



Contents lists available at ScienceDirect

European Journal of Combinatorics

journal homepage: www.elsevier.com/locate/ejc

The Burge correspondence and crystal graphs

Joseph Pappe^a, Digjoy Paul^b, Anne Schilling^a^a Department of Mathematics, UC Davis, One Shields Ave., Davis, CA 95616-8633, USA^b Chennai Mathematical Institute, Chennai, India

ARTICLE INFO

Article history:

Received 19 April 2022

Accepted 31 October 2022

Available online 21 November 2022

ABSTRACT

The Burge correspondence yields a bijection between simple labelled graphs and semistandard Young tableaux of threshold shape. We characterize the simple graphs of hook shape by peak and valley conditions on Burge arrays. This is the first step towards an analogue of Schensted's result for the RSK insertion which states that the length of the longest increasing subword of a word is the length of the largest row of the tableau under the RSK correspondence. Furthermore, we give a crystal structure on simple graphs of hook shape. The extremal vectors in this crystal are precisely the simple graphs whose degree sequence are threshold and hook-shaped.

© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license

(<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

The celebrated Robinson–Schensted (RS) correspondence [1,2] gives a bijection between words w in the alphabet $\{1, 2, \dots, n\}$ of length k and a pair of tableaux of the same shape λ , a partition of k with at most n parts, where the first tableau is a semistandard Young tableau in the same alphabet and the second tableau is a standard tableau. Schensted [2] proved that λ_1 (the biggest part of the partition λ) is the length of the longest increasing subword of w . Knuth's generalization of the RS correspondence [3], known as the RSK correspondence, provides a bijective proof of the *Cauchy identity* in symmetric function theory

$$\sum_{\lambda} s_{\lambda}(x)s_{\lambda}(y) = \prod_{i,j \geq 1} \frac{1}{1 - x_i y_j},$$

where the sum is over all partitions λ and $s_{\lambda}(x)$ is the Schur function in the variables x_1, x_2, \dots indexed by the partition λ .

E-mail addresses: jhpappe@ucdavis.edu (J. Pappe), dpaul@cmi.res.in (D. Paul), anne@math.ucdavis.edu (A. Schilling).

<https://doi.org/10.1016/j.ejc.2022.103640>

0195-6698/© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

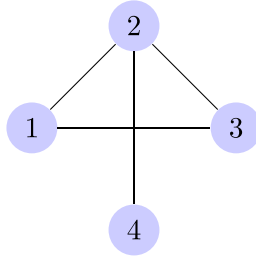


Fig. 1. A threshold graph.

In [4], W. Burge gives four variants of the RSK correspondence. In this paper, we focus on the correspondence in [4, Section 4], which gives a bijection between simple labelled graphs (graphs without loops or multiple edges) and semistandard Young tableaux of threshold shape. A partition $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)$ is called *threshold* if $\lambda_i^t = \lambda_i + 1$ for all $1 \leq i \leq d(\lambda)$, where λ_i^t is the length of i th column of the Young diagram of λ and $d(\lambda)$ is the maximal d such that $(d, d) \in \lambda$. We call this bijection the *Burge correspondence*. The Burge correspondence gives a bijective proof of the *Littlewood identity* [5, Exer. 1.5.9(a) and 1.8.6(c)]

$$1 + \sum_{\lambda} s_{\lambda}(x_1, x_2, \dots) = \prod_{i < j} (1 + x_i x_j), \quad (1.1)$$

where the sum runs over all threshold partitions. For more details about the representation theoretic significance of the Burge correspondence, see [6].

In this paper, we characterize the graphs whose shapes under the Burge correspondence are hook shapes in terms of *peak* and *valley* conditions. This is the first step towards an analogue for the Burge correspondence of Schensted's result for the RS correspondence, namely that increasing sequences under the RS correspondence give tableaux of single row shape.

Threshold partitions also play an important role in graph theory. Given a simple graph $G = ([n], E)$ with vertex set $[n] = \{1, 2, \dots, n\}$ and edge set E , the degree d_i of a vertex i is the number of neighbours of i . The *degree sequence* of G is the tuple $d_G = (d_1, d_2, \dots, d_n)$. One of the central questions in graph theory is the characterization of all sequences that appear as degree sequences of a simple graph (see for example [7–11]). The Erdős–Gallai theorem [7] gives a characterization of graphic partitions, that is, partitions that are degree sequences of a simple graph.

Define $X^G := x_1^{d_1} x_2^{d_2} \dots$, the monomial associated with a simple graph G with degree sequence $d_G = (d_1, d_2, \dots)$. It is known that the generating function $\sum_G X^G = \prod_{i < j} (1 + x_i x_j)$ is a symmetric function in x_1, x_2, \dots , where the sum runs over all simple graphs G . Note that the right hand side is the right hand side in Littlewood's identity (1.1). Hence, from (1.1) and the Burge correspondence, it follows that a sequence $d = (d_1 \geq d_2 \geq \dots \geq d_n)$ is a degree sequence of a simple graph G if and only if $d \leq \lambda$ for some threshold partition λ , where \leq is the dominance order for partitions. In [12], Gasharov proved that this condition is equivalent to the Erdős–Gallai theorem. Hence threshold partitions are the maximal graphic partitions.

The *degree partition* of G is the partition \bar{d}_G obtained by rearranging d_G in weakly decreasing manner. A graph G is called *threshold* if the associated degree partition is threshold. For alternative definitions and characterizations of threshold graphs, see [10, Chapter 3]. Threshold graphs can be interpreted in an extremal sense. First, the threshold partitions (the degree partitions of threshold graphs) are maximal among graphic partitions. Second, the threshold partitions of n are the extreme points of the degree partition polytope, the convex hull of all degree partitions of simple graphs on n , see [13] and references therein.

Example 1.1. The simple graph in Fig. 1 is threshold as its degree partition $(3, 2, 2, 1)$ is threshold.

The condition $\lambda_i^t = \lambda_i + 1$ for threshold partitions is in fact Merris' and Roby's reformulation [11] of conditions stated by Ruch and Gutman [8]. Klivans and Reiner [14] give some generalizations of these concepts to hypergraphs and related them to plethysm.

In this paper, we characterize when the shape λ_G of a simple graph G is a hook shape. This is done by analysing the Burge correspondence and imposing peak and valley conditions on the Burge array corresponding to G . In addition, we impose a crystal structure on simple graphs of hook shape. Crystal graphs are combinatorial skeletons of Lie algebra representations (see for example [15,16]). The extremal vectors in this crystal are precisely the simple graphs whose degree sequence is threshold.

The paper is organized as follows. In Section 2, we review the Burge correspondence and prove some results regarding standardization. In Section 3, we provide the characterization of simple graphs of hook shape. Finally in Section 4, we give the crystal structure on simple graphs of hook shape.

2. The Burge correspondence

In this section, we define the Burge correspondence [4]. We review some preliminaries in Section 2.1. We remind the reader of Schensted's result on longest increasing subwords of words in Section 2.2 before introducing the Burge correspondence in Section 2.3. In Section 2.4, we show that the Burge correspondence intertwines with standardization.

2.1. Preliminaries

A *partition* λ of a nonnegative integer n , denoted by $\lambda \vdash n$, is a weakly decreasing sequence $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_\ell)$ of positive integers λ_i such that $\sum_{i=1}^\ell \lambda_i = n$. The *length* of λ is ℓ . The *Young diagram* $Y(\lambda)$ of λ is a left-justified array of boxes with λ_i boxes in row i from the top. (This is also known as the English convention for Young diagrams of partitions). A partition λ is a *hook* if $Y(\lambda)$ does not contain any 2×2 squares.

Definition 2.1. Let λ be a partition. A *semistandard Young tableau* of shape λ in the alphabet $[n] := \{1, 2, \dots, n\}$ is a filling of the Young diagram of λ with letters in $[n]$ such that the numbers weakly increase along rows and strictly increase along columns. We denote by $\text{Tab}_n(\lambda)$ the set of all semistandard Young tableaux of shape λ in the alphabet $[n]$.

Let T be a semistandard Young tableau. The *shape* of T is denoted $\text{sh}(T)$. The *weight* of a semistandard Young tableau T , denoted $\text{wt}(T)$, is the integer vector (μ_1, \dots, μ_n) , where μ_i is the number of times the number i occurs. The subset of $\text{Tab}_n(\lambda)$ consisting of all semistandard Young tableaux of weight μ is denoted by $\text{Tab}(\lambda, \mu)$.

Given an integer vector $\mu = (\mu_1, \dots, \mu_n)$, let x^μ denote the monomial $x_1^{\mu_1} x_2^{\mu_2} \dots x_n^{\mu_n}$ in the n variables x_1, x_2, \dots, x_n .

Definition 2.2. For each integer partition λ , the *Schur polynomial* in n variables corresponding to λ is defined as

$$s_\lambda(x_1, \dots, x_n) = \sum_{T \in \text{Tab}_n(\lambda)} x^{\text{wt}(T)}.$$

2.2. Schensted algorithm and longest increasing subwords

The Burge correspondence (as well as the celebrated Robinson–Schensted–Knuth (RSK) correspondence) uses the *Schensted row insertion* algorithm. Given a semistandard Young tableau T in the alphabet $[n]$, a letter $i \in [n]$ can be inserted into T in the following way: if i is larger than or equal to all the entries of the first row of T , a new box containing i is added at the end of first row and the process stops. Otherwise, i replaces the smallest leftmost number j of the first row such that $j > i$. Then j is inserted in the second row of T in the same way and so on. The procedure stops

when a new box is added to T at the end of a row. The result is denoted $T \leftarrow i$. The shape of $T \leftarrow i$ contains one new box compared to the shape of T .

Given a tableau T and a letter x that lies at the end of some row of T , the *Schensted reverse bumping* algorithm generates a pair of a tableau T' and a letter y in the following way: Let t be the row index of x and x_1 be the rightmost entry of row $t - 1$ such that $x_1 < x$. Replace x_1 by x in T and output x_1 . Repeat the process for x_1 and continue until an element of the first row say y is obtained as output. The resulting tableau is T' . We shall denote the pair (T', y) by $T \rightarrow x$.

Given a word $w = w_1 w_2 \dots w_k$ in the alphabet $\{1, 2, \dots, n\}$, the *Schensted insertion tableau* is defined as $P(w) := \emptyset \leftarrow w_1 \leftarrow w_2 \leftarrow \dots \leftarrow w_k$. The shape of the semistandard Young tableau $P(w)$, denoted $\lambda(w) = (\lambda_1, \lambda_2, \dots)$, is called the *shape of the word* w . Schensted [2] proved that λ_1 is the length of the longest increasing subword of w . In particular, the shape $\lambda(w)$ is a single row if w is weakly increasing. Greene [17] extended the result of Schensted by interpreting the rest of the shape of λ . For a poset theoretic viewpoint of the map $w \mapsto \lambda(w)$ and various applications including in the context of flag varieties, see Britz and Fomin [18] and references therein. In the same spirit, we ask very similar questions about the Burge correspondence introduced in the next subsection.

2.3. The Burge correspondence

We begin by recalling the definition of a Burge array from [4, Section 4].

Definition 2.3. Given a simple graph $G = ([n], E)$ with $|E| = r$, define a two line array known as the *Burge array*

$$\mathcal{A}_G = \begin{bmatrix} a_1 & a_2 & \dots & a_r \\ b_1 & b_2 & \dots & b_r \end{bmatrix}$$

satisfying:

- (1) Each pair (a_k, b_k) is an edge of G and $a_k > b_k$ for each $1 \leq k \leq r$.
- (2) The top line is weakly increasing, that is, $a_k \leq a_{k+1}$ for all $1 \leq k < r$.
- (3) If $a_k = a_{k+1}$ for some $1 \leq k < r$, then $b_k > b_{k+1}$.

Notice that G is completely determined by the associated Burge array \mathcal{A}_G assuming that $[n]$ is known. Note that singletons in the simple graph do not appear in the Burge array \mathcal{A}_G .

Given a threshold partition λ , by definition, the Young diagram $Y(\lambda)$ of λ is divided into two symmetric pieces. The bottom piece consists of all boxes that lie strictly below the diagonal and the top piece consists of the rest. Each position in the top (bottom) piece of the Young diagram of $Y(\lambda)$ corresponds to a unique position, called the *opposite position*, in the bottom (top) piece of $Y(\lambda)$. The opposite position $\text{op}(s, t)$ of (s, t) is defined to be $(t + 1, s)$ if $s \leq t$ and $(t, s - 1)$ otherwise.

Having defined all the necessary tools, we now state the main algorithm of the *Burge correspondence*. Starting with the empty tableau T_0 , we shall insert all the edges of G , as ordered in \mathcal{A}_G , into T_0 . Let T_k be the tableau obtained by inserting the edge (a_k, b_k) into T_{k-1} in the following way:

- (1) First insert b_k into T_{k-1} using the Schensted insertion algorithm. This adds a new cell to the shape, say in position (s_k, t_k) .
- (2) Place the entry a_k in the cell $\text{op}(s_k, t_k)$. Observe that each addition of an edge transforms a tableau of threshold shape to another tableau of threshold shape.

Finally, the tableau $T_G := T_r$ is the threshold tableau associated with the graph G under the Burge correspondence.

Burge [4] proved that this tableau is semistandard. Given such a tableau T , we recover the Burge array in the following way: Let a_r be the largest entry of T with largest column index. Remove a_r from T . Let z_r be the value at the opposite position of the cell containing a_r . Let $b_r = y_r$ where $T \rightarrow z_r = (T_{r-1}, y_r)$. Repeat the process for T_{r-1} and continue until the empty tableau is obtained in the output.

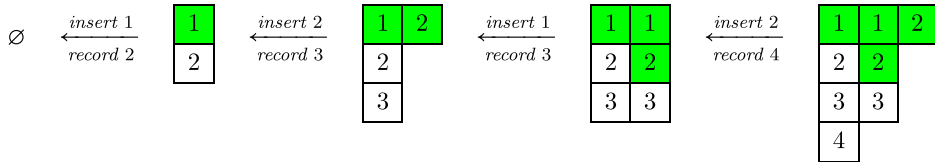


Fig. 2. Burge insertion associated to Example 2.4.

Example 2.4. The Burge array \mathcal{A}_G for the graph in Fig. 1 is $\begin{bmatrix} 2 & 3 & 3 & 4 \\ 1 & 2 & 1 & 2 \end{bmatrix}$. The associated tableau T_G of threshold shape $(3, 2, 2, 1)$ is obtained by inserting the edges of G as ordered in \mathcal{A}_G as depicted in Fig. 2.

In the same spirit as the shape of a word under the RSK correspondence, we can define the shape of a graph under the Burge correspondence.

Definition 2.5. The *shape* of a simple graph G is the partition $\lambda_G := \text{sh}(T_G)$, where T_G is the tableau associated with G by the Burge correspondence.

It can be observed that the shape of a threshold graph G is its degree partition \tilde{d}_G . Namely, let G be a threshold graph with degree sequence d_G . If T is the semistandard Young tableau of threshold shape λ and weight d_G , then d_G is less than or equal to λ in dominance order. Since d_G is a threshold sequence by assumption, we know that the only partition that dominates a threshold sequence is the corresponding partition. Hence $\lambda = \tilde{d}_G$.

We call a simple graph G a *hook-graph* if the associated tableau T_G has hook shape. Given the nature of the Burge algorithm, determining when a T_G has hook shape is analogous to asking when a tableau under the RSK algorithm has single row shape.

Problem 2.6. What is the shape of a simple graph?

In the next section, we characterize all hook-graphs.

2.4. Standardization

Both $\text{Tab}_m(\lambda)$ and words over $[m]$ with length n have the notion of *standardization*. Standardization intertwines with RSK in the sense that they form a commuting diagram. We show that an analogous result holds true for the Burge correspondence.

First, we review the standardization map for semistandard Young tableaux. Let $\lambda \vdash n$ and let $C = \{c_1 < \dots < c_n\}$ be a subset of \mathbb{N} . The standardization of $T \in \text{Tab}(\lambda, \mu)$ with respect to the alphabet C , denoted by $\text{stand}_C(T)$, is the map replacing all the 1's in T from left to right with the numbers c_1 through c_{μ_1} , replacing all the 2's from left to right with the numbers c_{μ_1+1} through $c_{\mu_1+\mu_2}$, etc.

The standardization map on words over the alphabet $[m]$ can be defined similarly. Let ω be a word using the alphabet $[m]$ with length n and let μ denote its content. In other words, let $\mu = (\mu_1, \mu_2, \dots, \mu_m)$ be an integer vector, where μ_i denotes the number of i 's in ω . The standardization of ω with respect to C , which we also denote by $\text{stand}_C(\omega)$, is defined by replacing the 1's in ω from left to right with the numbers c_1 through c_{μ_1} , replacing the 2's in ω from left to right with the numbers c_{μ_1+1} through $c_{\mu_1+\mu_2}$, etc. The following result formalizes the relationship between standardization and RSK, see for example [19, Lemma 7.11.6].

Proposition 2.7. Let ω be a word in the alphabet $[m]$ with length n . Then $\text{stand}_C(P(\omega)) = P(\text{stand}_C(\omega))$.

Analogously, we define the standardization of a Burge array \mathcal{A}_G . Given a Burge array \mathcal{A}_G with r columns and a subset $C = \{c_1 < \dots < c_{2r}\}$ of \mathbb{N} , define the standardization of \mathcal{A}_G with respect to C , denoted by $\text{stand}_C(\mathcal{A}_G)$, to be the map that replaces the 1's in \mathcal{A}_G from left to right with the numbers c_1 through c_{d_1} , replaces the 2's from left to right with the numbers c_{d_1+1} through $c_{d_1+d_2}$, etc. where $d_G = (d_1, d_2, \dots, d_n)$ is the degree sequence of G .

For T a semistandard Young tableau, the reading word of T denoted by $R(T)$ is obtained by reading the entries within a row from left to right starting with the bottommost row.

Proposition 2.8. *Let \mathcal{A}_G be a Burge array and $\mathcal{B}_G = \overline{\text{stand}}_C(\mathcal{A}_G)$. Let T_G (resp. S_G) be the tableau associated to \mathcal{A}_G (resp. \mathcal{B}_G) under the Burge correspondence. Then $S_G = \text{stand}_C(T_G)$.*

Proof. We proceed by induction on the number of columns of $\mathcal{A}_G = \begin{bmatrix} a_1 & a_2 & \dots & a_r \\ b_1 & b_2 & \dots & b_r \end{bmatrix}$. Note that the base case of $r = 0$ is trivial. Let \mathcal{A}_{r-1} (resp. \mathcal{B}_{r-1}) denote the Burge array formed by the first $r - 1$ columns of \mathcal{A}_G (resp. \mathcal{B}_G). We have $\mathcal{B}_{r-1} = \overline{\text{stand}}_{C-\{c_{2r}, c_{1r}\}}(\mathcal{A}_{r-1})$ where $[c_{2r}, c_{1r}]^T$ is the last column of \mathcal{B}_G . From our inductive hypothesis, $S_{r-1} = \text{stand}_{C-\{c_{1r}, c_{2r}\}}(T_{r-1})$. As the reading word $R(S_{r-1})$ is the standardization of $R(T_{r-1})$, we have $(R(S_{r-1}), c_{1r})$ is the standardization of $(R(T_{r-1}), b_r)$. By Proposition 2.7, $S_{r-1} \leftarrow c_{1r} = \text{stand}_{C-\{c_{2r}\}}(T \leftarrow b_r)$. Thus the a_r and c_{2r} must be placed in the same position of their respective tableau. From the inverse of the Burge correspondence, a_r is the largest value in T_G and lies in a column to the right of any equivalent letters. Therefore, a_r in T_G gets sent to c_{2r} by stand_C and does not affect the mapping of the other letters in the tableau. Hence, $S_G = \text{stand}_C(T_G)$. \square

3. Characterization of the shape of graphs

While answering Problem 2.6 in full generality seems far-achieving, we determine necessary and sufficient conditions of a hook graph in this section.

3.1. Trees

We begin by establishing a necessary condition for a connected graph to be of hook shape.

Proposition 3.1. *Let G be a simple graph and let k be the number of connected components of G that contain at least one edge. If G contains k edges e_1, \dots, e_k such that $G - \{e_1, \dots, e_k\}$ has the same number of connected components as G , then G is not a hook-graph.*

Proof. Let C_1, \dots, C_k denote the k connected components of G that contain at least one edge. Denote by n_i the number of vertices in C_i and let $n = \sum_{i=1}^k n_i$. If each C_i was minimally connected, it would contain $n_i - 1$ edges. Thus, G contains at least $\sum_{i=1}^k (n_i - 1) = n - k$ edges. The condition that “ G contains k edges e_1, \dots, e_k such that $G - \{e_1, \dots, e_k\}$ has the same number of connected components as G ” implies that G has at least $n - k + k = n$ edges. Observe that the length of d_G is precisely n . However, the length of the partition $(e, 1^e)$ is at least $n + 1$, where $e \geq n$ is the number of edges of G . This implies that the partition $(e, 1^e)$ is not weakly greater than \tilde{d}_G in dominance order. Thus, there are no semistandard Young tableaux of shape $(e, 1^e)$ with weight d_G , and T_G is not hook-shaped. \square

Recall that an undirected graph is called a *tree* if it is connected and does not contain any cycle. Setting $k = 1$ into Proposition 3.1, we obtain the following necessary condition for connected (excluding singletons) hook-graphs.

Corollary 3.2. *The only connected hook-graphs are trees.*

Remark 3.3. A tree need not be a hook-graph always. For example, the tree whose Burge array is $\begin{bmatrix} 2 & 4 & 4 \\ 1 & 3 & 2 \end{bmatrix}$ has the shape $(2, 2, 2)$.

3.2. Peak and valley condition

We now introduce peak and valley conditions which characterize when a graph has hook shape.

Definition 3.4 (Peak). A simple graph G with \mathcal{A}_G as in Definition 2.3 is said to have a *peak* if there exist $1 \leq i < j < k \leq r$ such that

- (1) $b_i \leq b_k$,
- (2) j is the minimum index with $b_k < b_j$,
- (3) $a_i \leq b_j$.

Definition 3.5 (Valley). A simple graph G with \mathcal{A}_G as in Definition 2.3 is said to have a *valley* if there exist $1 \leq i < j < k \leq r$ such that the following conditions hold:

- (1) $b_j \leq b_k < a_j$,
- (2) $b_j < b_i$.

Example 3.6.

- (1) The graph with Burge array $\begin{bmatrix} 2 & 4 & 4 \\ 1 & 3 & 2 \end{bmatrix}$ of Remark 3.3 has a peak with $i = 1, j = 2, k = 3$, but no valley.
- (2) The graph G with $\mathcal{A}_G = \begin{bmatrix} 4 & 5 & 6 & 7 \\ 1 & 3 & 5 & 2 \end{bmatrix}$ does not have a peak as $b_1 < b_4 < b_2$ but $a_1 > b_2$.

Note that $b_1 < b_4 < b_3$, but $j = 3$ is not the minimal j satisfying this condition. Also, \mathcal{A}_G does not have a valley.

- (3) The graph considered in Example 2.4 has both a peak and a valley.

We refer to a Burge array as being *PV-free* if the Burge array does not contain a peak or a valley.

Theorem 3.7. Let G be a simple graph on $[n]$. The graph G is a hook-graph if and only if its corresponding Burge array is PV-free.

Proof. We prove the equivalent statement that the shape of G is *non-hook* if and only if G has either a peak or a valley.

Proof of forward direction \Rightarrow :

Let T_G be the tableau of non-hook shape associated with the Burge array $\mathcal{A}_G = \begin{bmatrix} a_1 & a_2 & \dots & a_r \\ b_1 & b_2 & \dots & b_r \end{bmatrix}$. Let T_ℓ denote the tableau corresponding to the sub-array consisting of the first ℓ columns of \mathcal{A}_G . Choose k minimal such that the shape of T_k is non-hook, that is, let k be the column that creates the cells $(2, 2)$ and $(3, 2)$ when applying the Burge algorithm. We claim that there exist j_1, j_2 with $1 \leq j_1 < j_2 < k$ such that $\begin{bmatrix} a_{j_1} & a_{j_2} & a_k \\ b_{j_1} & b_{j_2} & b_k \end{bmatrix}$ is either a peak or a valley.

Let x be the first row entry of T_{k-1} that is bumped by b_k in k th step of the Burge algorithm. Let y be the entry in position $(2, 1)$ (first entry of 2nd row) of T_{k-1} , see Fig. 3. Note that $x > b_k$ and $y \leq x$. There are two different cases depending on whether x is an inserted letter or a recorded letter.

Case 1: Let x be the inserted letter b_j for some $1 < j \leq k - 1$.

Subcase A: If y is a recorded letter, then the only possibility is $y = a_1$. This implies that position $(1, 1)$ of T_{k-1} is b_1 and the arm of T_{k-1} consists of $b_2 b_3 \dots b_{k-1}$. So we have that j is the smallest

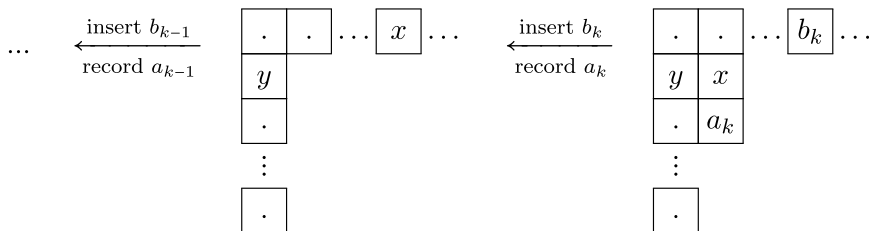


Fig. 3. Tableau after $(k - 1)$ insertions (left) and tableau after k insertions (right).

index between 1 and k satisfying $b_1 \leq b_k < b_j$. In addition, $a_1 = y \leq x = b_j$. Hence $\begin{bmatrix} a_1 & a_j & a_k \\ b_1 & b_j & b_k \end{bmatrix}$ is a peak.

Subcase B: Assume y is an inserted letter, say b_i for some i . Let b_m be the value which bumped b_i from the first row. So $b_m < b_i \leq x = b_j$. Also we must have $b_m \leq b_k$ as otherwise $b_k < b_m < x$ and in this case b_k would bump an entry strictly smaller than $x = b_j$. Now two cases arise: $m < j$ and $m > j$. When $m > j$, we have $a_j < a_m$. We use the relations $b_m \leq b_k < x = b_j < a_j < a_m$ and $b_m < b_i$ to show that $\begin{bmatrix} a_i & a_m & a_k \\ b_i & b_m & b_k \end{bmatrix}$ is a valley. When $m < j$, we claim that $\begin{bmatrix} a_m & a_j & a_k \\ b_m & b_j & b_k \end{bmatrix}$ is a peak. First observe that $a_m \leq b_j$. Since $m < j$, $x = b_j$ is inserted after b_m . If $a_m > b_j$ then b_j would bump an entry weakly larger than y and the shape of T_{k-1} becomes non-hook. This contradicts the minimality of k . Hence $a_m \leq b_j$. Also note that j is the smallest index between m and k satisfying $b_m \leq b_k < b_j$ as otherwise this would contradict the choice of y or x . Hence the claim is true.

Case 2: Let x be a recorded letter say a_j for $1 < j \leq k - 1$. This implies $y = b_i$ for some i as y can no longer be the recording letter a_1 .

Subcase A: $i \leq j$. Let b_m be the value which bumps b_i from the first row. Then we must have $m \geq j$. If $m < j$, then b_i would be placed at position $(2, 1)$ of T_m . Since $x = a_j$ is placed in the first row, b_j must bump a letter from the first row. That letter would bump b_i from position $(2, 1)$ of T_j . This contradicts that $y = b_i$. Now we have the relations $b_m \leq b_k < x = a_j \leq a_m$ and $b_m < b_i$. Hence $\begin{bmatrix} a_i & a_m & a_k \\ b_i & b_m & b_k \end{bmatrix}$ is a valley.

Subcase B: $i > j$. Since $x = a_j$ is placed in the first row, b_j must bump a letter from the first row. This letter must be an inserted letter since a_1 is already placed in the first column and hence a bumped recorded letter would create a non-hook shape. This would contradict the minimality of k . Let b_t with $t < j$ be the value bumped by b_j . So $b_t > b_j$. Let b_m denote the value that bumps $y = b_i$. Here $m > i$ but $b_m < b_i$. Observe that $b_i < b_t$, as b_i and b_t are in the first column. We also have $b_m \leq b_k$ otherwise b_k would bump something smaller than $x = a_j$. Moreover, $b_k < a_j \leq a_i \leq a_m$. Therefore $b_m \leq b_k < a_m$ and $b_t > b_m$, which shows that $\begin{bmatrix} a_t & a_m & a_k \\ b_t & b_m & b_k \end{bmatrix}$ is a valley.

Proof of backward direction \Leftarrow :

Now we shall prove that the shape of T_G is non-hook if G has either a valley or a peak. We start by proving the result for the case of a valley.

Suppose \mathcal{A}_G contains a **valley** $\begin{bmatrix} a_i & a_j & a_k \\ b_i & b_j & b_k \end{bmatrix}$. We may assume that there is no peak or valley in the first $k - 1$ columns of \mathcal{A}_G and that T_{k-1} has hook shape. We consider two cases.

Case 1: Assume that b_i is present in the first row of T_{j-1} . Since $b_i > b_j$, b_j bumps an element, say x , from the first row of T_{j-1} . So $b_j < x \leq b_i$. Since b_j bumps a letter in the first row and since by assumption T_j has hook shape, it follows that a_j must be in the first row of T_j . Note that a_j must also be in the first row of T_{k-1} . This is because if an element say b_ℓ (insertion letter) bumps a_j from

the first row before the insertion of b_k , then T_ℓ has non-hook shape since $a_j \geq a_1$, which is greater or equal to the letter in cell $(2, 1)$.

Since we have $b_j \leq b_k < a_j$, b_k bumps an element (which lies in the first row of T_{k-1}), say y . Then $b_j < y \leq a_j$. We claim that $x \leq y$. If $y < x$, note that y was not in T_{j-1} when b_j was inserted, since otherwise b_j would have bumped y . So $b_j < y < x \leq b_i < a_i < a_j$. This shows that y must be an inserted letter, say b_ℓ with $j < \ell < k$, and we have a valley formed by columns i, j, ℓ . This is a contradiction to our assumption. Therefore we must have $x \leq y$. In this case the shape of T_k becomes non-hook, because the entry in cell $(2, 1)$ of T_{k-1} is less than or equal to x and so y must be placed in the $(2, 2)$ position of T_k .

Case 2: Assume that b_i is not present in the first row of T_{j-1} . Let x be the element in T_{j-1} in the position where b_i was originally inserted. Since $x < b_i$ and x is inserted after step i , x must be an inserted letter, say b_ℓ with $i < \ell < j$. We claim that $b_\ell > b_j$. If not, we have $b_\ell \leq b_j$ and $\begin{bmatrix} a_i & a_\ell & a_j \\ b_i & b_\ell & b_j \end{bmatrix}$

is a valley since $b_j < b_i < a_i \leq a_\ell$. This is a contradiction. Hence $b_\ell > b_j$. This implies that b_j must bump an element from the first row of T_{j-1} . If z is that element, then $b_j < z \leq b_\ell$. Also, since b_j bumps an element, a_j must be in the first row in T_j . By the same arguments as in Case 1, a_j must be in the first row in T_{k-1} . Since by the valley condition $b_k < a_j$, b_k bumps an element in T_{k-1} . Call this element w . We claim that $z \leq w$. If $w < z$, note that w was not in T_{j-1} since then $b_j \leq b_k < w < z$ and hence b_j would have bumped w instead of z in T_{j-1} . If w is an inserted letter b_m with $j < m < k$, then $\begin{bmatrix} a_i & a_j & a_m \\ b_i & b_j & b_m \end{bmatrix}$ forms a valley as $b_m = w < z \leq b_\ell < a_\ell \leq a_j$. This is a contradiction. If w is a

recorded letter, then $w \geq a_j \geq a_\ell > b_\ell \geq z$, contradicting the assumption that $w < z$. This proves $z \leq w$. Hence w must be placed in position $(2, 2)$ of T_k . This implies that the shape of T_k is non-hook.

Considering the above cases, we conclude that T_G has non-hook shape when \mathcal{A}_G has a valley.

Suppose \mathcal{A}_G contains a **peak** $\begin{bmatrix} a_i & a_j & a_k \\ b_i & b_j & b_k \end{bmatrix}$. As before, we may assume that there is no valley or peak in first $k-1$ columns of \mathcal{A}_G and that T_{k-1} has hook shape.

Claim. b_j cannot be bumped from the first row before the k th step.

Proof. Assume that b_j is bumped from the first row before the k th column is inserted. This would create a non-hook shape as $b_j \geq a_i \geq a_1$. This contradicts the fact that T_{k-1} has hook shape. Hence we have proved the claim.

Since b_j is in the first row of T_{k-1} and $b_k < b_j$, b_k must bump an element, say z . So $b_k < z \leq b_j$.

Case 1: Suppose b_i is not present in the first row of T_{k-1} . Let y be the entry in position $(2, 1)$ of T_{k-1} . Then $y \leq b_i$. Also we have $b_i \leq b_k < z$. Since $y \leq z$, z would be placed at position $(2, 2)$ of T_k , hence the shape of T_k becomes non-hook.

Case 2: Assume b_i is present in the first row of T_{k-1} .

Subcase A: b_i is placed at the end of the first row of T_{i-1} . If $z = a_\ell$, then $z \geq a_1$ and T_k has non-hook shape since the element in position $(2, 1)$ in T_{k-1} is smaller or equal to a_1 .

Now consider the case when $z = b_\ell$ for some ℓ . If $\ell \leq j$, consider the array $\begin{bmatrix} a_i & a_\ell & a_j & a_k \\ b_i & b_\ell & b_j & b_k \end{bmatrix}$.

Note that indeed $i < \ell$. Namely, $z = b_\ell$ implies $b_i < b_\ell$ since $b_i \leq b_k$ and $b_k < z$. This implies that $i < \ell$; otherwise, b_i would bump a value from the first row of T_{i-1} . We have $b_i \leq b_k < z = b_\ell$.

By definition of the peak $\begin{bmatrix} a_i & a_j & a_k \\ b_i & b_j & b_k \end{bmatrix}$, we have $a_i \leq b_\ell = z$. Note that by the definition of a peak $j = \ell$. Again, $y \leq z$ implies that shape of T_k is non-hook.

Now let us consider the case when $\ell > j$. If $b_\ell = b_j$, then b_j would be bumped by b_k instead of b_ℓ , which is a contradiction. If $b_\ell < b_j$, then $\begin{bmatrix} a_i & a_j & a_\ell \\ b_i & b_j & b_\ell \end{bmatrix}$ becomes a peak. Observe that j is

minimal due to the minimality condition of the peak $\begin{bmatrix} a_i & a_j & a_k \\ b_i & b_j & b_k \end{bmatrix}$ and the inequality $b_k < b_\ell$. This contradicts our assumption that there should not be any other peak in the first k columns of \mathcal{A}_G .

Subcase B: b_i bumps an element from the first row of T_{i-1} when inserted. Then a_i is placed in the first row of T_i . Let u be the value bumped by b_i when inserted to T_{i-1} and let z be the value bumped by b_k . So, $b_i < u$ and $b_k < z$. Notice that the entry in position $(2, 1)$ of T_{k-1} is less than or equal to u . So, if $u \leq z$, the shape of T_k becomes non-hook.

On the contrary, suppose $u > z$. Then z is not in the first row of T_{i-1} since otherwise b_i would bump z instead of u as $b_i \leq b_k < z < u$. This contradicts the definition of u . Hence z is inserted after the i th step. As b_i bumps u from the first row of T_{i-1} , u is weakly less than a_i . As z is inserted after the i th step, it must be an inserted letter i.e. $z = b_\ell$ for some $i < \ell < k$. Now either $z < a_i$ or $z \geq a_i$.

Assume that $z < a_i$. We prove that there does not exist an inserted letter b_ℓ such that $i < t < k$ and $b_i \leq b_t < a_i$. Assume that there exists such an inserted letter b_t , where t is the smallest possible index. Observe that when b_t is inserted it bumps an element strictly right of b_i and weakly left of a_i in the arm of T_{t-1} . Denote this element by w . From the minimality of t , we have w is present in the first row of T_{i-1} and is strictly to the right of u . Thus, w is weakly larger than u , and when w is bumped, it will be weakly larger than the value in the $(2, 1)$ cell of T_{t-1} . This would contradict T_{k-1} being hook shaped. Therefore no such inserted letter b_t exists. Observe that z satisfies the condition of b_t namely $z = b_\ell$ where $i < \ell < k$ and $b_i \leq b_\ell < a_i$. By our claim this is impossible.

Next consider the case $z \geq a_i$. Note that $a_1 \geq u$ as both of them are placed in first column. Together with $u > z \geq a_i$ we get $a_1 > a_i$, which is not possible. Therefore the case $u > z$ does not arise.

Considering all the cases, we proved that the shape of T_k is non-hook when \mathcal{A}_G has a peak. This completes the proof. \square

Remark 3.8. Theorem 3.7 is the analogue to the statement for RSK that the shape of a word w under RSK is a single row if and only if w is weakly increasing.

Remark 3.9. In analogy with Schensted's result for the RSK insertion that the length of the longest increasing subsequence of a word w gives the length of the longest row in the Young tableaux under RSK, one might suspect that the longest PV-free subsequence of a Burge array gives the size of the largest hook in $\text{sh}(T_G)$. However, this is not true as the following counterexample shows. Take the graph G with Burge array

$$\mathcal{A}_G = \begin{bmatrix} 4 & 8 & 8 & 9 & 9 \\ 1 & 3 & 2 & 5 & 2 \end{bmatrix}.$$

The tableau under the Burge correspondence is

$$T_G = \begin{array}{|c|c|c|} \hline 1 & 2 & 2 \\ \hline 3 & 5 & 9 \\ \hline 4 & 8 & \\ \hline 8 & 9 & \\ \hline \end{array},$$

so that $\text{sh}(T_G) = (3, 3, 2, 2)$. However, the subsequence

$$\begin{bmatrix} 4 & 8 & 9 & 9 \\ 1 & 3 & 5 & 2 \end{bmatrix}$$

is PV-free and has hook shape $(4, 1, 1, 1)$, which is larger than the biggest hook $(3, 1, 1, 1)$ in $\text{sh}(T_G)$.

A *star graph* is a graph where one vertex i is connected to all other vertices by an edge and no other edges exist in the graph.

Corollary 3.10. *All star graphs are hook-graphs.*

Proof. Assuming that the vertices of the star graph are labelled $1, 2, \dots, n$ and the vertex connected to all other vertices is vertex i , the Burge array is

$$\begin{bmatrix} i & i & \dots & i & i+1 & i+2 & \dots & n \\ i-1 & i-2 & \dots & 1 & i & i & \dots & i \end{bmatrix},$$

which is PV-free. \square

4. Crystal structure on hook-graphs

We review the crystal structure on semistandard Young tableaux in Section 4.1 and then define the new crystal structure on hook-graphs in Section 4.2.

4.1. Review of crystal structure on semistandard Young tableaux

Crystal bases provide a combinatorial skeleton for $U_q(\mathfrak{g})$ -representations, where $U_q(\mathfrak{g})$ is the quantum group associated to the Lie algebra \mathfrak{g} . A *Kashiwara crystal* is a nonempty set B together with maps

$$\begin{aligned} e_i, f_i: B &\rightarrow B \cup \{\emptyset\} & \text{for } i \in I, \\ \text{wt}: B &\rightarrow \Lambda, \end{aligned}$$

where Λ is the weight lattice associated to the Lie algebra \mathfrak{g} and I is the index set of the Dynkin diagram for \mathfrak{g} . These maps have to satisfy certain conditions (see [15, Definition 2.13]).

For simply-laced Lie algebras \mathfrak{g} , a *Stembridge crystal* is a Kashiwara crystal for which the raising and lowering operators e_i and f_i satisfy certain local rules (see [15, Section 4.2]). Stembridge crystals are crystals corresponding to $U_q(\mathfrak{g})$ -representations.

In this paper, we only consider crystals of type A_{m-1} . For type A_{m-1} crystals we have $I = \{1, 2, \dots, m-1\}$. A particular model for type A_{m-1} crystals is given in terms of semistandard Young tableaux, that is, the set B is the set of all semistandard Young tableaux on the alphabet $[m]$. For further details, see [15, Section 3]. Here we review the crystal structure on semistandard Young tableaux.

Definition 4.1. Let T be a semistandard Young tableau. The *reading word* of T denoted by $R(T)$ is obtained by reading the entries within a row from left to right starting with the bottommost row. The *i th reading word* of T denoted by $R_i(T)$ is the induced subword of $R(T)$ containing only the entries i and $i+1$.

Definition 4.2. Let ω be a word with length n over the alphabet $[m]$. A *Knuth move* on ω is one of the following transformations:

- (1) $\omega_1 \dots bca \dots \omega_n \longrightarrow \omega_1 \dots bac \dots \omega_n$ if $a < b \leq c$,
- (2) $\omega_1 \dots bac \dots \omega_n \longrightarrow \omega_1 \dots bca \dots \omega_n$ if $a < b \leq c$,
- (3) $\omega_1 \dots acb \dots \omega_n \longrightarrow \omega_1 \dots cab \dots \omega_n$ if $a \leq b < c$,
- (4) $\omega_1 \dots cab \dots \omega_n \longrightarrow \omega_1 \dots acb \dots \omega_n$ if $a \leq b < c$.

Two words ω and ν are said to be *Knuth equivalent* if they differ by a sequence of Knuth moves.

It is well-known that ω and ν are Knuth equivalent if and only if $P(\omega) = P(\nu)$, that is, their insertion tableaux under Schensted insertion are equal. In addition, it is also known that the crystal operators f_i and e_i on words preserve Knuth equivalence. Thus, in order to define the crystal operators on semistandard Young tableaux it suffices to look at their reading words.

Definition 4.3. Let T be a semistandard Young tableau in $\text{Tab}_m(\lambda)$. Assign a ‘)’ to every i in $R_i(T)$ and a ‘(’ to every $i+1$ in $R_i(T)$. Successively pair every ‘(’ that is directly left of a ‘)’ which we call an *i -pair* and remove the i -paired terms. Continue this process until no more terms can be i -paired.

The *lowering operator* f_i for $1 \leq i < m$ acts on T as follows:

- (1) If there are no unpaired ')' terms left, then f_i annihilates T .
- (2) Otherwise locate the i in T corresponding to the rightmost unpaired ')' of $R_i(T)$ and replace it with an $i + 1$.

The raising operator e_i for $1 \leq i < m$ acts on T as follows:

- (1) If there are no unpaired '(' terms left, then e_i annihilates T .
- (2) Otherwise locate the $i + 1$ in T corresponding to the leftmost unpaired '(' of $R_i(T)$ and replace it with an i .

The weight $\text{wt}(T) = (a_1, a_2, \dots, a_m)$ is an m -tuple such that a_i is the number of letters i in T .

The crystal lowering and raising operators f_i and e_i for $1 \leq i < m$ together with the weight function wt define a crystal structure on $\text{Tab}_m(\lambda)$. The vertices of the crystal are the elements in $\text{Tab}_m(\lambda)$ for a fixed partition λ . There is an edge labelled i from $T \in \text{Tab}_m(\lambda)$ to $T' \in \text{Tab}_m(\lambda)$ if $f_i(T) = T'$. Note that f_i and e_i are partial inverses, that is, if $f_i(T) = T'$ then $e_i(T') = T$ and vice versa.

4.2. Crystal structure on hook-graphs

In this section, we assume that G is a hook-graph or equivalently by Theorem 3.7 that \mathcal{A}_G is a PV-free Burge array.

Definition 4.4. The i th reading word of \mathcal{A}_G , denoted by $\tilde{R}_i(\mathcal{A}_G)$, is obtained by the following algorithm:

- (1) Let a_k denote the leftmost $i + 1$ in the top row of \mathcal{A}_G . If $k = 1$ or $b_{k-1} \leq b_k$, then let a_k be the first letter of $\tilde{R}_i(\mathcal{A}_G)$.
- (2) Read all other i 's and $(i + 1)$'s in \mathcal{A}_G from left to right while appending the corresponding value to $\tilde{R}_i(\mathcal{A}_G)$.

Example 4.5. Let $\mathcal{A}_G = \begin{bmatrix} 3 & 3 & 4 \\ 2 & 1 & 3 \end{bmatrix}$. Then $\tilde{R}_1 = 21$, $\tilde{R}_2 = 3233$, and $\tilde{R}_3 = 4333$. For $\mathcal{A}_G = \begin{bmatrix} 3 & 4 \\ 2 & 1 \end{bmatrix}$, we have $\tilde{R}_3 = 34$.

Remark 4.6. Note that except for the column in \mathcal{A}_G containing a_k as in (1) of Definition 4.4, each column of \mathcal{A}_G contains either i or $i + 1$, but not both. Hence the algorithm to construct the reading word in Definition 4.4 is well-defined. Indeed, if \mathcal{A}_G contains the column $\begin{bmatrix} i + 1 \\ i \end{bmatrix}$, then it must be the leftmost column containing $i + 1$ in the top row since by the definition of a Burge array the bottom row is decreasing for equal top row elements. Hence a_k is this leftmost $i + 1$ and either $k = 1$ or $a_{k-1} \leq i$ and $b_{k-1} < i = b_k$, so that a_k is chosen as the first letter of $\tilde{R}_i(\mathcal{A}_G)$.

Definition 4.7. Assign a ')' to every i in $\tilde{R}_i(\mathcal{A}_G)$ and a '(' to every $i + 1$ in $\tilde{R}_i(\mathcal{A}_G)$. Successively pair every '(' that is directly left of a ')', called an i -pair, and remove the paired terms. Continue this process until no more terms can be paired.

The operator \tilde{f}_i acts on \mathcal{A}_G as follows:

- (1) If there are no unpaired ')' terms left, then \tilde{f}_i annihilates \mathcal{A}_G denoted by $\tilde{f}_i(\mathcal{A}_G) = 0$.
- (2) Otherwise locate the i in \mathcal{A}_G corresponding to the rightmost unpaired ')' and denote it by x .
 - (a) If there is no $i + 1$ in the same column as x , then \tilde{f}_i changes x in \mathcal{A}_G to an $i + 1$.
 - (b) If there is an $i + 1$ in the same column as x , then x is on the bottom row of \mathcal{A}_G . Let k be the index such that $b_k = x$. Let ℓ be the smallest index such that $b_\ell \leq b_{\ell+1} \leq \dots \leq b_{k-1}$. Let $\ell \leq m \leq k$ be the largest index such that $b_m < a_\ell$. In the top row of \mathcal{A}_G replace a_{k-1} with an $i + 1$ and replace a_s with a_{s+1} for $\ell \leq s \leq k - 2$. In the bottom row of \mathcal{A}_G replace b_m with a_ℓ and replace b_k with b_m . (Remark: Observe that in this case $k \neq 1$ as

$$\begin{array}{cccccccccccc} \dots & a_\ell & a_{\ell+1} & \dots & a_{m-1} & a_m & a_{m+1} & \dots & a_{k-1} & i+1 & \dots \\ \dots & b_\ell & b_{\ell+1} & \dots & b_{m-1} & b_m & b_{m+1} & \dots & b_{k-1} & i & \dots \end{array} \xrightarrow{\tilde{f}_i} \begin{array}{cccccccccccc} \dots & a_{\ell+1} & a_{\ell+2} & \dots & a_m & a_{m+1} & a_{m+2} & \dots & i+1 & i+1 & \dots \\ \dots & b_\ell & b_{\ell+1} & \dots & b_{m-1} & a_\ell & b_{m+1} & \dots & b_{k-1} & b_m & \dots \end{array}$$

Fig. 4. Action of \tilde{f}_i in Case (2b).

otherwise the $i+1$ and i in this column would form an i -pair in \tilde{R}_i . Thus, this operation is well-defined.) This procedure is illustrated in Fig. 4.

The operator \tilde{e}_i acts on \mathcal{A}_G as follows:

- (1) If there are no unpaired ‘(’ terms left, then \tilde{e}_i annihilates \mathcal{A}_G denoted by $\tilde{e}_i(\mathcal{A}_G) = 0$.
- (2) Otherwise locate the $i+1$ in \mathcal{A}_G corresponding to the leftmost unpaired ‘(’ and denote it by x .
 - (a) If x is in the top row of \mathcal{A}_G and there is an $i+1$ directly to the left of it, then let k be the index such that $a_k = x$. Let ℓ be the smallest index such that $b_\ell \leq b_{\ell+1} \leq \dots \leq b_{k-1}$. Let $\ell \leq m \leq k$ be the smallest index such that $b_k < b_m$. In the top row of \mathcal{A}_G replace a_ℓ with b_m and replace a_s with a_{s-1} for $\ell+1 \leq s \leq k-1$. In the bottom row of \mathcal{A}_G replace b_m with b_k and replace b_k with an i .
 - (b) Otherwise, \tilde{e}_i changes x in \mathcal{A}_G to an i .

Example 4.8. Examples of crystals on Burge arrays are given in Fig. 5. To illustrate the crystal operators \tilde{f}_i of Definition 4.7, consider \tilde{f}_2 on $\mathcal{A}_G = \begin{bmatrix} 2 & 3 & 4 \\ 1 & 2 & 3 \end{bmatrix}$. In this case $\tilde{R}_2(\mathcal{A}_G) = 3223$, x is the 2 in column two of \mathcal{A}_G , $k = 2$, $\ell = 1$, and $m = 1$. We obtain $\tilde{f}_2(\mathcal{A}_G) = \begin{bmatrix} 3 & 3 & 4 \\ 2 & 1 & 3 \end{bmatrix}$. For \tilde{f}_3 on \mathcal{A}_G we have $x = 3$, $k = 3$, $\ell = 1$, $m = 1$, and $\tilde{f}_3(\mathcal{A}_G) = \begin{bmatrix} 3 & 4 & 4 \\ 2 & 2 & 1 \end{bmatrix}$.

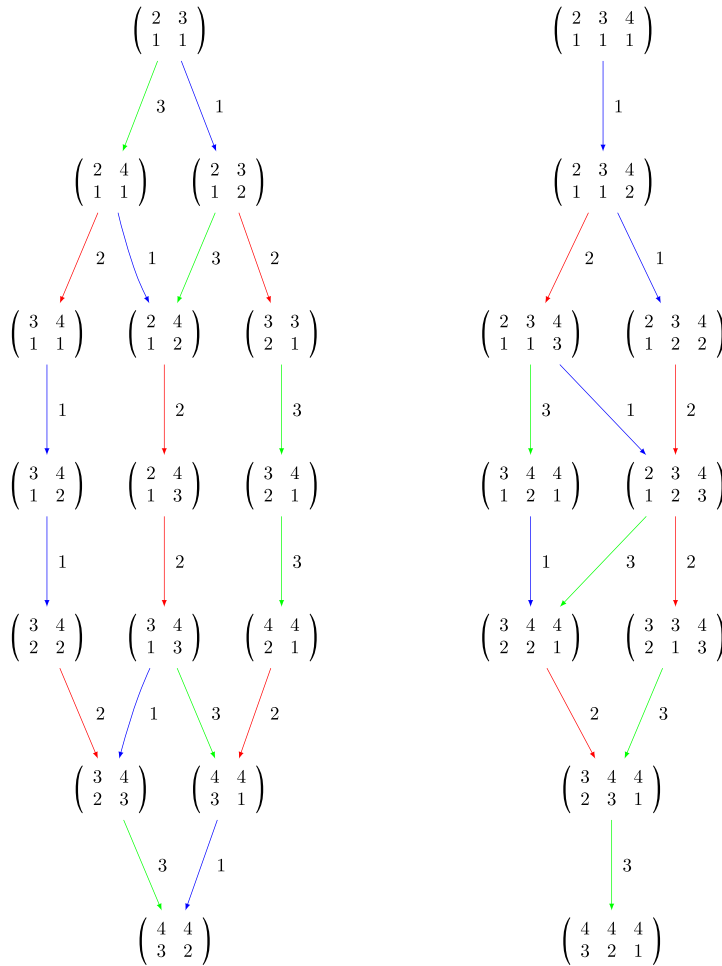
The fact that a non-annihilated $\tilde{f}_i(\mathcal{A}_G)$ (resp. $\tilde{e}_i(\mathcal{A}_G)$) is a PV-free Burge array or even a valid Burge array will be shown as a consequence of Proposition 4.11.

Lemma 4.9. $\tilde{R}_i(\mathcal{A}_G)$ has at most one i -pair.

Proof. Assume that $\tilde{R}_i(\mathcal{A}_G)$ has at least two i -pairs. This implies that \mathcal{A}_G contains at least two $(i+1)$ ’s. Let j be the index of the column containing the leftmost $i+1$ and k be the index of the column containing the second leftmost $i+1$. We break into cases based on the position of the $(i+1)$ ’s in columns j and k .

Case 1: Assume that $a_j = i+1$ and $a_k = i+1$. This implies that a_k is the second leftmost $i+1$ in $\tilde{R}_i(\mathcal{A}_G)$. Since \mathcal{A}_G is assumed to have at least two i -pairs, there must exist $\ell > k$ such that $b_\ell = i$. The subarray $\begin{bmatrix} a_j & a_k & a_\ell \\ b_j & b_k & b_\ell \end{bmatrix} = \begin{bmatrix} i+1 & i+1 & a_\ell \\ b_j & b_k & i \end{bmatrix}$ is then a valley as $b_k \leq i = b_\ell < i+1 = a_k$ and $b_k < b_j$. This contradicts \mathcal{A}_G being a PV-free Burge array.

Case 2: Assume that $a_j = i+1$ and $b_k = i+1$. This once again implies that b_k is the second leftmost $i+1$ in $\tilde{R}_i(\mathcal{A}_G)$. As \mathcal{A}_G contains at least two i -pairs, there must exist $\ell > k$ such that $b_\ell = i$. Let $j < s < \ell$ be the leftmost index such that $b_j \leq i = b_\ell < b_s$. Note that such an s exists as k satisfies



(A) Crystal of Burge arrays of shape (2, 1, 1) with letters in {1, 2, 3, 4}. (B) Crystal of Burge arrays of shape (3, 1, 1, 1) with letters in {1, 2, 3, 4}.

Fig. 5. Examples of crystals on Burge arrays.

the desired conditions. The subarray $\begin{bmatrix} a_j & a_s & a_\ell \\ b_j & b_s & b_\ell \end{bmatrix} = \begin{bmatrix} i+1 & a_s & a_\ell \\ b_j & b_s & i \end{bmatrix}$ is a peak as $b_j \leq i = b_\ell < b_s$ and $a_j = i+1 \leq b_s$. This contradicts \mathcal{A}_G being a PV-free Burge array.

Case 3: Assume that $b_j = i+1$ and $b_k = i+1$. This implies that b_j is the leftmost $i+1$ in $\tilde{R}_i(\mathcal{A}_G)$. To have at least two i -pairs, \mathcal{A}_G must contain columns ℓ and m such that $j < \ell < m$ and $b_\ell = b_m = i$. The subarray $\begin{bmatrix} a_j & a_\ell & a_m \\ b_j & b_\ell & b_m \end{bmatrix} = \begin{bmatrix} a_j & a_\ell & a_m \\ i+1 & i & i \end{bmatrix}$ is a valley as $b_\ell = i \leq i = b_m < a_\ell$ and $b_\ell = i < i+1 = b_j$. This contradicts \mathcal{A}_G being a PV-free Burge array. \square

Lemma 4.10. Let T_G be the threshold tableau associated to \mathcal{A}_G under the Burge correspondence. Then $\tilde{R}_i(\mathcal{A}_G)$ is Knuth equivalent to $R_i(T_G)$.

Proof. Since T_G is a hook tableau, $R_i(T_G)$ has at most one i -pair. Thus by Lemma 4.9 it suffices to prove that $\tilde{R}_i(\mathcal{A}_G)$ has an i -pair if and only if $R_i(T_G)$ has an i -pair, since the content of T_G and \mathcal{A}_G are the same.

Assume that $\tilde{R}_i(\mathcal{A}_G)$ has an i -pair. Let k be the index of the column containing the $i + 1$ that is in the i -pair. First assume that $a_k = i + 1$. If $k = 1$, then a_k is recorded in the leg of T_G and $R_i(T_G)$ has an i -pair. If $b_{k-1} \leq b_k$ then b_k , when inserted, does not bump an element. Otherwise it would bump an element greater than the element bumped by b_{k-1} which would create a non-hook shape. Thus, a_k is recorded in the leg of T_G and $R_i(T_G)$ has an i -pair. If $b_{k-1} > b_k$, then in order for $\tilde{R}_i(\mathcal{A}_G)$ to have an

i -pair, there must exist $\ell > k$ such that $b_\ell = i$. The subarray $\begin{bmatrix} a_{k-1} & a_k & a_\ell \\ b_{k-1} & b_k & b_\ell \end{bmatrix} = \begin{bmatrix} a_{k-1} & i+1 & a_\ell \\ b_{k-1} & b_k & i \end{bmatrix}$

is a valley as $b_k \leq i = b_\ell < i + 1 = a_k$ and $b_{k-1} > b_k$ which is a contradiction. Next assume that $b_k = i + 1$. In order for $\tilde{R}_i(\mathcal{A}_G)$ to have an i -pair, there must exist $\ell > k$ such that $b_\ell = i$. This implies that some $i + 1$ must be bumped into the leg by an insertion letter whose index is at most ℓ . Thus, T_G has an $i + 1$ in its leg and $R_i(T_G)$ has an i -pair. Thus, $R_i(T_G)$ has an i -pair whenever $\tilde{R}_i(\mathcal{A}_G)$ has an i -pair.

Assume that $R_i(T_G)$ has an i -pair. This implies that T_G has an $i + 1$ in its leg. If the $i + 1$ in the leg corresponds to a recording letter a_j for some j , then b_j does not bump an element when inserted. Otherwise $a_j = i + 1$ would get placed in the first row of T_G and cannot be bumped into the leg as $a_1 \leq a_j$. This implies that either $j = 1$ or $b_{j-1} \leq b_j$. In either case, $\tilde{R}_i(\mathcal{A}_G)$ contains an i -pair. Assume the $i + 1$ in the leg corresponds to an insertion letter b_j . Since b_j must be bumped into the leg and be i -paired with some i , there exists $j < k$ such that $b_k = i$. This implies that $\tilde{R}_i(\mathcal{A}_G)$ contains an i -pair. Thus, $\tilde{R}_i(\mathcal{A}_G)$ has an i -pair whenever $R_i(T_G)$ has an i -pair. \square

Proposition 4.11. Let \mathcal{A}_G be a PV-free Burge array and let T_G be its associated threshold tableau.

- (1) If $\tilde{f}_i(\mathcal{A}_G) \neq 0$, then $\tilde{f}_i(\mathcal{A}_G) = \mathcal{A}'_G$, where \mathcal{A}'_G is the associated Burge array of $f_i(T_G)$.
- (2) If $\tilde{e}_i(\mathcal{A}_G) \neq 0$, then $\tilde{e}_i(\mathcal{A}_G) = \tilde{\mathcal{A}}_G$, where $\tilde{\mathcal{A}}_G$ is the associated Burge array of $e_i(T_G)$.

Proof. As \tilde{f}_i and \tilde{e}_i are clearly partial inverses, it suffices to just prove part (1). From Lemma 4.10, we have that $f_i(T_G)$ is not annihilated. Let s be the column index of the rightmost i in \mathcal{A}_G and denote this rightmost i by \bar{i} . We claim that \bar{i} corresponds to the rightmost i in $R_i(T_G)$. If $b_s = \bar{i}$, then \bar{i} is inserted into the arm to the right of all preexisting i 's when column s is inserted and will remain the rightmost i by the properties of Schensted row insertions. If $a_s = \bar{i}$ and $a_{s-1} = i$, then when column s is inserted b_s will bump an element from the arm. As T_G is hook-shaped, $a_s = \bar{i}$ will be recorded into the arm to the right of all preexisting i 's and will remain the rightmost i . If $a_s = \bar{i}$ and $a_{s-1} \neq i$, then a_s is the only i in \mathcal{A}_G and is trivially the rightmost i in $R_i(T_G)$ which proves our claim. As $f_i(T_G)$ is not annihilated and T_G is hook-shaped, \bar{i} is the rightmost unpaired i of T_G and is changed to an $i + 1$ by f_i .

Let r be the column index of the last column of \mathcal{A}_G . We denote by S_t the tableau obtained by reverse inserting columns $t + 1$ through r of $f_i(T_G)$ and T_t the tableau obtained by inserting columns 1 through t of \mathcal{A}_G . We first assume that $r > s$ and prove that columns $s + 1$ through r are the same in both arrays so that we may take r to be the same as s later in the proof. More specifically using induction, we will prove that for $s + 1 \leq t \leq r$ column t of \mathcal{A}'_G is the same as column t in \mathcal{A}_G and $S_{t-1} = f_i(T_{t-1})$.

Let ℓ and a be the largest entries in the leg and arm of $T_r = T_G$, respectively. Note that ℓ and a are also the largest entries in the leg and arm of $S_r = f_i(T_G)$, respectively, as $r > s$. We break into subcases depending on whether $\ell > a$ or $\ell \leq a$.

When $\ell > a$, the column $[\ell, a]^T$ is obtained by reverse inserting both T_r and S_r implying the r th column of \mathcal{A}'_G is the same as the r th column of \mathcal{A}_G . We claim that $\ell > i + 1$. This is clearly true if \bar{i} is in the leg of T_G as we assume $r > s$. If \bar{i} is in the arm of T_r , then $a \geq i + 1$ as a is to the right of \bar{i} and \bar{i} is the rightmost i in T_r . Hence the claim is true, and \bar{i} remains the rightmost unpaired i in T_{r-1} . Thus, f_i acts on T_{r-1} by changing \bar{i} to an $i + 1$ which is precisely S_{r-1} .

When $\ell \leq a$, a is removed from T_r and a number which we denote by b_r is reverse inserted. This implies that the r th column of \mathcal{A}_G is $[a, b_r]^T$. Similarly, reverse inserting a column from S_r , a is

removed from S_r and a number which we denote by c_r is reverse inserted. Note that b_r and c_r are equal and are in the same cell of their corresponding tableau except if an i and $i + 1$ lie in the arm of S_r and the $(2, 1)$ cell of S_r is an $i + 1$. If an i and $i + 1$ lie in the arm of S_r , then the value of the $(2, 1)$ entry in T_r cannot be \bar{i} as this would contradict \bar{i} being the rightmost unpaired i of T_r . Thus, if an i and $i + 1$ lie in the arm of S_r and the $(2, 1)$ cell of S_r is an $i + 1$, then the $(2, 1)$ cell of T_r is an $i + 1$. However, this would imply $b_r = i$ as it would need to bump an $i + 1$ so that a is in the arm of T_r , an $i + 1$ is in the $(2, 1)$ cell, and the i in the arm is not bumped. This contradicts $r > s$. Therefore the r th column of \mathcal{A}_G and \mathcal{A}'_G match. As $b_r \neq i$, \bar{i} is still the rightmost unpaired i of T_{r-1} and $f_i(T_{r-1}) = S_{r-1}$. Assume that column t of \mathcal{A}'_G is the same as column t in \mathcal{A}_G and $S_{t-1} = f_i(T_{t-1})$ for some $s + 1 < t \leq r$. Repeating the argument in the base case, we see that this holds for the case $t - 1$ as well.

From the definition of \tilde{f}_i , we have that the columns $s + 1$ through r of $\tilde{f}_i(\mathcal{A}_G)$ are the same as the columns $s + 1$ through r of \mathcal{A}_G . By the previous paragraph, these columns are also equal to columns $s + 1$ through r of \mathcal{A}'_G . We now assume that $r = s$ and prove that the columns 1 through s of $\tilde{f}_i(\mathcal{A}_G)$ and \mathcal{A}'_G are equal by breaking into cases based off the position of \bar{i} in T_s .

Case 1: \bar{i} is in the leg of T_s .

Since \bar{i} is the rightmost unpaired i of T_s , this implies that T_s does not contain any other i except for \bar{i} . Also in order for \bar{i} to be in column s of \mathcal{A}_G , we must have $a_s = \bar{i}$. Thus, the column obtained from T_s by reverse inserting is of the form $[\bar{i}, a]^T$ where a is the largest value in the arm of T_s . In S_s , \bar{i} is replaced with an $i + 1$. Thus, the column obtained from S_s by reverse inserting is of the form $[i + 1, a]^T$. We see that T_{s-1} and S_{s-1} are equal implying columns 1 through $s - 1$ of \mathcal{A}_G and \mathcal{A}'_G are equal. Note that as \bar{i} is the only i in \mathcal{A}_G and $a_s = \bar{i}$, we have \tilde{f}_i acts by changing \bar{i} to an $i + 1$ in \mathcal{A}_G . This is precisely the form of \mathcal{A}'_G implying $\tilde{f}_i(\mathcal{A}_G) = \mathcal{A}'_G$.

Case 2: \bar{i} is in the arm of T_s .

Let ℓ and a be the largest elements in the leg and arm of T_s , respectively. We break into subcases.

Assume $\bar{i} \geq \ell$. In order for \bar{i} to be in the s th column of \mathcal{A}_G , \bar{i} must be a ; otherwise the column reverse inserted from T_s would be $[a, \bar{i}]^T$. This implies \bar{i} must bump an entry in the arm of T_{s-1} into its leg which would contradict $\bar{i} \geq \ell$. In particular, this implies that $a = \bar{i} \geq \ell$. Furthermore, the column obtained from T_s when reverse inserting is of the form $[\bar{i}, b_s]^T$, where b_s is the largest element in the arm of T_s strictly less than the entry in the cell $(2, 1)$. Since \bar{i} is the largest value in T_s , the $i + 1$ in S_s created by applying f_i to T_s is also the largest value. Thus, the column obtained from S_s when reverse inserting is of the form $[i + 1, c_s]^T$, where c_s is the largest element in the arm of S_s strictly less than the entry in the cell $(2, 1)$. Since $f_i(T_s) = S_s$, we have $b_s = c_s$ and $T_{s-1} = S_{s-1}$. Thus, columns 1 through $s - 1$ of \mathcal{A}_G are identical to the corresponding columns of \mathcal{A}'_G . As \bar{i} is the rightmost i in \mathcal{A}_G and is in the top row, it is also the rightmost unpaired i in \mathcal{A}_G . Thus, \tilde{f}_i acts by changing \bar{i} to an $i + 1$ in \mathcal{A}_G . Thus, $\tilde{f}_i(\mathcal{A}_G) = \mathcal{A}'_G$.

Assume now that $\ell > a$ and $\ell \neq i + 1$. In order for \bar{i} to be in the s th column of \mathcal{A}_G , \bar{i} must be equal to a . When reversing the Burge correspondence of T_s , the column obtained is then of the form $[\ell, \bar{i}]^T$. From our assumption, we also have $\ell > i + 1$ which implies that the column obtained from S_s when reversing the Burge correspondence is $[\ell, i + 1]^T$. Observe that $T_{s-1} = S_{s-1}$ which forces columns 1 through $s - 1$ of \mathcal{A}_G and \mathcal{A}'_G to be equal. We prove that \bar{i} is the rightmost unpaired i in \mathcal{A}_G . For there to be a hope that this claim is not true, then there must exist either a column $[i + 1, b_j]^T$ with $b_j \neq i$ or $[a_j, i + 1]^T$ in \mathcal{A}_G . Note that there can only be one column with an $i + 1$ in the top row; otherwise it would form a valley with columns $[i + 1, b_j]^T$ and $[\ell, \bar{i}]^T$. If there exists a column of the form $[i + 1, b_j]^T$, then in order for \bar{i} to be unpaired in T_s there must be an i in a column $j + 1$ through $s - 1$ or a column of the form $[i, b_j]^T$. Note that if there is an i in columns $j + 1$ through $s - 1$ this would imply \bar{i} is the rightmost unpaired i of \mathcal{A}_G . If there is no i in columns $j + 1$ through $s - 1$,

we must then have that column $j - 1$ is of the form $[i, b_{j-1}]^T$. If $b_{j-1} > b_j$, then $\begin{bmatrix} i & i + 1 & \ell \\ b_{j-1} & b_j & \bar{i} \end{bmatrix}$

is a valley implying $b_{j-1} \leq b_j$. Thus, if there exists a column of the form $[i + 1, b_j]^T$ with no i 's in columns j through $s - 1$, then \bar{i} must be the rightmost unpaired i . If there exists a column of the form $[a_j, i + 1]^T$ with no i 's in columns j through $s - 1$, this would imply \bar{i} when inserted would

bump an element from the arm of T_s contradicting that ℓ is in the leg of T_s . Thus, if there exists a column of the form $[a_j, i + 1]^T$, then there exists an i in columns j through $s - 1$. Therefore, \bar{i} is the rightmost unpaired i of \mathcal{A}_G and \tilde{f}_i acts by changing \bar{i} to an $i + 1$ in the s th column. This is precisely \mathcal{A}'_G .

Assume that $\ell > a$ and $\ell = i + 1$. In order for \bar{i} to be in column s of \mathcal{A}_G , a must be precisely \bar{i} . When reversing the Burge correspondence of T_s , the column obtained is then of the form $[\ell = i + 1, \bar{i}]^T$. Since $\ell = i + 1$ came from the leg of T_s , there must exist an i somewhere in columns 1 through $s - 1$; otherwise \bar{i} would not be the rightmost unpaired i in T_s . Thus, \bar{i} is also the rightmost unpaired i of \mathcal{A}_G . Let x be the value in the $(2, 1)$ position of T_s and let y be the rightmost element in the arm of T_s that is strictly less than x . Note that $x < i + 1 = \ell$; otherwise T_s would have shape $(1, 1)$ and $R_i(T_s) = i + 1$. This implies $y \neq i$. We also have y is not an element in the top row of \mathcal{A}_G . Otherwise if $a_m = y$ for some m , then $x > y \geq a_1$ which contradicts x being in the $(2, 1)$ cell. Thus, $y = b_m$ for some $1 \leq m \leq s - 1$.

Assume now that x is a recording letter. As x lies in the $(2, 1)$ cell of T_s , we must have $x = a_1$. This implies $b_1 \leq b_2 \leq \dots \leq b_s = \bar{i}$ and $a_1 < a_2 < \dots < a_s = \ell$. Recall that in S_s , $b_s = \bar{i}$ is replaced with an $i + 1$. Hence, when reverse inserting a column from S_s , the $i + 1$ that replaced \bar{i} is removed, $y = b_m$ is replaced by $x = a_1$, and the rest of the entries in the leg are shifted up. Thus column s of \mathcal{A}'_G is of the form $[i + 1, b_m]^T$. We see that S_{s-1} is then associated to the Burge array

$$\begin{bmatrix} a_2 & a_3 & \dots & a_m & a_{m+1} & a_{m+2} & \dots & a_s = \ell = i + 1 \\ b_1 & b_2 & \dots & b_{m-1} & a_1 & b_{m+1} & \dots & b_{s-1} \end{bmatrix} \text{ which mimics the action of } \tilde{f}_i \text{ on } \mathcal{A}_G \text{ as}$$

$b_1 \leq \dots \leq b_{s-1}$ and b_m is the rightmost entry such that $b_m < a_1$.

Assume now that x lies in the bottom row of \mathcal{A}_G . This implies that there exists b_n that bumped x in T_{n-1} to the $(2, 1)$ cell. Note that $b_{n-1} > b_n$; otherwise $\begin{bmatrix} a_z & a_{n-1} & a_n \\ x & b_{n-1} & b_n \end{bmatrix}$ would form a valley in \mathcal{A}_G . Since x is the value in the $(2, 1)$ cell, $b_n \leq b_{n+1} \leq \dots \leq b_s = \bar{i}$ in \mathcal{A}_G . We also have $b_{n+1} \geq a_n$. Otherwise $\begin{bmatrix} a_z & a_n & a_{n+1} \\ x & b_n & b_{n+1} \end{bmatrix}$ would form a valley. Thus, b_n is the largest element in the bottom row

from column n to $s - 1$ such that $b_n < a_n$ which implies $m = n$. Moreover, $a_n < \dots < a_s = \ell$ are all in the legs of both of T_s and S_s . As S_s differs from T_s by changing \bar{i} to an $i + 1$, we have that reversing the Burge correspondence removes the $i + 1$ from the arm of S_s , b_n is replaced by x , and the rest of the leg entries are shifted up. Thus column s of \mathcal{A}'_G is $[i + 1, b_n]^T$. As b_n bumped x out, we see that x is in the cell that it was originally inserted into. We see that reverse the Burge correspondence for

S_{s-1} up to S_{n-1} , we get the columns $\begin{bmatrix} a_{n+1} & a_{n+2} & \dots & a_{s-1} & a_s = \ell = i + 1 \\ a_n & b_{n+1} & \dots & b_{s-2} & b_{s-1} \end{bmatrix}$ and S_{n-1} is equal to

T_{n-1} as x is in its original cell. These changes to \mathcal{A}_G are seen to be the same as \tilde{f}_i as n is the leftmost column index such that $b_n \leq \dots \leq b_{s-1}$ and n is the rightmost column index between n and $s - 1$ such that $b_n < a_n$.

Assume that $\bar{i} < \ell \leq a$ and the $(2, 1)$ cell of T_s is not an $i + 1$. Let x be the value in the $(2, 1)$ position of T_s and let y be the rightmost element in the arm of T_s that is strictly less than x . Note that for \bar{i} to be in the s th column of \mathcal{A}_G , y must equal \bar{i} . As x is not equal to $i + 1$, we have also $i + 1 < x$. When reversing the Burge correspondence of T_s and S_s the columns obtained are $[a, \bar{i}]^T$ and $[a, i + 1]^T$ respectively where the $i + 1$ reverse bumped from S_s was the $i + 1$ created by f_i . We have T_{s-1} and S_{s-1} are equal implying columns 1 through $s - 1$ of \mathcal{A}_G and \mathcal{A}'_G are the same. Note that there cannot be an $i + 1$ in columns 1 through $s - 1$. Otherwise there would either be an $i + 1$ in the leg of T_{s-1} which would contradict $x > i + 1$ or $i + 1$ is in the arm of T_{s-1} in which case \bar{i} would bump an $i + 1$ instead of x . Thus, \bar{i} is the rightmost unpaired i of \mathcal{A}_G and \tilde{f}_i changes \bar{i} to an $i + 1$.

Assume that $\bar{i} < \ell \leq a$ and the $(2, 1)$ cell of T_s is an $i + 1$. Let $x = i + 1$ be the value in the $(2, 1)$ cell of T_s and y be the rightmost element in the arm of T_s that is strictly less than x . For \bar{i} to be in the s th column of \mathcal{A}_G , y must equal \bar{i} . As column s in \mathcal{A}_G is of the form $[a, \bar{i}]^T$, \bar{i} bumps $i + 1$ from the arm of T_{s-1} . Since x is bumped into the cell $(2, 1)$, we have $x = b_n$ for some n . Observe that no i can be strictly between columns n through s in \mathcal{A}_G ; otherwise T_{s-1} would contain an $i + 1$ in its

leg. From this observation and the fact that \bar{i} is the rightmost unpaired i in T_G , there must exist an i somewhere in columns 1 through $n - 1$ in \mathcal{A}_G .

Let m be the column of the index of the second rightmost i in \mathcal{A}_G which by the reasoning above satisfies $m < n$. We claim that i must be the insertion letter in column m , i.e. $b_m = i$. If $a_m = i$, then b_m must have bumped an element in T_{m-1} when inserted; otherwise an i would be present in the leg of T_{s-1} . Let b_z be the element bumped by b_m . This implies b_z is in the leg of T_{m-1} ; however, $b_z < a_z \leq a_m = i < i + 1$ which would be a contradiction. Thus our claim that $b_m = i$ holds. We also have $a_m \neq i + 1$ as $\tilde{R}_i(\mathcal{A}_G)$ can have at most one i -pair by Lemma 4.9. From these two facts, b_m cannot have bumped an element b_u when inserted into T_{m-1} . Otherwise $\begin{bmatrix} a_u & a_m & a \\ b_u & b_m & i \end{bmatrix}$ would be a valley. As $y = \bar{i}$ is turned into an $i + 1$ in S_s , the rightmost element in the arm of S_s that is strictly less than $x = i + 1$ is b_m . Thus, the column reverse inserted from S_s is $[a, b_m = i]^T$ while the column reverse inserted from T_s is $[a, \bar{i}]^T$. This implies the s th columns of \mathcal{A}_G and \mathcal{A}'_G are the same, but S_{s-1} differs from T_{s-1} . Note that columns $m + 1$ through $s - 1$ of both \mathcal{A}_G and \mathcal{A}'_G are the same as the reverse insertions of T_s and S_s in these steps do not involve b_m or the $i + 1$ created by f_i respectively. As b_m did not bump an element when inserted into T_{m-1} , we have a_m is in the leg of both T_m and S_m and is the largest entry. We see $[a_m, b_m]^T$ is reverse inserted from T_m and $[a_m, i + 1]^T$ is reverse inserted from S_m and $S_{s-1} = T_{s-1}$. Thus, the only difference between \mathcal{A}_G and \mathcal{A}'_G is that b_m is changed to an $i + 1$ in \mathcal{A}_G . Note that b_m is the rightmost unpaired i in \mathcal{A}_G since it is the second rightmost i and \bar{i} is i -paired with the b_n . Thus, \tilde{f}_i acts by changing b_m in \mathcal{A}_G to an $i + 1$ which is precisely \mathcal{A}'_G . \square

Corollary 4.12. Let C_m be the set of all PV-free Burge arrays with entries at most m . Then C_m together with the operators \tilde{f}_i and \tilde{e}_i forms a Stembridge crystal of type A_{m-1} .

Proof. The Burge correspondence is a crystal isomorphism between C_m and $\bigsqcup_{\substack{\lambda \text{ hook-shaped,} \\ \text{threshold}}} \text{Tab}_m(\lambda)$ by

Proposition 4.11, where $\text{Tab}_m(\lambda)$ is the set of all semistandard Young tableaux of shape λ and entries at most m together with the usual crystal operators as in Definition 4.3. Since $\text{Tab}_m(\lambda)$ forms a Stembridge crystal, so does C_m . \square

Corollary 4.13. Let \mathcal{A}_G be a PV-free Burge array corresponding to a graph G on n vertices. Then \mathcal{A}_G is highest weight if and only if G is star-shaped (up to singletons) such that the central vertex is labelled 1 and the other vertices have labels $\{2, \dots, n\}$.

For a crystal C , an element $b \in C$ is called *extremal* if either $f_i(b) = 0$ or $e_i(b) = 0$ for each i in the index set and its weight is in the Weyl orbit of the highest weight element in the crystal component (see [20]). Let the weight of the highest weight vector $u \in C$ (which satisfies $e_i(u) = 0$ for all i) be $\text{wt}(u) = \lambda$. The weight of the extremal vectors are permutations of λ . The tableaux under the Burge correspondence are threshold shapes. Hence, by the definition of threshold graphs, the extremal vectors of the crystal correspond to threshold graphs under the Burge correspondence.

Acknowledgements

We thank Amritanshu Prasad for fruitful discussions.

AS was partially supported by NSF, USA grants DMS-1760329 and DMS-2053350.

References

- [1] G.d.B. Robinson, On the representations of the symmetric group, Amer. J. Math. 60 (3) (1938) 745–760, <http://dx.doi.org/10.2307/2371609>.
- [2] C. Schensted, Longest increasing and decreasing subsequences, Canad. J. Math. 13 (1961) 179–191, <http://dx.doi.org/10.4153/CJM-1961-015-3>.
- [3] D.E. Knuth, Permutations, matrices, and generalized Young tableaux, Pacific J. Math. 34 (1970) 709–727.

- [4] W.H. Burge, Four correspondences between graphs and generalized Young tableaux, *J. Comb. Theory Ser. A* 17 (1974) 12–30, [http://dx.doi.org/10.1016/0097-3165\(74\)90024-7](http://dx.doi.org/10.1016/0097-3165(74)90024-7).
- [5] I.G. Macdonald, *Symmetric Functions and Hall Polynomials*, second ed., in: *Oxford Classic Texts in the Physical Sciences*, The Clarendon Press, Oxford University Press, New York, 2015, p. xii+475, With contribution by A. V. Zelevinsky and a foreword by Richard Stanley, Reprint of the 2008 paperback edition.
- [6] D. Paul, A. Prasad, A. Sadhukhan, Tableau correspondences and representation theory, in: *Contributions in Algebra and Algebraic Geometry*, in: *Contemp. Math.*, vol. 738, Amer. Math. Soc. Providence, RI, 2019, pp. 109–123, <http://dx.doi.org/10.1090/conm/738/14881>.
- [7] P. Erdős, T. Gallai, Graphs with prescribed degrees of vertices (in Hungarian), *Mat. Lapok* 11 (1960) 264–272.
- [8] E. Ruch, I. Gutman, The branching extent of graphs, *J. Comb. Inf. Syst. Sci.* 4 (4) (1979) 285–295.
- [9] U.N. Peled, M.K. Srinivasan, The polytope of degree sequences, *Linear Algebra Appl.* 114/115 (1989) 349–377, [http://dx.doi.org/10.1016/0024-3795\(89\)90470-9](http://dx.doi.org/10.1016/0024-3795(89)90470-9).
- [10] N.V.R. Mahadev, U.N. Peled, *Threshold Graphs and Related Topics*, in: *Annals of Discrete Mathematics*, vol. 56, North-Holland Publishing Co. Amsterdam, 1995, p. xiv+543.
- [11] R. Merris, T. Roby, The lattice of threshold graphs, *JIPAM. J. Inequal. Pure Appl. Math.* 6 (1) (2005) Article 2, 21.
- [12] V. Gasharov, The Erdos-Gallai criterion and symmetric functions, *European J. Combin.* 18 (3) (1997) 287–294, <http://dx.doi.org/10.1006/eujc.1996.0096>.
- [13] A. Bhattacharya, S. Sivasubramanian, M.K. Srinivasan, The polytope of degree partitions, *Electron. J. Combin.* 13 (1) (2006) Research Paper 46, 18.
- [14] C. Klivans, V. Reiner, Shifted set families, degree sequences, and plethysm, *Electron. J. Combin.* 15 (1) (2008) Research Paper 14, 35.
- [15] D. Bump, A. Schilling, *Crystal Bases*, World Scientific Publishing Co. Pte. Ltd. Hackensack, NJ, 2017, p. xii+279, <http://dx.doi.org/10.1142/9876>, Representations and combinatorics.
- [16] J. Hong, S.-J. Kang, Introduction to Quantum Groups and Crystal Bases, in: *Graduate Studies in Mathematics*, vol. 42, American Mathematical Society, Providence, RI, 2002, p. xviii+307, <http://dx.doi.org/10.1090/gsm/042>.
- [17] C. Greene, An extension of Schensted's theorem, *Adv. Math.* 14 (1974) 254–265, [http://dx.doi.org/10.1016/0001-8708\(74\)90031-0](http://dx.doi.org/10.1016/0001-8708(74)90031-0).
- [18] T. Britz, S. Fomin, Finite posets and Ferrers shapes, *Adv. Math.* 158 (1) (2001) 86–127, <http://dx.doi.org/10.1006/aima.2000.1966>.
- [19] R.P. Stanley, *Enumerative Combinatorics. Vol. 2*, in: *Cambridge Studies in Advanced Mathematics*, vol. 62, Cambridge University Press, Cambridge, 1999, p. xii+581, <http://dx.doi.org/10.1017/CBO9780511609589>, With a foreword by Gian-Carlo Rota and appendix 1 by Sergey Fomin.
- [20] M. Kashiwara, Crystal bases of modified quantized enveloping algebra, *Duke Math. J.* 73 (2) (1994) 383–413, <http://dx.doi.org/10.1215/S0012-7094-94-07317-1>.