i^{(\beta)}\mathfrak{G}^{(\beta)}_{ws_i}$ *whenever* $\ell(ws_i) = \ell(w)+1$,

where $\pi_i^{(\beta)}$ is the β-divided-difference operator *acting on $\mathbb{C}[x_1, x_2, \dots]$ by*

$$\pi_i^{(\beta)} f(x_1, x_2, \dots) := \frac{(1+\beta x_{i+1}) f(x_1, x_2, \dots) - (1+\beta x_i) f(\dots, x_{i+1}, x_i, \dots)}{x_i - x_{i+1}}.$$

It is not hard to see that $\mathfrak{G}^{(\beta)}_w$'s are well-defined since the $\pi_i^{(\beta)}$ satisfies the relations

$$\pi_i^{(\beta)} \circ \pi_i^{(\beta)} = \pi_i^{(\beta)}, \quad \pi_i^{(\beta)} \circ \pi_{i+1}^{(\beta)} \circ \pi_i^{(\beta)} = \pi_{i+1}^{(\beta)} \circ \pi_i^{(\beta)} \circ \pi_{i+1}^{(\beta)},$$
$$\pi_i^{(\beta)} \circ \pi_j^{(\beta)} = \pi_j^{(\beta)} \circ \pi_i^{(\beta)} \text{ when } |i-j| > 1.$$

Let $\mathcal{A}_n^{(\beta)}$ be the algebra with generators $u_1, u_2, \dots, u_{n-1}$ satisfying the relations

$$u_i^2 = \beta u_i, \quad u_i u_{i+1} u_i = u_{i+1} u_i u_{i+1},$$
$$u_i u_j = u_j u_i, |i-j| > 1.$$

Let $x_1, x_2, \dots, x_{n-1}$ be variables such that $x_i x_j = x_j x_i$ and $x_i u_j = u_j x_i$ for any $1 \leqslant i, j \leqslant n-1$.

Definition 1.2.8 (As coefficients of generating functions, [**8**]). *Define the polynomial*

$$\mathfrak{G}^{(\beta)}(x_1, x_2, \dots, x_{n-1}) = \prod_{j=1}^{n-1} \prod_{i=n-1}^{j} (1 + x_j u_i).$$

Then the Grothendieck polynomial $\mathfrak{G}^{(\beta)}_w(x_1, \dots, x_{n-1})$ is the coefficient of $w \in \mathbb{S}_n$. Here w is identified with $u_{i_1}u_{i_2}\dots u_{i_\ell}$ if $i_1 i_2 \dots i_\ell$ is a reduced expression of w.

Definition 1.2.9 (As coefficients of generating functions, stable version, [**8**]). *Define the polynomial*

$$G^{(\beta)}(x_1, x_2, \dots, x_{n-1}) = \prod_{j=1}^{n-1} \prod_{i=n-1}^{1} (1 + x_j u_i).$$

Then the stable Grothendieck polynomial $G_w^{(\beta)}(x_1,\ldots,x_{n-1})$ is the coefficient of w.

It is not hard to see that this definition is exactly Equation (1.1).

EXAMPLE 1.2.3. *Let $n=3$. We compute the Grothendieck polynomials.*

$$\mathfrak{G}^{(\beta)}(x_1,x_2)=(1+x_1u_2)(1+x_1u_1)(1+x_2u_2)=1+x_1u_1+(x_1+x_2+\beta x_1x_2)u_2+x_1x_2u_1u_2+x_1^2u_2u_1+x_1^2x_2u_2u_1u_2.$$

Thus $\mathfrak{G}_1^{(\beta)}=x_1+x_2+\beta x_1x_2$, $\mathfrak{G}_{21}^{(\beta)}=x_1^2$ and $\mathfrak{G}_{212}^{(\beta)}=x_1^2x_2$. Alternatively we can obtain them via the β-divided-difference operator.

$$\begin{aligned}
\mathfrak{G}_{212}^{(\beta)} &= x_1^2x_2\,,\\
\mathfrak{G}_{21}^{(\beta)} &= \pi_2^{(\beta)}x_1^2x_2=\frac{(1+\beta x_3)x_1^2x_2-(1+\beta x_2)x_1^2x_3}{x_2-x_3}=x_1^2\,,\\
\mathfrak{G}_{2}^{(\beta)} &= \pi_1^{(\beta)}x_1^2=\frac{(1+\beta x_2)x_1^2-(1+\beta x_1)x_2^2}{x_1-x_2}=x_1+x_2+\beta x_1x_2.
\end{aligned}$$

While the stable version can be computed with an extra factor.

$$\begin{aligned}
G^{(\beta)}(x_1,x_2)=&(1+x_1u_2)(1+x_1u_1)(1+x_2u_2)(1+x_2u_1)=1+(x_1+x_2+\beta x_1x_2)u_1+(x_1+x_2+\beta x_1x_2)u_2\\
&+x_1x_2u_1u_2+[x_1^2+x_1x_2+x_2^2+\beta(x_1x_2^2+x_1^2x_2)]u_2u_1+(x_1^2x_2+x_1x_2^2+\beta x_1^2x_2^2)u_2u_1u_2.
\end{aligned}$$

Thus

$$\begin{aligned}
G_1(x_1,x_2) &= x_1+x_2+\beta x_1x_2=s_1(x_1,x_2)+\beta s_{11}(x_1,x_2)\,,\\
G_{21}(x_1,x_2) &= x_1^2+x_1x_2+x_2^2+\beta(x_1x_2^2+x_1^2x_2)=s_2(x_1,x_2)+\beta s_{21}(x_1,x_2)\,,\\
G_{212}(x_1,x_2) &= x_1^2x_2+x_1x_2^2+\beta x_1^2x_2^2=s_{21}(x_1,x_2)+\beta s_{22}(x_1,x_2)\,.
\end{aligned}$$

Note that increasing n will not change $\mathfrak{G}_w$'s but may add more terms to G_w's.

CHAPTER 2

Crystal for stable Grothendieck polynomials

This chapter is based on work in collaboration with Jennifer Morse, Wencin Poh and Anne Schilling published in [**23**].

The chapter is organized as follows. In Section 2.1, we introduce the $\star$-crystal on decreasing factorizations in the 0-Hecke monoid and show that it intertwines with the crystal on semistandard set-valued tableaux [**22**] under the residue map. In Section 2.2, we discuss two insertion algorithms for decreasing factorizations. The first is the Hecke insertion introduced by Buch et al. [**4**] and the second is the new $\star$-insertion. In Section 2.3, properties of the $\star$-insertion are discussed. In particular, we prove that it intertwines with the crystal operators and that it relates to the uncrowding algorithm. We conclude in Section 2.4 with some discussions about the non-fully-commutative case.

2.1. The $\star$-crystal

In this section, we define the K-theoretic generalization of the crystal on decreasing factorizations by Morse and Schilling [**24**] when the associated word is fully-commutative. The underlying combinatorial objects are decreasing factorizations in the 0-Hecke monoid introduced in Section 2.1.1. The $\star$-crystal on these decreasing factorizations is defined in Section 3.1.2. We review the crystal structure on set-valued tableaux introduced by Monical, Pechenik and Scrimshaw [**22**] in Section 2.1.3. The residue map and the proof that it intertwines the $\star$-crystal and the crystal on set-valued tableaux is given in Section 2.1.4.

2.1.1. Decreasing factorizations in the 0-Hecke monoid.

Definition 2.1.1. *The 0-Hecke monoid $\mathcal{H}_0(n)$, where $n \geqslant 1$ is an integer, is the monoid of finite words generated by positive integers in the alphabet $[n-1]$ subject to the relations*

$$\begin{aligned} pq &= qp && \text{if } |p-q|>1, \\ pqp &= qpq && \text{for all } p,q, \\ pp &= p && \text{for all } p. \end{aligned} \tag{2.1}$$

We may form an equivalence relation $\equiv_{\mathcal{H}_0}$ on all words in the alphabet $[n-1]$ based on the relations (2.1). The equivalence classes are infinite since the last relation changes the length of the word. We say that a word $a = a_1 a_2 \ldots a_\ell$ is *reduced* if $\ell \geqslant 0$ is the smallest among all words in $\mathcal{H}_0(n)$ equivalent to a. In this case, ℓ is the length of a. Note that $\mathcal{H}_0(n)$ is in bijection with $\mathbb{S}_n$ by identifying the reduced word $a_1 a_2 \ldots a_\ell$ in $\mathcal{H}_0(n)$ with $s_{a_1} s_{a_2} \cdots s_{a_\ell} \in \mathbb{S}_n$. We say $w \in \mathcal{H}_0(n)$ or $\mathbb{S}_n$ is *fully-commutative* or *321-avoiding* if none of the reduced words equivalent to w contain a consecutive braid subword of the form $i\, i+1\, i$ or $i\, i-1\, i$ for any $i \in [n-1]$.

Remark 2.1.1. *Any (not necessarily reduced) word $w \in \mathcal{H}_0(0)$ containing a consecutive braid subword is not fully-commutative.*

Definition 2.1.2. *A decreasing factorization of $w \in \mathcal{H}_0(n)$ into m factors is a product of the form*

$$\mathbf{h} = h^m \ldots h^2 h^1,$$

where the sequence in each factor

$$h^i = h^i_1 h^i_2 \ldots h^i_{\ell_i}$$

is either empty (meaning $\ell_i = 0$) or strictly decreasing (meaning $h^i_1 > h^i_2 > \cdots > h^i_{\ell_i}$) for each $1 \leqslant i \leqslant m$ and $\mathbf{h} \equiv_{\mathcal{H}_0} w$ in $\mathcal{H}_0(n)$.

The set of all possible decreasing factorizations into m factors is denoted by $\mathcal{H}^m$ or $\mathcal{H}^m(n)$ if we want to indicate the value of n. We call $\mathsf{ex}(\mathbf{h}) = \mathsf{len}(\mathbf{h}) - \ell$ the excess of $\mathbf{h}$, where $\mathsf{len}(\mathbf{h})$ is the number of letters in $\mathbf{h}$ and ℓ is the length of w. We say $\mathbf{h}$ is fully-commutative (or 321-avoiding) if w is fully-commutative.

2.1.2. The $\star$-crystal. Let $\mathcal{H}^{m,\star}$ be the set of fully-commutative decreasing factorizations in $\mathcal{H}^m$. We introduce a type A_{m-1} crystal structure on $\mathcal{H}^{m,\star}$, which we call the *$\star$-crystal*. This generalizes the crystal for Stanley symmetric functions [**24**] (see also [**19**]).

DEFINITION 2.1.3. *For any $\mathbf{h} = h^m \ldots h^2h^1 \in \mathcal{H}^{m,\star}$, we define crystal operators $e_i^\star$ and $f_i^\star$ for $i \in [m-1]$ and a weight function $\mathrm{wt}(\mathbf{h})$. The weight function is determined by the length of the factors*

$$\mathrm{wt}(\mathbf{h}) = (\mathsf{len}(h^1), \mathsf{len}(h^2), \ldots, \mathsf{len}(h^m)).$$

To define the crystal operators $e_i^\star$ and $f_i^\star$, we first describe a pairing process:

- *Start with the largest letter b in h^{i+1}, pair it with the smallest $a \geqslant b$ in h^i. If there is no such a, then b is unpaired.*
- *The pairing proceeds in decreasing order on elements of h^{i+1} and with each iteration, previously paired letters of h^i are ignored.*

If all letters in h^i are paired, then $f_i^\star$ annihilates $\mathbf{h}$. Otherwise, let x be the largest unpaired letter in h^i. The crystal operator $f_i^\star$ acts on $\mathbf{h}$ in either of the following ways:

(1) If $x+1 \in h^i \cap h^{i+1}$, then remove $x+1$ from h^i, add x to h^{i+1}.
(2) Otherwise, remove x from h^i and add x to h^{i+1}.

If all letters in h^{i+1} are paired, then $e_i^\star$ annihilates $\mathbf{h}$. Let y be the smallest unpaired letter in h^{i+1}. The crystal operator $e_i^\star$ acts on $\mathbf{h}$ in either of the following ways:

(1) If $y-1 \in h^i \cap h^{i+1}$, then remove $y-1$ from h^{i+1}, add y to h^i.
(2) Otherwise, remove y from h^{i+1} and add y to h^i.

It is not hard to see that $e_i^\star$ and $f_i^\star$ are partial inverses of each other.

EXAMPLE 2.1.1. *Let $\mathbf{h} = (7532)(621)(6)$, then*

$$f_1^\star(\mathbf{h}) = 0, \qquad e_1^\star(\mathbf{h}) = (7532)(62)(61),$$
$$f_2^\star(\mathbf{h}) = (75321)(61)(6), \qquad e_2^\star(\mathbf{h}) = (753)(6321)(6).$$

REMARK 2.1.2. *Compared to [**24**], one pairs a letter b in h^{i+1} with the smallest letter $a \geqslant b$ in h^i rather than $a > b$.*

Proposition 2.1.1. *Let* $\mathbf{h} = h^m \dots h^1 \in \mathcal{H}^{m,\star}$ *such that* $f_i^\star(\mathbf{h}) \neq 0$. *Then* $f_i^\star(\mathbf{h}) \in \mathcal{H}^{m,\star}$, $f_i^\star(\mathbf{h}) \equiv_{\mathcal{H}_0} \mathbf{h}$, *and* $\mathsf{ex}(f_i^\star(\mathbf{h})) = \mathsf{ex}(\mathbf{h})$. *Furthermore, the* j*-th factor in* $f_i^\star(\mathbf{h})$ *and* $\mathbf{h}$ *agrees for* $j \notin \{i, i+1\}$. *Analogous statements hold for* $e_i^\star$.

Proof. Suppose $\tilde{\mathbf{h}} := f_i^\star(\mathbf{h}) \neq 0$. Then by definition of $f_i^\star$, $\tilde{\mathbf{h}} = h^m \dots h^{i+2}\tilde{h}^{i+1}\tilde{h}^i h^{i-1} \dots h^1$ and h^j is unchanged for $j \notin \{i, i+1\}$. In addition, the number of factors does not change.

To see $\mathbf{h} \equiv_{\mathcal{H}_0} \tilde{\mathbf{h}}$, it suffices to show that $h^{i+1}h^i \equiv_{\mathcal{H}_0} \tilde{h}^{i+1}\tilde{h}^i$. Let x be the largest unpaired letter in h^i. By the bracketing procedure this implies that $x \notin h^{i+1}$. We can write h^{i+1} as $w_1 w_2$, where w_1 is a word containing only letters greater than x, and w_2 is a word containing only letters smaller than x. We can write h^i as $w_3 x w_4$, where w_3 contains only letters greater than x and w_4 contains only letters smaller than x.

The pairing process will result in one of the two following cases:

(1) If $x+1 \in h^i \cap h^{i+1}$, then obtain $\tilde{h}^i$ by removing $x+1$ from h^i, and $\tilde{h}^{i+1}$ by adding x to h^{i+1}.
(2) Otherwise, obtain $\tilde{h}^i$ by removing x from h^i and obtain $\tilde{h}^{i+1}$ by adding x to h^{i+1}.

We first argue that in either case we must have $x-1 \notin w_2$. Assume $x-1 \in w_2$ and let k be the largest number such that the interval $[x-k, x-1] \subseteq w_2$. By assumption $k \geqslant 1$. In order for x to be the largest unpaired letter in h^i, $[x-k, x-1]$ must be contained in w_4. We can write $w_2 = (x-1)\dots(x-k)w_2'$ and $w_4 = (x-1)\dots(x-k)w_4'$, where all letters in w_2' are smaller than $x-k-1$. When $k=1$, we have the following subword

$$(x-1)w_2'w_3x(x-1) \equiv_{\mathcal{H}_0} w_2'w_3(x-1)x(x-1),$$

which contains a braid $(x-1)x(x-1)$. When $k>1$, we also have the following subword

$$(x-k)w_2'w_3x(x-1)\dots(x-k+1)(x-k) \equiv_{\mathcal{H}_0} w_2'w_3(x-1)\dots(x-k+2)(x-k)(x-k+1)(x-k),$$

which also contains a braid.

Case (1): Let k be the largest letter such that $[x+1, x+k] \subseteq w_3$. Clearly $k \geqslant 1$. Suppose $k>1$, then we can write $w_3 = w_3'(x+k)\dots(x+1)$. Since x is the largest unpaired letter in h^i, everything in $[x+1, x+k] \subseteq w_3$ must be paired. The letter $x+1$ in w_3 is paired with $x+1 \in w_1$, which implies that $x+i$ in w_3 is paired with $x+i \in w_1$ for all $1 \leqslant i \leqslant k$. This implies that $[x+1, x+k] \subseteq w_1$. Then we have the following subword

$$(x+1)w_2w_3'(x+k)\dots(x+2)(x+1) \equiv_{\mathcal{H}_0} w_2w_3'(x+k)\dots(x+1)(x+2)(x+1)$$

which contains a braid. Thus, we must have $k=1$, which implies that $x+2 \notin w_3$. Write $w_1 = w_1'(x+1)$. Then by direct computation

$$\begin{aligned} h^{i+1}h^i &\equiv_{\mathcal{H}_0} w_1'(x+1)w_2w_3'(x+1)xw_4 \equiv_{\mathcal{H}_0} w_1'(x+1)(x+1)w_2w_3'xw_4 \\ &\equiv_{\mathcal{H}_0} w_1'(x+1)w_2w_3'xxw_4 \equiv_{\mathcal{H}_0} \left(w_1'(x+1)xw_2\right)\left(w_3'xw_4\right) = \tilde{h}^{i+1}\tilde{h}^i. \end{aligned}$$

Case (2): We claim that if $x+1 \notin h^{i+1}$, then $x+1 \notin h^i$. Otherwise the $x+1 \in h^i$ must be paired with some $z \in h^{i+1}$, so we have $z \leqslant x+1$. But x is unpaired, which implies $z > x$, that gives us a contradiction. Hence $x+1 \notin w_3$. Recall that $x-1 \notin w_2$. Therefore, by a straightforward computation

$$h^{i+1}h^i = w_1w_2w_3xw_4 \equiv_{\mathcal{H}_0} (w_1xw_2)(w_3w_4) \equiv_{\mathcal{H}_0} \tilde{h}^{i+1}\tilde{h}^i.$$

The above arguments show that $h^{i+1}h^i \equiv_{\mathcal{H}_0} \tilde{h}^{i+1}\tilde{h}^i$, thus $\mathbf{h} \equiv_{\mathcal{H}_0} \tilde{\mathbf{h}}$, and the total length of the decreasing factorization are unchanged under $f_i^\star$. Furthermore, the excess remains unchanged under $f_i^\star$.

Similar arguments hold for $e_i^\star$. □

Remark 2.1.3. *Here we summarize several results from the proof that will be needed later. Namely, if x is the largest unpaired letter in h^i, then*

- $x-1 \notin h^{i+1}$.
- *One and only one of the three statements hold: $x+1 \in h^{i+1} \cap h^i$, $x+1 \notin h^{i+1} \cup h^i$, and $x+1 \in h^{i+1}, x+1 \notin h^i$.*

It will be shown in Section 2.1.4 that $\mathcal{H}^{m,\star}$ is indeed a Stembridge crystal of type A_{m-1} (for an introduction to crystal and terminology, see [**5**]).

2.1.3. The crystal on set-valued tableaux. We now review the crystal structure on semistandard set-valued tableaux given in [**22**]. We state the definition on skew shapes rather than just straight shapes.

Definition 2.1.4. *Let $T \in \mathsf{SVT}^m(\lambda/\mu)$. We employ the following pairing rule for letters i and $i+1$. Assign $-$ to every column of T containing an i but not an $i+1$. Similarly, assign $+$ to every column of T containing an $i+1$ but not an i. Then, successively pair each $+$ that is to the left of and adjacent to a $-$, removing all paired signs until nothing can be paired.*

The operator f_i changes the i in the rightmost column with an unpaired $-$ (if this exists) to $i+1$, except if the cell b containing that i has a cell to its right, denoted $b^{\rightarrow}$, that contains both i and $i+1$. In that case, f_i removes i from $b^{\rightarrow}$ and adds $i+1$ to b. Finally, if no unpaired $-$ exists, then f_i annihilates T.

Similarly, the operator e_i changes the $i+1$ in the leftmost column with an unpaired $+$ (if this exists) to i, except if the cell b containing that $i+1$ has a cell to its left, denoted $b^{\leftarrow}$, that contains both i and $i+1$. In that case, e_i removes $i+1$ from $b^{\leftarrow}$ and adds i to b. Finally, if no unpaired $+$ exists, then e_i annihilates T.

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

One can easily show that the crystal on $\mathsf{SVT}^m(\lambda/\mu)$ of Definition 2.1.4 defines a seminormal crystal (for definitions see [**5**]). It was proved in [**22**, Theorem 3.9] that the above described operators e_i and f_i define a type A_{m-1} Stembridge crystal structure on $\mathsf{SVT}^m(\lambda)$. We claim that their proof goes through also for skew shapes.

Theorem 2.1.1. *The crystal $\mathsf{SVT}^m(\lambda/\mu)$ of Definition 2.1.4 is a Stembridge crystal of type A_{m-1}.*

Proof. Since the proof is exactly the same as in [**22**, Theorem 3.9], we just state the outline and give a brief description. For details we refer to [**22**].

First note that the signature rule given by column-reading is compatible with the signature rule given by row-reading (top to bottom, left to right, and arrange the letters in the same cell by descending order) by semistandardness. Hence we may consider the crystal to live inside the tensor product of its rows. A single-row semistandard set-valued tableaux of a fixed shape is isomorphic to a Stembridge crystal, as shown in [**22**, Proposition 3.5]:

$$\Phi_s \colon \mathsf{SVT}^m(s\Lambda_1) \to \bigoplus_{k=1}^{m} B((s-1)\Lambda_1 + \Lambda_k),$$

where Λ_k are the fundamental weights of type A_{m-1}.

Let $\lambda = (\lambda_1, \ldots, \lambda_\ell)$ and $\mu = (\mu_1, \ldots, \mu_\ell)$ (the last couple μ_i could be zero) be two partitions such that $\mu \subseteq \lambda$. Construct the map below, which is a strict crystal embedding:

$$\Psi \colon \mathsf{SVT}^m(\lambda/\mu) \to \mathsf{SVT}^m((\lambda_1-\mu_1)\Lambda_1) \otimes \mathsf{SVT}^m((\lambda_2-\mu_2)\Lambda_1) \otimes \cdots \otimes \mathsf{SVT}^m((\lambda_\ell-\mu_\ell)\Lambda_1).$$

Thus, we have a strict crystal embedding:

$$(\Phi_{\lambda_1-\mu_1} \oplus \cdots \oplus \Phi_{\lambda_\ell-\mu_\ell}) \circ \Psi \colon \mathsf{SVT}^m(\lambda/\mu) \to \bigotimes_{j=1}^{\ell} \left(\bigoplus_{k=1}^{m} B((\lambda_j-\mu_j)\Lambda_1 + \Lambda_k) \right).$$

Since $\mathsf{SVT}^m(\lambda/\mu)$ is a seminormal crystal, we can conclude that it is a Stembridge crystal. □

2.1.4. The residue map. In this section, we define the residue map from set-valued tableaux of skew shape to fully-commutative decreasing factorizations in the 0-Hecke monoid. We then show in Theorem 2.1.2 that the residue map intertwines with the crystal operators, proving that $\mathcal{H}^{m,\star}$ is indeed a crystal of type A_{m-1} (see Corollary 2.1.1).

Definition 2.1.5. *Given $T \in \mathsf{SVT}^m(\lambda/\mu)$, we define the residue map $\mathsf{res} : \mathsf{SVT}^m(\lambda/\mu) \to \mathcal{H}^m$ as follows. Associate to each cell (i,j) in λ/μ its content $\ell(\lambda)+j-i$, where $\ell(\lambda)$ is the number of parts in λ. Produce a decreasing factorization $\mathbf{h} = h^m h^{m-1} \ldots h^2 h^1$ by declaring h^k to be the (possibly empty) sequence formed by taking the contents of all cells in T containing the entry k and then arranging the contents in decreasing order. This defines $\mathsf{res}(T) := \mathbf{h}$.*

Example 2.1.2. *Let T be the set-valued tableau of skew shape $(2,2)/(1)$*

$$T = \begin{array}{|c|c|} \hline 23 & 3 \\ \hline \multicolumn{1}{c|}{} & 12 \\ \cline{2-2} \end{array}.$$

The content of each cell in T is denoted by a subscript as follows:

$$\begin{array}{|c|c|}\hline 23_1 & 3_2 \\ \hline \multicolumn{1}{c|}{} & 12_3 \\ \cline{2-2}\end{array}\,.$$

To read off the third factor, we search for all cells with an entry 3; these cells have contents 1 and 2, so we have 21 in the third factor. Altogether, we obtain $\mathsf{res}(T) = (21)(31)(3) \in \mathcal{H}^3$.

The image of the residue map res is $\mathcal{H}^{m,\star}$, the set of fully-commutative decreasing factorizations into m factors. In fact, res is a bijection from semistandard set-valued skew tableaux on the alphabet $[m]$ to $\mathcal{H}^{m,\star}$ up to shifts in the skew shape.

For this purpose, let us describe the inverse of the residue map. Let $\mathbf{h} = h^m h^{m-1} \dots h^2 h^1 \in \mathcal{H}^{m,\star}$. Begin by filling the diagonals of content that appear in h^m by the entry m. As the resulting T is supposed to be of skew shape, the cells containing m along increasing diagonals need to go weakly down from left to right. If these diagonals are consecutive, then the cells have to be in the same row of T since T is semistandard. Continue the procedure above by putting entry i into the diagonals specified by h^i for all $i = m-1, m-2, \dots, 1$, applying the condition that the resulting filling should be semistandard.

Proposition 2.1.2. *If* $\mathbf{h} = h^m h^{m-1} \dots h^2 h^1 \in \mathcal{H}^{m,\star}$*, then the above algorithm is well-defined up to shifts along diagonals. It produces a skew semistandard set-valued tableau T such that* $\mathsf{res}(T) = \mathbf{h}$.

Proof. We shall show more generally that at any given stage in the algorithm for the inverse of the residue map above, the tableau T produced is of skew shape if and only if $\mathbf{h}$ is fully-commutative.

Assume that T is not of skew shape. Consider the earliest stage in the algorithm when the produced tableau is not of skew shape. Then, either one of the following cases must have occurred for the first time.

Case 1: There are adjacent cells with nonempty sets A and B (where $\max(A) \leqslant \min(B)$) in the same row on diagonals i and $i+1$ respectively with no cells appearing directly below these cells, as illustrated on the left side of Figure 2.1. Moreover, by minimality, we have an integer x with the following properties:

(1) $i+1 \in h^x$ and $x < \min(A)$,

(2) there does not exist a y with $x \leqslant y < \min(B)$ and $i+2 \in h^y$.

By applying semistandardness, a cell containing x is created directly below the cell containing the set A as in the right side of Figure 2.1. Furthermore, by (2), for all $x \leqslant y < \min(B)$, we have that every letter in h^y is either at most $i+1$ or at least $i+3$. It follows that, after possibly applying commutativity ($i+1$ with letters at most $i-1$ or at least $i+3$) and the idempotent relation, $h^{\min(B)} \dots h^{x+1} h^x$ is equivalent to one containing the braid subword $i+1\ i\ i+1$. This implies that $\mathbf{h}$ is equivalent to a Hecke word containing the same braid subword.

Case 2: There are adjacent cells with nonempty sets A and B in the same column on diagonals $i+1$ and i respectively with no cells appearing directly to the left of these cells, as illustrated on the left side of Figure 2.2. Moreover, by minimality, we have an integer x with the following properties:

(1) $i \in h^x$ and $x \leqslant \min(A)$,

(2) there does not exist a y with $x < y \leqslant \min(B)$ and $i-1 \in h^y$.

By applying semistandardness, a cell containing x is created directly to the left of the cell containing the set A as in the right side of Figure 2.2. Furthermore, by (2), for all $x < y \leqslant \min(B)$, we have that every letter in h^y is either at most $i-2$ or at least i. Similar to the argument in Case 1, $h^{\min(B)} \dots h^{x+1} h^x$ is equivalent to one containing the braid subword $i\ i+1\ i$. This implies that $\mathbf{h}$ is equivalent to a word in $\mathcal{H}_0(n)$ containing the same braid subword.

The above arguments imply that the image of res is contained in $\mathcal{H}^{m,\star}$. Conversely, if $\mathbf{h}$ is fully-commutative, then the algorithm for res^{-1} does not produce Case 1 or Case 2 above and hence the resulting tableau T is of skew shape which in turn implies that the algorithm is well-defined (up to shifts along the diagonal if a gap of size at least 3 occurs in the labels). $\square$

If the skew shape λ/μ of the tableau T is known, then one may simplify the procedure above noting that the filling of i specified by letters in h^i must occur along a horizontal strip for all $i = m, m-1, \dots, 1$. In this case, the recovered tableau T is unique and there is no shift ambiguity if a gap of size at least 3 occurs in the labels.

Example 2.1.3. *Let* $\mathbf{h} = (61)(752)(75)(762)$ *be a decreasing factorization of* $w = 651762$.

In the algorithm for the inverse of the residue map, the entry 4 is placed on diagonal 1 and 6, respectively. Due to semistandardness, the entry 3 in diagonal 2 must be placed below the 4 in

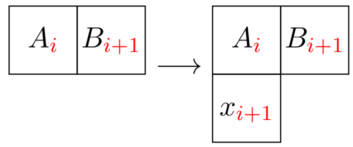


FIGURE 2.1. A forbidden case while inverting the residue map.

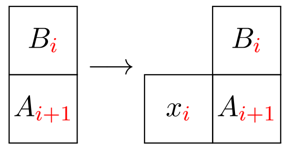
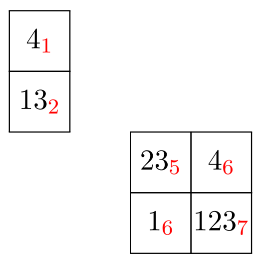
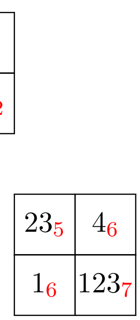


FIGURE 2.2. Another forbidden case while inverting the residue map.

diagonal 1, while the 3's in diagonals 5 and 7 are respectively to the left and below the 4 in diagonal 6. Continuing with the remaining fillings, we have two possibilities:

$T_1 =$  ,

or

$T_2 =$  ,

where $T_1 \in \mathsf{SVT}^4((4,4,1,1)/(2,2))$ *and* $T_2 \in \mathsf{SVT}^4((3,3,1,1,1)/(1,1,1))$. *Note that they indeed just differ by a shift along diagonals as stated in Proposition 2.1.2.*

EXAMPLE 2.1.4. *Let* $\mathbf{h} = (8431)(863)(8654)(941)$ *be a decreasing factorization of* $w = 84396541$. *Suppose that* $\mathbf{h} = \mathsf{res}(T)$, *where* $T \in \mathsf{SVT}^4(\lambda/\mu)$ *with* $\lambda/\mu = (5,5,4,2,1)/(4,4,1,1)$.

Then, we fill in 4 *along the diagonals with labels 1, 3, 4, 8 respectively, noting that the* 4 *in diagonal 4 is to the right of the* 4 *in diagonal* 3 *(due to the semistandardness of* T*). Continuing* *with the remaining fillings, we have*

$$T = \begin{array}{ccccc}
\boxed{14_1} & & & & \\
 & \boxed{34_3} & \boxed{4_4} & & \\
 & \boxed{12_4} & \boxed{2_5} & \boxed{23_6} & \\
 & & & & \boxed{234_8} \\
 & & & & \boxed{1_9}
\end{array}.$$

THEOREM 2.1.2. *The crystal on set-valued tableaux* $\mathsf{SVT}^m(\lambda/\mu)$ *and the crystal on decreasing factorizations* $\mathcal{H}^{m,\star}$ *intertwine under the residue map. That is, the following diagrams commute:*

$$\begin{tikzcd}
\mathsf{SVT}^m(\lambda/\mu) \arrow[r, "\mathsf{res}"] \arrow[d, "f_k"] & \mathcal{H}^{m,\star} \arrow[d, "f_k^\star"] \\
\mathsf{SVT}^m(\lambda/\mu) \arrow[r, "\mathsf{res}"] & \mathcal{H}^{m,\star}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathsf{SVT}^m(\lambda/\mu) \arrow[r, "\mathsf{res}"] \arrow[d, "e_k"] & \mathcal{H}^{m,\star} \arrow[d, "e_k^\star"] \\
\mathsf{SVT}^m(\lambda/\mu) \arrow[r, "\mathsf{res}"] & \mathcal{H}^{m,\star}.
\end{tikzcd}$$

PROOF. Let $T \in \mathsf{SVT}^m(\lambda/\mu)$, $\mathbf{h} = \mathsf{res}(T)$ and $\ell = \ell(\lambda)$. We prove the following three statements associated to $f_k(T)$ and $f_k^\star(\mathbf{h})$.

(1) We claim that if there is no unpaired k in T, then $f_k^\star$ annihilates $\mathbf{h}$. Furthermore, if the rightmost unpaired k in cell b of T has content x, then x is also the largest unpaired letter in h^k.

For the proof of (1) it suffices to notice that the signature rule on tableaux is equivalent to the pairing process for decreasing factorizations of $\mathcal{H}_0(n)$. We rephrase the pairing procedure for decreasing factorizations on tableaux:

- At the beginning, no letter is paired.
- Then start with the rightmost column and work westward.
- Successively, for each $k+1$, compute its content a, then pair it with the k of smallest content weakly greater than a that is yet unpaired.

Next, we argue that the signature rule yields the same result on the rightmost unpaired letter. Assume we are looking at cell b containing the current $k+1$ with content a.

Case (a): Suppose there is no unpaired k with content a but at least one unpaired k with strictly greater content(s). Then pair it with the current $k+1$. This is the direct signature rule.

Case (b): Suppose there is no unpaired k with content weakly greater than a, then this $k+1$ is unpaired. This is also the direct signature rule.

Case (c): Suppose there is an unpaired k with content a. Then it must be either in the same cell b, or one row below and one column to the left of b on the diagonal labeled a. If they are in the same cell, then the pairing is the direct signature rule.

Otherwise, there must be cells to the left and below b since the shape is skew. Suppose cell b is in row r. Consider the rightmost entry in cell (r,j) in row r containing a $k+1$, and the leftmost entry in cell $(r-1,q)$ in row $r-1$ containing a k. Considering this as the first of a consecutive occurrence, cell b is cell (r,j), so we have $\ell+j-r=a$. By semistandardness and the condition that the shape is skew, we can partially fill out the involved subtableau of T for rows $r-1,r$ from column q to j:

$k+1\ _{\ell+q-r}$	$k+1\ _{\ell+q+1-r}$	$\dots$	$k+1\ _{\ell+j-1-r}$	$k+1\dots_{\ell+j-r}$
$\dots k\ _{\ell+q-r+1}$	$k\ _{\ell+q-r}$	$\dots$	$k\ _{\ell+j-r}$	$k\ _{\ell+j-r+1}$

.

All the cells (s,t) with $q<t<j$ and $s\in\{r,r-1\}$ and the cells (r,q) and $(r-1,j)$ are single-valued by semistandardness as shown in the above figure.

From the $k+1$ in (r,j), we start the pairing process. First, we claim that the k in cell $(r-1,j)$ must be unpaired at this point. Suppose that there is a $k+1$ to the east of cell (r,j) with content smaller or equal to $\ell+j-r+1$, then it must be cell $(r,j+1)$, which violates that (r,j) is the rightmost cell in row r containing a $k+1$. Then the pairing says the $k+1$ in cell (r,t) pairs with the k in cell $(r-1,t-1)$ for $q<t\leqslant j$. Lastly, the $k+1$ in cell (r,q) has to pair with the previously unpaired k in cell $(r-1,j)$ since there are no unpaired k with label greater or equal to $\ell+q-r$ and smaller than $\ell+j-r+1$.

Although the pairing is different than the usual signature rule pairing, which pairs $k+1,k$ in the same column, the $2(j-q+1)$ letters end up being paired. Since it will not influence which one will be the rightmost unpaired letter, it is still equivalent to the signature rule.

So in any case, the pairing is equivalent to the signature rule. Thus, the rightmost unpaired k in T corresponds to the largest unpaired letter in h^k.

(2) We claim that if f_k changes the rightmost unpaired k in T to a $k+1$ (with content x) without moving it, then $f_k^\star$ moves a letter x from h^k to h^{k+1}.

Since f_k does not need to move any letter, it means the cell to the right of b, denoted by $b^\rightarrow$, does not contain a k. It is the only cell with content $x+1$ that could contain a k. This implies that $x+1 \notin h^k$. By Definition 2.1.3, $f_k^\star$ moves x from h^k to h^{k+1}.

(3) We claim the following. If f_k changes a k from $b^\rightarrow$ into a $k+1$ and moves to cell b, then $f_k^\star$ removes an $x+1$ from h^k and changes it to an x in h^{k+1}.

That f_k needs to move a number means that k and $k+1$ are in $b^\rightarrow$, which implies that $x+1 \in h^k \cap h^{k+1}$. By Definition 2.1.3, $f_k^\star$ removes the $x+1$ from h^k and adds an x to h^{k+1}.

We have proved the three statements and they complete the proof that f_k and $f_k^\star$ intertwine under the residue map. The proof is similar for e_k and $e_k^\star$. $\square$

Corollary 2.1.1. *The set $\mathcal{H}^{m,\star}$, together with crystal operators $e_i^\star$ and $f_i^\star$ for $1 \leqslant i < m$ and weight function* wt *defined in Definition 2.1.3, is a Stembridge crystal.*

Proof. By Theorem 2.1.2 and the fact that the residue map preserves the weight and is invertible, this follows from the fact that $\mathsf{SVT}^m(\lambda/\mu)$ is a Stembridge crystal proven in [**22**, Theorem 3.9] (see also Theorem 2.1.1). $\square$

Example 2.1.5. *Consider the tableau T (with labels in red) given by*

$$T = \begin{array}{|c|c} \hline 3_{\color{red}1} & \\ \hline 1_{\color{red}2} & \multicolumn{1}{c|}{123_{\color{red}3}} \\ \hline \end{array},$$

with $\mathsf{res}(T) = (31)(3)(32)$.

For the crystal operators on set-valued tableaux we obtain

$$f_1(T) = \begin{array}{|c|c} \hline 3_{\color{red}1} & \\ \hline 12_{\color{red}2} & \multicolumn{1}{c|}{23_{\color{red}3}} \\ \hline \end{array},$$

with $\mathsf{res}\,(f_1(T)) = (31)(32)(2)$*. Then it can be easily checked that the following diagram commutes:*

$$\begin{tikzcd}
T = \begin{array}{|c|c|} \cline{1-1} 3_1 & \multicolumn{1}{c}{} \\ \hline 1_2 & 123_3 \\ \hline \end{array} \arrow[r, "\mathsf{res}"] \arrow[d, "f_1"] & (31)(3)(32) \arrow[d, "f_1^\star"] \\
f_1(T) = \begin{array}{|c|c|} \cline{1-1} 3_1 & \multicolumn{1}{c}{} \\ \hline 12_2 & 23_3 \\ \hline \end{array} \arrow[r, "\mathsf{res}"] & (31)(32)(2).
\end{tikzcd}$$

2.2. Insertion algorithms

In this section, we discuss two insertion algorithms for decreasing factorizations in $\mathcal{H}^m$ (resp. $\mathcal{H}^{m,\star}$). The first is the Hecke insertion introduced by Buch et al. [**4**], which we review in Section 2.2.1. We prove a relationship between Hecke insertion and the residue map (see Theorem 2.2.1). In particular, this proves [**22**, Open Problem 5.8] for fully-commutative permutations. The second insertion is a new insertion, which we call $\star$-insertion, introduced in Section 2.2.2. It goes from fully-commutative decreasing factorizations in the 0-Hecke monoid to pairs of (transposes of) semistandard tableaux of the same shape and is well-behaved with respect to the crystal operators.

2.2.1. Hecke insertion.

Hecke insertion was first introduced in [**4**] as column insertion. Here we state the row insertion version as in [**27**]. In this section, we represent a decreasing factorization $\mathbf{h} = h^m h^{m-1} \cdots h^1$, where $h^i = h^i_1 h^i_2 \ldots h^i_{\ell_i}$, by a *decreasing Hecke biword*

$$\begin{bmatrix} \mathbf{k} \\ \mathbf{h} \end{bmatrix} = \begin{bmatrix} m & \ldots & m & \ldots & 1 & \ldots & 1 \\ h^m_1 & \ldots & h^m_{\ell_m} & \ldots & h^1_1 & \ldots & h^1_{\ell_1} \end{bmatrix}.$$

In addition, we say that $[\mathbf{k}, \mathbf{h}]^t$ is *fully-commutative* if $\mathbf{h}$ is fully-commutative.

Example 2.2.1. *Consider the decreasing factorization* $\mathbf{h} = (1)(2)(31)(\ )(32)$. *Then the corresponding biword* $[\mathbf{k}, \mathbf{h}]^t$ *is*

$$\begin{bmatrix} \mathbf{k} \\ \mathbf{h} \end{bmatrix} = \begin{bmatrix} 5 & 4 & 3 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 & 3 & 2 \end{bmatrix}.$$

Definition 2.2.1. *Starting with a decreasing Hecke biword* $[\mathbf{k}, \mathbf{h}]^t$, *we define* Hecke row insertion *from the right. The insertion sequence is read from right to left. Suppose there are* n *columns in* $[\mathbf{k}, \mathbf{h}]^t$.

Start the insertion with (P_0, Q_0) *being both empty tableaux. We recursively construct* (P_{i+1}, Q_{i+1}) *from* (P_i, Q_i). *Suppose the* $(n-i)$*-th column in* $[\mathbf{k}, \mathbf{h}]^t$ *is* $[y, x]^t$.

We describe how to insert x *into* P_i, *denoted* $P_i \leftarrow x$, *by describing how to insert* x *into a row* R. *The insertion may modify the row and may produce an output integer, which will be inserted into the next row. First, we insert* x *into the first row* R *of* P_i *following the rules below:*

(1) If $x \geqslant z$ *for all* $z \in R$, *the insertion terminates in either of the following ways:*
   (a) If we can append x *to the right of* R *and obtain an increasing tableau, the result* P_{i+1} *is obtained by doing so; form* Q_{i+1} *by adding a box with* y *in the same position where* x *is added to* P_i.
   (b) Otherwise row R *remains unchanged. Form* Q_{i+1} *by adding* y *to the existing corner of* Q_i *whose column contains the rightmost box of row* R.

(2) Otherwise, there exists a smallest z *in* R *such that* $z > x$.
   (a) If replacing z *with* x *results in an increasing tableau, then do so. Let* z *be the output integer to be inserted into the next row.*
   (b) Otherwise, row R *remains unchanged. Let* z *be the output integer to be inserted into the next row.*

The entire Hecke insertion terminates at (P_n, Q_n) *after we have inserted every letter from the Hecke biword. The resulting insertion tableau* P_n *is an increasing tableau, meaning that both rows and columns of* P_n *are strictly increasing. If* $\mathbf{k} = (n, n-1, \ldots, 1)$, *the recording tableau* Q_n *is a standard set-valued tableau.*

Example 2.2.2. *Take $[\mathbf{k},\mathbf{h}]^t$ from Example 2.2.1. Following the Hecke row insertion, we compute its insertion tableau and recording tableau:*

$$\emptyset \to \begin{ytableau} 2 \end{ytableau} \to \begin{ytableau} 2 & 3 \end{ytableau} \to \begin{ytableau} 2 & \none \\ 1 & 3 \end{ytableau} \to \begin{ytableau} 2 & \none \\ 1 & 3 \end{ytableau} \to \begin{ytableau} 2 & 3 \\ 1 & 2 \end{ytableau} \to \begin{ytableau} 3 & \none \\ 2 & 3 \\ 1 & 2 \end{ytableau} = P,$$

$$\emptyset \to \begin{ytableau} 1 \end{ytableau} \to \begin{ytableau} 1 & 1 \end{ytableau} \to \begin{ytableau} 3 & \none \\ 1 & 1 \end{ytableau} \to \begin{ytableau} 3 & \none \\ 1 & 13 \end{ytableau} \to \begin{ytableau} 3 & 4 \\ 1 & 13 \end{ytableau} \to \begin{ytableau} 5 & \none \\ 3 & 4 \\ 1 & 13 \end{ytableau} = Q.$$

Example 2.2.3. *Note that the recording tableau for the Hecke insertion of Definition 2.2.1 is not always a semistandard set-valued tableau. For example, for $\mathbf{h} = (21)(41)$ we have*

$$\begin{bmatrix} \mathbf{k} \\ \mathbf{h} \end{bmatrix} = \begin{bmatrix} 2 & 2 & 1 & 1 \\ 2 & 1 & 4 & 1 \end{bmatrix}$$

and

$$P = \begin{ytableau} 4 & \none \\ 1 & 2 \end{ytableau} \qquad \text{and} \qquad Q = \begin{ytableau} 22 & \none \\ 1 & 1 \end{ytableau}.$$

However, in Theorem 2.2.1 below we will see that in certain cases it is.

Theorem 2.2.1. *Let $T \in \mathsf{SVT}(\lambda)$ and $[\mathbf{k},\mathbf{h}]^t = \mathsf{res}(T)$. Apply Hecke row insertion from the right on $[\mathbf{k},\mathbf{h}]^t$ to obtain the pair of tableaux (P, Q). Then $Q = T$.*

Remark 2.2.1. *Combining Theorems 2.2.1 and 2.1.2 shows that Hecke insertion from right to left (as opposed to left to right in [**27**]) intertwines the crystal on set-valued tableaux and the $\star$-crystal, even though in general it is not always well-defined (see Example 2.2.3). This resolves [**22**, Open Problem 5.8] when the decreasing factorizations are fully-commutative. Even when $\mathbf{h}$ is fully-commutative, but does not correspond to a straight-shaped tableau under res^{-1} as in Example 2.2.3, one can fill the skew part with small enough numbers and apply the Hecke insertion on this tableau. In the above example*

$$\begin{bmatrix} \mathbf{k} \\ \mathbf{h} \end{bmatrix} = \begin{bmatrix} 2 & 2 & 1 & 1 & 0 & 0 \\ 2 & 1 & 4 & 1 & 3 & 2 \end{bmatrix} \qquad \text{with} \qquad Q = T = \begin{ytableau} 12 & 2 & \none \\ 0 & 0 & 1 \end{ytableau}.$$

Note, however, that unlike in [22] we use row Hecke insertion from right to left rather than column insertion from left to right (in analogy to [24] for Edelman–Greene insertion).

Since $k \in T(i,j)$ if and only if $\ell + j - i \in h^k$ under the residue map, where $\ell = \ell(\lambda)$ and h^k is the k-th factor of $\mathbf{h}$, the statement of Theorem 2.2.1 is equivalent to applying Hecke insertion on the entries of T sorted first by ascending order of entries, followed by ascending diagonal content.

Example 2.2.4. *Let T be the semistandard set-valued tableau*

$$T = \begin{array}{|c|c|}\hline 2_1 & 4_2 \\ \hline 1_2 & 23_3 \\ \hline \end{array}\,.$$

The insertion sequence by entry is listed in the table below:

Cell	*(1,1)*	*(2,1)*	*(1,2)*	*(1,2)*	*(2,2)*
Content	*2*	*1*	*3*	*3*	*2*
Entry	*1*	*2*	*2*	*3*	*4*

We will prove Theorem 2.2.1 by induction by considering all subtableaux of T, obtained by adding the entries in T one by one in the order above:

$$\emptyset \to \begin{array}{|c|}\hline 1_2 \\ \hline \end{array} \to \begin{array}{|c|}\hline 2_1 \\ \hline 1_2 \\ \hline \end{array} \to \begin{array}{|c|c}\cline{1-1} 2_1 & \\ \hline 1_2 & 2_3 \\ \hline \end{array} \to \begin{array}{|c|c}\cline{1-1} 2_1 & \\ \hline 1_2 & 23_3 \\ \hline \end{array} \to \begin{array}{|c|c|}\hline 2_1 & 4_2 \\ \hline 1_2 & 23_3 \\ \hline \end{array} = T.$$

In addition, the corresponding sequence of insertion tableaux and recording tableaux is listed here:

$$\emptyset \to \begin{array}{|c|}\hline 2 \\ \hline \end{array} \to \begin{array}{|c|}\hline 2 \\ \hline 1 \\ \hline \end{array} \to \begin{array}{|c|c}\cline{1-1} 2 & \\ \hline 1 & 3 \\ \hline \end{array} \to \begin{array}{|c|c}\cline{1-1} 2 & \\ \hline 1 & 3 \\ \hline \end{array} \to \begin{array}{|c|c|}\hline 2 & 3 \\ \hline 1 & 2 \\ \hline \end{array} = P.$$

$$\emptyset \to \begin{array}{|c|}\hline 1 \\ \hline \end{array} \to \begin{array}{|c|}\hline 2 \\ \hline 1 \\ \hline \end{array} \to \begin{array}{|c|c}\cline{1-1} 2 & \\ \hline 1 & 2 \\ \hline \end{array} \to \begin{array}{|c|c}\cline{1-1} 2 & \\ \hline 1 & 23 \\ \hline \end{array} \to \begin{array}{|c|c|}\hline 2 & 4 \\ \hline 1 & 23 \\ \hline \end{array} = Q.$$

Proof of Theorem 2.2.1. We prove the theorem by proving the following more specific statement.

For a given step in the insertion process, suppose that the entries of T that are involved so far form a nonempty subtableau T' of T with shape μ containing cell $(1,1)$, and the insertion tableau and recording tableau at the corresponding step are $P(T')$ and $Q(T')$. Then, they both have shape μ, and the entry of cell (i,j) of $P(T')$ is $\ell + j - \mu'_j + i - 1$, and $Q(T') = T'$, where μ' is the transpose of the partition μ and $\ell := \lambda'_1 = \ell(\lambda)$.

We prove this by induction on subtableaux of T.

Base step: Suppose T' only contains a single cell $(1,1)$ and $T'(1,1) = S$, where S is a subset of $T(1,1)$ with cardinality d. Then $P(T')$ is obtained by inserting d times the number ℓ. So we have $P(T') = \boxed{\ell}$ and $Q(T') = T'$. Here $\mu = (1)$, so for $(i,j) = (1,1)$, we have $\ell + j - \mu'_j + i - 1 = \ell$.

Inductive step: Suppose that the statements hold for some subtableau T' of shape μ. Assume the next insertion step involves adding the entry k in cell (p,q) of T to T' to obtain T''. There are two cases: (1) the cell (p,q) is already in T', or (2) the cell (p,q) is not in T'.

Case (1): We must have (p,q) to be an inner corner of T' (no cell is to its right or above it), so $p = \mu'_q$ and $p > \mu'_{q+1}$. In this case, k is recorded in $Q(T')$. Then by the induction on T', every cell (i,j) of $P(T')$ has value $\ell + j - \mu'_j + i - 1$. To determine the insertion path of $P(T') \leftarrow \ell + q - p$, we compute the columns q and $q+1$ of $P(T')$ as follows:

row number	q-th column	$(q+1)$-st column
p	$\ell + q - 1$	
	$\vdots$	
$\mu'_{q+1} < p$	$\ell + q - p + \mu'_{q+1} - 1$	$\ell + q$
	$\vdots$	$\vdots$
2	$\ell + q - p + 1$	$\ell + q + 2 - \mu'_{q+1}$
1	$\ell + q - p$	$\ell + q + 1 - \mu'_{q+1}$

Following Case 2(b) of Hecke insertion, the insertion path is vertically up column $q+1$. At the top of the column, $\ell + q$ is inserted into row $\mu'_{q+1} + 1$. Furthermore, $\ell + q$ is greater than $\ell + q - p + \mu'_{q+1}$ in cell $(\mu'_{q+1} + 1, q)$ because $p > \mu'_{q+1}$. By Hecke insertion Case 1(b), the insertion ends in row $\mu'_{q+1} + 1$. Also $P(T')$ is unchanged, and k is recorded in cell (p,q) of $Q(T')$ since it is the corner whose column contains the rightmost box of row $\mu'_{q+1} + 1$. In this case, we get $Q(T'') = T''$. Since the shape μ is unchanged, we have that $P(T'') = P(T')$ also satisfies the statement.

Case (2): If cell (p,q) is not in T', then it must be an outer corner of T', so $\mu'_q = p - 1$ and $\mu'_{q-1} > p - 1$. Specifically, two cases can happen: (a) $p = 1$ and $(1, q-1) \in T'$, (b) both $(p-1, q), (p, q-1) \in T'$, or $q = 1$ and $(p-1, 1) \in T$.

Case 2(a): The first row of $P(T')$ is $\ell + 1 - \mu'_1, \ldots, \ell + j - \mu'_j, \ldots, \ell + (q-1) - \mu'_{q-1}$. Since $\ell + q - p = \ell + q - 1 > \ell + (q-1) - \mu'_{q-1}$, it is appended to the end of the first row which is the cell $(1, q)$. The letter k is recorded in the same new cell of $Q(T')$. In this case, the only entry in P that is changed is $(1, q)$, and its entry $\ell + q - 1$ satisfies the statement. Also $Q(T'')$ equals T''.

Case 2(b): Since entry $(i, q-1)$ of $P(T')$ is $\ell + q - 1 - \mu'_{q-1} + i - 1$ and entry (i, q) of $P(T')$ is $\ell + q - \mu'_q + i - 1$, the number $q - p + \ell$ is in-between the two when $i = 1$. So the insertion starts by bumping $(1, q)$. To get the insertion path, we compute columns $q-1$ and q as follows:

row number	$(q-1)$-st column	q-th column
μ'_{q-1}	$\ell + q - 2$	
	...	
$p-1$	$\ell + q + p - \mu'_{q-1} - 3$	$\ell + q - 1$
	...	...
2	$\ell + q - \mu'_{q-1}$	$\ell + q - p + 2$
1	$\ell + q - 1 - \mu'_{q-1}$	$\ell + q - p + 1$

By Hecke insertion Case 2(a), $\ell + q - p$ is placed in cell $(1, q)$ and the original column q is shifted one position higher. By Hecke insertion Case 1(a), the insertion terminates at row p and the original entry in cell $(p-1, q)$ is appended at the rightmost box of row p. Thus, μ'_q increases by 1. The updated entries in column q still satisfy the statement. Since the entries in other columns of $P(T')$ are unchanged and μ'_j is unchanged for $j \neq q$, they also satisfy the statement. So we have $P(T'')$ satisfies the statement. The letter k is inserted into the new cell (p, q) of $Q(T')$, which makes $Q(T'') = T''$.

Thus, the statement holds, proving the theorem. $\square$

2.2.2. The $\star$-insertion. We define a new insertion algorithm, which we call $\star$-insertion, from fully-commutative decreasing Hecke biwords $[\mathbf{k}, \mathbf{h}]^t$ to pairs of tableaux P and Q, denoted by $\star([\mathbf{k}, \mathbf{h}]^t) = (P, Q)$, as follows.

Definition 2.2.2. *Fix a fully-commutative decreasing Hecke biword* $[\mathbf{k}, \mathbf{h}]^t$. *The insertion is done by reading the columns of this biword from right to left.*

Begin with (P_0, Q_0) being a pair of empty tableaux. For every integer $i \geqslant 0$, we recursively construct (P_{i+1}, Q_{i+1}) from (P_i, Q_i) as follows. Let $[q, x]^t$ be the i-th column (from the right) of $[\mathbf{k}, \mathbf{h}]^t$. Suppose that we are inserting x into row R of P_i.

Case 1: *If R is empty or $x > \max(R)$, then form P_{i+1} by appending x to row R and form Q_{i+1} by adding q in the corresponding position to Q_i. Terminate and return (P_{i+1}, Q_{i+1}).*

Case 2: *Otherwise, if $x \notin R$, locate the smallest y in R with $y > x$. Bump y with x and insert y into the next row of P_i.*

Case 3: *Otherwise, if $x \in R$, locate the smallest y in R with $y \leqslant x$ and interval $[y, x]$ contained in R. Row R remains unchanged and y is to be inserted into the next row of P_i.*

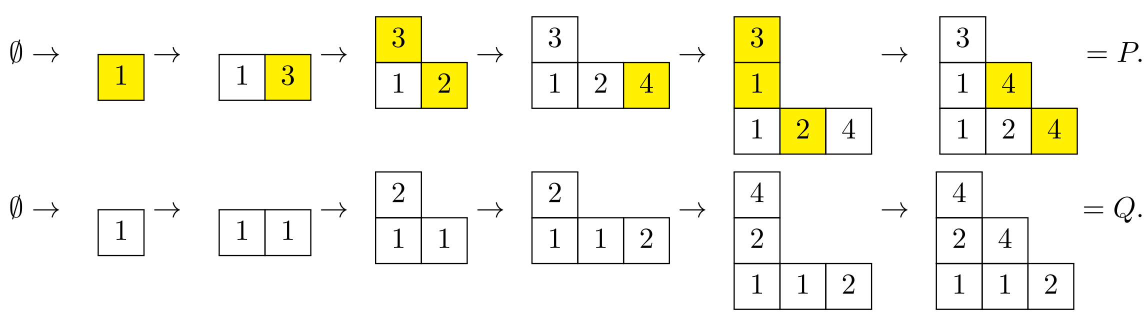
Denote $(P, Q) = (P_\ell, Q_\ell)$ if $[\mathbf{k}, \mathbf{h}]^t$ has length ℓ. We define the $\star$-insertion *by $\star([\mathbf{k}, \mathbf{h}]^t) = (P, Q)$.*

Furthermore, denote by $P \leftarrow x$ the tableau obtained by inserting x into P. The collection of all cells in $P \leftarrow x$, where insertion or bumping has occurred is called the insertion path *for $P \leftarrow x$. In particular, in Case 1 the newly added cell is in the insertion path, in Case 2 the cell containing the bumped letter y is in the insertion path, and in Case 3 the cell containing the same entry as the inserted letter is in the insertion path.*

Example 2.2.5. *Let*

$$\begin{bmatrix} \mathbf{k} \\ \mathbf{h} \end{bmatrix} = \begin{bmatrix} 4 & 4 & 2 & 2 & 1 & 1 \\ 4 & 2 & 4 & 2 & 3 & 1 \end{bmatrix}.$$

The corresponding sequence of insertion tableaux and recording tableaux under the $\star$-insertion is listed here:



Then we have $\star([\mathbf{k}, \mathbf{h}]^t) = (P, Q)$, and the cells in the insertion paths at each step are highlighted in yellow.

Example 2.2.6. *Let*

$$\begin{bmatrix} \mathbf{k} \\ \mathbf{h} \end{bmatrix} = \begin{bmatrix} 4 & 4 & 2 & 2 & 1 & 1 \\ 4 & 2 & 4 & 2 & 3 & 1 \end{bmatrix}.$$

The corresponding sequence of insertion tableaux and recording tableaux under the $\star$-insertion is listed here:

$$\emptyset \to \begin{ytableau} *(yellow) 1 \end{ytableau} \to \begin{ytableau} 1 & *(yellow) 3 \end{ytableau} \to \begin{ytableau} *(yellow) 3 \\ 1 & *(yellow) 2 \end{ytableau} \to \begin{ytableau} 3 \\ 1 & 2 & *(yellow) 4 \end{ytableau} \to \begin{ytableau} *(yellow) 3 \\ *(yellow) 1 \\ 1 & *(yellow) 2 & 4 \end{ytableau} \to \begin{ytableau} 3 \\ 1 & *(yellow) 4 \\ 1 & 2 & *(yellow) 4 \end{ytableau} = P.$$

$$\emptyset \to \begin{ytableau} 1 \end{ytableau} \to \begin{ytableau} 1 & 1 \end{ytableau} \to \begin{ytableau} 2 \\ 1 & 1 \end{ytableau} \to \begin{ytableau} 2 \\ 1 & 1 & 2 \end{ytableau} \to \begin{ytableau} 4 \\ 2 \\ 1 & 1 & 2 \end{ytableau} \to \begin{ytableau} 4 \\ 2 & 4 \\ 1 & 1 & 2 \end{ytableau} = Q.$$

Then we have $\star([\mathbf{k}, \mathbf{h}]^t) = (P, Q)$, and the cells in the insertion paths at each step are highlighted in yellow.

Lemma 2.2.1. *Let $[\mathbf{k}, \mathbf{h}]^t$ be a fully-commutative decreasing Hecke biword. Suppose that $\star([\mathbf{k}, \mathbf{h}]^t) = (P, Q)$. Then, the following statements hold:*

(1) P^t is semistandard and Q has the same shape as P.

(2) Let x be an integer such that $x \cdot \mathbf{h}$ is fully-commutative. Then the insertion path for $P \leftarrow x$ goes weakly to the left.

Proof. See Appendix A.1.1. $\square$

For the following results, given a tableau P with positive integer entries, $\mathsf{row}(P)$ denotes its row reading word, obtained by reading these entries row-by-row starting from the top row (in French notation), reading from left to right. We will consider $\mathsf{row}(P)$ as an element in a fixed 0-Hecke monoid.

Lemma 2.2.2. *Let P be a tableau such that P^t is semistandard and $\mathsf{row}(P)$ is fully-commutative. Let x be an integer such that $\mathsf{row}(P) \cdot x$ is fully-commutative. Then,*

$$\mathsf{row}(P \leftarrow x) \equiv_{\mathcal{H}_0} \mathsf{row}(P) \cdot x. \tag{2.2}$$

Proof. See Appendix A.1.2. $\square$

Remark 2.2.2. *Observe that the assumption that* $\mathsf{row}(P)$ *is fully-commutative implies that* $\mathsf{row}(R)$ *is fully-commutative for each row* R *of* P. *Moreover, in the proof of Lemma 2.2.2, if* x *is to be inserted into row* R *of* P *when computing* $P \leftarrow y$ *and* $x \in R$, *then the extra assumption that* $\mathsf{row}(P) \cdot x$ *is fully-commutative implies that* R *does not contain* $x + 1$.

Lemma 2.2.3. *Let* P *be a tableau such that* P^t *is semistandard and* $\mathsf{row}(P)$ *is fully-commutative. Let* x, x' *be integers such that* $\mathsf{row}(P) \cdot x$ *and* $\mathsf{row}(P) \cdot xx'$ *are fully-commutative.*

Denote the insertion paths of $P \leftarrow x$ *and* $(P \leftarrow x) \leftarrow x'$ *as* π *and* π' *respectively. Also, suppose that* $P \leftarrow x$ *and* $(P \leftarrow x) \leftarrow x'$ *introduce boxes* B *and* B' *respectively. Then the following statements about* $\star$*-insertion are true:*

(1) If $x < x'$, *then* π' *is strictly to the right of* π. *Moreover,* B' *is strictly to the right of and weakly below* B.

(2) If $x \geqslant x'$, *then* π' *is weakly to the left of* π. *Moreover,* B' *is weakly to the left of and strictly above* B.

Proof. See Appendix A.1.3. $\square$

Let U be a tableau such that U^t is semistandard and $\mathsf{row}(U)$ is fully-commutative. We describe the *reverse row bumping* for $\star$-insertion of U as follows. Locate an inner corner of U and remove entry y from that row. Perform the following operations until an entry is bumped out of the bottommost row. Suppose that we are reverse bumping y into a row R. If $y \notin R$, find the largest $x \in R$ with $x < y$; insert y and bump out x. Otherwise, $y \in R$, so find the largest $x \in R$ such that $[y, x]$ is the longest interval of consecutive integers. In this case, row R remains unchanged but x is bumped out. Then reverse bump x into the next row below unless there is no further row below. In this case, terminate and return the resulting tableau as T along with the bumped entry x. It is straightforward to see that reverse row bumping specified above reverses the bumping process specified by the $\star$-insertion.

Example 2.2.7. *Let U be the tableau*

$$U \;=\; \begin{array}{|c|c|c|} \cline{1-1} 5 & \multicolumn{2}{c}{} \\ \cline{1-1} 2 & \multicolumn{2}{c}{} \\ \cline{1-2} 2 & 5 & \multicolumn{1}{c}{} \\ \hline 2 & 3 & 5 \\ \hline 1 & 2 & 4 \\ \hline \end{array}\;.$$

By performing reverse row bumping on the topmost 5 in U, we obtain

$$T \;=\; \begin{array}{|c|c|c|} \cline{1-1} 5 & \multicolumn{2}{c}{} \\ \cline{1-2} 2 & 5 & \multicolumn{1}{c}{} \\ \hline 2 & 3 & 5 \\ \hline 1 & 3 & 4 \\ \hline \end{array}$$

and entry 2. It is also straightforward to check that $U = T \leftarrow 2$.

Corollary 2.2.1. *Let T be a tableau of shape λ such that T^t is semistandard and $row(T)$ is fully-commutative. Let k be a positive integer.*

Let $x_1 < x_2 < \cdots < x_k$ (similarly $x_k \leqslant \cdots \leqslant x_2 \leqslant x_1$) be integers such that $\mathsf{row}(T) \cdot x_1 x_2 \ldots x_i$ is fully-commutative for all $1 \leqslant i \leqslant k$. Then, the collection of boxes added to T to form the tableau

$$U = ((T \leftarrow x_1) \leftarrow x_2) \cdots \leftarrow x_k$$

has the property that no two boxes are in the same column (similarly row).

Conversely, if U is a tableau of shape μ such that $\lambda \subseteq \mu$ and μ/λ consists of k boxes with no two boxes in the same column, i.e, a horizontal strip of size k (similarly row, i.e., a vertical strip of size k), then there is a unique tableau T of shape λ and unique integers $x_1 < x_2 < \cdots < x_k$ (similarly $x_k \leqslant \cdots \leqslant x_2 \leqslant x_1$) such that

$$U = ((T \leftarrow x_1) \leftarrow x_2) \cdots \leftarrow x_k.$$

In particular, if $(P, Q) = \star([\mathbf{k}, \mathbf{h}]^t)$, where $[\mathbf{k}, \mathbf{h}]^t$ is a fully-commutative decreasing Hecke biword, then Q is semistandard.

Proof. Assume that $x_1 < x_2 < \cdots < x_k$. By statement (1) of Lemma 2.2.3, the sequence of added boxes in $U = ((T \leftarrow x_1) \leftarrow x_2) \cdots \leftarrow x_k$ moves weakly below and strictly to the right when computing U. In particular, no two of the added boxes can be in the same column.

To recover the required tableau T and integers $x_1 < x_2 < \cdots < x_k$, perform reverse row bumping on the boxes specified by the shape μ/λ within U starting from the rightmost box, working from right to left. The tableau T and the integers $x_1, x_2, \ldots, x_k$ are uniquely determined by the operations. Moreover, by Lemma 2.2.3, the integers $x_k, x_{k-1}, \ldots, x_1$ obtained in the given order of operations satisfy $x_1 < x_2 < \cdots < x_k$.

Now assume $x_k \leqslant \cdots \leqslant x_2 \leqslant x_1$. By statement (2) of Lemma 2.2.3, the sequence of added boxes moves strictly above and weakly to the right when computing U. In particular, no two of the added boxes can be in the same row.

Similarly, one may perform reverse row bumping on the boxes specified by the shape μ/λ within U starting from the topmost box, working from top to bottom. Again, the operations uniquely determine the tableau T and the integers $x_1, x_2, \ldots, x_k$. Moreover, by Lemma 2.2.3, the integers $x_k, x_{k-1}, \ldots, x_1$ obtained in the given order of operations satisfy $x_k \leqslant \cdots \leqslant x_2 \leqslant x_1$.

Finally, note that in a decreasing Hecke biword $[\mathbf{k}, \mathbf{h}]^t$, where $\mathbf{h} = h^m \ldots h^2 h^1$, entries within a fixed a^i are inserted in increasing order. It follows that the collection of all boxes with label i form a horizontal strip within the tableau Q. Collecting all these horizontal strips with values i from m to i in order by using the converse recovers Q, implying that Q is semistandard. □

Theorem 2.2.2. *The $\star$-insertion is a bijection from the set of all fully-commutative decreasing Hecke biwords to the set of all pairs of tableaux (P, Q) of the same shape, where both P^t and Q are semistandard and* $\mathsf{row}(P)$ *is fully-commutative.*

Proof. By successive applications of Lemma 2.2.2, if $(P, Q) = \star([\mathbf{k}, \mathbf{h}]^t)$, then as $\mathbf{h}$ is fully-commutative, $\mathsf{row}(P)$ is also fully-commutative. Hence, using Lemma 2.2.1 and Corollary 2.2.1, $\star$-insertion is a well-defined map from the set of all fully-commutative decreasing Hecke biwords to the set of all pairs of tableaux (P, Q) of the same shape with both P^t, Q semistandard and $\mathsf{row}(P)$ being fully-commutative.

It remains to show that the $\star$-insertion is an invertible map. Assume that P and Q are tableaux of the same shape with both P^t, Q semistandard and $\mathsf{row}(P)$ being fully-commutative. Since Q is semistandard, the collection of boxes with the same entry form a horizontal strip. Starting with the largest such entry m, perform reverse row bumping with the boxes in the strip from right to left. By Lemma 2.2.3, this recovers the entries in h^m in decreasing order. Repeating this procedure in decreasing order of entries recovers $\mathbf{h} = h^m \dots h^2 h^1$, which automatically yields a decreasing Hecke biword $[\mathbf{k}, \mathbf{h}]^t$. Furthermore, by repeated applications of Lemma 2.2.2, since $\mathsf{row}(P)$ was fully-commutative, then the reverse word of $\mathbf{h}$ is fully-commutative, so that $\mathbf{h}$ is fully-commutative too. Finally, by repeated applications of the converse stated in Corollary 2.2.1, the recovered decreasing Hecke biword $[\mathbf{k}, \mathbf{h}]^t$ is unique. $\square$

2.3. Properties of the $\star$-insertion

In this section, we show that the $\star$-insertion intertwines with the crystal operators. More precisely, the insertion tableau remains invariant on connected crystal components under the $\star$-insertion as shown in Section 2.3.1 by introducing certain micro-moves. In Section 2.3.2, it is shown that the $\star$-crystal on $\mathcal{H}^{m,\star}$ intertwines with the usual crystal operators on semistandard tableaux on the recording tableaux under the $\star$-insertion. In Section 3.2, we relate the $\star$-insertion to the uncrowding operation.

2.3.1. Micro-moves and invariance of the insertion tableaux. In this section, we introduce certain equivalence relations of the $\star$-insertion in order to establish its relation with the $\star$-crystal. From now on we are focusing on the sequence in the insertion order. Since each decreasing factorization $\mathbf{h}$ is inserted from right to left, we look at $\mathbf{h}$ read from right to left.

Definition 2.3.1. *We define an equivalence relation through micro-moves on fully-commutative words in $\mathcal{H}_0(n)$.*

(1) Knuth moves, for $x < z < y$:

(I1) $xyz \sim yxz$

(I2) $zxy \sim zyx$

(2) Weak Knuth moves, for $y > x + 1$:

(II1) $xyy \sim yxy$

(II2) $xxy \sim xyx$

(3) Hecke move*, for* $y = x+1$*:*

(III) $xxy \sim xyy$

Note that the micro-moves preserve the relation $\equiv_{\mathcal{H}_0}$.

Similar relations have appeared in [**7**, Eq. (1.2)].

EXAMPLE 2.3.1. *The* $13242 \in \mathcal{H}_0(5)$ *is equivalent to* 31242, 13422, 13224, 31224, *and itself.*

Next, we use the following notation on $\star$-insertion tableaux. For a single-row increasing tableau R, let R^x denote the first row of the tableau $R \leftarrow x$ and let $R(x)$ denote the output of the $\star$-insertion from the first row. If the $\star$-insertion outputs a letter, then denote it by $R(x)$; if x is appended to the end of the row R, then the output $R(x)$ is 0, which can be ignored. We always have $x \cdot 0 \sim x \sim 0 \cdot x$.

EXAMPLE 2.3.2. *Let* $R = \boxed{1}\boxed{3}\boxed{4}\boxed{6}\boxed{7}\boxed{8}$ *, then the first row of* $R \leftarrow 7$ *is*

$$R^7 = \boxed{1}\boxed{3}\boxed{4}\boxed{6}\boxed{7}\boxed{8}$$

and $R(7) = 6$*. Furthermore, the first row of* $R^7 \leftarrow 9$ *is* $R^{7,9} = \boxed{1}\boxed{3}\boxed{4}\boxed{6}\boxed{7}\boxed{8}\boxed{9}$ *and* $R^8(9) = 0$.

LEMMA 2.3.1. *Let R be a single-row increasing tableau, and x, y, z be letters such that* $\mathsf{row}(R) \cdot x \cdot y \cdot z$ *is fully-commutative. Let x', y', z' be letters such that $xyz \sim x'y'z'$. Following the above notation, we have*

$$R^{xyz} = R^{x'y'z'} \qquad \textit{and} \qquad R(x)R^x(y)R^{xy}(z) \sim R(x')R^{x'}(y')R^{x'y'}(z').$$

PROOF. See Appendix A.2.1. $\square$

PROPOSITION 2.3.1. *If two words in $\mathcal{H}_0(n)$ have the property that their reverse words are equivalent according to Definition 2.3.1, then they have the same insertion tableau under $\star$-insertion (inserted from right to left).*

PROOF. Let P be a $\star$-insertion tableau. By Lemma 2.2.1, P^t is a semistandard tableau. Let the rows of P be $R_1, \ldots, R_\ell$. Then each row is strictly increasing. The row R_j is considered to be empty for $j > \ell$.

Let x_1, y_1, z_1 and x_1', y_1', z_1' be letters such that $x_1y_1z_1 \sim x_1'y_1'z_1'$ and $\mathsf{row}(P) \cdot x_1 \cdot y_1 \cdot z_1$ is fully-commutative. Let the output of the $\star$-insertion algorithm of $P \leftarrow x_1 \leftarrow y_1 \rightarrow z_1$ (resp. $P \leftarrow x_1' \leftarrow y_1' \leftarrow z_1'$) from the row i be $x_{i+1}, y_{i+1}, z_{i+1}$ (resp. $x_{i+1}', y_{i+1}', z_{i+1}'$). That is:

- $R_i^{x_iy_iz_i}$ is the first row of $[(R_i \leftarrow x_i) \leftarrow y_i] \leftarrow z_i$ and the outputs in order are $x_{i+1}, y_{i+1}, z_{i+1}$.
- $R_i^{x_i'y_i'z_i'}$ is the first row of $[(R_i \leftarrow x_i') \leftarrow y_i'] \leftarrow z_i'$ and outputs in order are $x_{i+1}', y_{i+1}', z_{i+1}'$.

By Lemma 2.3.1, we have that $R_i^{x_iy_iz_i} = R_i^{x_i'y_i'z_i'}$ and $x_{i+1}y_{i+1}z_{i+1} \sim x_{i+1}'y_{i+1}'z_{i+1}'$ for all i (possibly some extra rows exceeding ℓ). Thus, we have the desired result. $\square$

EXAMPLE 2.3.3. *The four words in $\mathcal{H}_0(5)$ of Example 2.3.1 all have the same $\star$-insertion tableau:*

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

With some lemmas in Appendix A, we prove that the crystal operators $f_k^\star$ act by a composition of micro-moves as given in Definition 2.3.1. More precisely, for a fully-commutative decreasing factorization $\mathbf{h}$, we have $\mathbf{h}^{\mathsf{rev}} \sim f_k^\star(\mathbf{h})^{\mathsf{rev}}$ as long as $f_k^\star(\mathbf{h}) \neq 0$, where $\mathbf{h}^{\mathsf{rev}}$ is the reverse of $\mathbf{h}$.

REMARK 2.3.1. *By Definition 2.1.3 and Remark 2.1.3, there are two cases for the k-th and $(k+1)$-st factors under the crystal operator $f_k^\star$, where x is the largest unpaired letter in the k-th factor, $w_i, v_i > x$ and $u_i, b_i < x$:*

(1) $(w_1 \dots w_p u_1 \dots u_q)(v_1 \dots v_s x b_1 \dots b_t) \xrightarrow{f_k^\star} (w_1 \dots w_p x u_1 \dots u_q)(v_1 \dots v_s b_1 \dots b_t)$,
where $v_s \neq x+1$.

(2) $(w_1 \dots w_p u_1 \dots u_q)(v_1 \dots v_s x b_1 \dots b_t) \xrightarrow{f_k^\star} (w_1 \dots w_p x u_1 \dots u_q)(v_1 \dots v_{s-1} x b_1 \dots b_t)$,
where $v_s = w_p = x+1$.

In both cases, $u_i < x-1$ since if $u_1 = x-1$ then $b_1 = x-1$ due to the fact that x is unbracketed; but this would mean that the word is not fully-commutative. We also notice that since all u_i are paired with some b_j, we have that $t \geqslant q$ and $b_i \geqslant u_i$. Similarly, all v_i are paired with some w_j, so we have that $p \geqslant s$ and $v_i \geqslant w_{p-s+i}$. Let u denote the sequence $u_1 \dots u_q$ and let b denote the sequence $b_1 \dots b_t$.

Proposition 2.3.2. *Suppose* $\mathbf{h}$ *is a fully-commutative decreasing factorization such that* $f_k^\star(\mathbf{h}) \neq 0$ *(resp.* $e_k^\star(\mathbf{h}) \neq 0$*). Then* $f_k^\star(\mathbf{h})^{\mathsf{rev}} \sim \mathbf{h}^{\mathsf{rev}}$ *(resp.* $e_k^\star(\mathbf{h})^{\mathsf{rev}} \sim \mathbf{h}^{\mathsf{rev}}$*) for the equivalence relation* $\sim$ *of Definition 2.3.1.*

Proof. We prove the statement for $f_k^\star$. Since $e_k^\star$ is a partial inverse of $f_k^\star$, the result follows.

Let $\mathbf{h} = h^m \dots h^1 \in \mathcal{H}^{m,\star}$ and define $\widetilde{\mathbf{h}} = f_k^\star(\mathbf{h}) = h^m \dots \tilde{h}^{k+1}\tilde{h}^k h^{k-1} \dots h^1$. Specifically, $h^{k+1} = (w_1 \dots w_p u_1 \dots u_q)$ and $h^k = (v_1 \dots v_s x b_1 \dots b_t)$, where x is the largest unpaired letter in h^k. Then by Lemmas A.2.2 and A.2.3, we have the following sequence of equivalence moves:

$$(b_q \dots b_1 x v_s \dots v_1 u_q \dots u_1) w_p \dots w_{p-s+1} \sim (b_q u_q \dots b_1 u_1 x v_s \dots v_1) w_p \dots w_{p-s+1}$$

$$b_q u_q \dots b_1 u_1 x (v_s \dots v_1 w_p \dots w_{p-s+1}) \sim b_q u_q \dots b_1 u_1 x (v_s w_p \dots v_1 w_{p-s+1}).$$

Case (1): When $v_s \neq x+1$, $\tilde{h}^{k+1} = (w_1 \dots w_p x u_1 \dots u_q)$, $\tilde{h}^k = (v_1 \dots v_s b_1 \dots b_t)$. By Lemmas A.2.5 and A.2.6, we have

$$b_q u_q \dots b_1 u_1 (x v_s w_p \dots v_1 w_{p-s+1}) \sim b_q u_q \dots b_1 u_1 (v_s \dots v_1 x w_p \dots w_{p-s+1})$$

$$(b_q u_q \dots b_1 u_1 v_s \dots v_1) x w_p \dots w_{p-s+1} \sim (b_q \dots b_1 v_s \dots v_1 u_q \dots u_1) x w_p \dots w_{p-s+1}.$$

Thus, we have that

$$b_t \dots b_1 x v_s \dots v_1 u_q \dots u_1 w_p \dots w_1 \sim b_t \dots b_1 x v_s \dots v_1 u_q \dots u_1 x w_p \dots w_1.$$

Case (2): When $v_s = w_p = x+1$, $\tilde{h}^{k+1} = (w_1 \dots w_p x u_1 \dots u_q)$, $\tilde{h}^k = (v_1 \dots v_{s-1} x b_1 \dots b_t)$. Then by Lemmas A.2.4 and A.2.7, we have

$$b_q u_q \dots b_1 u_1 (x v_s w_p) v_{s-1} w_{p-1} \dots v_1 w_{p-s+1} \sim b_q u_q \dots b_1 u_1 (x x w_p) v_{s-1} w_{p-1} \dots v_1 w_{p-s+1}$$

$$b_q u_q \dots b_1 u_1 x (x w_p v_{s-1} w_{p-1} \dots v_1 w_{p-s+1}) \sim b_q u_q \dots b_1 u_1 x (v_{s-1} \dots v_1 x w_p \dots w_{p-s+1})$$

$$(b_q u_q \dots b_1 u_1 x v_{s-1} \dots v_1) x w_p \dots w_{p-s+1} \sim (b_q \dots b_1 x v_{s-1} \dots v_1 u_q \dots u_1) x w_p \dots w_{p-s+1}.$$

Thus, we have that

$$b_t \dots b_1 x v_s \dots v_1 u_q \dots u_1 w_p \dots w_1 \sim b_t \dots b_1 x v_{s-1} \dots v_1 u_q \dots u_1 x w_p \dots w_1.$$

Therefore, we have shown that in both cases, $f_k^\star(\mathbf{h})^{\mathsf{rev}} \sim \mathbf{h}^{\mathsf{rev}}$. $\square$

Proposition 2.3.3. *For $\mathbf{h} \in \mathcal{H}^{m,\star}$ such that $f_k^\star(\mathbf{h}) \neq 0$ for some $1 \leqslant k < m$, the $\star$-insertion tableau for $\mathbf{h}$ equals the $\star$-insertion tableau for $f_k^\star(\mathbf{h})$.*

Proof. By Proposition 2.3.2, the reverse words for $\mathbf{h}$ and $f_k^\star(\mathbf{h})$ are $\sim$-equivalent. By Proposition 2.3.1, the corresponding insertion tableaux are equal. $\square$

Proposition 2.3.4. *Let $\mathbf{h} \in \mathcal{H}^{m,\star}$ be a lowest weight element under Definition 2.1.3 of weight λ. Then there exists $r \geqslant 1$ where $\lambda_i = 0$ for $i < r$ and $\lambda_{i+1} \geqslant \lambda_i$ for $1 \leqslant i \leqslant m$. Suppose $\mathbf{h} = h^m \cdots h^r = (h^m_{\lambda_m} \dots h^m_1)(h^{m-1}_{\lambda_{m-1}} \cdots h^{m-1}_1) \dots (h^r_{\lambda_r} \dots h^r_1)$, then the i-th row of the $\star$-insertion tableau equals $h_1^{m+1-i}, h_2^{m+1-i}, \dots, h^{m+1-i}_{\lambda_{m+1-i}}$, that is,*

$$P^\star(\mathbf{h}) = \begin{array}{|c|c|c|c|c|c|} \cline{1-3} h_1^r & \dots & h^r_{\lambda_r} \\ \cline{1-4} \dots & \dots & \dots & \dots \\ \cline{1-5} h_1^{m-1} & h_2^{m-1} & \dots & \dots & h^{m-1}_{\lambda_{m-1}} \\ \hline h_1^m & h_2^m & \dots & \dots & \dots & h^m_{\lambda_m} \\ \hline \end{array}\,. \tag{2.3}$$

Proof. Without loss of generality, we may assume that $r = 1$. We prove the statement by induction on m. The case $m = 1$ is trivial.

Let $m \geqslant 1$ be arbitrary and suppose that the statement holds for this m. We prove the statement for $m+1$. We need to insert $P^\star(\mathbf{h}) \leftarrow h_1^{m+1} \leftarrow h_2^{m+1} \leftarrow \cdots \leftarrow h^{m+1}_{\lambda_{m+1}}$, where $P^\star(\mathbf{h})$ is as in (2.3) with $r = 1$. Note that $h_i^{m+1} \leqslant h_i^m$ for $1 \leqslant i \leqslant \lambda_m$. Specifically, $h_1^{m+1} \leqslant h_1^m$, so its insertion path is vertical along the first column and we obtain

$$P^\star(\mathbf{h}) \leftarrow h_1^{m+1} = \begin{array}{|c|c|c|c|c|c|} \cline{1-1} h_1^1 \\ \cline{1-3} h_1^2 & \dots & h^r_{\lambda_r} \\ \cline{1-4} \dots & \dots & \dots & \dots \\ \cline{1-5} h_1^m & h_2^{m-1} & \dots & \dots & h^{m-1}_{\lambda_{m-1}} \\ \hline h_1^{m+1} & h_2^m & \dots & \dots & \dots & h^m_{\lambda_m} \\ \hline \end{array}\,.$$

Since $h_1^{m+1} < h_2^{m+1} \leqslant h_2^m$, the insertion path of h_2^{m+1} is strictly to the right of the insertion path of h_1^{m+1} and weakly left of the second column by Lemma 2.2.3, so it is vertical along the second column. Similar arguments show that the insertion path for h_i^{m+1} is just vertical along the i-th column. Thus, the result holds for $m+1$. $\square$

REMARK 2.3.2. *For a lowest weight element $\mathbf{h} \in \mathcal{H}^{m,\star}$ of weight $\mathbf{a}$, the corresponding insertion tableau must have shape $\mu = \mathsf{sort}(\mathbf{a})$, which is the partition obtained by reordering $\mathbf{a}$.*

PROPOSITION 2.3.5. *Let $T \in \mathsf{SSYT}(\lambda)$ and $(P, Q) = \star \circ \mathsf{res}(T)$. Then $Q = T$.*

PROOF. The proof is done by induction on subtableaux of T similarly to the proof of Theorem 2.2.1.

For a given step in the insertion process, suppose that the entries of T that are involved so far form a nonempty subtableau T' of T with shape μ containing cell $(1,1)$. Furthermore, assume that the insertion and recording tableau at the corresponding step are $P(T')$ and $Q(T')$. Then they both have shape μ, and the entry of cell (i,j) of $P(T')$ is $\ell + j - \mu'_j + i - 1$. In addition, $Q(T') = T'$, where μ' is the conjugate of the partition μ and $\ell := \lambda'_1 = \ell(\lambda)$.

Note that we do not encounter Case (1) in the proof of Theorem 2.2.1. All other arguments still hold since for every insertion the letter is not contained in the row it is inserted into, that is, the insertion always bumps the smallest letter that is greater than itself. Thus, we omit the detail of the proof. $\square$

2.3.2. The $\star$-insertion and crystal operators. In this section, we prove that the $\star$-insertion and the crystal operators on fully-commutative decreasing factorizations and semistandard Young tableaux intertwine.

THEOREM 2.3.1. *Let $\mathbf{h} \in \mathcal{H}^{m,\star}$. Let $(P^\star(\mathbf{h}), Q^\star(\mathbf{h})) = \star(\mathbf{h})$ be the insertion and recording tableaux under the $\star$-insertion of Definition 2.2.2. Then*

(1) $f_i^\star(\mathbf{h})$ is defined if and only if $f_i(Q^\star(\mathbf{h}))$ is defined.

(2) If $f_i^\star(\mathbf{h})$ is defined, then $Q^\star(f_i^\star(\mathbf{h})) = f_i Q^\star(\mathbf{h})$.

In other words, the following diagram commutes:

$$\begin{array}{ccc}
\mathcal{H}^{m,\star} & \xrightarrow{Q^\star} & \mathsf{SSYT}^m \\
\Big\downarrow f_i^\star & & \Big\downarrow f_i \\
\mathcal{H}^{m,\star} & \xrightarrow{Q^\star} & \mathsf{SSYT}^m.
\end{array}$$

PROOF. The crystal operator $f_i^\star$ acts only on factors h^{i+1} and h^i. Hence it suffices to prove the statement for $\mathbf{h} = h^{i+1}h^i \ldots h^1$ with $i+1$ factors.

Suppose $f_i^\star(\mathbf{h}) \neq 0$. By Proposition 2.3.3, $P^\star(\mathbf{h}) = P^\star(f_i^\star(\mathbf{h}))$. Furthermore, by Lemma 2.2.1 $P^\star(\mathbf{h})$ and $Q^\star(\mathbf{h})$ have the same shape. Hence in particular, $Q^\star(\mathbf{h})$ and $Q^\star(f_i^\star(\mathbf{h}))$ have the same shape and therefore the letters i and $i+1$ in $Q^\star(\mathbf{h})$ and $Q^\star(f_i^\star(\mathbf{h}))$ occupy the same skew shape.

Recall from Definition 2.1.3 that $f_i^\star$ removes precisely one letter from factor $h^i = (h^i_\ell h^i_{\ell-1} \ldots h^i_1)$, say h^i_k. By Lemma 2.2.3, the insertion paths of $h^i_1, \ldots, h^i_\ell$ into $P^\star(h^{i-1} \cdots h^1)$ move strictly to the right and the newly added cells form a horizontal strip. In addition, the letters $h^i_1, \ldots, h^i_\ell$ appear in the first row of $P^\star(h^i \cdots h^1)$. Now compare this to the insertion paths for $h^i_1, \ldots, \widehat{h^i_k}, \ldots, h^i_\ell$ into $P^\star(h^{i-1} \ldots h^1)$, where h^i_k is missing. Up to the insertion of h^i_{k-1}, everything agrees. Suppose that h^i_k bumps the letter x in the first row and h^i_{k+1} bumps the letter $y > x$ in the first row by Lemma 2.2.3. Then when h^i_{k+1} gets inserted without prior insertion of h^i_k, the letter h^i_{k+1} either still bumps y or h^i_{k+1} bumps x (in which case x and y are adjacent in the first row in $P^\star(h^{i-1} \cdots h^1)$). There are no other choices, since if there are letters between x and y in the first row and h^i_{k+1} bumps one of these, it would have already bumped a letter to the left of y in $P^\star(h^i \cdots h^1)$. If h^i_{k+1} bumps x without prior insertion of h^i_k, then its insertion path is the same as the insertion path of h^i_k previously. If h^i_{k+1} bumps y, then the letter inserted into the second row by similar arguments either bumps the same letter as in the previous insertion path of h^i_{k+1} or h^i_k and so on. The last cell added is hence the same cell added in the previous insertion path of either h^i_k or h^i_{k+1}. Repeating these arguments, exactly one cell containing i in $Q^\star(h^i \cdots h^1)$ is missing in $Q^\star((h^i_\ell \ldots \widehat{h^i_k} \ldots h^i_1)h^{i-1} \cdots h^1)$ and all other cells containing i are the same. Hence, $Q^\star(f_i^\star(\mathbf{h}))$ is obtained from $Q^\star(\mathbf{h})$ by changing exactly one letter i to $i+1$.

It remains to prove that $f_i^\star(\mathbf{h}) \neq 0$ if and only if $f_i(Q^\star(\mathbf{h})) \neq 0$ and, if $f_i^\star(\mathbf{h}) \neq 0$, then the letter i that is changed to $i+1$ from $Q^\star(\mathbf{h})$ to $Q^\star(f_i^\star(\mathbf{h}))$ is the rightmost unbracketed i in $Q^\star(\mathbf{h})$. First assume that under the bracketing rule for $f_i^\star$, all letters in the factor h^i are bracketed, so that

$f_i^\star(\mathbf{h}) = 0$. This means that each letter in h^i is paired with a weakly smaller letter in h^{i+1}. Then by similar arguments as in Lemma 2.2.3 (2), for each insertion path for the letters in h^i, there is an insertion path for the letters in h^{i+1} that is weakly to the left and the resulting new cell is weakly to the left and strictly above of the corresponding new cell for the letter in h^i. This means that each i in $Q^\star(\mathbf{h})$ is paired with an $i+1$ and hence $f_i(Q^\star(\mathbf{h})) = 0$.

Now assume that $f_i^\star(\mathbf{h}) \neq 0$. Let us use the same notation as in Remark 2.3.1 (with k replaced by i). Since all letters $u_q, \ldots, u_1 < x$ are paired with some letters $b_j < x$, their insertion paths (again by similar arguments as in Lemma 2.2.3) lie strictly to the left of the insertion path for x. First assume that $v_s \neq x+1$. Recall that by Proposition 2.3.3, $P^\star(\mathbf{h}) = P^\star(f_i^\star(\mathbf{h}))$. Also, by the above arguments, moving letter x to factor h^{i+1} under $f_i^\star$, changes one i to $i+1$ (precisely the i that is missing when removing x from h^i). Now the letters $w_p, \ldots, w_1 > x$ are inserted after the letter x in the $(i+1)$-th factor in $f_i^\star(\mathbf{h})$ and by Lemma 2.2.3 their insertion paths are strictly to the right of the insertion path of x in $f_i^\star(\mathbf{h})$. But this means that the corresponding $i+1$ in $Q^\star(\mathbf{h})$ cannot bracket with the i that changes to $i+1$ under $f_i^\star$. This proves that $f_i(Q^\star(\mathbf{h})) \neq 0$. Furthermore, each $v_s, \ldots, v_1$ is paired with some w_j and hence the insertion path of this w_j is weakly to the left of the insertion path of the corresponding v_h. Hence all i to the right of the i that changes to an $i+1$ under $f_i^\star$ are bracketed. This proves that this i is the rightmost unbracketed i, proving the claim. The case $v_s = x+1$ is similar. $\square$

Remark 2.3.3. *Proposition 2.3.4 and Theorem 2.3.1 provide another proof via $\star$-insertion, in the case where w is fully-commutative, of the Schur positivity of G_w of Fomin and Greene [**7**]*

$$G_w = \sum_\mu \beta^{|\mu|-\ell(w)} g_w^\mu s_\mu,$$

where $g_w^\mu = |\{T \in \mathsf{SSYT}^n(\mu') \mid w_C(T) \equiv w\}|$.

2.3.3. Uncrowding set-valued skew tableaux.

Lemma 2.3.2. *For skew shape λ/μ, the crystal operators on $\mathsf{SVT}^m(\lambda/\mu)$ intertwine with those on $\mathsf{SSYT}^m(\nu/\mu)$, for $\lambda \subseteq \nu$, under* uncrowd*.*

Proof. Chan and Pflueger [**6**] proved that the image of $T \in \mathsf{SVT}(\lambda/\mu)$ under the uncrowding map is a pair (P, Q), where P is a semistandard tableau of shape ν/μ and Q is a flagged increasing tableau of shape ν/λ. Monical, Pechenik and Scrimshaw in [**22**, Theorem 3.12] proved that the crystal operators on $\mathsf{SVT}^m(\lambda)$ intertwine with those on $\mathsf{SSYT}^m(\nu)$ under $\mathsf{uncrowd}$. Since $\mathsf{uncrowd}$ is defined equally on skew shapes, the result follows. $\square$

2.3.4. Compatibility of $\star$-insertion with uncrowding. For a partition μ, let T_μ be the unique tableau of shape μ with μ_i letters i in each row i. Note that $\mathsf{uncrowd}(T_\mu) = (T_\mu, \emptyset)$ since $\mathsf{ex}(T_\mu) = 0$.

LEMMA 2.3.3. *For $T \in \mathsf{SVT}^m(\lambda/\mu)$, if $(P, Q) = \star(\mathbf{h}\mathbf{h}')$ where $\mathbf{h} = \mathsf{res}(T)$ and $\mathbf{h}' = \mathsf{res}(T_\mu)$, then T_μ is contained in Q.*

PROOF. For $T \in \mathsf{SVT}^m(\lambda/\mu)$, let $T*$ be the set-valued tableau of shape λ obtained from T by adding $\ell(\mu)$ to each entry and filling in the cells of μ with T_μ. By Proposition 2.3.5, we have

$$\star \circ \mathsf{res}(T_\mu) = (P_\mu, T_\mu)\,, \tag{2.4}$$

where P_μ is the semistandard tableau specified in the proof of Proposition 2.3.5. The claim follows by noting that $\mathsf{res}(T*) = \mathsf{res}(T)\mathsf{res}(T_\mu)$. $\square$

DEFINITION 2.3.2. *A modification of $\star$-insertion is defined on $\mathcal{H}^{*,m}$ as follows: for $\mathbf{h} \in \mathcal{H}^{*,m}$, let λ/μ be the shape of $\mathsf{res}^{-1}(\mathbf{h})$ (which is well-defined up to a shift by Proposition 2.1.2). For $\mathbf{h}' = \mathsf{res}(T_\mu)$, let $(P*, Q*) = \star(\mathbf{h}\mathbf{h}')$. Define $\tilde{\star}(\mathbf{h}) = (P, Q)$ where P is obtained from $P*$ by deleting all entries in cells of μ and Q is defined from $Q*$ by deleting T_μ from it and decreasing all other letters by $\ell(\mu)$.*

Note that this is well-defined by Lemma 2.3.3 and the fact that each $\mathbf{h} \in \mathcal{H}^{*,m}$ can be associated to a skew shape λ/μ which is the shape of $\mathsf{res}^{-1}(\mathbf{h})$ by Proposition 2.1.2. Also note that $\tilde{\star}(\mathbf{h}) = \star(\mathbf{h})$ if $\mu = \emptyset$.

THEOREM 2.3.2. *Let $T \in \mathsf{SVT}^m(\lambda/\mu)$, $(\tilde{P}, \tilde{Q}) = \mathsf{uncrowd}(T)$, and $(P, Q) = \tilde{\star} \circ \mathsf{res}(T)$. Then $Q = \tilde{P}$.*

Proof. We start by addressing the straight-shape case; for $T* \in \mathsf{SVT}^m(\lambda)$, consider the following compositions of maps:

$$\begin{tikzcd}
(\tilde{P},\tilde{Q}) \arrow[d, "f_k"] & T* \arrow[l, "\mathsf{uncrowd}"'] \arrow[r, "\mathsf{res}"] \arrow[d, "f_k"] & \mathbf{h} \arrow[r, "\star"] \arrow[d, "f_k^\star"] & (P,Q) \arrow[d, "f_k"] \\
(f_k(\tilde{P}),\tilde{Q}) & f_k(T*) \arrow[l, "\mathsf{uncrowd}"'] \arrow[r, "\mathsf{res}"] & f_k^\star(\mathbf{h}) \arrow[r, "\star"] & (P, f_k(Q)).
\end{tikzcd}$$

By Lemma 2.3.2, the left square commutes. By Theorem 2.1.2 the center square commutes. By Proposition 2.3.3 and Theorem 2.3.1 the right square commutes. Hence it suffices to prove that $Q = \tilde{P}$ when $T*$ is a lowest weight element in the crystal.

Suppose $T* \in \mathsf{SVT}^m(\lambda)$ is of lowest weight with $\mathrm{wt}(T*) = \mathbf{a}$ and $\mathsf{ex}(T*) = \ell$. Then the decreasing factorization $\mathbf{h} \in \mathcal{H}^{m,\star}$ is lowest weight by Theorem 2.1.2. By Remark 2.3.2, P and hence Q has to be of shape $\nu = \mathsf{sort}(\mathbf{a})$. By Theorem 2.3.1, Q is the unique lowest weight element in SSYT^m of shape ν.

Consider the uncrowding operator on $T*$ and record each tableau during the process of uncrowding as in Definition 1.2.5 by a sequence of set-valued tableaux $T* = \tilde{P}_0 \to \tilde{P}_1 \to \cdots \to \tilde{P}_\ell = \tilde{P}$. Since $T*$ is of lowest weight, so are all the $\tilde{P}_i$. Furthermore, all $\tilde{P}_i$ have the same weight $\mathbf{a}$. Let $(P_i, Q_i) = \star \circ \mathsf{res}(\tilde{P}_i)$. For all $0 \leqslant i \leqslant \ell$, Q_i is the unique lowest weight element in SSYT^m of shape ν. Hence in particular $Q_i = Q$ for all $0 \leqslant i \leqslant \ell$. By Proposition 2.3.5, $Q = Q_\ell = \tilde{P}$, proving the claim for straight shapes.

Now take $T \in \mathsf{SVT}^m(\lambda/\mu)$ and construct $T*$ from T by adding $\ell(\mu)$ to each entry and filling in the cells of μ with T_μ. Note that $T*$ is a set-valued tableaux of shape λ. Let $(P, Q) = \star \circ \mathsf{res}(T*)$ and $(P*, Q*) = \tilde{\star} \circ \mathsf{res}(T)$. Since $\mathsf{res}(T*) = \mathsf{res}(T)\mathsf{res}(T_\mu)$, Lemma 2.3.3 implies that $Q* = Q/T_\mu$. On the other hand, since $T*$ has straight shape, the preceding paragraph gives that $\mathsf{uncrowd}(T*) = (Q, \tilde{Q})$ for some $\tilde{Q}$. We then note that $\mathsf{uncrowd}(T)$ and $\mathsf{uncrowd}(T*)$ are identical on cells of λ/μ up to a shift of the entries by $\ell(\mu)$; in particular, applying $\mathsf{uncrowd}$ to $T*$ does not involve any cell of μ since none of these are multicells and their entries are the smallest $\ell(\mu)$ letters. $\square$

2.4. Results on the non-fully-commutative case

In this section, we discuss some aspects when we generalize to the non-fully-commutative case. In Section 2.4.1, we describe a local crystal on $\mathcal{H}^m(3)$. In Section 2.4.2, we show that under very mild assumptions it is not possible to expect a local crystal for $n > 3$.

2.4.1. The case $n = 3$.

We provide a description of a type A_{m-1} crystal structure on $\mathcal{H}^m(3)$.

Definition 2.4.1. *Let $\mathbf{h} = h^m h^{m-1} \dots h^2 h^1 \in \mathcal{H}^m(3)$. Fix $1 \leqslant k < m$. Define the pairing process of $\mathbf{h}$ and the number of pairs in $h^{k-1} \dots h^{j+1} h^j$, denoted $p([j, k-1])$, recursively as follows:*

(1) The empty factorization, denoted $\emptyset$, has no pairs and $p(\emptyset) = 0$.

(2) If $p([1, j-1])$ is defined for all $1 \leqslant j \leqslant k$, then we have $p([j, k-1]) = p([1, k-1]) - p([1, j-1])$.

(3) If $h^k = (\,)$, then set $p([1, k]) = p([1, k-1])$.

(4) Otherwise, if $h^k = (21)$, pair the 2 with the 1 in h^k and set $p([1, k]) = p([1, k-1]) + 1$.

(5) Otherwise, if $h^k = (2)$ and $p([1, k-1])$ is even, ignoring all previously paired letters, locate the leftmost unpaired letter in $h^{k-1} \dots h^2 h^1$.

(a) If this letter is in $h^j = (1)$ and $p([j+1, k-1])$ is even, then pair the 2 in h^k with the 1 in h^j and set $p([1, k]) = p([1, k-1]) + 1$.

(b) If this letter is in $h^j = (2)$ and $p([j+1, k-1])$ is odd, then pair the 2 in h^k with the 2 in h^j and set $p([1, k]) = p([1, k-1]) + 1$.

(c) Else, set $p([1, k]) = p([1, k-1])$.

(6) Otherwise, if $h^k = (1)$ and $p([1, k-1])$ is odd, ignoring all previously paired letters, locate the leftmost unpaired letter in $h^{k-1} \dots h^2 h^1$.

(a) If this letter is in $h^j = (2)$ and $p([j+1, k-1])$ is even, then pair the 1 in h^k with the 2 in h^j and set $p([1, k]) = p([1, k-1]) + 1$.

(b) If this letter is in $h^j = (1)$ and $p([j+1, k-1])$ is odd, then pair the 1 in h^k with the 1 in h^j and set $p([1, k]) = p([1, k-1]) + 1$.

(c) Else, set $p([1, k]) = p([1, k-1])$.

(7) Else, set $p([1, k]) = p([1, k-1])$.

Example 2.4.1. *Let $m = 8$ and consider $\mathbf{h} = (\ )(2)(\ )(21)(1)(1)(2)(21) \in \mathcal{H}^8(3)$. The pairing process results in $(\ )(2)(\ )(21)(1)(1)(2)(21)$, where the paired letters are indicated with braces. Hence, we have the following values of $p([1,k])$ for $1 \leqslant k \leqslant 8$: 0, 1, 1, 2, 2, 3, 3, 3. Note that the letters in the fourth and seventh factors are left unpaired.*

Similarly, if we take $\mathbf{h} = (\ )(2)(2)(21)(2)(1)(21)(21) \in \mathcal{H}^8(3)$, we obtain $(\ )(2)(2)(21)(2)(1)(21)(21)$. Thus, we have the following values of $p([1,k])$ for $1 \leqslant k \leqslant 8$: 0, 1, 2, 2, 2, 3 ,4, 5. In this case all the letters in $\mathbf{h}$ are paired.

Definition 2.4.2. *Let $\mathbf{h} = h^m \dots h^2 h^1 \in \mathcal{H}^m(3)$. The crystal operator f_i for $1 \leqslant i < m$ on $\mathbf{h}$ is defined as follows. The operator f_i only depends on $h^{i+1}h^i$ and the parity of $p([1, i-1])$ of Definition 2.4.1. In the following cases, we indicate only the changes in $h^{i+1}h^i$ under f_i as the remainder of $\mathbf{h}$ remains invariant:*

(1) $(21)(x) \xrightarrow{i} 0$, *where* $(x) \in \{(\ ), (1), (2), (21)\}$,
(2) $(x)(\ ) \xrightarrow{i} 0$, *where* $(x) \in \{(\ ), (1), (2), (21)\}$,
(3) $(x)(x) \xrightarrow{i} 0$, *where* $(x) \in \{(\ ), (1), (2)\}$,
(4) $(1)(21) \xrightarrow{i} (21)(2)$,
(5) $(2)(21) \xrightarrow{i} (21)(1)$,
(6) $(\ )(x) \xrightarrow{i} (x)(\ )$, *where* $(x) \in \{(1), (2)\}$,
(7) $(\ )(21) \xrightarrow{i} (2)(1) \xrightarrow{i} (21)(\ )$, *if* $p([1, i-1])$ *is even,*
(8) $(\ )(21) \xrightarrow{i} (1)(2) \xrightarrow{i} (21)(\ )$, *if* $p([1, i-1])$ *is odd.*

The operator e_i is defined similarly. One reverses the changes introduced in cases (4) to (8) and annihilates $\mathbf{h}$ when the following occurs at $h^{i+1}h^i$:

(1)' $(x)(21) \xrightarrow{i} 0$, *where* $(x) \in \{(\ ), (1), (2), (21)\}$,
(2)' $(\ )(x) \xrightarrow{i} 0$, *where* $(x) \in \{(\ ), (1), (2), (21)\}$,
(3)' $(x)(x) \xrightarrow{i} 0$, *where* $(x) \in \{(\ ), (1), (2)\}$.

Similar to Definition 2.1.3, the weight map is defined as $\mathrm{wt}(\mathbf{h}) = (\mathsf{len}(h^1), \mathsf{len}(h^2), \dots, \mathsf{len}(h^m))$. Meanwhile, $\varphi_i(\mathbf{h})$ (resp. $\varepsilon_i(\mathbf{h})$) is defined to be the largest nonnegative integer k such that $f_i^k(\mathbf{h}) \neq 0$ (resp. $e_i^k(\mathbf{h}) \neq 0$).

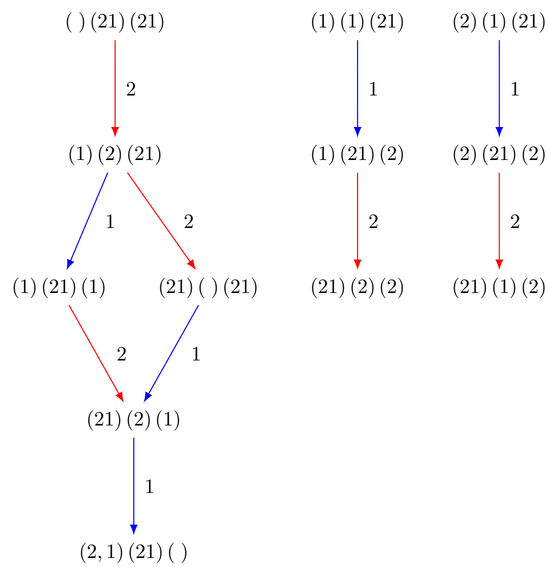


FIGURE 2.3. The crystal graph for $\mathcal{H}^3(3)$ restricted to decreasing factorizations with four letters.

It is not difficult to check that the operators f_i and e_i defined above preserve the relation $\equiv_{\mathcal{H}_0}$ on $\mathcal{H}^m(3)$ whenever they do not annihilate the decreasing factorizations. Furthermore, the structure above defines an abstract, seminormal A_{m-1} crystal on $\mathcal{H}^m(3)$.

We note that one may also verify that the crystal is a Stembridge crystal by checking that the axioms formulated in [**31**] are satisfied. Figure 2.3 displays the crystal graph on $\mathcal{H}^3(3)$ restricted to decreasing factorizations that use exactly 4 letters.

2.4.2. Nonlocality. In this subsection, we show that it is impossible to construct a crystal on $\mathcal{H}^m$ with the following properties for f_i:

(1) f_i only changes the i-th and $(i+1)$-th decreasing factors;
(2) f_i is determined by the first $(i+1)$ factors;
(3) $f_i(\mathbf{h}) \equiv_{\mathcal{H}_0} \mathbf{h}$ and $\mathsf{ex}[f_i(\mathbf{h})] = \mathsf{ex}(\mathbf{h})$, for all $\mathbf{h} \in \mathcal{H}^m$ with $f_i(\mathbf{h}) \neq 0$.

Let $\mathbf{h}_1 = h_1^m \dots h_1^2 h_1^1 \in \mathcal{H}^m$ and suppose that $f_i(\mathbf{h}_1) \neq 0$. If we write $f_i(\mathbf{h}_1) = h_2^m \dots h_2^2 h_2^1$, then the above assumptions imply that $h_1^{i+1} h_1^i \dots h_1^1 \equiv_{\mathcal{H}_0} h_2^{i+1} h_2^i \dots h_2^1$. Obviously the crystal on $\mathcal{H}^m(3)$ defined in Section 2.4.1 satisfies these assumptions.

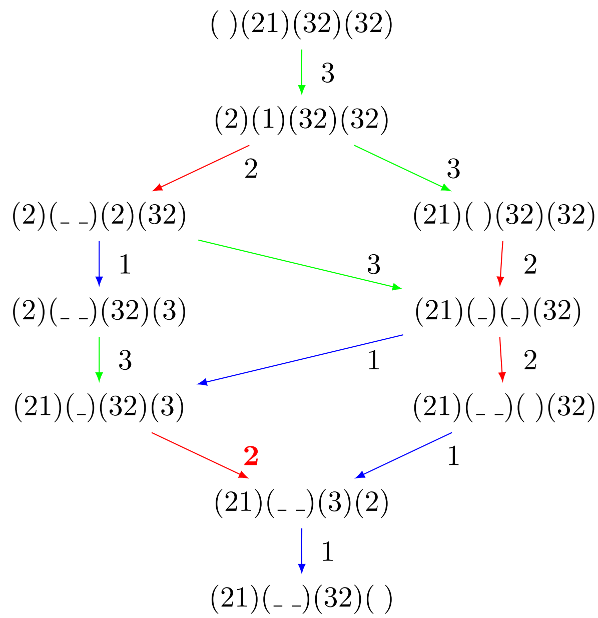


FIGURE 2.4. Partial filling of the connected component of $\mathcal{H}^4(3)$ containing highest weight element ( )(21)(32)(32).

Suppose that a crystal structure with the above assumptions exists on $\mathcal{H}^4(4)$. Consider the Schur expansion of the stable Grothendieck polynomial in 4 variables for $w = 12132$:

$$G_{12132}(x_1, x_2, x_3, x_4; \beta) = s_{221} + \beta(2s_{222} + 3s_{2211}) + \beta^2(6s_{2221} + 6s_{22111}) + \cdots .$$

(Note that s_{22111} is zero in four variables and hence could be omitted). The linear term in β implies that there are two connected components with highest weight $(2,2,2,0)$ (lowest weight $(0,2,2,2)$) for the crystal $\mathcal{H}^4(4)$ with excess 1. All decreasing factorizations mentioned below are those of $w = 12132$ with 4 factors and excess 1.

There are two decreasing factorizations of weight $(2,2,2,0)$: ( )(21)(21)(32) and ( )(21)(32)(32). Focus on the connected component with highest weight ( )(21)(32)(32) and try to complete the crystal graph from top to bottom. Since the only decreasing factorization of weight $(2,2,1,1)$ with the first and second factors both being (32) is (2)(1)(32)(32), we can compute the action of f_3 on this highest weight element. By some similar arguments we can fill in part of the crystal graph as indicated in Figure 2.4 with the above assumptions. The dashed spaces are undetermined.

Yet note that the red f_2 highlighted in the graph changed the first factor from (3) to (2). Hence, Condition (1) is violated, providing a counterexample that crystals with the above conditions always exist on $\mathcal{H}^m(n)$ for $n > 3$.

CHAPTER 3

Uncrowding algorithm for hook-valued tableaux

This chapter is based on work in collaboration with Joseph Pappe, Wencin Poh and Anne Schilling in preprint [**25**].

The chapter is organized as follows. In Section 3.1, we review the definition of semistandard hook-valued tableaux of [**32**] and the crystal structure on them [**11**]. In Section 3.2, 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 Section 3.3, we consider applications of the uncrowding algorithm, in particular expansions of the canonical Grothendieck polynomials using techniques developed in [**1**].

3.1. Hook-valued tableaux

In Section 3.1.1, we define hook-valued tableaux [**32**] and in Section 3.1.2 we review the crystal structure on hook-valued tableaux as introduced in [**11**].

3.1.1. 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. 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 $\mathsf{L}^+(U) = (x, \ell_1, \ell_2, \dots, \ell_p)$. We denote by $\max(U)$ (resp. $\min(U)$) the maximal (resp. minimal) entry in U.

Definition 3.1.1. *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) $\max(A) \leqslant \min(B)$ *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 3.1.1. *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.*

3.1.2. Crystal structure on hook-valued tableaux. Hawkes and Scrimshaw [**11**] defined a crystal structure on hook-valued tableaux. We review their definition here.

Definition 3.1.2 ([**11**], 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,*

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

where ℓ *is the number of columns of* T *and* C_i *is the* i*th column of* T.

Example 3.1.1. *Let T be the hook-valued tableau*

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

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

$$R(C) = 432113543344445.$$

Definition 3.1.3. *[**11**, Definition 4.3] Let $T \in \mathsf{HVT}^m(\lambda)$. For any $1 \leqslant i < m$, 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 [**22**] or the one on MVT in [**11**] when the arm excess or leg excess of the tableaux is set to 0, respectively.

Example 3.1.2. *Consider the following hook-valued tableau T:*

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

Then, e_3 annihilates T, whereas

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

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 $T(r, c)$.

3.2. Uncrowding map on hook-valued tableaux

In Section 3.2.1, we give a new uncrowding map on hook-valued tableaux and prove some of its properties in Section 3.2.2. The relation to the uncrowding map on multiset-valued tableaux is given in Section 3.2.3. In Section 3.2.4, we give the inverse of the uncrowding map on hook-valued tableaux, called the crowding map. In Section 3.2.5, an alternative definition of the uncrowding map on hook-valued tableaux is provided.

3.2.1. Uncrowding map on hook-valued tableaux. In [**11**], 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 Section 3.2.3.

Definition 3.2.1. *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 Figures 3.1 and 3.2 for illustration.

FIGURE 3.1. When $\tilde{r} \neq r$. Left: $(\tilde{r}, c+1)$ is a new cell; Right: $(\tilde{r}, c+1)$ is an existing cell.

FIGURE 3.2. When $\tilde{r} = r$. Left: $(r, c+1)$ is a new cell; Right: $(r, c+1)$ is an existing cell.

LEMMA 3.2.1. *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. See Appendix B.1.1. □

DEFINITION 3.2.2. *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 $d \geqslant 1$ 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 3.2.3. *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 $1 \leqslant i \leqslant \alpha$, 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 $\mathcal{U}(T) = (P(T), Q(T)) := (P_\alpha, Q_\alpha)$.

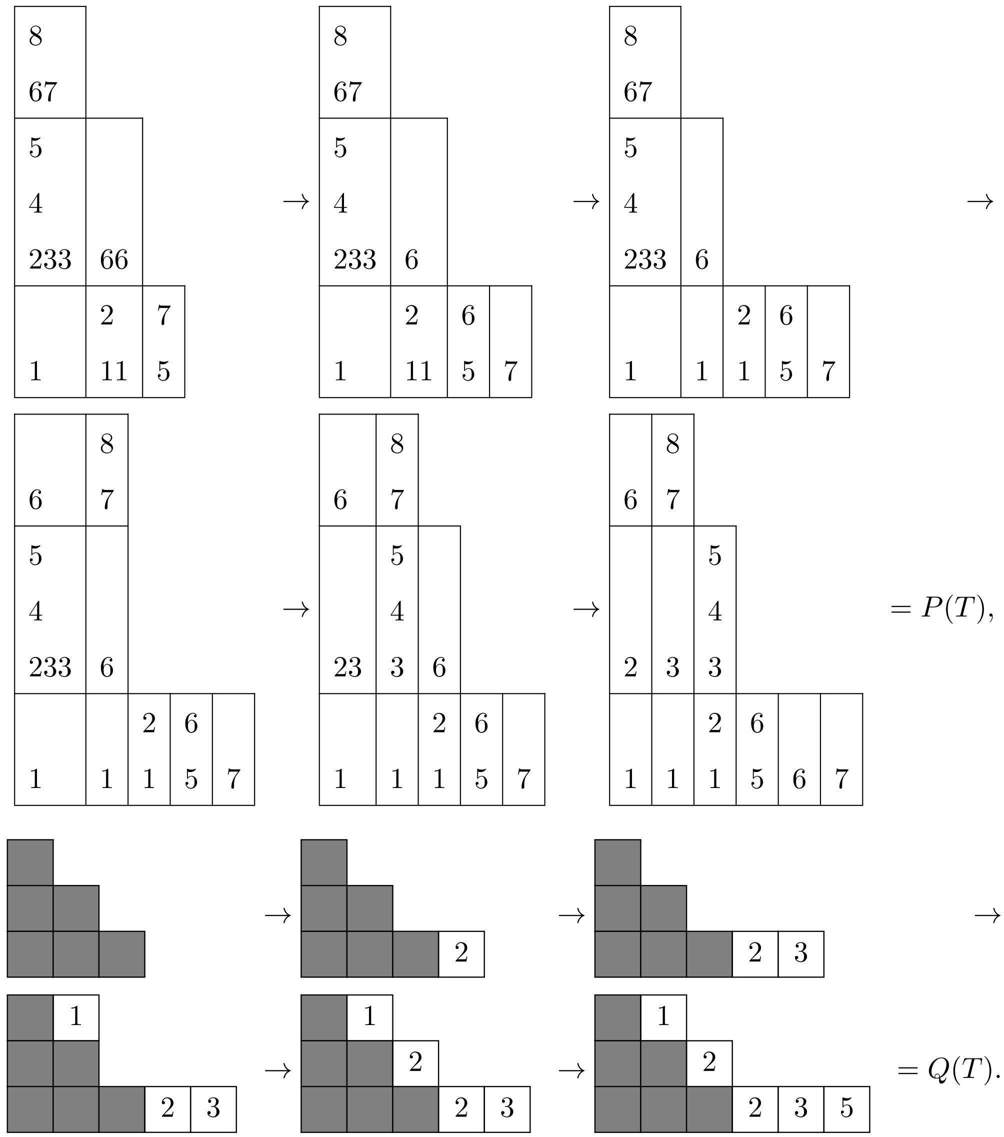
EXAMPLE 3.2.1. *Let T be the hook-valued tableau*

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

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

$$\begin{array}{|c|c|c|}\cline{1-1} \begin{matrix}8\\67\end{matrix} & \multicolumn{2}{c}{} \\ \cline{1-2} \begin{matrix}5\\4\\233\end{matrix} & 66 & \multicolumn{1}{c}{} \\ \hline \begin{matrix}\\1\end{matrix} & \begin{matrix}2\\11\end{matrix} & \begin{matrix}7\\5\end{matrix} \\ \hline\end{array} \to \begin{array}{|c|c|c|}\cline{1-1} \begin{matrix}8\\67\end{matrix} & \multicolumn{2}{c}{} \\ \cline{1-2} \begin{matrix}5\\4\\233\end{matrix} & 6 & \multicolumn{1}{c}{} \\ \hline \begin{matrix}\\1\end{matrix} & \begin{matrix}2\\11\end{matrix} & \begin{matrix}6\\57\end{matrix} \\ \hline\end{array} \to \begin{array}{|c|c|c|c|}\cline{1-1} \begin{matrix}8\\67\end{matrix} & \multicolumn{3}{c}{} \\ \cline{1-2} \begin{matrix}5\\4\\233\end{matrix} & 6 & \multicolumn{2}{c}{} \\ \hline \begin{matrix}\\1\end{matrix} & \begin{matrix}2\\11\end{matrix} & \begin{matrix}6\\5\end{matrix} & \begin{matrix}\\7\end{matrix} \\ \hline\end{array} = \mathcal{V}(T).$$

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



Corollary 3.2.1. *Let $T \in \mathsf{HVT}$. Then $P(T)$ is a set-valued tableau.*

Proof. By Lemma 3.2.1 and Definition 3.2.2, 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 3.2.4. *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 $1 \leqslant j \leqslant d$, let $c_j = c_0 + j$ and let $r_j = \tilde{r}$ be $\tilde{r}$ in Definition 3.2.1 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 3.2.2. *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 $0 \leqslant i < m$ be the insertion path of $\mathcal{V}^i(T) \to \mathcal{V}^{i+1}(T)$. Since $c_0^{i+1} \leqslant c_0^i$ for all $0 \leqslant i < m$, 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 $r_j^{i+1} \leqslant r_j^i$ for all $0 \leqslant j \leqslant d_i$. We prove this by induction on j. Note that $r_0^{i+1} \leqslant r_0^i$ by the definition of $\mathcal{U}$. Assume by induction that $r_j^{i+1} \leqslant r_j^i$. 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 $r_{j+1}^{i+1} \leqslant r_{j+1}^i$. □

3.2.2. 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 3.2.1. Furthermore, two words are *Knuth equivalent* if one can be transformed to the other by a sequence of Knuth equivalences on three consecutive letters

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

Lemma 3.2.3. *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. See Appendix B.1.2. $\square$

Remark 3.2.1. *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:*

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

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

Proposition 3.2.1. *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). $\square$

Lemma 3.2.4. *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. See Appendix B.1.3. $\square$

Theorem 3.2.1. *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 3.2.4, 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 3.2.3 and denote the pair of insertion and recording tableaux produced at the j-th step for $0 \leqslant j \leqslant \alpha$ 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 $\mathsf{shape}(P_j(f_i T)) = \mathsf{shape}(f_i(P_j(T))) = \mathsf{shape}(P_j(T))$ for all $0 \leqslant j \leqslant \alpha$. Hence

(3.1)

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

Next we show $Q_j(T) = Q_j(f_i(T))$ for all $0 \leqslant j \leqslant \alpha$ 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 $j \geqslant 0$. It suffices to show that the cells

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

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

in $Q_{j+1}(T)$ and $Q_{j+1}(f_i(T))$ are at the same position with the same entry. By (3.1), the cells are in the same position, say in column $\tilde{c}$. By Definition 3.1.3, 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 3.2.3. 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. $\square$

Hawkes and Scrimshaw [**11**, 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 3.2.2. *The crystal* $\mathsf{HVT}^m(\lambda)$ *of Definition 3.1.3 is a Stembridge crystal of type* A_{m-1}.

Proof. According to [**22**], $\mathsf{SVT}^m(\mu)$ is a Stembridge crystal of type A_{m-1}. By Theorem 3.2.1, 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 [**5**, Chapter 2]). The statement follows. □

3.2.3. 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 [**11**, 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 [**11**, Section 3.2].

Definition 3.2.5. *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 $1 \leqslant k \leqslant \lambda_1$ 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 j-th column of T for every $1 \leqslant j \leqslant \lambda_1$. In other words, if we denote by $T_{\geqslant k}$ the tableau formed by the columns weakly to the right of the k-th column of T, U_k is obtained by performing RSK row insertion using the column reading word of $T_{\geqslant k}$.

(b) Form the tableau F_k of shape $\mathsf{shape}(U_k)/\mathsf{shape}(T_{\geqslant k})$ 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 3.2.2. *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 3.2.2. *Let $T \in \mathsf{MVT}(\lambda)$. Then $\mathcal{U}(T) = \Upsilon(T)$. In other words, the uncrowding map as defined in Definition 3.2.3 is equivalent to the uncrowding map of Definition 3.2.5 in [**11**, Section 3.2].*

Proof. Recall from Definition 3.2.3, 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 $i \geqslant 1$. By [**11**, 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 3.2.1 and Theorem 3.2.1, $P(S) \in \mathsf{SSYT}$ 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 [**11**, 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 [**11**, Theorem 3.17] and $P(f_i T) = f_i P(T)$ by Theorem 3.2.1. 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,

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

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 3.2.5 are in the same positions as those for the uncrowding insertion in Definition 3.2.3. For every $Y \in \mathsf{MVT}(\mu)$ and for every $1 \leqslant j \leqslant \mu_1$, denote by $Y_{\geqslant j}$ the tableau formed by the rightmost j columns of Y; here $Y_{\geqslant \mu_1+1}$ is the empty tableau.

Let $T \in \mathsf{MVT}(\lambda)$ be arbitrary. For $1 \leqslant k \leqslant \lambda_1 + 1$, 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 k-th column of T; here $P^{(\lambda_1+1)} = T$. In other words, $P^{(k)} = \mathcal{V}^{\alpha_k}(T)$ as in Definition 3.2.2, where α_k is the arm excess of $T_{\geqslant k}$. As the entries to the left of column k of T are untouched by the uncrowding insertion in Definition 3.2.2, for every $1 \leqslant k \leqslant \lambda_1 + 1$, we have $(P^{(k)})_{\geqslant k} = P(T_{\geqslant k}) = U(T_{\geqslant k})$. It follows that for every $1 \leqslant k \leqslant \lambda_1$, up to horizontal shifts, the newly formed boxes in $\mathsf{shape}(P^{(k)})/\mathsf{shape}(P^{(k+1)}) = \mathsf{shape}[(P^{(k)})_{\geqslant k+1}]/\mathsf{shape}[(P^{(k+1)})_{\geqslant k+1}]$ and $\mathsf{shape}([U(T_{\geqslant k})]_{\geqslant k+1})/\mathsf{shape}([U(T_{\geqslant k+1})]_{\geqslant k+1})$ are in the same positions. Since the entries in these boxes both record the difference in column indices relative to the k-th column for each $1 \leqslant k \leqslant \lambda_1$ 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. □

3.2.4. 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 3.2.6. *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 $b = \max\{x \in \mathsf{L}_h^+(r, c) \mid x \leqslant m\}$. Otherwise, set $m = \mathsf{H}_h(r, c)$ and $b = \max(\mathsf{L}_h^+(r, c))$.*

(2) *Find the largest r' such that $\mathsf{H}_h(r', c-1) \leqslant b$. 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 $\{x \in \mathsf{L}_h(r, c) \mid q < x \leqslant m\}$ 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 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 Figures 3.3 and 3.4 for illustration.*

Figure 3.3. When $r' = r$. Left: (i) $\mathsf{A}_h(r, c) \neq \emptyset$. Right: (ii) $\mathsf{A}_h(r, c) = \emptyset$.

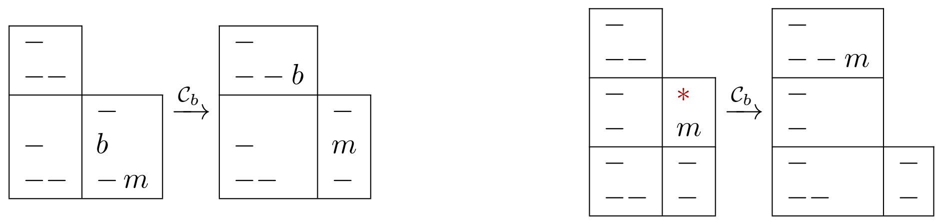
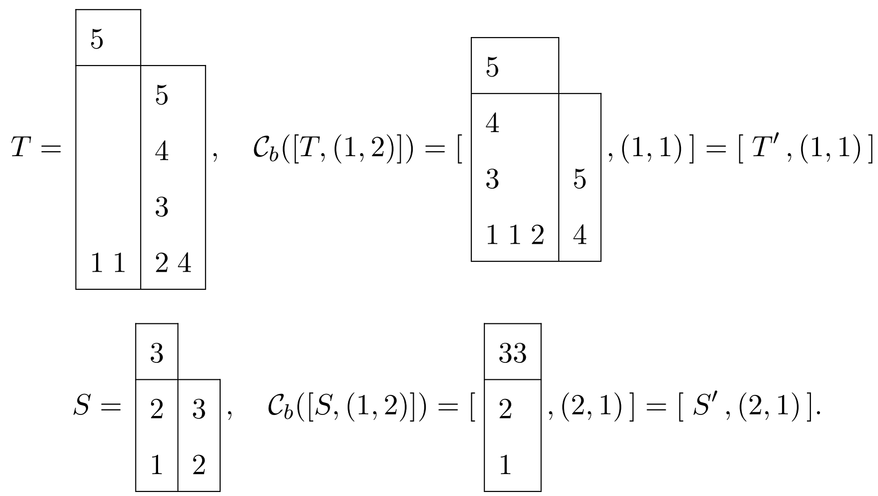


FIGURE 3.4. When $r' \neq r$. Left: $\mathsf{A}_h(r,c) \neq \emptyset$. Right: $\mathsf{A}_h(r,c) = \emptyset$.

EXAMPLE 3.2.3. *We compute $\mathcal{C}_b$ in two examples:*



REMARK 3.2.2. *In Definition 3.2.6,*

- *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 3.2.3, 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 3.2.3 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 $(r_1,c_1),(r_2,c_2),\ldots,(r_e,c_e)$ and $\alpha(F) = (\alpha_1,\alpha_2,\ldots,\alpha_e)$. For all $0 \leqslant j \leqslant e-1$ and for all $0 \leqslant s \leqslant \alpha_{j+1}$, 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

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

REMARK 3.2.3. *The tableaux $T_j^{(s)}$ are well-defined. We check the conditions in Definition 3.2.6. Let $h = T_j^{(s)}$ for some $0 \leqslant j \leqslant e-1$ and for some $0 \leqslant s < \alpha_{j+1}$, 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 $\mathsf{A}_h(r, c)$.*

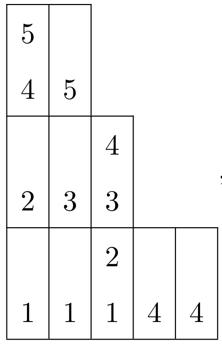
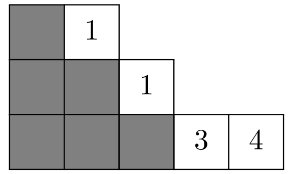
DEFINITION 3.2.7. *With the same notation as above, define the* insertion path *of $T_{j-1}^{(0)} \to T_j^{(0)}$ for $1 \leqslant j \leqslant e$ to be*

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

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

EXAMPLE 3.2.4. *Consider the following pair of tableaux*

$$(S, F) \in \mathsf{HVT}((5,3,2)) \times \hat{\mathcal{F}}((5,3,2)/((3,2,1))),$$

$S =$  , $F =$  .

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

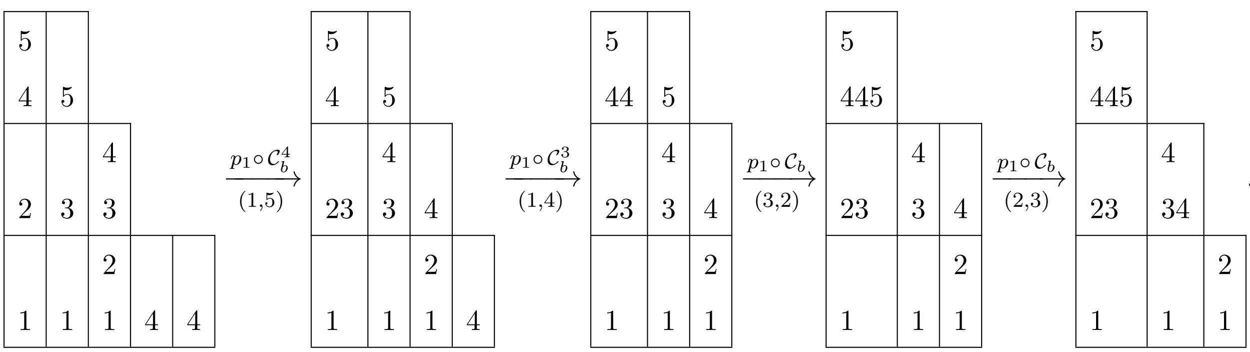
$$\mathsf{path}_1 = ((1,5),(1,4),(2,3),(2,2),(2,1)),\ d(1,5) = 1,$$

$$\mathsf{path}_2 = ((1,4),(2,3),(2,2),(3,1)),\ d(1,4) = 1,$$

$$\mathsf{path}_3 = ((3,2),(3,1)),\ d(3,2) = 1,$$

$$\mathsf{path}_4 = ((2,3),(2,2)),\ d(2,3) = 2.$$

We obtain the sequence from the algorithm:



Lemma 3.2.5. *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 $0 \leqslant s \leqslant \alpha_{j+1}$. We need to show that $r_{j+1}^{(s)} \geqslant r_j^{(s+z_j)}$ for $0 \leqslant s \leqslant \alpha_{j+1}$. Computing $T_{j-1}^{(s)}$ from $T_{j-1}^{(s-1)}$ for $1 \leqslant s \leqslant \alpha_j$, we denote b and q in Step 1 and Step 2 of Definition 3.2.6 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 $r_{j+1}^{(0)} \geqslant r_j^{(z_j)}$. We prove that, for $1 \leqslant s \leqslant \alpha_{j+1}$, we have that $q_{j+1}^{(s)} \geqslant q_j^{(s+z_j)}$, which implies $b_{j+1}^{(s)} \geqslant b_j^{(s+z_j)}$ and thus $r_{j+1}^{(s)} \geqslant r_j^{(s+z_j)}$.

We prove $q_{j+1}^{(s)} \geqslant q_j^{(s+z_j)}$ 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 $y' = \max\{z \in \mathsf{L}^+_{T_{j-1}^{(z_j)}}(r_{j+1}, c_{j+1}) \mid z \leqslant q_j^{(z_j)}\}$. See Figure 3.5 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 $r_{j+1}^{(1)} \neq r_{j+1}^{(0)}$, then $q_{j+1}^{(1)}$ equals y when $y > y'$ and $q_j^{(z_j)}$ when $y = y'$. In both cases $q_{j+1}^{(1)} \geqslant x = q_j^{(z_j+1)}$.

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

FIGURE 3.5. 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).

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

Now we have proved the base case $s = 1$. Next, suppose it holds for some $s \geqslant 1$ that $q_{j+1}^{(s)} \geqslant q_j^{(s+z_j)}$ and $r_{j+1}^{(s)} \geqslant r_j^{(s+z_j)}$. 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 $r_{j+1}^{(s+1)} \geqslant r_j^{(s+1+z_j)}$. 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 $y' = \max\{z \in \mathsf{L}^+_{T_{j-1}^{(s+z_j)}}(r_j^{(s+z_j)}, c_j^{(s+z_j)}) \mid z \leqslant q_j^{(s+z_j)}\}$. See Figure 3.6 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 $q_{j+1}^{(s+1)} = q_j^{(s+z_j)} \geqslant x$. If $r_{j+1}^{(s+1)} \neq r_{j+1}^{(s)}$, then $q_{j+1}^{(s+1)} = \max\{z \in \mathsf{L}^+_{T_j^{(s)}}(r_{j+1}^{(s)}, c_{j+1}^{(s)}) \mid z \leqslant q_{j+1}^{(s)}\} \geqslant q_j^{(s+z_j)} \geqslant x$. So in either case we have $q_{j+1}^{(s+1)} \geqslant q_j^{(s+1+z_j)}$.

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 $\mathsf{H}_{T_{j-1}^{(s+z_j)}}(r_j^{(s+z_j)} + 1, c_j^{(s+z_j)} - 1) \leqslant y' \leqslant q_j^{(s+z_j)}$. 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

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

FIGURE 3.6. 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).

that $r_{j+1}^{(s+1)} \neq r_{j+1}^{(s)}$. Therefore $q_{j+1}^{(s+1)} = \max\{z \in \mathsf{L}^+_{T_j^{(s)}}(r_{j+1}^{(s)}, c_{j+1}^{(s)}) \mid z \leqslant q_{j+1}^{(s)}\}$. By induction we have $q_j^{(s+z_j)} \leqslant q_{j+1}^{(s)}$, thus $q_{j+1}^{(s+1)} \geqslant q_j^{(s+z_j)} \geqslant y' = q_j^{(s+1+z_j)}$. This completes the proof. $\square$

LEMMA 3.2.6. *With the notations as above, let* $0 \leqslant j \leqslant e-1$, $0 \leqslant s < \alpha_{j+1}$ *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 3.2.5 that the cell $(r_j^{(\alpha_j)}, c_j^{(\alpha_j)})$ is also the topmost cell containing an arm. $\square$

PROPOSITION 3.2.3. *The tableau* $T_j^{(s+1)}$ *is a semistandard hook-valued tableau for all* $0 \leqslant j \leqslant e-1$ *and for all* $0 \leqslant s < \alpha_{j+1}$.

PROOF. We only need to check that the q in Step 2 of Definition 3.2.6 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.

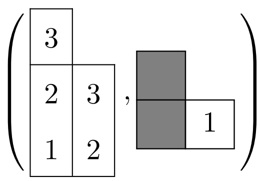
The case when q is not the only arm element can only happen when we reach the destination column of the path. By the proof of Lemma 3.2.5, we have that for $q_{j+1}^{(s)} \geqslant q_j^{(s+z_j)}$ for $s \geqslant 1$ and for j such that $d(r_j, c_j) = d(r_{j+1}, c_{j+1})$. Hence the statement follows 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:

$$\mathsf{K}_\lambda(\mu) := \{(S, F) \in \mathsf{SVT}(\mu) \times \hat{\mathcal{F}}(\mu/\lambda) \mid \mathsf{weight}(T_j^{(s)}) = \mathsf{weight}(S), \forall\, 0 \leqslant j \leqslant e-1, \forall\, 0 \leqslant s \leqslant \alpha_{j+1}\},$$

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

Remark 3.2.4. *From the perspective of the uncrowding map, the set-valued tableau S in Example 3.2.3 cannot be obtained from a shape $(1,1)$ hook-valued tableau via the uncrowding map as explained in Remark 3.2.2. We say the cell $(1,2)$ in S practices social distancing. In this case,*


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

The (S, F) in Example 3.2.4 is in $\mathsf{K}_{(3,2,1)}(5,3,2)$.

Definition 3.2.8. *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 3.2.4. *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 3.2.1 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 3.2.1 and 3.2.6. Thus $a = \max(\mathsf{A}_h(r, c))$. We have that $\mathsf{H}_h(\tilde{r}, c) \leqslant a$. 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.

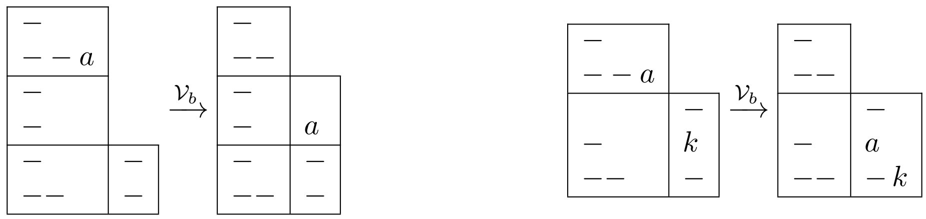


FIGURE 3.7. Left: case (1A): $(\tilde{r}, c+1)$ is not in h. Right: case (1B): $(\tilde{r}, c+1)$ is in h.

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 3(b)i, $q = b = a$, 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.

FIGURE 3.8. Left: Case (1A): $(r, c+1)$ is not in h. Right: Case (1B): $(r, c+1)$ is in 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

$\{x \in \mathsf{L}_{\tilde{h}}(r,c) \mid a < x \leqslant k\}$ 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 $0 \leqslant j \leqslant e-1$, for all $0 \leqslant s \leqslant \alpha_{j+1}$. Thus $\mathsf{image}(\mathcal{U}) \subset \mathsf{K}_\lambda$ and $\mathcal{C} \circ \mathcal{U} = \mathbf{1}_{\mathsf{HVT}(\lambda)}$. □

Proposition 3.2.5. 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 $0 \leqslant j < e$ and for all $0 \leqslant s < \alpha_{j+1}$, $\mathcal{C}_b([T_j^{(s)},(r,c)]) = [T_j^{(s+1)},(r',c-1)]$ for some r,c,r'. We show that $\mathcal{V}_b(T_j^{(s+1)}) = T_j^{(s)}$ for all $0 \leqslant j < e$ and for all $0 \leqslant s < \alpha_{j+1}$. Following the notation in Definition 3.2.1, we first locate the rightmost column that contains nonzero arm excess, 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 3.2.6, 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 $T_j^{(s)}$.

$$\begin{array}{|c|c|} \hline & b \\ - & * \\ -- & m \\ \hline \end{array} \xrightarrow{\mathcal{C}_b} \begin{array}{|l|} \hline b \\ * \\ - \\ --m \\ \hline \end{array} \qquad\qquad \begin{array}{|c|c|} \hline & - \\ & b \\ - & * \\ -- & q\,m \\ \hline \end{array} \xrightarrow{\mathcal{C}_b} \begin{array}{|l|c|} \hline b & \\ * & \\ - & - \\ --q & m \\ \hline \end{array}$$

Figure 3.9. Left: Case (1A): $\mathsf{A}_{T_j^{(s)}}(r,c) = \emptyset$. Right: Case (1B): $\mathsf{A}_{T_j^{(s)}}(r,c) \neq \emptyset$.

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 $b = \max\{x \in \mathsf{L}^+_{T_j^{(s)}} \mid x \leqslant m\}$. $T_j^{(s+1)}$ 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 $\{x \in \mathsf{L}_{T_j^{(s)}(r,c)} \mid q < x \leqslant m\}$ 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 $m \geqslant q$, $\tilde{r} = r$. $\mathcal{V}_b$ 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 $T_j^{(s)}$.

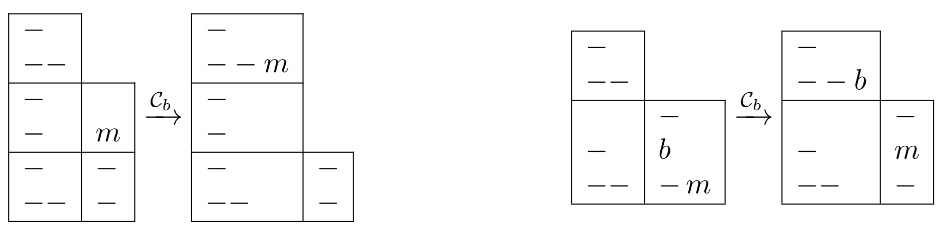


FIGURE 3.10. Left: case (2A): $\mathsf{A}_{T_j^{(s)}}(r,c) = \emptyset$. Right: case (2B): $\mathsf{A}_{T_j^{(s)}}(r,c) \neq \emptyset$.

Case (2B): $\mathsf{A}_{T_j^{(s)}}(r,c) \neq \emptyset$. m is the only element in $\mathsf{A}_{T_j^{(s)}}(r,c)$, $q = b = \max\{x \in \mathsf{L}^+_{T_j^{(s)}}(r,c) \mid x \leqslant m\}$. $T_j^{(s+1)}$ 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 $0 \leqslant j \leqslant e-1$, for all $0 \leqslant s < \alpha_j$, and $\mathcal{V}(T_{j+1}^{(0)}) = T_j^{(0)}$. It follows that we also recover the recording tableau F. Thus $\mathcal{U}(T_e^{(0)}) = (S,F)$. $\square$

Corollary 3.2.3. *The uncrowding map $\mathcal{U}$ is a bijection between $\mathsf{HVT}(\lambda)$ and K_λ with inverse $\mathcal{C}$.*

3.2.5. Alternative uncrowding on hook-valued tableaux. In Section 3.2.1, 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 3.2.9. *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* $\{z \in \mathsf{A}_T(r, c) \mid \ell \leqslant z \leqslant a\}$. *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 3.2.10. *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* $d \geqslant 1$ *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 3.2.11. *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.

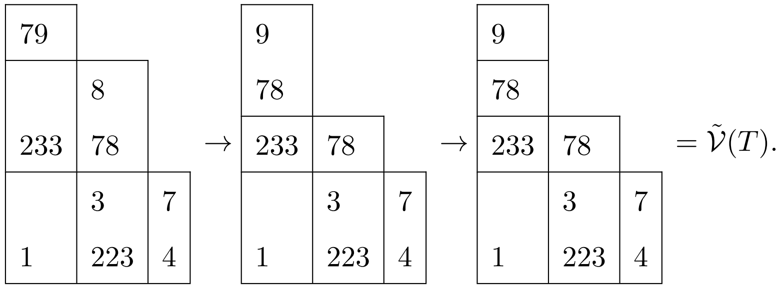
(2) *For* $1 \leqslant i \leqslant \alpha$, *let* $\tilde{P}_{i+1} = \tilde{\mathcal{V}}(\tilde{P}_i)$. *Let* r *be the index of the topmost row of* $\tilde{P}_i$ *containing a cell with nonzero leg excess and let* $\tilde{r}$ *be the row index of the cell* $\mathsf{shape}(\tilde{P}_{i+1})/\mathsf{shape}(\tilde{P}_i)$. *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 $\tilde{\mathcal{U}}(T) = (\tilde{P}(T), \tilde{Q}(T)) := (\tilde{P}_\alpha, \tilde{Q}_\alpha)$.

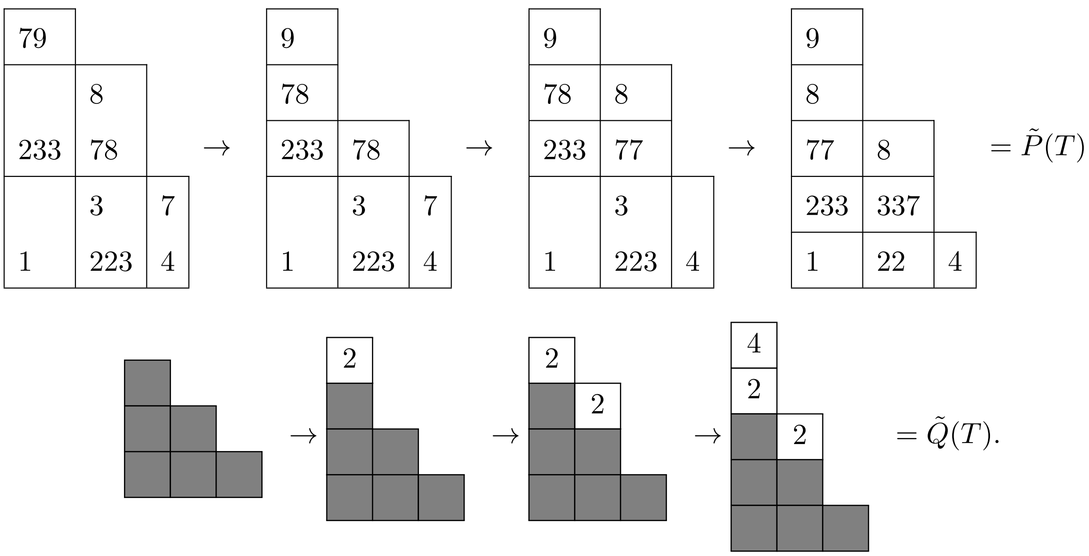
Example 3.2.5. *Let* T *be the hook-valued tableau*

$$T = \begin{array}{|l|l|l|}
\cline{1-1}
79 & \multicolumn{2}{l}{} \\
\cline{1-2}
\begin{array}{l} \\ 233 \end{array} & \begin{array}{l} 8 \\ 78 \end{array} & \multicolumn{1}{l}{} \\
\hline
\begin{array}{l} \\ 1 \end{array} & \begin{array}{l} 3 \\ 223 \end{array} & \begin{array}{l} 7 \\ 4 \end{array} \\
\hline
\end{array} .$$

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



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



PROPOSITION 3.2.6. *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 3.2.1 and Lemma 3.2.2. $\square$

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

THEOREM 3.2.2. *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 3.2.1, Lemma 3.2.4, and Theorem 3.2.1. □

3.3. Applications

In this section, we provide the expansion of the canonical Grothendieck 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 (1.2), 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 [**22**]

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

Definition 3.3.1. *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 3.3.1. *The reading word of $P(T)$ in Example 3.2.1 is $\mathsf{word}(P(T)) = 8675423362111567$.*

Example 3.3.2. *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 3.3.2. *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 $\alpha = (\alpha_i)_{i \geqslant 1}$, 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 3.3.3.

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

Lam and Pylyavskyy [**15**] showed that the dual stable symmetric Grothendieck polynomials $g_\mu(x;1)$ are generating functions of reverse plane partitions of shape μ

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

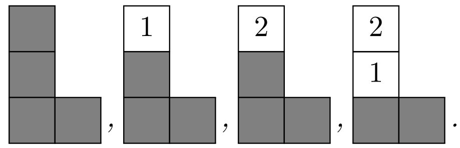
They also provided the Schur expansion of the dual stable symmetric Grothendieck polynomials [**15**, Theorem 9.8]

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

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

EXAMPLE 3.3.4. *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 3.3.1. *[**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*

$$S \in \mathbb{S}(\alpha) \text{ if and only if } P(\mathsf{word}(S)) \in \mathbb{T}(\alpha), \text{ and}$$
$$R \in \mathbb{R}(\alpha) \text{ if and only if } P(\mathsf{word}(R)) \in \mathbb{T}(\alpha),$$

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*

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

Proposition 3.3.1. *The canonical Grothendieck polynomials have a tableaux Schur expansion.*

Proof. Recall the uncrowding map on set-valued tableaux of Definition 1.2.5

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

By Corollary 3.2.3, 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 [**11**, 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

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

$\square$

Corollary 3.3.1. *The canonical Grothendieck polynomials have $G_\mu(x;-1)$ and $g_\mu(x;1)$ expansions:*

$$G_\lambda(x;\alpha,\beta) = \sum_{R\in\mathbb{R}(\lambda)} \mathsf{wt}_\lambda(R) G_{\mathsf{shape}(R)}(x;-1),$$

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

Example 3.3.5. *We compute the first couple of 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*

$$\mathbb{T}((2)) = \left\{ \begin{ytableau} 1 & 1 \end{ytableau}, \begin{ytableau} 2 \\ 1 & 1 \end{ytableau}, \begin{ytableau} 1 & 1 & 1 \end{ytableau}, \begin{ytableau} 2 \\ 1 & 1 & 1 \end{ytableau}, \ldots \right\}.$$

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:

$$\{R \in \mathbb{R}((2)) \mid P(\mathsf{word}(R)) = T_1\} = \{ \begin{ytableau} 1 & 1 \end{ytableau}, \begin{ytableau} 1 \\ 1 & 1 \end{ytableau}, \begin{ytableau} 1 & 1 \\ 1 & 1 \end{ytableau}, \begin{ytableau} 1 \\ 1 \\ 1 & 1 \end{ytableau}, \ldots \}$$

$$\{R \in \mathbb{R}((2)) \mid P(\mathsf{word}(R)) = T_2\} = \{ \begin{ytableau} 2 \\ 1 & 1 \end{ytableau}, \begin{ytableau} 1 & 2 \\ 1 & 1 \end{ytableau}, \begin{ytableau} 2 \\ 1 \\ 1 & 1 \end{ytableau}, \begin{ytableau} 2 \\ 2 \\ 1 & 1 \end{ytableau}, \ldots \}$$

$$\{S \in \mathbb{S}((2)) \mid P(\mathsf{word}(S)) = T_1\} = \{ \begin{ytableau} 1 & 1 \end{ytableau} \}$$

$$\{S \in \mathbb{S}((2)) \mid P(\mathsf{word}(S)) = T_2\} = \{ \begin{ytableau} 2 \\ 1 & 1 \end{ytableau}, \begin{ytableau} 1 & {\begin{array}{c} 2 \\ 1 \end{array}} \end{ytableau} \}.$$

Applying the expansion formulas, we obtain

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

APPENDIX A

Proofs for Crystal for stable Grothendieck polynomials

A.1. Proofs for $\star$-insertion

A.1.1. Proof of Lemma 2.2.1.

Proof. We will prove (1) by induction on the number of cells of P. Statement (2) will follow by some results in the proof of statement (1).

Consider the leftmost column $[q, x]^t$ of $[\mathbf{k}, \mathbf{h}]^t$ and let $[\mathbf{k}', \mathbf{h}']^t$ be the Hecke biword formed by taking the remaining columns in the same order. If the $\star$-insertion of $[\mathbf{k}', \mathbf{h}']^t$ yields (P', Q'), note that we have $P = P' \leftarrow x$. For all integers $j \geqslant 1$, denote by R_j the (possibly empty) j-th row of P'. Denote by u the entry to be inserted into R_j and B_j as the cell in the insertion path at R_j, where $1 \leqslant j \leqslant k$. Additionally, if bumping occurs at R_j, denote the entry bumped out as y.

(1) We will prove that if $(P')^t$ is semistandard, then the transpose of the updated tableau is semistandard.

Case (a): Suppose that the insertion terminates at R_1. Then Case 1 of the $\star$-insertion has occurred, with a cell containing x appended at the end of the row. If R_1 is nonempty, then $x > \max(R_1)$. Additionally, as $(P')^t$ is semistandard, integers strictly increase along R_1 but weakly increase along the column containing B_1. Hence, the transpose of the resulting tableau P is semistandard.

Case (b): Suppose that insertion terminates at R_k, where $k > 1$. We will show that for all $1 \leqslant j \leqslant k$, the changes introduced at row R_j of P' maintain the property that the transpose of the updated tableau is semistandard.

Case (b)(i): Suppose that $j = k$. In this case, a new cell containing u is appended at the end of R_k and $u > \max(R_k)$ if the row is nonempty, proving that the integers increase strictly along R_k.

If Case 2 occurs at R_{k-1}, then u is the entry bumped out of R_{k-1} with the property that when u' is inserted into R_{k-1}, $u \in R_{k-1}$ is the smallest entry with $u > u'$. Let z be the entry below cell B_k. We claim that $z \leqslant u$. If we assume instead that $z > u$, then the cell containing z is strictly to the right of B_{k-1}. However, the cell above B_{k-1} has value greater than u since $(P')^t$ is semistandard and $u \notin R_k$. This contradicts the minimality of u', as u' is greater than this value, hence proving the claim.

If Case 3 occurs at R_{k-1}, then u is bumped out of R_{k-1} with the property that when u' is inserted into R_{k-1}, $u \in R_{k-1}$ is the smallest entry with $[u, u'] \subseteq R_{k-1}$. Let z be the entry below cell B_k. Then, similar to the argument immediately before, $z \leqslant u'$. Hence, we have established that the integers weakly increase along the column containing B_k after u is appended at the end of R_k.

Case (b)(ii): Suppose that $1 \leqslant j < k$ and Case 2 occurs at R_j. Then y is the entry bumped out of R_j with the property that when u is inserted into R_j, $y \in R_j$ is the smallest entry with $y > u$. Thus, as $u \notin R_j$, for all entries z and z' respectively to the left and to the right of B_j, we have $z < u < y < z'$.

If Case 2 occurs at R_{j-1}, then u is bumped out of R_{j-1} with the property that when u' is inserted into R_{j-1}, $u \in R_{j-1}$ is the smallest entry with $u > u'$. Let z be the entry below cell B_j. Then by repeating the same argument as in the first subcase of in Case (b)(i), we obtain $z \leqslant u$.

If Case 3 occurs at R_{j-1}, then u was bumped out of R_{j-1} with the property that when u' is inserted into R_{j-1}, $u \in R_{j-1}$ is the smallest entry with $[u, u'] \subseteq R_{j-1}$. Let z be the entry below cell B_j. Then by repeating the same argument as in the second subcase of in Case (b)(i), we obtain $z \leqslant u'$.

Hence, we have established that integers increase weakly along the column containing B_j but increase strictly along R_j after u bumps out y.

Case (b)(iii): Suppose that $1 \leqslant j < k$ and Case 3 occurs at R_j. In this case, there are no changes to row R_j after inserting u and bumping y. Hence, it is trivial that integers increase weakly along the column containing B_j but increase strictly along R_j after u bumps out y.

In all cases, we have shown that if $(P')^t$ is semistandard, then the transpose of the updated tableau remains semistandard. Therefore, by induction on the number of added cells, we have proved that the insertion tableau P under $\star$-insertion satisfies the property that P^t is semistandard.

Finally, note that the shape of the recording tableau is modified only when Case 1 of the $\star$-insertion has occurred. In this case, a cell is added to form Q at the same position as the cell added to form P. Since we always begin with a pair of empty tableaux, by inducting on the number of added cells, the shapes of P and Q are the same.

(2) Suppose that the insertion terminates at R_k, where $k \geqslant 1$. We shall prove that B_j is weakly to the left of B_{j-1} for all $1 < j \leqslant k$ by revisiting the cases explored in the proof of part (1) (note that P should replace the role of P').

If Case 2 occurs at R_{j-1}, then u is the entry bumped out of R_{j-1} with the property that when u' is inserted into R_{j-1}, $u \in R_{j-1}$ is the smallest entry with $u > u'$. As in the proof of the first subcase of Case (b)(i) in part (1), we conclude that the entry z of the cell below B_k satisfies $z \leqslant u$, showing that B_j is weakly to the left of B_{j-1}.

If Case 3 occurs at R_{j-1}, then u was bumped out of R_{j-1} with the property that when u' is inserted into R_{j-1}, $u \in R_{j-1}$ is the smallest entry with $[u, u'] \subseteq R_{j-1}$. As in the proof of the second subcase of Case (b)(i) in part (1), we conclude that the entry z of the cell below B_j satisfies $z \leqslant u'$, B_j is weakly to the left of B_{j-1}.

This completes the proof. □

A.1.2. Proof of Lemma 2.2.2.

Proof. To prove (2.2), let us first prove the following statements for all row tableaux P:

- With the assumptions in lemma, if insertion terminates at row P while computing $P \leftarrow x$, then

$$\mathsf{row}(P \leftarrow x) \equiv_{\mathcal{H}_0} \mathsf{row}(P) \cdot x.$$

- With the assumptions in lemma, if y is bumped from row P and P changes to P' while computing $P \leftarrow x$, then

$$\mathsf{row}(P \leftarrow x) \equiv_{\mathcal{H}_0} y \cdot \mathsf{row}(P').$$

Assume that insertion terminates at row P while computing $P \leftarrow x$. Then, Case 1 must have occurred and P changes to P', where P' is P appended by a cell containing x. Hence, we have

$$\mathsf{row}(P \leftarrow x) \equiv_{\mathcal{H}_0} \mathsf{row}(P') \equiv_{\mathcal{H}_0} \mathsf{row}(P) \cdot x.$$

Assume that y is bumped from row P and P changes to P' while computing $P \leftarrow x$. Then, either Case 2 or Case 3 must have occurred.

If Case 2 occurs at P, then $x \notin P$ and there is a $y \in P$ with $y > x$; furthermore, y is the smallest value with such property. Write P as AyB, where A and B are the row subtableaux of P formed by entries to the left and to the right of y, respectively. Then, $P \leftarrow x$ is the tableau with row Axb followed by row y. As $x \notin P$, we have $\max(A) < x < y < \min(B)$. Hence by commutativity relations, for all $z \in B$, we have $z \cdot x \equiv_{\mathcal{H}_0} x \cdot z$ and for all $z \in A$, we have $z \cdot y \equiv_{\mathcal{H}_0} y \cdot z$, so that regarding A and B as words in $\mathcal{H}_0(n)$, we obtain

$$A \cdot y \equiv_{\mathcal{H}_0} y \cdot A, \quad B \cdot x \equiv_{\mathcal{H}_0} x \cdot B.$$

It follows that

$$\mathsf{row}(P) \cdot x \equiv_{\mathcal{H}_0} \mathsf{row}(AyB) \cdot x \equiv_{\mathcal{H}_0} A \cdot y \cdot B \cdot x \equiv_{\mathcal{H}_0} y \cdot A \cdot x \cdot B \equiv_{\mathcal{H}_0} y \cdot \mathsf{row}(AxB) \equiv_{\mathcal{H}_0} \mathsf{row}(P \leftarrow x).$$

If Case 3 occurs at P, then $x, y \in P$ with y being the smallest value such that $[y, x] \subseteq P$. Write P as ABC, where $B = [y, x]$, A and C are respectively the row subtableaux of P formed by entries to the left and to the right of B. Then, $P \leftarrow x$ is the tableau with row ABC followed by row y. As $\mathsf{row}(P) \cdot x$ was assumed to be fully-commutative, $x + 1 \notin P$. Furthermore, by minimality of y, $y > \max(A) + 1$. Hence, by commutativity relations, for all $z \in A$, we have $z \cdot y \equiv_{\mathcal{H}_0} y \cdot z$ and for all $z \in C$, we have $x \cdot z \equiv_{\mathcal{H}_0} z \cdot x$, so that

$$A \cdot y \equiv_{\mathcal{H}_0} y \cdot A, \quad C \cdot x \equiv_{\mathcal{H}_0} x \cdot C.$$

Moreover, by using the relations $p-1\ p\ p = p-1\ p-1\ p$, we have $y \cdot B \equiv_{\mathcal{H}_0} B \cdot x$. It follows that

$$\mathsf{row}(P) \cdot x \equiv_{\mathcal{H}_0} \mathsf{row}(ABC) \cdot x \equiv_{\mathcal{H}_0} A \cdot B \cdot C \cdot x \equiv_{\mathcal{H}_0} A \cdot y \cdot B \cdot C \equiv_{\mathcal{H}_0} y \cdot \mathsf{row}(ABC) \equiv_{\mathcal{H}_0} \mathsf{row}(P \leftarrow x).$$

Hence, the two statements above hold for all row tableaux P.

We are now ready to prove (2.2) in full generality. The result follows once we prove by induction on the number of rows of P, with the given setup above, that the following statements hold:

- If the insertion terminates within tableau P while computing $P \leftarrow x$, then

$$\mathsf{row}(P \leftarrow x) \equiv_{\mathcal{H}_0} \mathsf{row}(P) \cdot x.$$

- If y is bumped from tableau P and P changes to P' while computing $P \leftarrow x$, then

$$\mathsf{row}(P \leftarrow x) \equiv_{\mathcal{H}_0} y \cdot \mathsf{row}(P').$$

Indeed, if P is a (possibly empty) row tableau, then we are done by the two previous statements that have been proved. Let $k \geqslant 1$ be an arbitrary integer. Assume that both statements mentioned above hold for all such tableaux P with k rows.

Let P be a tableau with $k+1$ rows with the setup as above. Then, we may consider the subtableau P^* formed from its first k rows and denote the final row as R. Note that $\mathsf{row}(P) = \mathsf{row}(R) \cdot \mathsf{row}(P^*)$ and $\mathsf{row}(R)$ is fully-commutative.

Assume that the changes from P to $P \leftarrow x$ involve at most the first k rows of P. Then $P \leftarrow x$ is the same tableau as $P^* \leftarrow x$ with an extra row R, so that by the inductive hypothesis,

$$\mathsf{row}(P \leftarrow x) \equiv_{\mathcal{H}_0} \mathsf{row}(R) \cdot \mathsf{row}(P^* \leftarrow x) \equiv_{\mathcal{H}_0} \mathsf{row}(R) \cdot \mathsf{row}(P^*) \cdot x \equiv_{\mathcal{H}_0} \mathsf{row}(P) \cdot x.$$

Now assume that the changes from P to $P \leftarrow x$ involves all $k+1$ rows of P. Let P' be the resulting tableau after performing these changes on P^* and let y be the entry bumped from the final row of P^*. Then, $P \leftarrow x$ is the tableau obtained by concatenating tableau $R \leftarrow y$ after P'.

If the insertion terminates at row R, then by the previous statements for all row tableaux and the inductive hypothesis, we obtain

$$\mathsf{row}(P \leftarrow x) \equiv_{\mathcal{H}_0} \mathsf{row}(R \leftarrow y) \cdot \mathsf{row}(P') \equiv_{\mathcal{H}_0} \mathsf{row}(R) \cdot y \cdot \mathsf{row}(P')$$
$$\equiv_{\mathcal{H}_0} \mathsf{row}(R) \cdot \mathsf{row}(P^* \leftarrow x) \equiv_{\mathcal{H}_0} \mathsf{row}(R) \cdot \mathsf{row}(P^*) \cdot x \equiv_{\mathcal{H}_0} \mathsf{row}(P) \cdot x.$$

Otherwise, if the insertion bumps z from R and R changes to R' while computing $R \leftarrow y$, then it holds that the insertion bumps z from P while computing $P \leftarrow x$. In this case, if we denote P'' as the tableau P' concatenated by row R', then

$$\mathsf{row}(P) \cdot x \equiv_{\mathcal{H}_0} \mathsf{row}(R \leftarrow y) \cdot \mathsf{row}(P') \equiv_{\mathcal{H}_0} z \cdot \mathsf{row}(R') \cdot \mathsf{row}(P') \equiv_{\mathcal{H}_0} z \cdot \mathsf{row}(P'') \equiv_{\mathcal{H}_0} \mathsf{row}(P \leftarrow x).$$

This completes the induction. □

A.1.3. Proof of Lemma 2.2.3.

Proof. Similar to Fulton's proof [**9**] of the Row Bumping Lemma, we will keep track of the entries as they are bumped from a row. Consider a row R of tableau P and suppose that u and u' are to be inserted into R when computing $P \leftarrow x$ and $(P \leftarrow x) \leftarrow x'$ respectively, where $u < u'$. Denote by C (similarly C') the box in π (similarly π') that is also in R.

Case 1: $x < x'$. We will prove that the following assertions hold for R:

(a) If the insertion terminates at R while computing $P \leftarrow x$, then the insertion terminates at R while computing $(P \leftarrow x) \leftarrow x'$.

(b) C' is strictly to the right of C.

Note that the insertion terminates at R when computing $P \leftarrow x$ precisely when Case 1 of the $\star$-insertion occurs at R. Box C containing u is appended at the end of R. As $u' > u$, Case 1 occurs again at R with box C' containing u' appended to the right of C, so bumping does not occur at R when computing $(P \leftarrow x) \leftarrow x'$. This proves (a) and simultaneously, (b) for this case.

Let us assume that bumping occurs at R with y bumped out when computing $P \leftarrow x$.

Case A: If y is bumped from R because Case 2 occurs, the insertion at row R introduced to box C' occurs strictly to the right of C (containing u) because:

(i) If $u' > \max(R)$, then box C' containing u' is appended to the end of R by Case 1. In particular, C' appears strictly to the right of C.

(ii) Otherwise, since $u' > u$, the letter u' is inserted into a box C' strictly to the right of C with y' bumped out. If $u' \notin R$, Case 2 occurs and $y' > y$ because C' and C originally contained y' and y respectively. Else, $u' \in R$ and Case 3 occurs. Suppose that $[y', u']$ is the longest interval of consecutive integers contained in R. Since box C that originally contained y is strictly to the left of C', we have $u < y < u'$. Therefore, $[u, u']$ cannot be contained in R, so $y < y'$.

Case B: Otherwise, y is bumped from R because Case 3 occurs when computing $P \leftarrow x$ and $[y, u]$ is the longest interval of consecutive integers contained in R by Remark 2.2.2. The insertion at row R introduced to box C' occurs strictly to the right of C (containing u) because:

(i) If either $u' > \max(R)$ or $u' \notin R$, then by similar arguments as in Case A(i) and Case A(ii), C' appears to the right of C. Furthermore, in the latter situation, by a similar argument in Case A(ii), we have $y < y'$.

(ii) Otherwise, $u' \in R$ and Case 3 occurs. As $u' > u$, u' is inserted into box C' strictly to the right of C with y' bumped out. In addition, $[y', u']$ is the longest interval of consecutive integers contained in R. As $\mathsf{row}(R)$ is fully-commutative before computing $P \leftarrow x$, $u + 1 \notin R$. Hence $[u, u']$ cannot be contained in R. It follows that $y \leqslant u < u + 1 < y'$.

Note that in the arguments above, we have also shown that if y and y' are bumped from R when computing $P \leftarrow x$ and $(P \leftarrow x) \leftarrow x'$ respectively, then $y < y'$. It follows that we may apply similar arguments in the rows following R. Since assertion (b) now holds for all rows, we conclude that π' is strictly to the right of π. In addition, π' cannot continue after π ends because of assertion (a). Considering that π' goes weakly left by Lemma 2.2.1, we conclude that box B' is strictly to the right of and weakly below B.

Case 2: $x \geqslant x'$. We will prove that the following assertions hold for R:

(1) If the insertion terminates at R while computing $P \leftarrow x$, then bumping occurs at R while computing $(P \leftarrow x) \leftarrow x'$.

(2) C' is weakly to the left of C.

If the insertion terminates at row R when computing $P \leftarrow x$, then Case 1 occurs and box C containing u is appended at the end of R. If $u' \in R$, Case 3 occurs at R with $y' \leqslant u' \leqslant u$ bumped out. Furthermore, box C' containing u' is weakly to the left of C. If $u' \notin R$, Case 2 occurs at R with $y' > u'$ bumped out and $u' < u$. We have $y' \leqslant u$ by minimality of y', so that box C' is weakly to the left of C. In either of the subcases, bumping occurs at R when computing $(P \leftarrow x) \leftarrow x'$. This proves (a) and simultaneously, (b) for this case.

Let us assume that bumping occurs at R with y bumped out when computing $P \leftarrow x$.

Case A: If y is bumped from R because Case 2 occurs when computing $P \leftarrow x$, the insertion at row R introduced to box C' occurs weakly to the left of C (containing u) because:

(i) If $u' \notin R$, then u' is inserted into box C' containing y' by Case 2, while bumping out this y'. As $u' < u$, we have $y' \leqslant u < y$ and that C' appears weakly to the left of C.

(ii) Otherwise, $u' \in R$ and Case 3 occurs. The letter u' is inserted into box C' weakly to the left of C as $u' \leqslant u$. In addition, if $[y', u']$ is the longest interval of consecutive integers in R, then y' is bumped out. Furthermore, we have $y' < y$ as C, which originally contained y before computing $P \leftarrow x$, is to the right of the box containing y'.

Case B: Otherwise, y is bumped from R because Case 3 occurs when computing $P \leftarrow x$. Let $[y, u]$ be the longest interval of consecutive integers that is contained in R. The insertion at row R introduced to box C' occurs weakly to the left of C (containing u) because:

(i) If $u' \notin R$, then $u' < u$, u' is inserted into box C' containing y' and y' is bumped out by Case 2. As $\mathsf{row}(P) \cdot x$ is fully-commutative, in particular $\mathsf{row}(R)$ is fully-commutative. Hence $u' < y$, so that C' is weakly to the left of box containing y (hence also weakly to the left of C). Furthermore, we have $y' \leqslant y$ by the minimality of y'.

(ii) If $u' \in R$, then either $u' = u$ or $u < u'$. The former case is easy as Case 3 occurs again with u' inserted into $C' = C$ and $y' = y$ is bumped out. If $u < u'$, then as $\mathsf{row}(P) \cdot x$ is fully-commutative, $\mathsf{row}(R)$ is fully-commutative, so that $u' < y - 1$. It follows that C' is strictly to the left of box containing y (hence also strictly to the left of C). Furthermore, we have $y' \leqslant u' < y-1 < y$.

Note that in the arguments above, we have also shown that if y and y' are bumped from R when computing $P \leftarrow x$ and $(P \leftarrow x) \leftarrow x'$ respectively, then $y \geqslant y'$. It follows that we may apply similar arguments in the rows following R. Since assertion (b) now holds for all rows, we conclude that π' is weakly to the left of π. In addition, π' must continue after π ends because of assertion (a). Considering that π' goes weakly left by Lemma 2.2.1, we conclude that box B' is weakly to the left of and strictly above B. $\square$

A.2. Proofs of micro-moves

A.2.1. Proof of Lemma 2.3.1.

Proof. Let R be a single-row increasing tableau and M be the largest letter in R. First note that if $a \in R$ and $\mathsf{row}(R) \cdot a$ is fully-commutative, then $a+1 \notin R$, see also Remark 2.2.2.

There are five types of equivalence triples, so we discuss them in 3 groups.

1. Cases (I1) and (II1): We have $x < z < y$, or $x < z = y$ and $y > x+1$. In both cases $x' = y, y' = x, z' = z$.

Case (1A): $M < x < z \leqslant y$. In this case, the first resulting tableau is $R^{xyz} = \boxed{R \mid x \mid z}$ and the outputs are $R(x) = R^x(y) = 0$ and $R^{xy}(z) = y$. The second resulting tableau is $R^{yxz} = \boxed{R \mid x \mid z}$ and the outputs are $R(y) = 0 = R^{yx}(z)$ and $R^y(x) = y$. So we have $R^{xyz} = R^{yxz}$ and also $0 \cdot 0 \cdot y \sim 0 \cdot y \cdot 0$.

Case (1B): $x \leqslant M < z \leqslant y$. In this case, we have $R^{xy} = R^{yx}$ and $R(x) = R^y(x)$ since y is just appended to the end of R and does not influence how x is inserted. This gives $R^{xyz} = R^{yxz}$. The related outputs are $R^x(y) = R(y) = 0$, $R^{xy}(z) = R^{yx}(z) = y$. Thus, $R(x) \cdot 0 \cdot y \sim 0 \cdot R(x) \cdot y$.

Case (1C): $x < z \leqslant M < y$. In this case, we also have that $R^{xy} = R^{yx}$ and $R(x) = R^y(x)$, for the same reason as case (1B). Thus, we have $R^{xyz} = R^{yxz}$ and $R^{xy}(z) = R^{yx}(z)$. Since we have $R^x(y) = R(y) = 0$, $R(x) \cdot 0 \cdot R^{xy}(z) \sim 0 \cdot R^y(x) \cdot R^{yx}(z)$.

Case (1D): $x < z \leqslant y \leqslant M$. If x is the maximal letter in R^x, then it follows as case (1B). Otherwise, this case needs further separation into subcases.

Case 1D-(i): $x, y \notin R$. Then $x < R(x), y < R(y)$ and $R(x) \neq y$.

(1) If $R(x) < y$, then $R^x(y) = R(y)$ and $R^y(x) = R(x)$, which implies $R^{xy} = R^{yx}$, thus $R^{xyz} = R^{yxz}$ and $R^{xy}(z) = R^{yx}(z)$. Hence $R(x)R^x(y)R^{xy}(z) = R^y(x)R(y)R^{yx}(z)$. Since $R(x) < R^{xy}(z) \leqslant y < R(y)$, we have $R(y) > R^y(x) + 1$ and for the outputs $R(x)R^x(y)R^{xy}(z) = R^y(x)R(y)R^{yx}(z) \sim R(y)R^y(x)R(y)R^{yx}(z)$ by move type (I1) or (II1).

(2) If $R(x) > y$, let the letter to the right of $R(x)$ in R be $R(x)^{\rightarrow}$. Then both R^{xyz} and R^{yxz} are obtained by replacing $R(x)$ with x and $R(x)^{\rightarrow}$ with z. For the output, we have $R(x) = R(y)$, $R^x(y) > R(y)$, $R^y(x) = y$, $R^{yx}(z) = R^x(y)$ and $R^{xy}(z) = y$. Since $y < R(x) < R^x(y)$, we have that $R^x(y) = R(x)^{\rightarrow} > y + 1$. Hence the outputs $R(x)R^x(y)R^{xy}(z) = R(x)R(x)^{\rightarrow}y \sim R(x)yR(x)^{\rightarrow} = R(y)R^y(x)R^{yx}(z)$ by move of type (I2).

Case 1D-(ii): $x \in R, y \notin R$. Then $R(x) \leqslant x, R(y) > y$ and $x + 1 \notin R$. In this case, we have $R^x(y) = R(y)$ and $R^y(x) = R(x)$, thus $R^{xy} = R^{yx}$, $R^{xy}(z) = R^{yx}(z)$ and $R^{xyz} = R^{yxz}$. Since $x + 1 \notin R$, we have $R^{xy}(z) > x + 1$. This implies $R(x) \leqslant x < R^{xy}(z) \leqslant y < R(y)$, thus $R(x)R^x(y)R^{xy}(z) \sim R(y)R^y(x)R^{yx}(z)$ as it is a type (I1) move.

Case 1D-(iii): $x \notin R, y \in R$. Then $x < R(x)$, $y \geqslant R(y)$, $y + 1 \notin R$, $R(x) - 1 \notin R$, $R(x) \leqslant y$, $R(y) \leqslant R^x(y)$ and $R^y = R$.

(1) If $R(x) = y$, denote the box to the right of y in y as $y^{\rightarrow}$. Note that $y^{\rightarrow} > y + 1$. Then $R^x(y) = y^{\rightarrow}$, $R^{xy}(z) = y$, $R^y(x) = y$ and $R^{yx}(z) = y^{\rightarrow}$. Note $y - 1 \notin R$, otherwise $R(x) \leqslant y - 1$. Thus, $R(y) = y$. Both R^{xyz} and R^{yxz} are obtained by replacing $y \in R$ with x and $y^{\rightarrow}$ with z, so $R^{xyz} = R^{yxz}$. The outputs $R(x)R^x(y)R^{xy}(z) = yy^{\rightarrow}y \sim yyy^{\rightarrow} = R(y)R^y(x)R^{yx}(z)$ as it is a type (II2) move.

(2) Suppose $R(x) < y$ and $R(x) = R(y)$. Then $[R(x), y] \subset R$ and $R^x(y) = R(x) + 1$. Since $R^y = R$ and $R^{xy} = R^x$, we have that both R^{xy} and R^{yx} equal R^x and furthermore $R^y(x) = R(x)$. Note that z can either be equal to y or $z < R^x(y)$, otherwise $z \in R^{xy}$ and $z + 1 \in R^{xy}$, which will give us a braid from $\mathsf{row}(R^{xy}) \cdot z$. Thus, we have $R^{xy}(z) = R^{yx}(z) = R(x) + 1$. In either case, the outputs are $R(x)R^x(y)R^{xy}(z) = R(x)(R(x) + 1)(R(x) + 1) \sim R(x)R(x)(R(x) + 1) = R(y)R^y(x)R^{yx}(z)$ as they are type (III) moves.

(3) Suppose $R(x) < y$ and $R(x) < R(y)$. Then $R(y) > R(x)+1$ and $R^x(y) = R(y)$. Similar to the previous case, both R^{xy} and R^{yx} are equal to R^x, and z is either y or $z < R(y)$. In either case, $R^{xy}(z) \leqslant R(y)$.

Then the outputs are $R(x)R^x(y)R^{xy}(z) = R(x)R(y)R^x(z) \sim R(y)R(x)R^x(z) = R(y)R^y(x)R^{yx}(z)$ as they are type (I1) or (II1) moves.

Case 1D-(iv): $x, y \in R$. In this case $x \geqslant R(x)$, $y \geqslant R(y)$, $x+1 \notin R$ and $y+1 \notin R$. Since $x+1 \notin R$, $[x,y]$ is not contained in R and hence $R(y) > x+1 > x \geqslant R(x)$.

Then $R^x(y) = R(y)$, $R^y(x) = R(x)$ and $R^{xy} = R^{yx} = R$. Since $z > x$ and $x+1 \notin R$, we have $R(z) > x+1 \geqslant R(x)+1$. By similar reasons to the previous two subcases of Case 1D-(iii), z can either be y or $z < R(y)$ in order to avoid a braid in $\mathsf{row}(R^{xy})z$. So, we have $R^{xy}(z) \leqslant R(y)$. Then the outputs are $R(x)R^x(y)R^{xy}(z) = R(x)R(y)R(z) \sim R(y)R(x)R(z) = R(y)R^y(x)R^{yx}(z)$ as they are type (I1) moves.

2. Cases (I2) and (II2): We have $z < x < y$, or $z = x < y$ and $y > z+1$. In both cases $x' = x, y' = z, z' = y$. By definition, $x \in R^x$.

Case (2A): $M < x < y$, then $R(x) = R^x(y) = 0$. $R^{xy} = \begin{array}{|c|c|c|}\hline R & x & y \\ \hline\end{array}$ is obtained by appending x and y to the end of R. Since $x \in R^x$ and $z \leqslant x < y$, we have $R^{xy}(z) = R^x(z)$. Moreover, R^{xzy} is obtained by appending y to the end of R^{xz} and hence $R^{xyz} = R^{xzy}$. The outputs are $R(x)R^x(y)R^{xy}(z) = 00R^x(z) \sim 0R^x(z)0 = R(x)R^x(z)R^{xz}(y)$.

Case (2B): $z \leqslant x \leqslant M < y$, then $R^x(y) = R^{xz}(y) = 0$. Since $R^{xy} = \begin{array}{|c|c|}\hline R^x & y \\ \hline\end{array}$, $x \in R^x$ and $z \leqslant x$, we have $R^{xy}(z) = R^x(z)$, thus $R^{xyz} = \begin{array}{|c|c|}\hline R^{xz} & y \\ \hline\end{array} = R^{xzy}$. The output $R(x)R^x(y)R^{xy}(z) = R(x)0R^x(z) \sim R(x)R^x(z)0 = R(x)R^x(z)R^{xz}(y)$.

Case (2C): $z \leqslant x < y \leqslant M$, then we have $R^x(z) \leqslant x$. We discuss the following subcases.

Case 2C-(i): $x, y \notin R$, then we have $R(x) > x$ and $R^x(y) > y$. Since $y > x$ and x replaces $R(x)$ in R, we have $R^x(y) > R(x)$ from row strictness. Since $R^x(y) > R(x)$ and $R^x(z) \leqslant x$, we have $R^{xy}(z) = R^x(z)$ and $R^{xz}(y) = R^x(y)$. Furthermore, $R^{xyz} = R^{xzy}$. Moreover, we have $R^x(z) \leqslant x < R(x) < R^x(y)$, which implies $R^x(y) > R^x(z)+1$. Hence $R(x)R^x(y)R^{xy}(z) = R(x)R^x(y)R^x(z) \sim R(x)R^x(z)R^x(y) = R(x)R^x(z)R^{xz}(y)$ by type (I2) moves.

Case 2C-(ii): $x \in R, y \notin R$. Then $R^x = R$, $R(x) \leqslant x, R^x(y) > y$. Since $z \leqslant x$ and $[R(x), x] \subset R^x$, we have that $R^x(z) \leqslant R(x)$. Since $R^x(y) > y > x$ and $R^x(z) \leqslant R(x)$, we have that $R^{xy}(z) = R^x(z)$ and $R^{xz}(y) = R^x(y)$, thus $R^{xyz} = R^{xzy}$. Since $R^x(z) \leqslant R(x) \leqslant x < y < R^x(y)$, we have $R^x(y) > R^x(z) + 1$. The outputs are $R(x)R^x(y)R^{xy}(z) = R(x)R^x(y)R^x(z) \sim R(x)R^x(z)R^x(y) = R(x)R^x(z)R^{xz}(y)$ by type (I2) or (II2) moves.

Case 2C-(iii): $x \notin R, y \in R$. Then $R^{xy} = R^x$, $R(x) > x$ and $R^x(y) \leqslant y$. Let the letter to the right of $R(x)$ in R be $R(x)^{\rightarrow}$. Then $R(x)^{\rightarrow} > R(x) > x$ implies $R(x)^{\rightarrow} > x + 1$. This also shows that $x + 1 \notin R^x$ and thus $R^x(y) > R(x)$. Since $R^{xy} = R^x$ and $R^{xzy} = R^{xz}$, we have $R^{xyz} = R^{xz} = R^{xzy}$. Since $R^x(z) \leqslant x < R(x) < R^x(y)$, we have $R^x(y) > R^x(z) + 1$. Since $z \leqslant x$, we also have that $R^x(y) = R^{xz}(y)$. Thus, the outputs are $R(x)R^x(y)R^{xy}(z) = R(x)R^x(y)R^x(z) \sim R(x)R^x(z)R^x(y) = R(x)R^x(z)R^{xz}(y)$ by a type (I2) move.

Case 2C-(iv): $x \in R, y \in R$. Then $R^x = R$, $R^{xy} = R$, $R(x) \leqslant x$, $R^x(y) \leqslant y$, $x + 1 \notin R$ and $y + 1 \notin R$. Thus, $R^x(y) > x + 1$. Since $z \leqslant x$ and $[R(x), x] \subset R^x$, we have that $R^x(z) \leqslant R(x)$. Since $R^{xy} = R$, $R^{xyz} = R^z$. Since $R^x(z) \leqslant x$, $R^{xz}(y) = R^z(y) = R(y)$ and thus $R^{xzy} = R^z$. This implies $R^{xyz} = R^{xzy}$. Now we have $R(z) \leqslant R(x) \leqslant x < x + 1 < R(y)$. Therefore, the outputs are $R(x)R^x(y)R^{xy}(z) = R(x)R(y)R(z) \sim R(x)R(z)R(y) = R(x)R^x(z)R^{xz}(y)$ by type (I2) or (II2) moves.

3. Case (III): We have $y = x$, $z = x + 1$ and hence $x' = x$, $y' = x + 1$ and $z' = x + 1$.

Case (3A): $x > M$. Then R^x is obtained by appending x to the end of R and $R(x) = 0$. Also $R^{xx} = R^x$ with output $R^x(x)$. Note $R^{x,x+1}(x + 1) = R^x(x)$. Both $R^{xx,x+1}$ and $R^{x,x+1,x+1}$ are obtained by appending $x + 1$ to the end of R^x, thus they are the same. The outputs are $R(x)R^x(x)R^{xx}(x + 1) = 0R^x(x)0 \sim 00R^x(x) = R(x)R^x(x + 1)R^{x,x+1}(x + 1)$.

Case (3B): $x \leqslant M, x + 1 > M$. Both $R^{xx,x+1}$ and $R^{x,x+1,x+1}$ are obtained by appending $x + 1$ to the end of R^x, so they are equal. Since $x \in R^x$, we have $R^{x,x+1}(x + 1) = R^x(x)$. Thus, the outputs are $R(x)R^x(x)R^{xx}(x + 1) = R(x)R^x(x)0 \sim R(x)0R^{x,x+1}(x + 1)$.

Case (3C): $x + 1 \leqslant M$. It is clear that $x \in R^x$. If x is the maximal letter in R^x, then the rest follows as case (3B).

Otherwise, let $x^{\to}$ be the letter to the right of x in R^x. Since $x \in R^x$, we must have $x+1 \notin R^x$, thus $x^{\to} > x+1$. Moreover, we have $R^{xx} = R^x$, $R^x(x+1) = R^{xx}(x+1) = x^{\to}$. Since $R^{x,x+1}$ is obtained from R^x by replacing $x^{\to}$ with $x+1$ and $x, x+1 \in R^{x,x+1}$, we have $R^{x,x+1}(x+1) = R^x(x)$. Both $R^{xx,x+1}$ and $R^{x,x+1,x+1}$ are obtained from R^x by replacing $x^{\to}$ with $x+1$, thus they are the same. Furthermore, since $R^x(x) \leqslant x$ and $x^{\to} > x+1$, we have that $R(x)R^x(x)R^{xx}(x+1) = R(x)R^x(x)x^{\to} \sim R(x)x^{\to}R^x(x) = R(x)R^x(x+1)R^{x,x+1}(x+1)$ by a type (I2) or (II2) move. $\square$

A.2.2. Lemmas for the proof of Proposition 2.3.2.

LEMMA A.2.1.

(1) For $2 \leqslant i \leqslant q$, $b_{i-1} > u_i + 1$.

(2) For $1 \leqslant i < s$, $v_i > w_{p-s+i+1} + 1$.

PROOF. **(1):** When $b_{i-1} > b_i + 1$ or $u_i < b_i$, the result follows directly.

Consider the case that $u_i = b_i = a$ and $b_{i-1} = b_i + 1 = a + 1$ for some letter a. Since $a = u_i < u_{i-1} \leqslant b_{i-1} = a+1$, we must have $u_{i-1} = a+1$. Let c be the largest letter such that $[a, c] \subseteq b$. Then $c \geqslant a+1$ and $c+1 \notin b$. Moreover, since all u_i are paired, $u_i \leqslant b_i$ and $u_{j-1} > u_j$, it is not hard to see that $[a, c] \subseteq u$ and $c, c-1 \in u$. Since $c+1 \notin b$, we can use commutativity to move $c \in b$ to the left and obtain a subword $c(c-1)c$, which contradicts that the original word is fully-commutative.

(2): The proof is almost identical to the first part. When $w_{p-s+i} > w_{p-s+i+1} + 1$ or $v_i > w_{p-s+i}$, the result follows.

Consider the case $w_{p-s+i} = w_{p-s+i+1} + 1 = a + 1$ and $v_i = w_{p-s+i} = a + 1$ for some letter a. Since $a = w_{p-s+i+1} \leqslant v_{i+1} < v_i = a+1$, we must that $v_{i+1} = a$. Let c be the smallest letter such that $[c, a+1] \subseteq w$. Then $c \leqslant a$ and $c - 1 \notin w$. Moreover, since all v_i are paired, $v_j \geqslant w_{p-s+j}$ and $v_{j+1} < v_j$, we can see that $[c, a+1] \subseteq v$ and $c, c+1 \in v$. Since $c-1 \notin w$, we can use commutativity to move $c \in w$ to the right and form a subword $c(c+1)c$, which contradicts that the original word is fully-commutative. $\square$

We now summarize several observations that will be used later.

Remark A.2.1. *For both types of actions of $f_k^\star$ as in Remark 2.3.1, we have the following equivalence relations:*

(1) *For* $1 \leqslant i \leqslant q, 1 \leqslant j \leqslant s-1$, $v_{j+1}v_ju_i \sim v_{j+1}u_iv_j$, *since* $u_i < v_{j+1} < v_j$.

(2) *For* $1 \leqslant i \leqslant q$, $xv_su_i \sim xu_iv_s$, *since* $u_i < x < v_s$.

(3) *For* $1 \leqslant i \leqslant q$, $b_1xu_i \sim b_1u_ix$, *since* $u_i \leqslant u_1 \leqslant b_1 < x$, *and* $u_i < x-1$.

(4) *For* $1 \leqslant j < i-1$, $1 \leqslant i \leqslant q$, $b_{j+1}b_ju_i \sim b_{j+1}u_ib_j$, *since* $u_i \leqslant b_i < b_{j+1} < b_j$.

(5) *For* $2 \leqslant i \leqslant q$, $b_ib_{i-1}u_i \sim b_iu_ib_{i-1}$, *since* $u_i \leqslant b_i < b_{i-1}$ *and* $b_{i-1} > u_i+1$ *by Lemma A.2.1.*

(6) *For* $1 \leqslant i \leqslant s$, $p-s+i-1 \leqslant j \leqslant p-1$, $w_{j+1}v_iw_j \sim v_iw_{j+1}w_j$, *since* $w_{j+1} < w_j < w_{p-s+i} \leqslant v_i$.

(7) *For* $1 \leqslant i \leqslant s-1, w_{p-s+i+1}v_iw_{p-s+i} \sim v_iw_{p-s+i+1}w_{p-s+i}$, *since* $w_{p-s+i+1} < w_{p-s+i} \leqslant v_i$ *and* $v_i > w_{p-s+i+1}+1$ *by Lemma A.2.1.*

(8) *For all* $1 \leqslant j \leqslant s-1$, $1 \leqslant i \leqslant q, v_{j+1}u_iv_j \sim v_{j+1}v_ju_i$, *since* $u_i < v_{j+1} < v_j$.

(9) *For* $1 < i \leqslant q$, $b_1u_iv_s \sim b_1v_su_i$, *since* $u_i < u_1 \leqslant b_1 < v_s$.

(10) *For* $1 \leqslant i \leqslant q$, $1 \leqslant j \leqslant s$, $xu_iv_j \sim xv_ju_i$, *since* $u_i < x < v_j$.

(11) *For* $1 \leqslant j \leqslant s-1$, $xv_jw_p \sim v_jxw_p$, *since* $x < w_p \leqslant v_s < v_j$.

Remark A.2.2. *When $v_s \neq x+1$, we have the following equivalence relations:*

(1) $1 \leqslant i \leqslant s$, $xv_iw_p \sim v_ixw_p$, *since* $x < w_p \leqslant v_s$ *and* $v_s > x+1$.

(2) $b_1u_1v_s \sim b_1v_su_1$, *since* $u_1 \leqslant b_1 < v_s$ *and* $v_s > x+1 > u_1+1$.

Lemma A.2.2. *We have that* $b_q \dots b_1xv_s \dots v_1u_q \dots u_1$ *is equivalent to* $b_qu_q \dots b_2u_2b_1u_1xv_s \dots v_1$.

Proof. With the equivalence relations from Remark A.2.1 (1)-(5), we can make the sequences of equivalence moves as follows:

$$\begin{aligned}
&b_q \dots b_1xv_s \dots v_2v_1u_qu_{q-1} \dots u_1 \sim b_q \dots b_1xv_s \dots v_2u_qv_1u_{q-1} \dots u_1 \sim \\
&b_q \dots b_1xv_su_q \dots v_2v_1u_{q-1} \dots u_1 \sim b_q \dots b_1xu_qv_s \dots v_2v_1u_{q-1} \dots u_1 \sim \\
&b_q \dots b_1u_qxv_s \dots v_2v_1u_{q-1} \dots u_1 \sim b_qu_q \dots b_1xv_s \dots v_2v_1u_{q-1} \dots u_1 \sim \\
&b_qu_qb_{q-1}u_{q-1} \dots b_1u_1xv_s \dots v_2v_1.
\end{aligned}$$

□

LEMMA A.2.3. *We have that $v_s \dots v_1 w_p \dots w_{p-s+1}$ is equivalent to $v_s w_p v_{s-1} w_{p-1} \dots v_1 w_{p-s+1}$.*

PROOF. With the equivalence relations from Remark A.2.1 (6)-(7), we can make the following equivalence moves:

$$v_s \dots v_2 v_1 w_p w_{p-1} \dots w_{p-s+1} \sim v_s \dots v_2 w_p w_{p-1} \dots w_{p-s+2} v_1 w_{p-s+1} \sim$$

$$v_s \dots v_3 w_p w_{p-1} \dots v_2 w_{p-s+2} v_1 w_{p-s+1} \sim v_s w_p \dots v_1 w_{p-s+1}.$$

□

LEMMA A.2.4. *We have*

$$x w_p v_{s-1} w_{p-1} \dots v_2 w_{p-s+2} v_1 w_{p-s+1} \sim v_{s-1} \dots v_1 x w_p \dots w_{p-s+1}.$$

PROOF. With the equivalence relations from Remark A.2.1 (6),(7) and (11), we can make the following equivalent moves:

$$x w_p v_{s-1} w_{p-1} \dots v_2 w_{p-s+2} v_1 w_{p-s+1} \sim x v_{s-1} w_p w_{p-1} \dots v_2 w_{p-s+2} v_1 w_{p-s+1} \sim$$

$$v_{s-1} x w_p w_{p-1} \dots v_2 w_{p-s+2} v_1 w_{p-s+1} \sim v_{s-1} \dots v_1 x w_p w_{p-1} \dots w_{p-s+2} w_{p-s+1}.$$

□

LEMMA A.2.5. *When $v_s \neq x+1$, we have*

$$x v_s w_p v_{s-1} w_{p-1} \dots v_1 w_{p-s+1} \sim v_s \dots v_1 x w_p \dots w_{p-s+1}.$$

PROOF. With the equivalence relations from Remark A.2.1 (6)-(7) and Remark A.2.2 (1), we can make the following equivalence moves:

$$x v_s w_p v_{s-1} w_{p-1} v_{s-2} \dots v_1 w_{p-s+1} \sim v_s x w_p v_{s-1} w_{p-1} v_{s-2} \dots v_1 w_{p-s+1} \sim$$

$$v_s x v_{s-1} w_p w_{p-1} v_{s-2} \dots v_1 w_{p-s+1} \sim v_s v_{s-1} x w_p w_{p-1} v_{s-2} \dots v_1 w_{p-s+1} \sim$$

$$v_s v_{s-1} v_{s-2} \dots v_1 x w_p w_{p-1} \dots w_{p-s+1}.$$

□

Lemma A.2.6. *When $v_s \neq x{+}1$, we have $b_q u_q \dots b_1 u_1 v_s \dots v_1$ is equivalent to $b_q \dots b_1 v_s \dots v_1 u_q \dots u_1$.*

Proof. With the equivalence relations from Remark A.2.1 (4), (5), (8)-(9) and Remark A.2.2 (2), we can make the following equivalence moves:

$$b_q u_q \dots b_1 u_1 v_s \dots v_1 \sim b_q u_q \dots b_1 v_s u_1 \dots v_1 \sim$$
$$b_q u_q \dots b_1 v_s \dots v_1 u_1 \sim b_q \dots b_1 v_s \dots v_1 u_q \dots u_1.$$

□

Lemma A.2.7. *We have $b_q u_q \dots b_1 u_1 x v_{s-1} \dots v_1$ is equivalent to $b_q \dots b_1 x v_{s-1} \dots v_1 u_q \dots u_1$.*

Proof. With the equivalence relations from Remark A.2.1 (1), (3), (5) and (10) we have the following equivalence moves:

$$b_q u_q \dots b_1 u_1 x v_{s-1} \dots v_1 \sim b_q u_q \dots b_1 x u_1 v_{s-1} \dots v_1 \sim$$
$$b_q u_q \dots b_1 x v_{s-1} \dots v_1 u_1 \sim b_q \dots b_1 x v_{s-1} \dots v_1 u_q \dots u_1.$$

□

APPENDIX B

Proofs for Uncrowding algorithm for hook-valued tableaux

B.1. Proofs of Lemma

B.1.1. Proof of Lemma 3.2.1.

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 (4)a or (4)b in Definition 3.2.1 is applied.

Case 1: Assume Step (4)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 $\mathsf{A}_T(r, c)$. 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 (4)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.

$\square$

B.1.2. Proof of Lemma 3.2.3.

Proof. Let $R_i(T) = r_1 r_2 \dots r_m$. We break into cases based on the value of a.

Case 1: Assume $a \neq i, i+1$.

Assume Step (4)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 (4)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 $k \geqslant i+1$ 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 $\mathsf{A}_T(r, c+1)$ does 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 $R_i(T) = r_1 \ldots i\ i{+}1\ k \ldots r_m$ and $R_i(\mathcal{V}_b(T)) = r_1 \ldots i{+}1\ i\ k \ldots r_m$. We see that $R_i(T)$ and $R_i(\mathcal{V}_b(T))$ only differ by a Knuth relation implying 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 (4)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 $R_i(T) = r_1 \ldots i\ i+1\ i \ldots r_m$ and $R_i(\mathcal{V}_b(T)) = r_1 \ldots i+1\ i\ i \ldots r_m$ which are Knuth equivalent.

Assume Step (4)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 $R_i(T) = r_1 \ldots i+1\ \underbrace{i \ldots i\, a}_{\gamma} \ldots r_m$ and $R_i(\mathcal{V}_b(T)) = r_1 \ldots \underbrace{i \ldots i}_{\gamma-1}\ i+1\ a$ where $\gamma \geqslant 1$ 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 (4)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 $R_i(T) = r_1 \ldots i{+}1\ i{+}1\ i \ldots r_m$ and $R_i(\mathcal{V}_b(T)) = r_1 \ldots i{+}1\ i\ i{+}1 \ldots r_m$ which are Knuth equivalent.

Assume Step (4)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 $R_i(T) = R_i(\mathcal{V}_b(T))$. □

B.1.3. Proof of Lemma 3.2.4.

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 3.2.1 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 $R_i(T) = r_1 r_2 \ldots r_m$ and $R_i(\mathcal{V}_b(T)) = r'_1 r'_2 \ldots r'_m$ 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 $\tilde{r} \leqslant r$. We break into subcases based upon 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 $k \geqslant i+1$ 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 $a \geqslant i$ 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' is 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 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 $a \geqslant i+1$ 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 $k \geqslant i+1$ 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 $k \geqslant i+1$ 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 appying 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 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 $k\geqslant j>i+1$ 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 3.2.3 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 3.2.3, 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 $\hat{r} \leqslant r$ 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 $\hat{c} \geqslant c+2$.

Let $\tilde{i}$ denote the rightmost unpaired i of T. From the proof of Lemma 3.2.3, 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 $f_i(\mathcal{V}_b(T)) = \mathcal{V}_b(f_i(T))$.

$\square$

Bibliography

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

[2] J. Blasiak, J. Morse, and A. Pun, *Demazure crystals and the schur positivity of catalan functions*, arXiv preprint arXiv:2007.04952, (2020).

[3] A. S. Buch, *A Littlewood-Richardson rule for the K-theory of Grassmannians*, Acta Math., 189 (2002), pp. 37–78.

[4] A. S. Buch, A. Kresch, M. Shimozono, H. Tamvakis, and A. Yong, *Stable Grothendieck polynomials and K-theoretic factor sequences*, Math. Ann., 340 (2008), pp. 359–382.

[5] D. Bump and A. Schilling, *Crystal bases*, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2017. Representations and combinatorics.

[6] M. Chan and N. Pflueger, *Combinatorial relations on skew Schur and skew stable Grothendieck polynomials*, arXiv preprint arXiv:1909.12833, (2019).

[7] S. Fomin and C. Greene, *Noncommutative Schur functions and their applications*, vol. 193, 1998, pp. 179–200. Selected papers in honor of Adriano Garsia (Taormina, 1994).

[8] S. Fomin and A. N. Kirillov, *Grothendieck polynomials and the Yang-Baxter equation*, in Formal power series and algebraic combinatorics/Séries formelles et combinatoire algébrique, DIMACS, Piscataway, NJ, 1994, pp. 183–189.

[9] W. Fulton, *Young Tableaux:*, London Mathematical Society Student Texts, Cambridge University Press, 1996. With Applications to Representation Theory and Geometry.

[10] A. Gunna and P. Zinn-Justin, *Vertex models for canonical Grothendieck polynomials and their duals*, arXiv preprint arXiv:2009.13172, (2020).

[11] G. Hawkes and T. Scrimshaw, *Crystal structures for canonical Grothendieck functions*, Algebraic Combinatorics, 3 (2020), pp. 727–755.

[12] J. Hong and S.-J. Kang, *Introduction to quantum groups and crystal bases*, vol. 42 of Graduate Studies in Mathematics, American Mathematical Society, Providence, RI, 2002.

[13] M. Kashiwara, *Crystalizing the q-analogue of universal enveloping algebras*, Comm. Math. Phys., 133 (1990), pp. 249–260.

[14] ———, *On crystal bases of the Q-analogue of universal enveloping algebras*, Duke Math. J., 63 (1991), pp. 465–516.

[15] T. Lam and P. Pylyavskyy, *Combinatorial Hopf algebras and K-homology of Grassmannians*, Int. Math. Res. Not. IMRN, (2007), pp. Art. ID rnm125, 48.

[16] A. Lascoux and M.-P. Schützenberger, *Structure de Hopf de l'anneau de cohomologie et de l'anneau de Grothendieck d'une variété de drapeaux*, C. R. Acad. Sci. Paris Sér. I Math., 295 (1982), pp. 629–633.

[17] ———, *Symmetry and flag manifolds*, in Invariant theory (Montecatini, 1982), vol. 996 of Lecture Notes in Math., Springer, Berlin, 1983, pp. 118–144.

[18] C. Lenart, *Combinatorial aspects of the K-theory of Grassmannians*, Ann. Comb., 4 (2000), pp. 67–82.

[19] ———, *A unified approach to combinatorial formulas for Schubert polynomials*, J. Algebraic Combin., 20 (2004), pp. 263–299.

[20] G. Lusztig, *Canonical bases arising from quantized enveloping algebras*, 3 (1990), p. 447–498.

[21] ———, *Introduction to Quantum Groups*, Birkhäuser Boston, 1993.

[22] C. Monical, O. Pechenik, and T. Scrimshaw, *Crystal structures for symmetric Grothendieck polynomials*, Transformation Groups, doi:10.1007/S00031-020-09623-y (2020).

[23] J. Morse, J. Pan, W. Poh, and A. Schilling, *A crystal on decreasing factorizations in the* 0*-Hecke monoid*, Electron. J. Combin., 27 (2020), pp. Paper 2, 29.

[24] J. Morse and A. Schilling, *Crystal approach to affine Schubert calculus*, Int. Math. Res. Not. IMRN, (2016), pp. 2239–2294.

[25] J. Pan, J. Pappe, W. Poh, and A. Schilling, *Uncrowding algorithm for hook-valued tableaux*, arXiv preprint arXiv:2012.14975, (2020).

[26] R. Patrias, *Antipode formulas for some combinatorial Hopf algebras*, Electron. J. Combin., 23 (2016), pp. Paper 4, 30.

[27] R. Patrias and P. Pylyavskyy, *Combinatorics of K-theory via a K-theoretic Poirier-Reutenauer bialgebra*, Discrete Math., 339 (2016), pp. 1095–1115.

[28] V. Reiner, B. E. Tenner, and A. Yong, *Poset edge densities, nearly reduced words, and barely set-valued tableaux*, J. Combin. Theory, Ser. A, 158 (2018), pp. 66–125.

[29] R. P. Stanley, *On the number of reduced decompositions of elements of Coxeter groups*, European J. Combin., 5 (1984), pp. 359–372.

[30] J. R. Stembridge, *On the fully commutative elements of Coxeter groups*, J. Algebraic Combin., 5 (1996), pp. 353–385.

[31] ———, *A local characterization of simply-laced crystals*, Trans. Amer. Math. Soc., 355 (2003), pp. 4807–4823.

[32] D. Yeliussizov, *Duality and deformations of stable Grothendieck polynomials*, J. Algebraic Combin., 45 (2017), pp. 295–344.