A Bijection Between $K$-Kohnert Diagrams and Reverse Set-Valued Tableaux

Jianping Pan\textsuperscript{a}  \quad Tianyi Yu\textsuperscript{b}

Submitted: Aug 8, 2022; Accepted: Oct 1, 2023; Published: Nov 17, 2023
© The authors. Released under the CC BY-ND license (International 4.0).

Abstract

Lascoux polynomials are $K$-theoretic analogues of the key polynomials. They both have combinatorial formulas involving tableaux: reverse set-valued tableaux (RSVT) rule for Lascoux polynomials and reverse semistandard Young tableaux (RSSYT) rule for key polynomials. Furthermore, key polynomials have a simple algorithmic model in terms of Kohnert diagrams, which are in bijection with RSSYT. Ross and Yong introduced $K$-Kohnert diagrams, which are analogues of Kohnert diagrams. They conjectured a $K$-Kohnert diagram rule for Lascoux polynomials. We establish this conjecture by constructing a weight-preserving bijection between RSVT and $K$-Kohnert diagrams.

Mathematics Subject Classifications: 05E05

1 Introduction

Fix a positive integer $n$ throughout this paper. A weak composition of length $n$ is a sequence of $n$ non-negative integers. If $\alpha$ is a weak composition, we use $\alpha_i$ to denote its $i^{th}$ entry.

Key polynomials $\kappa_\alpha$ are homogeneous polynomials labeled by weak compositions. They were first introduced by Demazure [7] as the characters of the Demazure modules. Further studies [11, 16, 15, 20, 10, 13, 14, 3, 4] provided several combinatorial formulas.

Lascoux polynomials $L_\alpha^{(\beta)}$ are $K$-theoretic generalizations of key polynomials [12]. They are inhomogeneous polynomials with an extra variable $\beta$. Setting $\beta = 0$ in $L_\alpha^{(\beta)}$ yields $\kappa_\alpha$. There are several existing combinatorial formulas for $L_\alpha^{(\beta)}$ involving set-valued skyline fillings [18] and set-valued tableaux [6, 26]. In this paper, we will define Lascoux polynomials by a combinatorial formula involving reverse set-valued tableaux (RSVT). It first appeared implicitly in [6] and was rediscovered by Shimozono and the second

\textsuperscript{a}Department of Mathematics, North Carolina State University, Raleigh, North Carolina, U.S.A.  \hspace{1cm} (jpan9@ncsu.edu).

\textsuperscript{b}Department of Mathematics, University of California San Diego, La Jolla, California, U.S.A.  \hspace{1cm} (tiy059@ucsd.edu).

\url{https://doi.org/10.37236/11434}
author [24]. Specifically, for each weak composition $\alpha$, there is a set $\text{RSVT}(\alpha)$, which consists of certain $\text{RSVT}$ satisfying a left key condition (see subsection 2.1). Then $\mathcal{L}_\alpha^{(\beta)}$ can be written as a sum over $\text{RSVT}(\alpha)$:

$$\mathcal{L}_\alpha^{(\beta)} := \sum_{T \in \text{RSVT}(\alpha)} \beta^{\text{ex}(T)} x^{\text{wt}(T)}.$$

Ross and Yong [21] defined a generalization of Kohnert’s move on diagrams [11]. We call them $K$-Kohnert moves. Repeatedly applying $K$-Kohnert moves on the key diagram of $\alpha$ yields a set of diagrams, which is denoted as $\text{KKD}(\alpha)$ (see subsection 2.3).

**Conjecture 1.** [21] The Lascoux polynomials indexed by $\alpha$, is given by

$$\mathcal{L}_\alpha^{(\beta)} = \sum_{D \in \text{KKD}(\alpha)} \beta^{\text{ex}(D)} x^{\text{wt}(D)}.$$

Pechenik and Scrimshaw [19] proved a special case of this conjecture where all positive numbers in $\alpha$ are the same. This paper will prove the conjecture for all $\alpha$.

**Theorem 2.** Conjecture 1 is true.

To prove this theorem, we define two maps: $\Psi_\alpha$ on $\text{KKD}(\alpha)$ (see subsection 3.1) and $\Phi_\alpha$ on $\text{RSVT}(\alpha)$ (see subsection 3.2). We will show $\Psi_\alpha$ (resp. $\Phi_\alpha$) is a well-defined map to $\text{RSVT}(\alpha)$ (resp. $\text{KKD}(\alpha)$). Finally, we establish the following.

**Theorem 3.** The maps $\Psi_\alpha : \text{KKD}(\alpha) \rightarrow \text{RSVT}(\alpha)$ and $\Phi_\alpha : \text{RSVT}(\alpha) \rightarrow \text{KKD}(\alpha)$ are mutually inverses of each other. Moreover, they preserve $\text{wt}(\cdot)$ and $\text{ex}(\cdot)$.

Kohnert rules were originally developed by Kohnert [11] to compute key polynomials, which repeatedly applies Kohnert moves on the key diagram of a weak composition. There exist other Kohnert rules. For example, Kohnert also defined a Kohnert rule for the Schubert polynomials, which repeatedly applies Kohnert moves on the Rothe diagram of a permutation. This gave the first monomial positive formula for Schubert polynomials. Armon, Assaf, Bowling, and Ehrhard [1] proved the characters of flagged Schur modules for northwest diagrams can be computed using Kohnert rules.

The $K$-Kohnert rule discussed in this paper is a $K$-theoretic generalization of the Kohnert rule for key polynomials, which is also the first proven $K$-Kohnert rule. It would be interesting to extend Kohnert technology to the rest of the $K$-world. For example, the $K$-theoretic generalization of Schubert polynomials are the Grothendieck polynomials and the corresponding $K$-Kohnert rule was also conjectured by Ross and Yong [21].

The paper is organized as follows. In Section 2, we review related combinatorial rules for $\kappa_\alpha$ and $\mathcal{L}_\alpha^{(\beta)}$. In Section 3, we define two maps $\Psi_\alpha$ and $\Phi_\alpha$ on $\text{KKD}(\alpha)$ and $\text{RSVT}(\alpha)$ respectively. The following sections will prove Theorem 3. In Section 4, we introduce a partial order on all weak compositions. We call it the Bruhat order and show it is equivalent to the left swap order in [3]. In Section 5, we describe the sets $\text{KKD}(\alpha)$ and $\text{RSVT}(\alpha)$ recursively using the Bruhat order. In Section 6, we introduce two auxiliary
operators $\sharp_g$ and $\flat_e$ on $\text{KD}(\alpha)$ and discuss their properties. In Section 7, we give recursive descriptions of maps $\Psi_\alpha$ and $\Phi_\alpha$ in terms of $\sharp_g$ and $\flat_e$. Then we show $\Psi_\alpha$ (resp. $\Phi_\alpha$) is a well-defined map to $\text{RSVT}(\alpha)$ (resp. $\text{KKD}(\alpha)$) using the recursive definitions developed in Section 5. Finally we prove Theorem 3.

2 Background

2.1 $\text{RSSYT}(\alpha)$ and $\text{RSVT}(\alpha)$

Given a partition $\lambda = (\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_\ell \geq 0)$, a Young diagram of shape $\lambda$ is a finite collection of boxes, aligned at the left, in which the $i^{th}$ row has $\lambda_i$ boxes. We use English convention for our Young diagrams and tableaux, so the first row is the highest row.

A reverse semistandard Young tableau of shape $\lambda$ is a filling of the Young diagram $\lambda$ with positive numbers such that

(i) each box contains exactly one number,
(ii) the entries in each row weakly decrease from left to right, and
(iii) the entries of each column strictly decrease from top to bottom.

Let $\text{RSSYT}_\lambda$ be the set of all the reverse semistandard Young tableaux of shape $\lambda$.

Following [22], we introduce another set of tableaux where a box might have more than one number. A reverse set-valued tableau of shape $\lambda$ is a filling of the Young diagram $\lambda$ with positive numbers such that

(i) each box contains a finite and non-empty set of positive integers,
(ii) if a set $A$ is to the left of a set $B$ in the same row, then $\min(A) \geq \max(B)$, and
(iii) if a set $C$ is below a set $A$ in the same column, then $\min(A) > \max(C)$.

Let $\text{RSVT}_\lambda$ be all the reverse set-valued tableaux of shape $\lambda$.

Let the weight vector for $T$ be the weak composition whose $i^{th}$ component is the total number of appearances of $i$ in $T$, denoted by $\text{wt}(T)$. Given any weak composition $\alpha$, let $|\alpha| = \sum_{i \geq 1} \alpha_i$. Given $T \in \text{RSVT}_\lambda$, define $L(T)$ to be the element in $\text{RSSYT}_\lambda$ constructed by only keeping the largest number in each box of $T$. We call these numbers the leading numbers of $T$. Any number in $T$ that is not a leading number is called an extra number. Let the excess of $T$ be the number of extra numbers in $T$, so we can denote it by $\text{ex}(T) = |\text{wt}(T)| - |\lambda|$.

Next we give the definition of left key of $T$, denoted by $K_-(T)$, where $T$ is a $\text{RSSYT}$. It was first given in [25, Section 5]. We give the description as in [24, Definition 3.11].

**Definition 4.** Let $C_1, C_2$ be two adjacent columns of a $\text{RSSYT}$ with $C_1$ on the left. We may view $C_1$ and $C_2$ as sets. We define $C_1 \prec C_2$ as follows. Assume $C_2 = \{a_1 < a_2 < \cdots < a_m\}$. Start by finding the smallest $b_1 \in C_1$ such that $b_1 \geq a_1$. Then find the smallest
Given a reverse semistandard Young tableau $T$ with columns $C_1, C_2, \ldots, C_n$. Then its left key $K_-(T)$ is a RSVT constructed by taking $C_1 < C_2 \cdots < C_k$ as its $k^{th}$ column.

Example 5. Consider the following $T \in \text{RSVT}_{(3,2)}$. We have $\text{wt}(T) = (2, 2, 2, 1, 0, 1)$ and $\text{ex}(T) = 3$. We can also compute $L(T)$ and its left key.

$$T = \begin{bmatrix} 64 & 32 & 2 \\ 31 & 1 \end{bmatrix}, \quad L(T) = \begin{bmatrix} 6 & 3 & 2 \\ 3 & 1 \end{bmatrix}, \quad K_-(T) = K_-(L(T)) = \begin{bmatrix} 6 & 6 & 3 \\ 3 & 3 \end{bmatrix}.$$

Given a weak composition $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, let $\alpha^+$ be the partition obtained from $\alpha$ by sorting the numbers in decreasing order and ignoring the trailing 0’s. Define the key tableau for $\alpha$ to be the unique element in $\text{RSSYT}_{\alpha^+}$ whose $j^{th}$ column consists of the numbers $\{i | \alpha_i \geq j\}$. Denote this tableau by $\text{key}(\alpha)$.

Remark 6. For any reverse set-valued tableau $T$, $K_-(T)$ is a key tableau for some weak composition $\alpha$.

With the above concepts, we now define the subsets of $\text{RSSYT}_{\alpha^+}$ and $\text{RSVT}_{\alpha^+}$ that will be used to compute $\kappa_\alpha$ and $\xi^{(\beta)}_\alpha$.

$$\text{RSSYT}(\alpha) := \{ T \in \text{RSSYT}_{\alpha^+} : K_-(T) \leq \text{key}(\alpha) \}$$

$$\text{RSVT}(\alpha) := \{ T \in \text{RSVT}_{\alpha^+} : K_-(T) \leq \text{key}(\alpha) \}$$

Here the $\leq$ relation means entry-by-entry comparison. For example, $T$ in Example 5 is in $\text{RSVT}((0, 0, 2, 0, 0, 3))$ but not in $\text{RSVT}((0, 2, 0, 0, 3))$.

We now list the combinatorial formulas in [16, 15, 20] for key polynomials, and in [6, 24] for Lascoux polynomials labeled by a weak composition $\alpha$:

$$\kappa_\alpha := \sum_{T \in \text{RSSYT}(\alpha)} \beta^\text{wt}(T), \quad \xi^{(\beta)}_\alpha := \sum_{T \in \text{RSVT}(\alpha)} \beta^\text{ex}(T) \beta^\text{wt}(T).$$

2.2 Viewing $\text{RSVT}(\alpha)$ as a pair of diagrams

In this subsection, we introduce another perspective on $\text{RSVT}(\alpha)$. A diagram is a finite subset of $\mathbb{N} \times \mathbb{N}$. We may represent a diagram by putting a box at row $r$ and column $c$ for each $(c, r)$ in the diagram. We adopt the convention where columns begin at 1 from the left and rows begin at 1 from the bottom. The weight of a diagram $D$, denoted as $\text{wt}(D)$, is a weak composition whose $i^{th}$ entry is the number of boxes in its $i^{th}$ row.

A diagram pair is an ordered pair $D = (D_1, D_2)$ such that $D_1$ and $D_2$ are disjoint diagrams. We may represent $D$ by putting a box at $(c, r)$ for each $(c, r) \in D_1$ and putting
a box with label $X$ at $(c,r)$ for each $(c,r) \in D_2$. Cells in $D_1$ are called Kohnert cells. Cells in $D_2$ are called ghost cells. The weight of $D$, denoted as $\text{wt}(D)$, is a weak composition whose $i^{th}$ entry is the number of Kohnert cells and ghost cells in its $i^{th}$ row. Let the excess of $D$, denoted by $\text{ex}(D)$, be $|D_2|$.

Now we embed the set of RSVT into the set of diagram pairs. Given an RSVT $T$, we send it to $(L,E)$. The set $L$ (resp. $E$) consists of all $(r,c)$ such that $r$ is a leading (resp. extra) number in column $c$ of $T$. This map is injective. If we know $(L,E)$ is the image of some RSVT $T$, we can uniquely recover $T$: First, for each $c$, build a column that consists of $r$ such that $(c,r) \in L$. The column should be decreasing from top to bottom. Then for each $(c,r) \in E$, put $r$ in the lowest cell whose largest number is larger than $r$. This will be column $c$ of $T$. Now we may view each RSVT as a diagram pair. We write $T = (L,E)$ to denote this correspondence. It is clear that this correspondence preserves $\text{wt}(\cdot)$ and $\text{ex}(\cdot)$.

**Example 7.** Let $T$ be the RSVT in the previous example. It corresponds to the diagram pair $(\{(1,3), (1,6), (2,1), (2,3), (3,2)\}, \{(1,1), (1,4), (2,2)\})$, which can be presented as

```
   6 .
  5   
  4 X
  3 . 
  2 X .
  1 X .
```

We put a circled number on the left of the diagram to indicate the row number. Viewing $T$ as a diagram pair, we have $\text{wt}(T) = (2,2,1,0,1)$ and $\text{ex}(T) = 3$, which agrees with Example 5.

We may also view $\text{RSSYT}(\alpha)$ as a subset of $\text{RSVT}(\alpha)$. Thus, $\text{RSSYT}(\alpha)$ is the set of diagram pairs $(L,\emptyset) \in \text{RSVT}(\alpha)$. With this convention, we have the following observation.

**Remark 8.** If the diagram pair $(L,E)$ is in $\text{RSVT}(\alpha)$ then $(L,\emptyset) \in \text{RSSYT}(\alpha)$.

### 2.3 $\text{KD}(\alpha)$ and $\text{KKD}(\alpha)$

We give another combinatorial definition of key polynomials due to Kohnert [11]. A diagram pair is called a key diagram pair if its Kohnert cells are left-justified and has no ghost cells. Given a weak composition $\alpha$, we let $D_\alpha$ the key diagram pair associated to $\alpha$: On its row $i$, there are $\alpha_i$ left-justified Kohnert cells and no ghost cells.

Next, we define a Kohnert move on a diagram pair with no ghost cells: Select the rightmost box in any row and move it downward to the first position available, possibly jumping over other cells as needed. Let $\text{KD}(\alpha)$ be the closure of $\{D_\alpha\}$ under all possible Kohnert moves.

**Theorem 9.** [11] The key polynomial indexed by $\alpha$, is given by

$$
\kappa_\alpha = \sum_{D \in \text{KD}(\alpha)} x^{\text{wt}(D)}
$$

---

The Electronic Journal of Combinatorics 30(4) (2023), #P4.26
Remark 10. There is a natural identification between $\text{KD}(\alpha)$ and $\text{RSSYT}(\alpha)$ which yields $\text{KD}(\alpha) = \text{RSSYT}(\alpha)$. Take $T \in \text{RSSYT}(\alpha)$. By our convention in the previous subsection, $T$ is viewed as a diagram pair $(L, \emptyset)$. This result is well-known to experts. For example, it follows from work done in [3]. For completeness, we will recover this result in Section 5.

Example 11. Let $\alpha = (0, 2, 1)$, then

$$\text{key}(\alpha) = \begin{array}{c} 3 \\ 2 \\ \end{array}, \quad \text{D}(\alpha) = \begin{array}{c} 3 \\ 2 \\ \end{array}, \quad \text{thus we obtain:}$$

$$\text{KD}(\alpha) = \left\{ \begin{array}{c} 3 \\ 2 \\ \end{array}, \begin{array}{c} 3 \\ 2 \\ 1 \\ \end{array}, \begin{array}{c} 3 \\ 1 \\ 2 \\ \end{array}, \begin{array}{c} 2 \\ 1 \\ 1 \\ \end{array}, \begin{array}{c} 3 \\ 1 \\ 1 \\ \end{array} \right\}, \quad \text{and} \quad \text{RSSYT}(\alpha) = \left\{ \begin{array}{c} 3 \\ 2 \\ \end{array}, \begin{array}{c} 2 \\ 2 \\ 1 \\ \end{array}, \begin{array}{c} 3 \\ 1 \\ 2 \\ \end{array}, \begin{array}{c} 2 \\ 1 \\ 1 \\ \end{array}, \begin{array}{c} 3 \\ 1 \\ 1 \\ \end{array} \right\}.$$ 

If we view each RSSYT as a diagram pair, it is clear that $\text{RSSYT}(\alpha) = \text{KD}(\alpha)$.

Ross and Yong [21, Section 1.2] generalized Kohnert moves. We state their construction below.

A $K$-Kohnert move is an operation on a diagram pair. It selects the rightmost cell in a row. The selected cell cannot be a ghost cell. Then move this cell downward to the first position available. It can jump over other Kohnert cells, but cannot jump over any ghost cells. After the move, it may or may not leave a ghost cell at the original position. When a $K$-Kohnert move leaves a ghost cell, we also refer to it as a ghost move.

Let $KKD(\alpha)$ be the closure of $\{D_\alpha\}$ under all possible $K$-Kohnert moves. We make the following observations.

Remark 12. Let $\alpha$ be a weak composition. We have

- $\text{KD}(\alpha) \subseteq KKD(\alpha)$.
- If $(K, G) \in KKD(\alpha)$, then $(K, \emptyset) \in \text{KD}(\alpha)$.

Remark 13. Usually, an element of $\text{KD}(\alpha)$ is viewed as a diagram. We defined $\text{KD}(\alpha)$ as a set of diagram pairs so we can work with $\text{KD}(\alpha)$ and $KKD(\alpha)$ using the same technique. In particular, with our convention, $\text{KD}(\alpha)$ is viewed as a subset of $KKD(\alpha)$.


Conjecture 14. [21] The Lascoux polynomial indexed by $\alpha$ is given by

$$L_{\alpha}^{(\beta)} = \sum_{D \in KKD(\alpha)} \beta_{\text{ex}}(D) x^{\text{wt}(D)}.$$
We prove this conjecture by establishing bijections between $\text{KKD}(\alpha)$ and $\text{RSVT}(\alpha)$ that preserve $\text{wt}(\cdot)$ and $\text{ex}(\cdot)$. Moreover, when restricted to $\text{KD}(\alpha) \subseteq \text{KKD}(\alpha)$ and $\text{RSSYT}(\alpha) \subseteq \text{RSVT}(\alpha)$, our bijections restrict to the identity maps. We will describe our bijections in the next two subsections.

**Example 15.** Continuing Example 11 for $\alpha = (0, 2, 1)$, we get

$$\text{KKD}(\alpha) = \text{KD}(\alpha) \cup \left\{ \begin{array}{c} \begin{array}{c} 3 \end{array} \end{array} \right\}, \text{ and }$$

$$\text{RSVT}(\alpha) = \text{RSSYT}(\alpha) \cup \left\{ \begin{array}{c} 3 \ 2 \ 21, \ 2 \ 21, \ 32 \ 1, \ 3 \ 1, \ 3 \ 21 \end{array} \right\}. \right.$$

Viewing the 6 elements in $\text{RSVT}(\alpha)$ with at least one extra number as diagram pairs, we obtain the following. Note they are different from the elements in $\text{KKD}(\alpha)$ with at least one ghost cell.

Thus, Conjecture 1 is correct when $\alpha = (0, 2, 1)$ since

$$\sum_{D \in \text{KKD}(\alpha)} \beta_{\text{ex}(D)} x^{\text{wt}(D)} = \sum_{T \in \text{RSVT}(\alpha)} \beta_{\text{ex}(T)} x^{\text{wt}(T)}.$$

### 2.4 Kohnert Tableaux

Assaf and Searles introduced Kohnert tableaux in [3]. We will use Kohnert tableaux to prove the correctness of our bijections.

**Definition 16.** [3, Definition 2.3] Let $\alpha$ be a weak composition. A **Kohnert tableau with content** $\alpha$ is a key diagram filled by numbers such that:

1. Column $c$ of the tableau consists of numbers $\{ i : \alpha_i \geq c \}$, with each number appearing exactly once.

2. If a number $i$ appears in row $r$, then $i \geq r$.

3. If a number $i$ appears in $(c, r)$ and $(c + 1, r')$, then $r \geq r'$.

4. Let $i, j$ appear in column $c$ with $j > i$ and $j$ is lower than $i$. Then there is an $i$ in column $c + 1$ that is strictly above the $j$ in column $c$.

Let $\text{KT}(\alpha)$ be the set of all Kohnert tableaux with content $\alpha$. 

---

*The Electronic Journal of Combinatorics 30(4) (2023), #P4.26*
Assaf and Searles constructed bijections between $\text{KT}(\alpha)$ and $\text{KD}(\alpha)$ in [3]. For each $T \in \text{KT}(\alpha)$, we may ignore its numbers and view each cell as a Kohnert cell. By [3, Lemma 2.4], the resulting diagram pair is in $\text{KD}(\alpha)$.

The inverse of the map above is called *Kohnert Labeling with respect to* $\alpha$, denoted as $\text{Label}_\alpha(\cdot)$. We may describe it as the following algorithm on certain diagram pairs.

Let $D$ be an arbitrary diagram pair such that column $c$ of $D$ has $|\{i : \alpha_i \geq c\}|$ Kohnert cells and no ghost cells. Initialize sets $S_1, S_2, \ldots$ as $S_c = \{i : \alpha_i \geq c\}$. Iterate through boxes of $D$ from right to left, and from bottom to top within each column. For the box $(c,r)$, find the smallest $i \in S_c$ such that $i$ does not appear at $(c+1,r')$ for all $r' > r$. We remove $i$ from $S_c$ and fill $i$ in $(c,r)$ of $D$. If no such $i$ exists or $i < r$, terminate the algorithm. After all boxes are filled, output the final tableau.

By [3, Lemma 2.6, Lemma 2.7, Theorem 2.8], the labeling algorithm on $D$ produces an output if and only if $D \in \text{KD}(\alpha)$. Moreover, if we restrict the algorithm on $\text{KD}(\alpha)$, then this is a bijection from $\text{KD}(\alpha)$ to $\text{KT}(\alpha)$ whose inverse is described above.

**Example 17.** Let $\alpha = (0,2,1)$, we have

$$\text{KT}(\alpha) = \begin{cases} \begin{array}{ccc} 3 & 3 & 2 \\ 2 & 2 & 2 \\ 1 & & \end{array}, & \begin{array}{ccc} 2 & 2 & 3 \\ 2 & 2 & 2 \\ 3 & & \end{array}, & \begin{array}{ccc} 3 & & \\ 3 & 2 & 2 \\ 2 & 2 & \end{array} \end{cases}$$

where the relative order in the sets corresponds to $\text{KD}(\alpha)$ from Example 11 under the above labeling algorithm.

### 3 Describing the maps

For each composition $\alpha$, we have introduced two sets of diagram pairs: $\text{KKD}(\alpha)$ and $\text{RSVT}(\alpha)$. We will define two maps: $\Psi_\alpha$ on $\text{KKD}(\alpha)$ and $\Phi_\alpha$ on $\text{RSVT}(\alpha)$. In Section 7, we will show the image of $\Psi_\alpha$ (resp. $\Phi_\alpha$) lies in $\text{RSVT}(\alpha)$ (resp. $\text{KKD}(\alpha)$).

### 3.1 An informal description of $\Psi_\alpha$

In this section, we describe a map $\Psi_\alpha$ from $\text{KKD}(\alpha)$ to the set of all diagram pairs. First, we describe an operator on $\text{KD}(\alpha)$. Let $G$ be a diagram. Then $\sharp_G(\cdot)$ acts on $\text{KD}(\alpha)$ in the following way: Take $D \in \text{KD}(\alpha)$. Iterate through cells of $G$ from right to left. Within each column, go from bottom to top. For $(c,r) \in G$, search for the largest $r' \leq r$ such that $(c,r')$ satisfies both of the following:

- The cell $(c,r')$ is a Kohnert cell in $D$, and
- If we raise the cell $(c,r')$ to $(c,r)$, the resulting diagram is still in $\text{KD}(\alpha)$.

After finding such $r'$, we move cell $(c,r')$ to $(c,r)$. After iterating over all cells in $G$, we denote the final Kohnert diagram by $\sharp_G(D)$. If we cannot find such an $r'$ during an iteration, then $\sharp_G(D)$ is undefined.
Example 18. Let $D$ be the fourth Kohnert diagram in Example 11. Let $G$ be the diagram $\{(1, 3), (2, 2)\}$. We may compute $\sharp_G(D)$ as follows. We label $(c, r)$ and $(c, r')$ involved in each step above and below the arrows.

Now, we may describe the map $\Psi_\alpha$. Take $D = (K, G) \in \text{KKD}(\alpha)$. Compute $(L, \emptyset) = \sharp_G((K, \emptyset)) \in \text{KD}(\alpha)$. Then $\Psi_\alpha(D) := (L, (K \cup G) - L)$.

Example 19. Let $D = (K, G)$ be the last diagram pair in Example 15. The previous example shows $\sharp_G(K) = (\{(1, 2), (1, 3), (2, 2)\}, \emptyset)$. Thus, $D$ is sent to the diagram pair $((1, 2), (1, 3), (2, 2)), \{(1, 1), (2, 1)\})$. Notice that this diagram pair corresponds to the following RSVT:

We say this is an informal description of $\Psi_\alpha$ since the map is not obviously well-defined. It seems possible that $\sharp_G((K, \emptyset))$ is undefined. In Section 7, we will provide an alternative description of the map $\Psi_\alpha$ and check the following.

Lemma 20. The map $\Psi_\alpha$ is a well-defined map from $\text{KKD}(\alpha)$ to $\text{RSVT}(\alpha)$.

3.2 An informal description of $\Phi_\alpha$

The map $\Phi_\alpha$ can be described similarly on $\text{RSVT}(\alpha)$. First, we need an analogue of the $\sharp_G(\cdot)$ operator. Let $E$ be a diagram. Then $\flat_E(\cdot)$ acts on $\text{KD}(\alpha)$ in the following way: Take $D \in \text{KD}(\alpha)$. Iterate through cells of $E$ from left to right. Within each column, go from top to bottom. For $(c, r) \in E$, search for the smallest $r' \geq r$ such that $(c, r')$ satisfies both of the following:

- The cell $(c, r')$ is a Kohnert cell in $D$, and
- If we drop the cell $(c, r')$ to $(c, r)$, the resulting diagram is still in $\text{KD}(\alpha)$.

After finding such $r'$, we move cell $(c, r')$ to $(c, r)$. After iterating over all cells in $E$, we denote the final Kohnert diagram by $\flat_E(D)$. If we cannot find such an $r'$ during an iteration, then $\flat_E(D)$ is undefined.

Example 21. Let $D$ be the first Kohnert diagram in Example 11. Let $E$ be the diagram $\{(1, 1), (2, 1)\}$. We may compute $\flat_E(D)$ as follows. We label $(c, r)$ and $(c, r')$ involved in each step above and below the arrows.
Now, we may describe the map $\Phi_\alpha$. Take $T \in \text{RSVT}(\alpha)$ and we may write $T$ as a diagram pair $(L, E)$. Compute $(K, \emptyset) = b_E((L, \emptyset))$. Then $\Phi_\alpha(T) := (K, (L \sqcup E) - K)$.

**Example 22.** Let $T$ be the last RSVT in Example 15. We may write $T = (L, E)$ as $(\{(1,2), (1,3), (2,2)\}, \{(1,1), (2,1)\})$. The previous example computes $(K, \emptyset) = b_E((L, \emptyset))$. Thus, $(L \sqcup E) - K = \{(1,3), (2,2)\}$ and $T$ is sent to

\[
\begin{array}{|c|}
\hline
\text{3} & \text{X} \\
\text{2} & \cdot & \text{X} \\
\text{1} & \cdot & \cdot \\
\hline
\end{array}
\]

Notice that this is an element of $\text{KKD}(\alpha)$.

Again, $\Phi_\alpha$ is not obviously well-defined. In Section 7, we will provide an alternative description of the map $\Phi_\alpha$ and check the following.

**Lemma 23.** The map $\Phi_\alpha$ is a well-defined map from $\text{RSVT}(\alpha)$ to $\text{KKD}(\alpha)$.

Now we restate our main result. The proof is also in Section 7.

**Theorem 24.** The maps $\Psi_\alpha : \text{KKD}(\alpha) \to \text{RSVT}(\alpha)$ and $\Phi_\alpha : \text{RSVT}(\alpha) \to \text{KKD}(\alpha)$ are mutually inverses of each other. Moreover, they preserve $\text{wt}(\cdot)$ and $\text{ex}(\cdot)$.

### 4 Bruhat Order on Weak Compositions

Partial order on weak compositions has been studied in [3, 2, 8, 9]. In this section, we give a definition via the key tableau associated to the weak composition. In subsection 4.1, we also study $m(\alpha, S)$ (resp. $M(\alpha, S)$) which is the unique minimum (resp. maximum) weak composition in a certain set of weak compositions. In subsection 4.2, we use properties of $M(\alpha, S)$ to show that our Bruhat order is equivalent to the left swap order defined in [3, 2], which implies that the Bruhat order is equivalent to the inclusion order on $\text{KD}(\alpha), \text{KKD}(\alpha), \text{RSSYT}(\alpha)$ and $\text{RSVT}(\alpha)$.

#### 4.1 Bruhat order

We may define a partial order on all weak compositions.

**Definition 25.** Let $\alpha, \gamma$ be two weak compositions. We define $\alpha \leq \gamma$ if $\text{key}(\alpha)$ and $\text{key}(\gamma)$ have the same shape and $\text{key}(\alpha) \leq \text{key}(\gamma)$ entry-wise. This order is called the **Bruhat order**.

**Remark 26.** The Bruhat order is originally defined on permutations of $[n]$. We may view such permutations as weak compositions via their one-line notations. From this perspective, if $u, v$ are two permutations of $[n]$, then $\text{key}(u)$ and $\text{key}(v)$ both have shape $(n, n-1, \cdots, 1)$. The following is a well-known result (see for instance [5, Theorem 2.6.3],[17, Theorem 2.1.11] and [23, Lemma 3.1]): the permutation $u$ is less than or equal to $v$ in
the classical Bruhat order if and only if $\text{key}(u) \leq \text{key}(v)$ entry-wise. In other words, the Bruhat order on weak compositions is a generalization of the classical Bruhat order on permutations.

**Definition 27.** For $S \subseteq [n]$, let $1_S$ be the weak composition whose $i^{th}$ entry is 1 if $i \in S$ and 0 otherwise. For a weak composition $\alpha$, the **support** of $\alpha$ is the set $\{i | \alpha_i > 0\}$, denoted as $\text{supp}(\alpha)$.

Weak compositions with only 0s and 1s are in natural bijection with subsets of $[n]$. The bijections are $S \mapsto 1_S$ and $\alpha \mapsto \text{supp}(\alpha)$. With the Bruhat order above, we may define a partial order on subsets of $[n]$.

**Definition 28.** For $S, S' \subseteq [n]$, we say $S \leq S'$ if $1_S \leq 1_{S'}$.

Throughout the paper, the binary operator $\leq$ on sets refers to their Bruhat order of their indicator function.

We have other alternative descriptions of this order.

**Lemma 29.** Take $S, S' \subseteq [n]$. The following are equivalent:

1. $S \leq S'$
2. $|S| = |S'|$ and for each $j \in [|S|]$, the $j^{th}$ largest number in $S$ is at most the $j^{th}$ largest number in $S'$.
3. $|S| = |S'|$ and for each $s \in S$, $|[s, n] \cap S| \leq |[s, n] \cap S'|$.

**Proof.** We first prove statement 1 and 2 are equivalent. By definition $S \leq S'$ if and only if $\text{key}(1_S) \leq \text{key}(1_{S'})$. The number on row $j$ column 1 of $\text{key}(1_S)$ (resp. $\text{key}(1_{S'})$) is the $j^{th}$ largest number in $S$ (resp. $S'$). Thus, $\text{key}(1_S) \leq \text{key}(1_{S'})$ is equivalent to statement 2.

Next, we show the statements 2 and 3 are equivalent. Assume the statement 2 is true. Let $s \in S$ be the $j^{th}$ largest number in $S$. Then $|[s, n] \cap S| = j$. Since the $j^{th}$ largest number in $S'$ is at least $s$, $|[s, n] \cap S'| \geq j$. Now assume statement 3 is true. Let $s$ be the $j^{th}$ largest number in $S$. We know there are at least $j$ numbers in $[s, n] \cap S'$, so the $j^{th}$ largest number in $S'$ is at least $s$.

Take $S \subseteq [n]$ and a weak composition $\alpha$. Consider the set $\{\gamma : \gamma \geq \alpha, \text{supp}(\gamma) \subseteq S\}$. Let $m(\alpha, S)$ be the unique minimum element in the set, if it exists. Later, we will show $m(\alpha, S)$ exists as long as the set is non-empty. First, we introduce an algorithm to compute $m(\alpha, S)$ or assert it does not exist. Initialize $\text{list}$ to be an empty list and initialize $\sigma$ to be the weak composition with all 0s. Iterate over $i = 1, \ldots, n$. Perform the following two processes in each iteration:

- **(Adding process):** If $\alpha_i > 0$, then add $\alpha_i$ to $\text{list}$.
- **(Removing process):** If $i \in S$ and $\text{list}$ is non-empty, then remove $\max(\text{list})$ from $\text{list}$ and assign it to $\sigma_i$. 

The Electronic Journal of Combinatorics 30(4) (2023), #P4.26 11
After all iterations, if \( \text{list} \) is empty, then \( m(\alpha, S) = \sigma \). Otherwise, such \( m(\alpha, S) \) does not exist.

**Example 30.** Let \( n = 7, \alpha = (1, 3, 0, 2, 0, 0, 2) \) and \( S = \{3, 4, 5, 6, 7\} \). Then we trace \( \sigma \) and \( \text{list} \) during the algorithm:

- Before iteration 1: \( \sigma = (0, 0, 0, 0, 0, 0, 0) \); \( \text{list} \) is empty.
- After iteration 1: \( \sigma = (0, 0, 0, 0, 0, 0, 0) \); \( \text{list} \) contains 1.
- After iteration 2: \( \sigma = (0, 0, 0, 0, 0, 0, 0) \); \( \text{list} \) contains 1, 3.
- After iteration 3: \( \sigma = (0, 0, 3, 0, 0, 0, 0) \); \( \text{list} \) contains 1.
- After iteration 4: \( \sigma = (0, 0, 3, 2, 0, 0, 0) \); \( \text{list} \) contains 1.
- After iteration 5: \( \sigma = (0, 0, 3, 2, 1, 0, 0) \); \( \text{list} \) is empty.
- After iteration 6: \( \sigma = (0, 0, 3, 2, 1, 0, 0) \); \( \text{list} \) is empty.
- After iteration 7: \( \sigma = (0, 0, 3, 2, 1, 0, 2) \); \( \text{list} \) is empty.

Since the list is empty after the iterations, the algorithm outputs

\[
m(\alpha, S) = (0, 0, 3, 2, 1, 0, 2).
\]

We take steps to show this algorithm is correct. We start with the following observation, which connects this algorithm with the \(< \) operator in Definition 4.

**Lemma 31.** Assume the algorithm outputs \( m(\alpha, S) = \sigma \). Let \( T_c \) (resp. \( A_c \)) be the set consisting of numbers in column \( c \) of \( \text{key}(\sigma) \) (resp. \( \text{key}(\alpha) \)). Then \( T_c = S < A_c \).

*Proof.* We know \( i \in A_c \) if and only if during the adding process of the iteration \( i \), a number at least \( c \) is added to \( \text{list} \). Similarly, \( i \in T_c \) if and only if during the removing process of the iteration \( i \), a number at least \( c \) is removed from \( \text{list} \).

Assume \( T_c = \{t_1 < t_2 < \cdots < t_s\} \) and \( A_c = \{a_1 < a_2 < \cdots < a_s\} \). During the adding process of iteration \( a_1 \), the algorithm puts \( a_{a_1} \) into \( \text{list} \). This is the first time that \( \text{list} \) gains a number at least \( c \). Thus, \( t_1 \geq a_1 \). Moreover, assume there exists \( t \in S \) such that \( a_1 \leq t < t_1 \). During the removing process of iteration \( t \), \( \text{list} \) has a number at least \( c \). The algorithm will remove a number at least \( c \) from \( \text{list} \), contradicting to \( t \notin T_c \). Thus, \( t_1 \) is the smallest in \( S \) with \( t_1 \geq a_1 \).

Now consider \( t_j \) with \( j > 1 \). During the removing process of \( t_j \), we remove a number at least \( c \) for the \( j^{th} \) time. Thus, we have added at least \( j \) such numbers to \( \text{list} \), so \( a_j \leq t_j \). Now assume there is \( t < t_j \) such that \( t \in S, t \geq a_j \), and \( t > t_{j-1} \). During the removing process of iteration \( t \), there is a number at least \( c \) in \( \text{list} \), so such a number will be removed. We have a contradiction since \( t \notin T_c \). Thus, \( t_j \) is the smallest in \( S \) such that \( t_j \geq a_j \) and \( t_j > t_{j-1} \). \( \square \)
Next, we investigate the condition for the algorithm to assert $m(\alpha, S)$ does not exist.

**Lemma 32.** The following are equivalent:

1. $|S| \geq |\text{supp}(\alpha)|$. In addition, if we let $S' \subseteq S$ consists of the largest numbers in $S$ with $|S'| = |\text{supp}(\alpha)|$, then $S' \supseteq \text{supp}(\alpha)$.

2. The algorithm asserts $m(\alpha, S)$ exists. (i.e. The list is empty when the algorithm ends.)

3. There is a weak composition $\gamma$ such that $\text{supp}(\gamma) \subseteq S$ and $\gamma \geq \alpha$.

**Proof.** First, assume $|S| < |\text{supp}(\alpha)|$. We check the last two statements do not hold.

2. Consider the $i^{th}$ iteration. First, the size of list is increased by one if $i \in \text{supp}(\alpha)$. Next, the size of list is fixed or decreased by one if $i \in S$. Throughout this algorithm, list gains $|\text{supp}(\alpha)|$ numbers and loses at most $|S|$ numbers. It is not empty when the algorithm ends.

3. Assume such $\gamma$ exists. We know $|\text{supp}(\gamma)| = |\text{supp}(\alpha)| > |S|$, contradicting to $\text{supp}(\gamma) \subseteq S$.

Now assume $|S| \geq |\text{supp}(\alpha)|$ and define $S'$ as above. Suppose $S' \not\supseteq \text{supp}(\alpha)$. We check the last two statements in the lemma do not hold.

2. By Lemma 29, there exists $j \in \text{supp}(\alpha)$ such that

$$|[j, n] \cap \text{supp}(\alpha)| > |[j, n] \cap S'| = |[j, n] \cap S|.$$  

Between the $j^{th}$ iteration and the last iteration inclusively, list gains $|[j, n] \cap \text{supp}(\alpha)|$ numbers and loses at most $|[j, n] \cap S|$ numbers. It is not empty when the algorithm ends.

3. Assume such $\gamma$ exists. By $\text{supp}(\gamma) \subseteq S$, we have $S' \supseteq \text{supp}(\gamma) \supseteq \text{supp}(\alpha)$. Contradiction.

Finally, assume $S' \supseteq \text{supp}(\alpha)$. We check the last two statements in the lemma are true.

2. Assume when the algorithm ends, list is not empty. Find the largest $j$ such that list is not empty since the $j^{th}$ iteration. First, we know list is empty right before the $j^{th}$ iteration. Second, we know a number is added to list during the $j^{th}$ iteration, so $j \in \text{supp}(\alpha)$. By Lemma 29, $|[j, n] \cap \text{supp}(\alpha)| \leq |[j, n] \cap S'| = |[j, n] \cap S|$. Between the $j^{th}$ iteration and the last iteration inclusively, list gains $|[j, n] \cap \text{supp}(\alpha)|$ numbers. Since list is not empty since iteration $j$, list loses $|[j, n] \cap S|$ numbers. Thus, list is empty after the last iteration.
3. From the previous statement, we know the algorithm will produce a weak composition \( \sigma \). Just need to check \( \text{supp}(\sigma) \subseteq S \) and \( \sigma \geq \alpha \). It is apparent that \( \text{supp}(\sigma) \subseteq S \). Since all positive numbers in \( \alpha \) are assigned into \( \sigma \), \( \text{key}(\sigma) \) and \( \text{key}(\alpha) \) are of the same shape. Next we show that \( \sigma \geq \alpha \) by comparing \( \text{key}(\sigma) \) and \( \text{key}(\alpha) \) column-by-column: Let \( T_c \) (resp. \( A_c \)) be the set consisting of numbers in column \( c \) of \( \text{key}(\sigma) \) (resp. \( \text{key}(\alpha) \)). By Lemma 31, \( T_c = S \triangleleft A_c \). Thus, \( T_c \geq A_c \). 

Now we can prove the correctness of our algorithm.

**Lemma 33.** The algorithm correctly computes \( m(\alpha, S) \). In other words, consider the set \( \{ \gamma : \gamma \geq \alpha, \text{supp}(\gamma) \subseteq S \} \).

- If list is empty after the iterations, then the output \( \sigma \) is the unique minimum in the set.

- Otherwise, the set is empty.

Moreover, the second case happens only when the set is empty.

**Proof.** If list is not empty when the algorithm ends, \( \{ \gamma : \gamma \geq \alpha, \text{supp}(\gamma) \subseteq S \} = \emptyset \) by the previous lemma.

Now assume list is empty when the algorithm ends. Then the set is non-empty: By the proof of the previous lemma, the output \( \sigma \) is in this set. We check \( \sigma \) is the least element. Assume there is a \( \gamma \) in the set with \( \gamma \not\geq \sigma \). Let \( T'_c \) consists of numbers in column \( c \) of \( \text{key}(\gamma) \). Define \( T_c \) and \( A_c \) similarly for \( \text{key}(\sigma) \) and \( \text{key}(\alpha) \) respectively. Then we can find \( c \) such that \( T'_c \not\subseteq T_c \). Let \( t'_j \) be the \( j^{th} \) smallest number in \( T'_c \). Define \( t_j \) and \( a_j \) similarly for \( T_c \) and \( A_c \). Then we can find smallest \( j \) such that \( t'_j < t_j \). Notice that \( t'_j \geq a'_j \) and \( t'_j \in S \). Moreover, \( t'_j > t'_{j-1} \geq t_{j-1} \) if \( j > 1 \). We have a contradiction to the fact \( T_c = S \triangleleft A_c \) from Lemma 31.

**Corollary 34.** Let \( \alpha \) be a weak composition and \( S \subseteq [n] \). \( m(\alpha, S) \) exists if and only if

- \( |S| \geq |\text{supp}(\alpha)| \), and

- \( S' \geq \text{supp}(\alpha) \), where \( S' \) consist of the largest numbers in \( S \) with \( |S'| = |\text{supp}(\alpha)| \).

**Proof.** Follows from the previous two lemmas.

Analogously, we may also look at the set \( \{ \gamma : \gamma \leq \alpha, \text{supp}(\gamma) \subseteq S \} \). Similarly, if it is non-empty, it will contain a unique maximum element. Let \( M(\alpha, S) \) be this element. To compute it, we only need to slightly change our algorithm above: Let \( i \) go from \( n \) to 1, instead of 1 to \( n \). Similar to Corollary 34, we have the following for \( M(\alpha, S) \).

**Corollary 35.** Let \( \alpha \) be a weak composition and \( S \subseteq [n] \). \( M(\alpha, S) \) exists if and only if

- \( |S| \geq |\text{supp}(\alpha)| \), and

- \( S' \leq \text{supp}(\alpha) \), where \( S' \) consist of the smallest numbers in \( S \) with \( |S'| = |\text{supp}(\alpha)| \).
Proof. The proof is similar to the proof of Corollary 34.

We end this subsection by a property that connects \( m(\gamma, S) \) and \( M(\alpha, S) \). Let \( \gamma \) be a weak composition. We use \( \gamma \) to denote the weak composition obtained by decreasing each positive entry of \( \gamma \) by 1.

**Lemma 36.** Let \( \alpha, \gamma \) be two weak compositions. Take \( S \subseteq [n] \) with \( |S| = |\text{supp}(\alpha)| \). Then the following are equivalent:

- \( m(\gamma, S) \) exists and \( \alpha \geq 1_S + m(\gamma, S) \).
- \( M(\alpha, S) \) exists and \( M(\alpha, S) \geq \gamma \).

**Proof.** Assume the first statement is true. Notice \( \text{supp}(1_S + m(\gamma, S)) = S \), so \( M(\alpha, S) \) exists and

\[ M(\alpha, S) \geq 1_S + m(\gamma, S). \]

Decrease each positive entry by 1 on both sides and get

\[ \overline{M(\alpha, S)} \geq m(\gamma, S). \]

Then we get the second statement since \( m(\gamma, S) \geq \gamma \).

Now assume the second statement is true. Notice \( \text{supp}(\overline{M(\alpha, S)}) \subseteq S \), so \( m(\gamma, S) \) exists and

\[ \overline{M(\alpha, S)} \geq m(\gamma, S). \]

By \( |S| = |\text{supp}(\alpha)| \), \( \text{supp}(M(\alpha, S)) = S \), so

\[ M(\alpha, S) \geq 1_S + m(\gamma, S). \]

Then we get the first statement since \( \alpha \geq M(\alpha, S) \).

\( \square \)

### 4.2 Left swap order

Assaf and Searles [3] also defined a partial order on weak compositions called the *left swap order*. In this subsection, we introduce this order and show that it is equivalent to the Bruhat order.

**Definition 37.** [3, 2, Definition 2.3.4] A *left swap* on a weak composition \( \alpha \) exchanges two parts \( \alpha_i < \alpha_j \) with \( i < j \). The left swap order on weak compositions is the transitive closure of the relation \( \gamma \preceq \alpha \) whenever \( \gamma \) is a left swap of \( \alpha \).

When \( \gamma \) is obtained from \( \alpha \) by exchanging the \( i^{th} \) and \( j^{th} \) parts of \( \alpha \), we write \( \gamma = (ij)\alpha \).

**Proposition 38.** [2, Prop. 2.3.9] Given weak compositions \( \alpha, \gamma \), we have \( \gamma \preceq \alpha \) if and only if the key diagram pair of \( \gamma \) is in \( \text{KD}(\alpha) \).

To show the equivalence between the left swap order and the Bruhat order, we need a few lemmas. We start with the following, which summarizes how \( M(\alpha, S) \) is changed when we changes \( S \) in a nice way.
Lemma 39. Let \( \alpha \) be a weak composition and \( S \subseteq [n] \) with \( S \subseteq \text{supp}(\alpha) \). Take \( g \notin S \) such that \( |[g, n] \cap \text{supp}(\alpha)| > |[g, n] \cap S| \). Then there exists \( s \in S \) such that \( s < g \) and

\[
M(\alpha, S') = (s \ g)M(\alpha, S),
\]

where \( S' = (S \cup \{g\}) - \{s\} \). In particular, \( M(\alpha, S) \) is a left swap of \( M(\alpha, S') \).

Proof. Run the algorithm that computes \( M(\alpha, S) \). After initialization, the algorithm iterates from \( i = n \) to \( i = 1 \). Right after the iteration with \( i = g \), the algorithm has put \( |[g, n] \cap \text{supp}(\alpha)| \) numbers to \( \text{list} \) and has removed at most \( |[g, n] \cap S| \) numbers from \( \text{list} \). Since \( |[g, n] \cap \text{supp}(\alpha)| > |[g, n] \cap S| \), the list is non-empty. Let \( x \) be the largest number in the current list. Then this \( x \) will be picked sometime in the future. Let \( s \in S \cap [1, g) \) be the largest such that in the iteration \( i = s \), the algorithm assigns \( x \) to \( \sigma_s \). We may run the algorithm to compute \( M(\alpha, (S \cup \{g\}) - \{s\}) \). It behaves the same as on \( M(\alpha, S) \), except it assigns \( x \) to \( \sigma_g \) and keeps \( \sigma_s = 0 \). Thus, \( M(\alpha, (S \cup \{g\}) - \{s\}) = (g \ s)M(\alpha, S) \). \( \square \)

We know from definition that \( M(\alpha, S) \leq \alpha \). The next lemma will describe their relationship in the left swap order.

Lemma 40. Let \( \alpha \) be a weak composition. Let \( S \) be a set such that \( S \subseteq \text{supp}(\alpha) \). Then \( M(\alpha, S) \preceq \alpha \).

Proof. Find the smallest \( g \geq 1 \) such that \( [g, n] \cap S = [g, n] \cap \text{supp}(\alpha) \). Prove this lemma by induction on \( g \). For the base case, we assume \( g = 1 \). Then \( \text{supp}(\alpha) = S \) which implies \( M(\alpha, S) = \alpha \).

Next, assume \( g > 1 \). Since \( [g - 1, n] \cap S \neq [g - 1, n] \cap \text{supp}(\alpha) \) and \( S \subseteq \text{supp}(\alpha) \), we know \( g-1 \in \text{supp}(\alpha) - S \). Thus, \( |[g-1, n] \cap S| < |[g-1, n] \cap \text{supp}(\alpha)| \). By Lemma 39, there exists \( s \in S \) such that if we let \( S' = (S - \{s\}) \cup \{g - 1\} \), we have \( M(\alpha, S) \leq M(\alpha, S') \). Notice that \( [g - 1, n] \cap S' = [g - 1, n] \cap \text{supp}(\alpha) \). We may apply our inductive hypothesis and get \( M(\alpha, S) \leq M(\alpha, S') \leq \alpha \). \( \square \)

Finally, we need the following intuitive lemma, which says both partial orders are preserved by the operator \( \alpha \mapsto \overline{\alpha} \).

Lemma 41. Given weak compositions \( \alpha \) and \( \gamma \) with \( \text{supp}(\alpha) = \text{supp}(\gamma) \). Then we have

- \( \gamma \preceq \alpha \) if and only if \( \overline{\gamma} \preceq \overline{\alpha} \).
- \( \gamma \preceq \alpha \) if and only if \( \overline{\gamma} \preceq \overline{\alpha} \).

Proof. Immediate from definitions. \( \square \)

Now we are ready to prove the equivalence of these two partial orders.

Proposition 42. Given weak compositions \( \alpha \) and \( \gamma \), we have \( \gamma \preceq \alpha \) if and only if \( \gamma \leq \alpha \).
Proof. First we show if $\gamma \preceq \alpha$, then $\gamma \leq \alpha$. It suffices to show when $\gamma$ is a left swap of $\alpha$. Say $\gamma = (i \, j)\alpha$ where $i < j$ and $\alpha_i < \alpha_j$. Let $T_c$ (resp. $T'_c$) consists of numbers in column $c$ of $\text{key}(\alpha)$ (resp. $\text{key}(\gamma)$). When $c \leq \alpha_i$ or $c > \alpha_j$, we have $T_c = T'_c$. When $\alpha_i < c \leq \alpha_j$, $T'_c$ is obtained from $T_c$ by replacing $j$ with $i$. Therefore $\text{key}(\gamma) \leq \text{key}(\alpha)$ and $\gamma \leq \alpha$.

Next we assume $\gamma \preceq \alpha$ and show $\gamma \preceq \alpha$. We prove by induction on $\max(\alpha)$. If $\max(\alpha) = 0$, then $\alpha, \gamma$ only contain 0s. Our claim is immediate. Now assume $\max(\alpha) > 0$. We consider two cases.

- If $\text{supp}(\gamma) = \text{supp}(\alpha)$, then $\gamma \preceq \alpha$. By our inductive hypothesis, $\gamma \preceq \alpha$. By Lemma 41 again, $\gamma \preceq \alpha$.

- Assume $\text{supp}(\gamma) \neq \text{supp}(\alpha)$. Let $S = \text{supp}(\gamma)$. First, notice that $\gamma \preceq M(\alpha, S)$ and these two weak compositions have the same support. By the previous case, $\gamma \preceq M(\alpha, S)$. It remains to check $M(\alpha, S) \preceq \alpha$, which follows from $S \leq \text{supp}(\alpha)$ and Lemma 40.

Consequently, we know several statements are equivalent to $\gamma \preceq \alpha$.

Corollary 43. Given weak compositions $\alpha, \gamma$, the following are equivalent:

1. $\gamma \preceq \alpha$;
2. $\text{KD}(\gamma) \subseteq \text{KD}(\alpha)$;
3. $\text{KKD}(\gamma) \subseteq \text{KKD}(\alpha)$;
4. $\text{RSSYT}(\gamma) \subseteq \text{RSSYT}(\alpha)$;
5. $\text{RSVT}(\gamma) \subseteq \text{RSVT}(\alpha)$.

Proof. We show the following directions.

- (1) $\iff$ (2) This follows from Propositions 38 and 42.

- (1) $\iff$ (4) This is true by definition.

- (1) $\iff$ (5) This is true by definition.

- (2) $\implies$ (3) This is true by definition.

- (3) $\implies$ (2) Since $\text{KKD}(\gamma) \subseteq \text{KKD}(\alpha)$ and $\text{KD}(\gamma) \subseteq \text{KKD}(\gamma)$, we have $\text{KD}(\gamma) \subseteq \text{KKD}(\alpha)$. Since there is no ghost cells in elements of $\text{KD}(\gamma)$, we have $\text{KD}(\gamma) \subseteq \text{KD}(\alpha)$.

\[\square\]
5 Recursive descriptions of $\text{KKD}(\alpha)$ and $\text{RSVT}(\alpha)$

To check our maps $\Phi_\alpha$ and $\Psi_\alpha$ are well-defined, we need to study their domains. In this section, we give necessary and sufficient criteria on when a diagram pair is in $\text{KKD}(\alpha)$ (resp. $\text{RSVT}(\alpha)$). Our criteria will be recursive.

For a diagram pair $D = (K, G)$, we can send it to a triple $(K_1, G_1, d)$ where $K_1$, $G_1$ are disjoint subsets of $[n]$ and $d$ is a diagram pair. They are defined as follows:

- $K_1 := \{r : (1, r) \in K\}$
- $G_1 := \{r : (1, r) \in G\}$
- $d$ is the diagram pair with a Kohnert cell (resp. ghost cell) at $(c, r)$ if $D$ has a Kohnert cell (resp. ghost cell) at $(c+1, r)$ with $c \geq 1$.

The map $D \to (K_1, G_1, d)$ is invertible: given disjoint $K_1, G_1 \subseteq [n]$ and a diagram pair $d$, we can uniquely recover $D$. Thus, we may identify a diagram pair with its image and write $D = (K_1, G_1, d)$.

Fix a weak composition $\alpha$ in this section. An element of $\text{KKD}(\alpha)$ or $\text{RSVT}(\alpha)$ can be written as $(K_1, G_1, d)$. We will find conditions on this triple to determine when $(K_1, G_1, d) \in \text{KKD}(\alpha)$ and $\text{RSVT}(\alpha)$.

5.1 Describing $\text{KKD}(\alpha)$

First, we describe the condition for a diagram pair $(K_1, G_1, d)$ to live in $\text{KKD}(\alpha)$.

**Theorem 44.** The diagram pair $(K_1, G_1, d)$ is in $\text{KKD}(\alpha)$ if and only if it satisfies:

1. $K_1$ and $G_1$ are disjoint subsets of $[n]$.
2. $K_1 \subseteq \text{supp}(\alpha)$.
3. For each $g \in G_1$, $|[g, n] \cap \text{supp}(\alpha)| > |([g, n] \cap K_1)|$.
4. $d \in \text{KKD}(\overline{M}(\alpha, K_1))$

The rest of this subsection proves it. First, we want to show if a diagram pair $D$ satisfies these conditions, then $D \in \text{KKD}(\alpha)$. This can be implied from the following lemma:

**Lemma 45.** If $D = (K_1, G_1, d)$ satisfies conditions $(1)$-$(4)$ in Theorem 44, then we can find $\gamma \leq \alpha$ such that $D \in \text{KKD}(\gamma)$. Moreover, $\gamma_i = 0$ if $i \notin K_1 \sqcup G_1$.

Notice $D \in \text{KKD}(\gamma)$ by Corollary 43. Thus, this lemma implies the reverse direction of Theorem 44.

The proof involves doing *reverse K-Kohnert* moves on diagram pairs $(K, G)$. We describe it below. Given a Kohnert cell $(c, r) \in K$, we look for the smallest $r' > r$ such that...
• \((c, r') \notin K\)

• there does not exist a \(c' > c\) such that \((c', r') \in K \cup G\)

• \((c, r'') \notin G\) for all \(r < r'' < r'\)

If not such \(r'\) exist, then the reverse \(K\)-Kohnert move is undefined on \((c, r)\). If such \(r'\) exist, we turn \((c, r')\) into a Kohnert cell and remove \((c, r)\) from \(K\). When \((c, r') \notin G\), this is the reverse Kohnert move for \((c, r)\); when \((c, r') \in G\), this is the reverse ghost move for \((c, r)\).

Clearly, the reverse \(K\)-Kohnert move reverses the effect of a \(K\)-Kohnert move.

**Proof.** We describe an algorithm that turns \(D\) into a key diagram pair of some weak composition \(\gamma\) via reverse K-Kohnert moves. First, we consider \(d\). Since \(d \in KKD(M(\alpha, K_1))\), we may do reverse \(K\)-Kohnert moves on \(d\) to obtain the key diagram pair of \(M(\alpha, K_1)\).

Now, Kohnert cells in \(D\) form the key diagram pair of \(M(\alpha, K_1)\). If \(G_1\) is empty, we are done. Otherwise, let \(g = \min(G_1)\). By Lemma 39, we can find \(k \in K_1 \cap [1, g]\) such that \(M(\alpha, (K_1 \cup \{g\}) - \{k\}) = (g, k)M(\alpha, K_1)\). We perform reverse K-Kohnert moves to lift the entire row \(k\) of \(D\) to row \(g\), and remove the ghost at \((1, g)\). Next we assign \((K_1 \cup \{g\}) - \{k\}\) to \(K_1\) and assign \(G_1 - \{g\}\) to \(G_1\). Now, Kohnert cells in \(D\) still form the key diagram pair of \(M(\alpha, K_1)\). We may repeat the steps above until \(G_1\) is empty. The resulting diagram pair is a key diagram pair for some weak composition \(\gamma\). Clearly, \(\gamma_i = 0\) if \((1, i)\) was not a cell in \(D\).

**Proof of Theorem 44.** The reverse direction is already shown. For each \(D = (K_1, G_1, d) \in KKD(\alpha)\), we need to show the four conditions are satisfied. We prove by induction on \(\max(\alpha)\). When \(\max(\alpha) = 0\), the claim is immediate.

Assume the statement is true for any weak composition whose maximum entry is less than \(\max(\alpha)\). For any \(D \in KKD(\alpha)\), we want to check it satisfies the four conditions. Condition (1) is immediate. We prove the other conditions by an induction on K-Kohnert moves. If \(D\) is the key diagram pair of \(\alpha\), then the last three conditions are immediate. Now assume \((K_1, G_1, d) \in KKD(\alpha)\) satisfies the last three conditions. Perform one K-Kohnert move and obtain \((K'_1, G'_1, d')\). We want to show \((K'_1, G'_1, d')\) also satisfies the last three conditions. If the K-Kohnert move is not on column 1, then \(K_1 = K'_1\) and \(G_1 = G'_1\), which gives us the second and the third condition. Notice that \(d'\) is obtained from \(d\) by one K-Kohnert move, so the last condition is also satisfied.

Now suppose the K-Kohnert move is on column 1. \(K'_1 = (K_1 - \{i\}) \cup \{j\}\) with \(i > j\), and \(G'_1\) is either \(G_1\) or \(G_1 \cup \{i\}\). We check the last three conditions below.

2. \(\supp(\alpha) \supseteq K_1 \supseteq K'_1\).

3. We check \(|[g, n] \cap \supp(\alpha)| > |[g, n] \cap K'_1|\), for each \(g \in G_1 \cup \{i\}\).
   If \(g \in G_1\), we have
   \[|[g, n] \cap \supp(\alpha)| > |[g, n] \cap K_1| \geq |[g, n] \cap K'_1|\]
For \( g = i \), we have
\[
\|\{i, n\} \cap \text{supp}(\alpha)\| \geq \|\{i, n\} \cap K_1\| > \|\{i, n\} \cap K'_1\|.
\]

4. We have \( d' = d \in \text{KKD}(M(\alpha, K'_1)) \). By our inductive hypothesis, \( d \) satisfies all four conditions of \( \text{KKD}(M(\alpha, K'_1)) \). Using Lemma 45, we may perform reverse K-Kohnert moves on \( d \) and get the key diagram pair of \( \gamma \leq M(\alpha, K'_1) \). Since we can make a K-Kohnert move on \( (1, i) \), \( d \) has no cells on row \( i \). Thus, we know \( \gamma_i = 0 \).

To show \( d \in \text{KKD}(M(\alpha, K'_1)) \), we show \( \gamma \leq M(\alpha, K'_1) \).

Let \( T = \text{key}(M(\alpha, K'_1)) \). For each column of \( T \) that contains \( i \), we replace \( i \) by the largest \( i' \) such that \( i' < i \) and \( i' \) is not in this column. Then we sort the column into strictly decreasing order. Let \( T' \) be the resulting tableau. It is clear that \( T' \) is a key. Let \( \sigma = \text{wt}(T') \). We make two observations about \( T' \) and \( \sigma \):

- Column 1 of \( T' \) consists of \( K'_1 \), so \( \text{supp}(\sigma) = K'_1 \).
- The tableau \( T' \) is entry-wise less than \( T \). Thus, \( \sigma \leq M(\alpha, K'_1) \leq \alpha \).

By definition, these two observations yield \( \sigma \leq M(\alpha, K'_1) \).

It remains to check \( \gamma \leq \sigma \). Let \( g_1 > \cdots > g_s \) be the numbers in column \( c \) of \( \text{key}(\gamma) \).

By \( \gamma_i = 0 \), none of these numbers is \( i \). Let \( t_1 > \cdots > t_s \) (resp. \( t'_1 > \cdots > t'_s \)) be numbers in column \( c + 1 \) of \( T \) (resp. \( T' \)). By \( \gamma \leq \overline{M(\alpha, K'_1)} \), we know \( g_k \leq t_k \) for \( 1 \leq k \leq s \).

If \( i \) is not in column \( c + 1 \) of \( T \), then we are done. Otherwise, assume \( t_a = i \). We know \( t'_1 > \cdots > t'_s \). Let \( a, b \) be the set of numbers in column \( c + 1 \) of \( T \).

Thus, we know \( t_a = i \) for \( 1 \leq k \leq s \). Thus, \( \gamma \leq \sigma \leq M(\alpha, K'_1) \).

\[\square\]

5.2 Describing \( \text{RSVT}(\alpha) \)

Take \( T \in \text{RSVT}(\alpha) \) and write \( T \) as \((L_1, E_1, t)\). First, we notice the following relation between \( K_-(T) \), \( K_-(t) \) and \( L_1 \):

**Lemma 46.** Assume \( T = (L_1, E_1, t) \in \text{RSVT}(\alpha) \). Assume \( K_-(t) = \text{key}(\gamma) \). Then \( \text{wt}(K_-(T)) = L_1 + m(\gamma, L_1) \).

**Proof.** Let \( S = \text{key}(1_{L_1} + m(\gamma, L_1)) \). View \( S \) and \( T \) as tableaux. Let \( T_c \) be the set consisting of leading numbers in column \( c \) of \( T \), so \( T_1 = L_1 \). It suffices to show that \( K_-(T) = S \). We compare these two tableaux column by column. Apparently, column 1 of \( K_-(T) \) and column 1 of \( S \) both consist of \( L_1 \).

Consider the column \( c \) of \( K_-(T) \) and \( S \) for \( c > 1 \). Column \( c \) of \( S \) agrees with column \( c - 1 \) of \( \text{key}(m(\gamma, L_1)) \). Let \( \text{key}(\gamma)_{c-1} \) consists of numbers in column \( c - 1 \) of \( \text{key}(\gamma) \). By Lemma 31, column \( c - 1 \) of \( \text{key}(m(\gamma, L_1)) \) consists of \( L_1 \triangleleft \text{key}(\gamma)_{c-1} \). Since \( \text{key}(\gamma) = K_-(t) \), we have \( \text{key}(\gamma)_{c-1} = T_2 \triangleleft \cdots \triangleleft T_c \). Thus, column \( c \) of \( S \) consists of \( L_1 \triangleleft T_2 \triangleleft \cdots \triangleleft T_c = T_1 \triangleleft T_2 \triangleleft \cdots \triangleleft T_c \), which agrees with column \( c \) of \( K_-(T) \). \[\square\]
Theorem 47. The diagram pair \((L_1, E_1, t)\) is in \(RSVT(\alpha)\) if and only if it satisfies:

1. \(L_1\) and \(E_1\) are disjoint subsets of \([n]\).
2. \(L_1 \leq \text{supp}(\alpha)\).
3. Let \(L_1'\) be the set of row indices of Kohnert cells in column 1 of \(t\). For each \(e \in E_1\), \(|\{(e, n) \cap L_1\}| > |\{(e, n) \cap L_1'\}|.
4. \(t \in RSVT(\overline{M(\alpha, L_1)})\)

Proof. First, we show that if \(T = (L_1, E_1, t) \in RSVT(\alpha)\), then the four conditions are satisfied.

1. \(L_1\) and \(E_1\) are clearly disjoint.
2. Notice that column 1 of \(K_-(T)\) consists of numbers in \(L_1\), while column 1 of \(\text{key}(\alpha)\) consists of numbers in \(\text{supp}(\alpha)\). By \(K_-(T) \leq \text{key}(\alpha)\), \(L_1 \leq \text{supp}(\alpha)\).
3. View \(T\) as a tableau. For \(e \in E_1\), assume it is on row \(j\) column 1 of \(T\). Then \(|L_1 \cap (e, n)| = j\). The set \(L_1'\) consists of leading numbers in column 2 of \(T\). The \(j\)th largest number in \(L_1'\), if exists, is the leading number at row \(j\) column 2 of \(T\). Thus, it is weakly less than \(e\), so \(|\{(e, n) \cap L_1'\}| < j\).
4. By \(T \in RSVT(\alpha)\), \(K_-(T) \leq \text{key}(\alpha)\). Let \(\gamma = \text{wt}(K_-(t))\). By Lemma 46, \(1_{L_1} + m(\gamma, L_1) \leq \alpha\). By Lemma 36,
\[\text{wt}(K_-(t)) = \gamma \leq \overline{M(\alpha, L_1)}.\]

Thus \(t \in RSVT(\overline{M(\alpha, L_1)})\).

Now, we check if we have \(T = (L_1, E_1, t)\) satisfying these four conditions, then \(T \in RSVT(\alpha)\). We may construct \(T\) as a tableau. First, we build column 1 of \(T\). We arrange numbers in \(L_1\) into a strictly decreasing column. For each \(e \in E_1\), by condition 2, \(|(e, n) \cap L_1| > 0\). Then we put \(e\) on row \(|(e, n) \cap L_1|\). Clearly, this column is an RSVT with leading numbers from \(L_1\) and extra numbers from \(E_1\).

Next, we may build the tableau corresponding to \(t\) recursively. Let \(L_1'\) be the set of leading numbers in column 1 of \(t\). Put the column we just constructed on the left of \(t\). Only need to check all numbers in row \(j\) of our new column are weakly larger than the leading number in row \(j\) column 1 of \(t\), which is the \(j\)th largest number in \(L_1'\).

- Let \(e\) be an extra number on row \(j\) of our new column, By condition 3, \(|(e, n) \cap L_1'| < j\). Thus, the \(j\)th largest number in \(L_1'\) is at most \(e\).
- By condition 4, if we let \(\gamma = \text{wt}(K_-(t))\), then \(\gamma \leq \overline{M(\alpha, L_1)}\). By Lemma 36, \(m(\gamma, L_1)\) exists. Then by Corollary 34, the \(j\)th largest number in \(\text{supp}(\gamma) = L_1'\) is weakly less than the \(j\)th largest number in \(L_1\), which is the leading number in row \(j\) of our new column.
Now we have constructed a tableau $T$ which if viewed as a diagram pair, corresponds to $(L_1, E_1, t)$. It remains to check $K_-(T) \leq key(\alpha)$. Notice that $K_-(T) = key(1_{L_1} + m(\gamma, L_1))$. By Lemma 36 and $\gamma \leq M(\alpha, L_1)$, we have $K_-(T) \leq key(\alpha)$. Thus, $(L_1, E_1, t) \in RSVT(\alpha)$. □

The recursive descriptions of $KKD(\alpha)$ and $RSVT(\alpha)$ share many similarities. They only differ at the third condition, which is the condition on the positions of ghost cells. From this observation, we get the following result which was discussed in Remark 10.

**Corollary 48.** We have $RSSYT(\alpha) = KD(\alpha)$.

**Proof.** We know $KD(\alpha)$ is the subset of $KKD(\alpha)$ containing all diagram pairs with no ghost cells. By Theorem 44, $KD(\alpha)$ consists of $(K_1, \emptyset, d)$ such that

- $K_1 \leq supp(\alpha)$.
- $d \in KD(M(\alpha, K_1))$.

On the other hand, $RSSYT(\alpha)$ is the subset of $RSVT(\alpha)$ containing all diagram pairs with no ghost cells. By Theorem 47, $RSSYT(\alpha)$ consists of $(L_1, \emptyset, t)$ such that

- $L_1 \leq supp(\alpha)$.
- $t \in RSSYT(M(\alpha, L_1))$.

An induction on max$(\alpha)$ yields $KKD(\alpha) = RSSYT(\alpha)$. □

This recursive description of $RSVT(\alpha)$ leads to the following lemma.

**Lemma 49.** Let $T = (L, E) = (L_1, E_1, t)$ be an element in $RSVT(\alpha)$. If $(L, \emptyset) \in KD(\gamma)$ for another weak composition $\gamma$, then $T \in RSVT(\gamma)$.

**Proof.** Prove by induction on max$(\gamma)$. Notice that $t$ is in $RSVT(M(\alpha, K_1))$. If we ignore ghost cells of $t$, it is in $KD(M(\gamma, K_1))$. By our inductive hypothesis, $t \in RSVT(M(\gamma, K_1))$.

Now we check $T = (L_1, E_1, t)$ satisfies the four conditions of $RSVT(\gamma)$. Condition 1 and 3 are implied by $T \in RSVT(\alpha)$. Condition 2 follows from $(L, \emptyset) \in KD(\gamma)$. The last condition is checked in the previous paragraph. □

**Remark 50.** Alternatively, we may prove Lemma 49 while viewing $T$ as a tableau. Let $T'$ be the tableau we get after keeping the smallest number in each cell of $T$. By $(L, \emptyset) \in KD(\gamma)$, we know $T' \in RSSYT(\gamma)$. Consequently, $K_-(T) = K_-(T') \leq \gamma$, so $T \in RSVT(\gamma)$.

### 6 Two operators on Kohnert diagrams

In order to prove the well-definedness and bijectivity of $\Psi_\alpha$ and $\Phi_\alpha$ defined in Section 3, we introduce two auxiliary operators $\sharp_g$ and $\flat_e$ on $KD(\alpha)$ and study their properties. Later in Section 7, we will use these two operators to give alternative descriptions of $\Psi_\alpha$ and $\Phi_\alpha$. 
6.1 Introducing the ♯_g operator

We define an operator ♯_g on KD(α) for each g ∈ [n].

**Definition 51.** For each g ∈ [n], define ♯_g : KD(α) → KD(α) × [n]. Take D = (K_1, ∅, d) ∈ KD(α). Find the largest k ∈ K_1 ∩ [1, g] such that D′ = (K_1 − {k} ⊔ {g}, ∅, d) is still in KD(α). If such k exists, then ♯_g(D) := (D′, k). Otherwise, ♯_g(D) is undefined.

We would like to determine when ♯_g(D) is defined. This is partially answered by the following lemma:

**Lemma 52.** Assume D = (K_1, ∅, d) ∈ KD(α). If ♯_g(D) is defined, then |[g, n] ∩ supp(α)| > |(g, n] ∩ K_1|.

The proof involves Kohnert tableaux from subsection 2.4.

**Proof.** Assume ♯_g(D) = (D′, k). Then in column 1 of D′, there are |(g, n] ∩ K_1| + 1 cells weakly above row g. In Label_α(D′), these cells are filled by distinct number in [g, n] ∩ supp(α). Thus, |[g, n] ∩ supp(α)| > |(g, n] ∩ K_1|. □

Next, we will show the converse of this lemma. First, we introduce an algorithm called **sharp algorithm.** Its input is a number g and D = (K_1, ∅, d) ∈ KD(α) such that |[g, n] ∩ supp(α)| > |(g, n] ∩ K_1|. It will output a diagram pair D′ with only Kohnert cells. It will also output a filling of D′. Later, we will check the filling is a Kohnert tableau with content α, which implies D′ ∈ KD(α). Finally, we will check D′ is the first component of ♯_g(D).

The sharp algorithm consists of five steps:

- **Step 1:** Compute Label_α(D).
- **Step 2:** Since |[g, n] ∩ supp(α)| > |(g, n] ∩ K_1|, there is a number m such that m ≥ g but m is weakly below row g in column 1. Find the highest such m. Let k be the row index of this m.
- **Step 3:** Let D′ = ((K_1 − {k}) ⊔ {g}, ∅, d). This is the first output.
- **Step 4:** To compute the filling, we start from Label_α(D) and move the m from (1, k) to (1, g). The resulting filling satisfies the first three conditions from Definition 16.
- **Step 5:** If there is an u < m such that u, m violates condition four in column 1, we find the smallest such u and swap it with m. Repeat this step until no such u exists. The final filling will be the second output.

**Example 53.** Consider α = (0, 0, 0, 2, 2, 1, 1), and D ∈ KD(α) as shown below. Let g = 3. The sharp algorithm gives m = 6 and k = 1. The output D′ is obtained by moving the
Kohnert cell \((1, k)\) to \((1, g)\) in \(D\). To obtain the filling, we need to first move \(m\) to row \(g\) in \(\text{Label}_\alpha(D)\). Next, swap \(m\) with 4 and then 5.

\[
D = \begin{array} {ccccc}
7 & \cdot & \cdot & \cdot & \cdot \\
6 & \cdot & \cdot & \cdot & \cdot \\
5 & \cdot & \cdot & \cdot & \cdot \\
4 & \cdot & \cdot & \cdot & \cdot \\
3 & \cdot & \cdot & \cdot & \cdot \\
2 & \cdot & \cdot & \cdot & \cdot \\
1 & \cdot & \cdot & \cdot & \cdot \\
\end{array}
\]

\[
\text{Label}_\alpha(D) = \begin{array} {ccccc}
5 & 6 & 4 & 4 & \cdot \\
4 & 6 & 5 & 4 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\end{array}
\]

\[
\text{Label}_\alpha(D) = \begin{array} {ccccc}
5 & 6 & 4 & 4 & \cdot \\
4 & 6 & 5 & 4 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\end{array}
\]

\[
\text{Label}_\alpha(D') = \begin{array} {ccccc}
6 & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\end{array}
\]

\[
D' = \begin{array} {ccccc}
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\end{array}
\]

**Lemma 54.** The filling produced by the sharp algorithm is a Kohnert tableau with content \(\alpha\). Consequently, the filling is \(\text{Label}_\alpha(D')\) and \(D' \in KD(\alpha)\).

**Proof.** We claim that after Step 4 and after each iteration of Step 5, the filling satisfies the first three conditions of Definition 16. Moreover, if \(i < j\) violates the last condition in column \(c\), then \(j = m\) and \(c = 1\).

After Step 4, the filling clearly satisfies the first three conditions. Now assume \(i < j\) violates the last condition in column \(c\). Clearly, \(c = 1\) and \(m\) is \(i\) or \(j\). Assume \(m\) is \(i\), then \(j\) is below row \(g\) in column 1. If \(j\) were below row \(k\), then \(m, j\) would have violated condition 4 before this move. On the other hand, if \(j\) were above row \(k\), then we would have picked \(j\) instead of \(m\). In either case, we reach a contradiction. Thus, \(j\) must be \(m\).

If there is a \(u\) such that \(u < m\) violates condition 4, we pick the smallest such \(u\). Assume our claim holds now. We need to show our claim is still true after we swap \(u\) and \(m\). We check the first three conditions:

1. Condition 1 clearly holds.

2. Only need to check condition 2 for \(m\). Recall \(u < m\). Since \(u\) satisfies condition 2 before the move, so does \(m\) after the move.

3. Only need to check condition 3 for \(u\). Since \(u, m\) violate condition 4 before this swap, and that \(u\) satisfies condition 3 before moving \(m\) to \((1, g)\), there is no \(u\) in column 2 strictly above the \(m\) in column 1. Thus, after the move, \(u\) satisfies condition 3.

Now assume \(i < j\) violates condition 4 in column \(c\). Clearly, \(c = 1\) and one of \(i, j\) is \(u\) or \(m\). We just need to check \(u\) cannot be \(i\) or \(j\) and \(m\) cannot be \(i\):

- Assume \(i = u\). Then \(u, j\) would have violated condition 4 before the move, contradicting to our claim.

- Assume \(j = u\). Then \(i < u < m\). Before the move, \(i, m\) violates condition 4. Then we would have picked \(i\) and swapped it with \(m\), rather than \(u\). Contradiction.

- Assume \(i = m\). If \(j\) were below \(m\) before the move, then \(m, j\) would have violated condition 4 before the move. Now assume \(j\) were between \(u\) and \(m\) before the move. Then \(u, j\) would have violated condition 4 before the move.
Now our claim holds after each move. When the sharp algorithm terminates, there is no violation of condition 4 with the form $i, m$. Thus, the filling satisfies condition 4, so it is in $\text{KT}(\alpha)$.

Lemma 55. The $D'$ yielded by the sharp algorithm is the first component of $\sharp_g(D)$.

Proof. The lemma is trivial if $g \in K_1$. Thus, we may assume $g \notin K_1$. We already showed $D' \in \text{KD}(\alpha)$ by constructing $\text{Label}_\alpha(D')$. Take $r \in K_1$ with $k < r < g$. It suffices to show $(K_1 - \{r\} \cup \{g\}, \emptyset, d) \notin \text{KD}(\alpha)$.

In column 1 of $\text{Label}_\alpha(D)$, assume $s_1, \ldots, s_p$ are the numbers below row $g$ and weakly above row $r$. By how we picked $m$, we know $s_1, \ldots, s_p < g$. We may run the labeling algorithm on $((K_1 - \{r\}) \cup \{g\}, \emptyset, d)$. It behaves the same as on $D$ on cells prior to $(1, r)$. After filling all these cells before $(1, r)$, we know $s_1, \ldots, s_p$ are still in the set $S_1$. However, there remains only $p - 1$ empty cells below row $g$. Thus, at least one number of $s_1, \ldots, s_p$ will be placed weakly above row $g$. Since this number is less than $g$, the labeling algorithm will terminate and produce no output. Thus, this diagram is not in $\text{KD}(\alpha)$.

Thus, we know the sharp algorithm outputs the first component of $\sharp_g(D)$, together with its Kohnert Labeling. Now we can tell when $\sharp_g(D)$ is well-defined.

Lemma 56. Assume $D = (K_1, \emptyset, d) \in \text{KD}(\alpha)$. Then $\sharp_g(D)$ is well-defined if and only if $|\langle g, n \rangle \cap \text{supp}(\alpha)| > |\langle g, n \rangle \cap K_1|$.

Proof. The forward direction is given by Lemma 52. The other direction follows from the sharp algorithm.

Corollary 57. Take $D \in \text{KD}(\alpha)$. Assume $\sharp_g(D) = (D', k)$. Assume $(1, k)$ is filled by $m$ in $\text{Label}_\alpha(D)$. Then $\text{Label}_\alpha(D)$ and $\text{Label}_\alpha(D')$ agree at the cell $(c, r)$, if $(c, r)$ satisfies one of the following:

- $c > 1$;
- $r < g$ and $r \neq k$;
- $r > m$.

Proof. By the behavior of the sharp algorithm, $\text{Label}_\alpha(D')$ is obtained from $\text{Label}_\alpha(D)$ by moving $m$ from $(1, k)$ to $(1, g)$ and repeatedly swapping $m$ with a number above it. The number $m$ will not go above row $m$, so only $(1, k)$ and cells between row $g$ and row $m$ in column 1 are affected.

6.2 Commutativity of $\sharp_g$ operators

Next, we observe that two $\sharp_g$ operators might “commute” under certain conditions. Consider the following example:
Example 58. Let $\alpha = (0,0,2,1)$. Let $D$ be the following element in $\text{KD}(\alpha)$. We first apply $\sharp_3$ and get $(D^1, 2)$. Then apply $\sharp_4$ on $D^1$ and get $(D^{\text{final}}, 1)$.

$$D = \begin{array}{cccc}
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\end{array} = D^{\text{final}}$$

We can try to swap the order of these two operators. We first apply $\sharp_4$ on $D$ and get $(D^2, 1)$. Then we apply $\sharp_3$ on $D^2$ and get $(D^{\text{final}}, 2)$.

$$D = \begin{array}{cccc}
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\end{array} = D^{\text{final}}$$

Observe that changing the order of these two operators will not affect the final Kohnert diagram.

This phenomenon is captured by the following two lemmas.

Lemma 59. Take $D \in \text{KD}(\alpha)$. Take $g_1, g_2 \in [n]$ with $g_1 < g_2$. Assume $\sharp_{g_1}(D) = (D^1, k_1)$ and $\sharp_{g_2}(D^1) = (D^{\text{final}}, k_2)$. If $k_1 > k_2$, then the two operators “commute”. That is:

- $\sharp_{g_2}(D^1) = (D^2, k_2)$ for some $D^2 \in \text{KD}(\alpha)$, and
- $\sharp_{g_1}(D^2) = (D^{\text{final}}, k_1)$.

Proof. Let $C$ be the first column of $\text{Label}_g(D)$. Define $C^1$ and $C^{\text{final}}$ similarly. In $C$, let $m_1$ (resp. $m_2$) be the number at row $k_1$ (resp. $k_2$).

First, we claim $m_1$ is below row $g_2$ in $C^1$. If not, then we may find a number $u$ that is weakly above row $g_2$ in $C$ but below row $g_2$ in $C^1$. By Corollary 57, the $u$ is above row $g_1$ in $C^1$, so $u$ is higher than $m_2$. By $u \geq g_2$, we should pick $u$ rather than $m_2$ when computing $\sharp_{g_2}(D^1)$. Contradiction.

Now consider $C^1$. By Corollary 57, $m_2$ is still at row $k_2$. All numbers between $m_2$ and row $g_2$ will be less than $g_2$. Thus, $m_1 < g_2$. Since $C$ and $C^1$ only differ between row $g_1$ and row $m_1$, all numbers between $m_2$ and row $g_2$ will be less than $g_2$ in $C$. Thus, $\sharp_{g_2}$ will also pick $m_2$ when acting on $D$. $\sharp_{g_2}(D) = (D^2, k_2)$. Column 1 of $\text{Label}_a(D^2)$ agrees with $C$ between row $k_1$ and $g_1$. Thus, $\sharp_{g_1}$ will pick $m_1$ when acting on $D^2$. \(\square\)

Lemma 60. Take $D \in \text{KD}(\alpha)$. Take $g_1, g_2 \in [n]$ with $g_1 < g_2$. Assume $\sharp_{g_2}(D) = (D^2, k_2)$ and $\sharp_{g_1}(D^2) = (D^{\text{final}}, k_1)$. If $k_1 > k_2$, then the two operators “commute”. That is:

- $\sharp_{g_1}(D) = (D^1, k_1)$ for some $D^1 \in \text{KD}(\alpha)$, and
- $\sharp_{g_2}(D^1) = (D^{\text{final}}, k_2)$.
Proof. Let $C$ be the first column of $\text{Label}_\alpha(D)$. Define $C^2$ and $C^{\text{final}}$ similarly. In $C$, let $m_1$ (resp. $m_2$) be the number at row $k_1$ (resp. $k_2$).

First, $C$ and $C^2$ agree between row $k_2$ and row $g_2$. Thus, $\sharp_{g_1}$ would also pick $m_1$ when acting on $D$, so $\sharp_{g_1}(D) = (D^1, g_2)$.

Since $\sharp_{g_2}$ picks $m_2$ when acting on $D$, all numbers between row $k_2$ and row $g_2$ in $C$ are less than $g_2$. In particular, $m_1 < g_2$. We know column 1 of $D^1$ is obtained from $C$ by changing cells between row $k_1$ and row $m_1$. Thus, in column 1 of $D^1$, all numbers between row $k_2$ and row $g_2$ are still less than $g_2$. When acting on $D^1$, $\sharp_{g_2}$ would still pick $m_2$. \hfill $\square$

6.3 Introducing the $b_k$ operator

Next, we define the operator $b_k$, which can be viewed as the (partial) inverse of $\sharp_g$.

**Definition 61.** For each $k \in [n]$, define $b_k : \text{KD}(\alpha) \to \text{KD}(\alpha) \times [n]$. Take $D = (K_1, \emptyset, d) \in \text{KD}(\alpha)$. Find the smallest $g \in K_1 \cap [k, n]$ such that $D' = ((K_1 - \{g\}) \cup \{k\}, \emptyset, d)$ is still in $\text{KD}(\alpha)$. If such $g$ exists, then $b_k(D) := (D', k)$. Otherwise, $b_k(D)$ is undefined.

**Lemma 62.** Take $D \in \text{KD}(\alpha)$.

- Take $g \in [n]$. If $\sharp_g(D) = (D', k)$, then $b_k(D') = (D, g)$.
- Take $k \in [n]$. If $b_k(D) = (D', g)$, then $\sharp_g(D') = (D, k)$.

In other words, $b_k$ and $\sharp_g$ are (partial) inverses of each other.

Proof. Assume $D = (K_1, \emptyset, d)$. Consider the first statement. If $g \in K_1$, then $k = g$ and $b_k(D') = (D, g)$ trivially. Now assume $g \notin K_1$. Then $((K_1 - \{r\}) \cup \{g\}, \emptyset, d)$ is not in $\text{KD}(\alpha)$ for all $r \in K_1$ with $k < r < g$. Thus, $b_k(D') = (D, g)$. The second statement can be proved similarly. \hfill $\square$

We would like to determine when $b_k(D)$ is well-defined. This is answered by the following lemma:

**Lemma 63.** Assume $D = (K_1, \emptyset, d) \in \text{KD}(\alpha)$. Let $K_2$ be the set of row indices for cells in column 2 of $D$. Then $b_k(D)$ is well-defined if and only if $k \in K_1$ or $|K_1 \cap (k, n]| > |K_2 \cap (k, n]|$.

Proof. First, assume the condition fails. We show $b_k(D)$ is undefined. Assume by contradiction that $b_k(D) = (D', g)$. Define $K'_1$ and $K'_2$ similarly for $D'$. Then $K'_1 = (K_1 - \{g\}) \cup \{k\}$ and $K'_2 = K_2$. Thus,

$$|K'_1 \cap (k, n]| < |K_1 \cap (k, n]| \leq |K_2 \cap (k, n]| = |K'_2 \cap (k, n]|.$$

Then consider $\text{Label}_\alpha(D')$. There are $|K'_2 \cap (k, n]|$ distinct numbers above row $k$ in column 2. They all must appear above row $k$ in column 1, but there are not enough cells for them. Contradiction.

Now assume the condition holds, we show $b_k(D)$ is well-defined. Clearly, we are done if $k \in K_1$. Now assume $|K_1 \cap (k, n)] > |K_2 \cap (k, n]|$ and consider $\text{Label}_\alpha(D)$. By our
assumption, we can find $m$ above row $k$ in column 1 such that there is no $m$ above row $k$ in column 2. Pick the lowest such $m$ and move it to $(1, k)$. Then the resulting filling is in $\text{KT}(\alpha)$:

1. Condition 1 of $\text{KT}(\alpha)$ is clear.
2. Since we moved a cell down, condition 2 is clear.
3. Condition 3 holds for $m$ since there is no $m$ above row $k$ in column 2.
4. Only need to check there is no violations of condition 4 in column 1. Let $i < j$ be a violation. Then $m$ must be $i$ or $j$. If $i = m$, then $m, j$ would have violated condition 4 before the move. Now assume $j = m$. If $i$ were above $m$ in $\text{Label}_\alpha(D)$, then $i, m$ would have violated condition 4 before the move. On the other hand, if $i$ were below $m$ in $\text{Label}_\alpha(D)$, then there is an $i$ above row $k$ in column 2, so $i, m$ cannot be a violation.

Thus, after moving one cell down to $(1, k)$ in $D$, the resulting diagram is still a Kohnert diagram, so $b_k(D)$ is well-defined.

Remark 64. In the previous proof of well-definedness, we choose the cell containing $m$ and move it down to row $k$. The resulting diagram is still in $\text{KD}(\alpha)$. Notice that this might not be the lowest cell that can do this job. See the following example.

Example 65. Following Example 53. We would like to compute $b_1(D')$. In $D'$, there are 4 cells in column 1 above row 1 and there are 2 cells in column 2 above row 1. Thus, the condition in Lemma 63. is satisfied. We want to check $b_1(D')$ is well-defined. The proof of well-definedness gives $m = 6$. After moving the 6 to row 1, the resulting filling is in $\text{KT}(\alpha)$, which implies the underlying diagram is in $\text{KD}(\alpha)$.

$$\text{Label}_\alpha(D') = \begin{array}{cccc}
4 & 5 & 4 & 5 \\
4 & 5 & 4 & 5 \\
6 & & & \\
7 & 7 & & \\
6 & & & \\
5 & & & \\
4 & & & \\
3 & & & \\
2 & & & \\
1 & & & \\
\end{array}$$

$D' = \begin{array}{cccc}
7 & 6 & & \\
6 & & & \\
5 & & & \\
4 & & & \\
3 & & & \\
2 & & & \\
1 & & & \\
\end{array}$

However, moving the cell $(1, 3)$ to $(1, 1)$ in $D'$ will also make the resulting diagram in $\text{KD}(\alpha)$.

Similar to $\sharp_g$, the $b_k$ operator can commute under certain conditions:

Lemma 66. Take $D \in \text{KD}(\alpha)$. Take $k_1, k_2 \in [n]$ with $k_1 > k_2$. Assume $b_{k_1}(D) = (D^1, g_1)$ and $b_{k_2}(D^1) = (D_{\text{final}}, g_2)$. If $g_1 < g_2$, then the two operators “commute”. That is:

- $b_{k_2}(D) = (D^2, g_2)$ for some $D^2 \in \text{KD}(\alpha)$, and
- $b_{k_1}(D^2) = (D_{\text{final}}, g_1)$.

Proof. It follows directly from Lemma 60 and Lemma 62.
6.4 Relations between $\sharp_g$ and $\flat_e$

In this section, we investigate the relationship between the two operators introduced above. We already know the effect of $\sharp_g$ can be reversed by the $\flat_e$ operator, and vice versa. Next, we show that a sequence of $\sharp_g$ can also be reversed by a sequence of $\flat_e$, and vice versa.

**Lemma 67.** Let $D^0 = (K_1, \emptyset, d) \in \text{KD}(\alpha)$, and $1 < g_1 < g_2 < \ldots < g_m \leq n$ with $g_i \notin K_1$. For $i = 1, 2, \ldots, m$, compute $\sharp_g_i(D^{i-1}) = (D^i, k_i)$. Assume $D^1, \ldots, D^m$ are all well-defined.

Find the permutation $\sigma$ such that $k_{\sigma(1)} < \ldots < k_{\sigma(m)}$ and define $T^m = D^m$. For $i = m, m - 1, \ldots, 1$, compute $\flat_{k_{\sigma(i)}}(T^i) = (T^{i-1}, g'_i)$. Then $T^0 = D^0$ and $\{g_1, \ldots, g_m\} = \{g'_1, \ldots, g'_m\}$.

**Proof.** We may represent $D^0, \ldots, D^m$ using the following diagram:

$$
\begin{array}{cccc}
D^0 & \overset{\sharp_g_1}{\rightarrow} & D^1 & \overset{\sharp_g_2}{\rightarrow} & \ldots & \overset{\sharp_g_m}{\rightarrow} & D^m \\
\longrightarrow_{k_1} & \longrightarrow_{k_2} & \ldots & \longrightarrow_{k_m}
\end{array}
$$

We put the operator above the arrow and put the second output under the arrow.

Suppose we find $k_i > k_{i+1}$. By Lemma 59, we can swap the order of $\sharp_{g_i}$ and $\sharp_{g_{i+1}}$, not affecting the last diagram $D^m$. Thus, after sorting the output numbers into increasing order, we have

$$
\begin{array}{cccc}
D^0 & \overset{\sharp_g_{\sigma(1)}}{\rightarrow} & \tilde{D}^1 & \overset{\sharp_g_{\sigma(2)}}{\rightarrow} & \ldots & \overset{\sharp_g_{\sigma(m)}}{\rightarrow} & D^m \\
\longrightarrow_{k_{\sigma(1)}} & \longrightarrow_{k_{\sigma(2)}} & \ldots & \longrightarrow_{k_{\sigma(m)}}
\end{array}
$$

where $\tilde{D}^i$ are some diagrams in $\text{KD}(\alpha)$. Finally, we have

$$
\begin{array}{cccc}
D^0 & \overset{\flat_{g_{\sigma(1)}}}{\leftarrow} & \tilde{D}^1 & \overset{\flat_{g_{\sigma(2)}}}{\leftarrow} & \ldots & \overset{\flat_{g_{\sigma(m)}}}{\leftarrow} & D^m \\
\longrightarrow_{g_{\sigma(1)}} & \longrightarrow_{g_{\sigma(2)}} & \ldots & \longrightarrow_{g_{\sigma(m)}}
\end{array}
$$

By $T^m = D^m$, we have $D^0 = T^0$. \hfill \Box

**Example 68.** Consider $\alpha = (0, 0, 2, 0, 3, 1, 2)$, $g_1 = 3, g_2 = 5, g_3 = 6$ and $D^0 \in \text{KD}(\alpha)$. Starting with $D^0$, we compute $\sharp_{g_1}, \sharp_{g_2}$ and then $\sharp_{g_3}$ to obtain $D^3$.

$$
D^0 = \begin{array}{cccc}
7 & \cdot & \cdot & \cdot \\
6 & \cdot & \cdot & \cdot \\
5 & \cdot & \cdot & \cdot \\
4 & \cdot & \cdot & \cdot \\
3 & \cdot & \cdot & \cdot \\
2 & \cdot & \cdot & \cdot \\
1 & \cdot & \cdot & \cdot \\
\end{array}
\overset{\sharp_{g_1}}{\rightarrow} \begin{array}{cccc}
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\end{array}
\overset{\sharp_{g_2}}{\rightarrow} \begin{array}{cccc}
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\end{array}
\overset{\sharp_{g_3}}{\rightarrow} \begin{array}{cccc}
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\end{array} = D^3.
$$

We obtain $k_1 = 2, k_2 = 4$ and $k_3 = 1$, which are highlighted in the above figure. We can pick the permutation $\sigma$ with one-line notation 312 and obtain $k_{\sigma(1)} = 1, k_{\sigma(2)} = 2$ and
$k_{\sigma(3)} = 4$. Then we have $g_{\sigma(1)} = 6, g_{\sigma(2)} = 3$ and $g_{\sigma(3)} = 5$, which yields the following sequence of operation with the same $D^3$ as the final output.

\[
D^0 = \begin{array}{cccc}
\begin{array}{ccc}
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\end{array}
\end{array}
\]

Applying $b_4, b_2$ then $b_1$ on $D^3$, we will recover $D^0$.

**Lemma 69.** Let $D^0 = (K_1, \emptyset, d) \in \text{KD}(\alpha)$, and $n > e_1 > e_2 > \cdots > e_m \geq 1$, with $e_i \notin K_1$. For $i = 1, 2, \ldots, m$, compute $b_{e_i}(D^{i-1}) = (D^i, k_i)$. Assume $D^1, \ldots, D^m$ are all well-defined.

Find the permutation $\sigma$ such that $k_{\sigma(1)} > \cdots > k_{\sigma(m)}$. Now define $T^m = D^m$. For $i = m, m-1, \ldots, 1$, compute $\sharp_{k_{\sigma(i)}}(T^i) = (T^{i-1}, e'_i)$. Then $T^0 = D^0$ and $\{e_1, \ldots, e_m\} = \{e'_1, \ldots, e'_m\}$.

**Proof.** The proof is the same as the previous proof, using Lemma 66 instead of Lemma 59.

\[\square\]

## 7 Recursive descriptions of the maps

We have described our maps $\Psi_\alpha$ and $\Phi_\alpha$ via $\sharp_G(K)$ and $b_E(L)$ in Section 3. These descriptions are simple to state but hard to work with. Now, we will describe the $\Psi_\alpha$ and $\Phi_\alpha$ recursively, involving definitions from Section 6 and Section 7. Using the new alternative descriptions, we can establish Lemma 20, Lemma 23 and Theorem 3.

### 7.1 Recursive description of $\Psi_\alpha$

Let $G$ be an arbitrary diagram. First, we can recursively describe the operator $\sharp_G(\cdot)$. If $D$ is empty, then $\sharp_G(D)$ is also empty if $G = \emptyset$, or undefined otherwise. If $D$ is not empty, write $D$ as $(K_1, \emptyset, d)$. Let $G_{\geq 2}$ be the diagram $\{(c - 1, r) : (c, r) \in G, c \geq 2\}$. View $d$ as an element of $\text{KD}(M(\alpha, K_1))$ and find $d' = \sharp_G(d)$ recursively. Let $G_1$ be the set $\{r : (1, r) \in G\}$ and assume $G_1 = \{g_1 < \cdots < g_{|G_1|}\}$. Let $D^0$ be the Kohnert diagram $(K_1, \emptyset, d')$. Then compute $\sharp_{g_i}(D^{i-1}) = (D^i, k_i)$ for $1 \leq i \leq |G_1|$. The final output is $D^{G_1}$.

**Lemma 70.** The description is equivalent to the description of $\sharp_G(K)$ in subsection 3.1.

**Proof.** Recall that $\sharp_G(K)$ iterates over cells of $G$ from right to left. Within each column, it goes from bottom to top. For a cell $(c, r) \in G$, it picks the highest cell weakly below $(c, r)$ such that once this cell is raised to $(c, r)$, the diagram is still in $\text{KD}(\alpha)$. Then it moves the chosen cell to $(c, r)$.
Let $D = (K_1, \emptyset, d)$ be the Kohnert diagram at the beginning of the iteration of $(c, r) \in G$. Assume $c \geq 2$. By the recursive description of $KD(\alpha)$, the following two statements are equivalent:

- $(c, r')$ is a cell in $D$ such that if we move it to $(c, r)$, the diagram is still in $KD(\alpha)$.
- $(c - 1, r')$ is a cell in $d$ such that if we move it to $(c - 1, r)$, the diagram is still in $KD(M(\alpha, K_1))$.

Thus, iterations of $(c, r) \in G$ with $c \geq 2$ will behave the same as if $\sharp_{G_{\geq 2}}$ acts on $d \in KD(M(\alpha, K_1))$. Then iterations of $(1, g) \in G$ can be characterized by the $\sharp_g$ operator. □

Now we can recursively describe the map $\Psi_\alpha$. To make our description concise, we extend $\sharp_g(\cdot)$ to diagram pairs $(K, G)$ such that $(K, \emptyset) \in KD(\alpha)$ and $G$ has no cells in column 1. The operator $\sharp_g(\cdot)$ acts as if acting on $(K, \emptyset)$.

Now take $D = (K_1, G_1, d) \in KKD(\alpha)$. If $D$ is the empty pair, we have $\Psi_\alpha(D) = D$. Otherwise, let $t = \Psi_\gamma(d)$, where $\gamma = M(\alpha, K_1)$. Assume $G_1 = \{g_1 < \cdots < g_{|G_1|}\}$. Let $D^0$ be the diagram pair $(K_1, \emptyset, t)$. Then compute $\sharp_\alpha(D^i - 1) = (D^i, k_i)$ and write $D^i$ as $(K^i_1, \emptyset, t)$. Finally, $\Psi_\alpha(D)$ is $(K^{i[G_1]}_1, \{k_1, \ldots, k_{|G_1|}\}, t)$.

**Example 71.** Consider $\alpha = (0, 0, 2, 0, 3, 1, 2)$. Let $D = (K, G)$ be the following element in $KKD(\alpha)$.

\[
\begin{array}{cccccc}
7 & \cdot \\
6 & \times & \times \\
5 & \times & \cdot \\
4 & \cdot & \cdot & \cdot & \cdot \\
3 & \times \\
2 & \cdot & \cdot \\
1 & \cdot \\
\end{array}
\]

D = 

If we compute $\Psi_\alpha(D)$ using the description in subsection 3.1, we would go through the following iterations.

\[
\begin{array}{cccc}
\begin{array}{ccccccc}
7 & \cdot \\
6 & \times & \times \\
5 & \times & \cdot \\
4 & \cdot & \cdot & \cdot & \cdot \\
3 & \times \\
2 & \cdot & \cdot \\
1 & \cdot \\
\end{array} & \rightarrow & \begin{array}{ccccccc}
\cdot & \cdot \\
\cdot & \cdot \\
\underbrace{\underbrace{\ldots}_{(2, 6)} & \ldots} & \cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\end{array} & \rightarrow & \begin{array}{ccccccc}
\cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\underbrace{\underbrace{\ldots}_{(1, 3)} & \ldots} & \cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\end{array} & \rightarrow & \begin{array}{ccccccc}
\cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\underbrace{\underbrace{\ldots}_{(1, 5)} & \ldots} & \cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\end{array} & \rightarrow & \begin{array}{ccccccc}
\cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\underbrace{\underbrace{\ldots}_{(1, 6)} & \ldots} & \cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\end{array} = L.
\end{array}
\]
Thus, we have
\[ \Psi_\alpha(D) = (L, (K \sqcup G) - L) = \]
\[
\begin{array}{cccccccc}
7 & \cdot \\
6 & \cdot \\
5 & \cdot \\
4 & X & \cdot \\
3 & \cdot \\
2 & X & \cdot \\
1 & X \\
\end{array}
\]

Now we try our new recursive description. We may write
\[ D \] as \((K_1, G_1, d)\), where \(K_1 = \{1, 2, 4, 7\}, G_1 = \{3, 5, 6\}\) and \(d\) is illustrated below. Our new description would first view \(d\) as an element of \(\text{KKD}(\overline{M(\alpha, K_1)}) = \text{KKD}((0, 1, 0, 2, 0, 0, 0, 1))\) and send it to \(t\):
\[
\begin{array}{cccccccc}
6 & X \\
5 & \cdot \\
4 & \cdot \\
3 & \cdot \\
2 & \cdot \\
1 & \cdot \\
\end{array} \rightarrow \begin{array}{cccc}
\cdot \\
\cdot \\
\cdot \\
\cdot \\
\cdot \\
\cdot \\
\end{array} = t.
\]

It remains to perform \(\sharp_3, \sharp_5\) and \(\sharp_6\).

\[
D^0 = \begin{array}{cccccccc}
7 & \cdot \\
6 & \cdot \\
5 & \cdot \\
4 & \cdot & \cdot & \cdot \\
3 & \cdot \\
2 & \cdot \\
1 & \cdot \\
\end{array} \rightarrow \begin{array}{cccccccc}
\cdot \\
\cdot \\
\cdot \\
\cdot & \cdot \\
\cdot \\
\cdot \\
\cdot \\
\end{array} \rightarrow \begin{array}{cccccccc}
\cdot \\
\cdot \\
\cdot \\
\cdot & \cdot \\
\cdot \\
\cdot \\
\cdot \\
\end{array} = D^3.
\]

Finally, the image is just \((\{3, 5, 6, 7\}, \{1, 2, 4\}, t)\), which agrees with the computation above.

It is clear that this recursive description agrees with the original description of \(\Psi_\alpha\).

To prove Lemma 20, we need to show \(t, D^0, \ldots, D^{[G_1]}\) exist and satisfy our assumptions. Besides, we need to check the final output is a diagram pair in \(\text{RSVT}(\alpha)\).

**Proof of Lemma 20.** Prove by induction on \(\text{max}(\alpha)\). We may assume \(\Psi_\gamma\) is a well-defined map from \(\text{KKD}(\gamma)\) to \(\text{RSVT}(\gamma)\), where \(\gamma = \overline{M(\alpha, K_1)}\). Thus, we know \(t \in \text{RSVT}(\gamma)\).

Then clearly if we ignore ghost cells in \(D^0\), it is in \(\text{KD}(\alpha)\). Moreover, \(D^0\) has no ghost cells in column 1. Next, we need to show the diagram pairs \(D^i\) are well-defined. By Theorem 44, we know for each \(g_i\), \([|g_i, n] \cap \text{supp}(\alpha)| > |[g_i, n] \cap K_1|\). Notice that the first \(i - 1\) iterations will not move any cells above row \(g_i - 1\). Thus, \([g_i, n] \cap K_1 = [g_i, n] \cap K_1^{-1}\).

By Lemma 56, \(D^i\) exists.

Next, we need to check the image is in \(\text{RSVT}(\alpha)\). In other words, we need to check \(T = (K^{[G_1]}, \{k_1, \ldots, k_{[G_1]}\}, t)\) should satisfy all four conditions in Theorem 47. Let \(L'_1\) be the set of row indices of Kohnert cells in column 1 of \(t\).
1. The first condition is immediate.

2. Since Kohnert cells of $D^{[G_1]} = (K_1^{[G_1]}, \emptyset, t)$ is in $KD(\alpha)$, we have $K_1^{[G_1]} \subseteq \text{supp}(\alpha)$.

3. For each $k_i$, we show $|(k_i, n] \cap K_1^{[G_1]}| > |(k_i, n] \cap L'_i|$. Since Kohnert cells of $D^{i-1} = (K_1^{i-1}, \emptyset, t)$ is in $KD(\alpha)$, we know the Kohnert cells of $t$ is in $KD(M(\alpha, K_1^{i-1}))$. Thus, $L'_i \subseteq \text{supp}(M(\alpha, K_1^{i-1})) \subseteq K_1^{i-1}$. We have

$$|(k_i, n] \cap K_1^{i-1}| > |(k_i, n] \cap L'_i|.$$ 

Since $K_1^{i-1}$ is obtained from $K_1^{i-1}$ by replacing $k_i$ with a larger number, we have $|(k_i, n] \cap K_1^{i-1}| = |(k_i, n] \cap K_1^{i-1}| + 1 > |(k_i, n] \cap L'_i|$. To obtain $K_1^{[G_1]}$, we replace each of $k_{i+1}, \ldots, k_{[G_1]}$ in $K_1^{i}$ with a larger number. Therefore, $|(k_i, n] \cap K_1^{[G_1]}| > |(k_i, n] \cap L'_i|$. 

4. Kohnert cells of $t$ is in $KD(M(\alpha, K_1^{[G_1]}))$ and $t \in \text{RSVT}(M(\alpha, K_1^{[G_1]}))$. By Lemma 49, $t \in \text{RSVT}(M(\alpha, K_1^{[G_1]}))$. 

\[\square\]

### 7.2 Recursive description of $\Phi_\alpha$

Let $E$ be an arbitrary diagram. We can recursively describe the operator $b_E(\cdot)$. If $D$ is empty, then $b_E(D)$ is also empty if $E = \emptyset$, or undefined otherwise. If $D$ is not empty, write $D$ as $(L_1, \emptyset, t)$. Let $E_{\geq 2}$ be the diagram $\{(c - 1, r) : (c, r) \in E, c \geq 2\}$. Let $E_1$ be the set $\{r : (1, r) \in E\}$. Assume $E_1 = \{e_1 > \cdots > e_{|E_1|}\}$. Let $D^0$ be the Kohnert diagram $(L_1, \emptyset, t)$. Then compute $b_{e_i}(D^{i-1}) = (D^i, g_i)$ for $1 \leq i \leq |E_1|$ and write $D^i$ as $(L_1^i, \emptyset, t)$. View $t$ as an element of $KD(M(\alpha, L_1^{[E_1]}))$ and find $t' = b_{E_{\geq 2}}(t)$ recursively. Finally, $b_E(D) = (L_1^{[E_1]}, \emptyset, t')$.

**Lemma 72.** The description is equivalent to the description of $b_E(L)$ in subsection 3.2.

**Proof.** Recall that $b_E(L)$ iterates over cells of $E$ from left to right. Within each column, it goes from top to bottom. For a cell $(c, r) \in E$, it picks the lowest cell weakly above $(c, r)$ such that once this cell is lowered to $(c, r)$, the diagram is still in $KD(\alpha)$. Then it moves the chosen cell to $(c, r)$.

Let $D = (L_1, \emptyset, t)$ be the Kohnert diagram at the beginning of the iteration of $(c, r) \in E$. The iterations of $(1, e) \in E$ can be characterized by the $b_e$ operator. Assume $c \geq 2$. By the recursive description of $KD(\alpha)$, the following two statements are equivalent for any $r' < r$:

- $(c, r')$ is a cell in $D$ such that if we move it to $(c, r)$, the diagram is still in $KD(\alpha)$.
- $(c - 1, r')$ is a cell in $t$ such that if we move it to $(c - 1, r)$, the diagram is still in $KD(M(\alpha, L_1^{[E_1]})))$.

Thus, iterations of $(c, r) \in E$ with $c \geq 2$ will behave the same as if $b_{E_{\geq 2}}$ acts on $t \in KD(M(\alpha, L_1^{[E_1]}))$. \[\square\]
Now we can recursively describe the map $\Phi_\alpha$. To make our description concise, we extend $b_e(\cdot)$ to diagram pairs $(K, G)$ such that $(K, \emptyset) \in KD(\alpha)$ and $G$ has no cells in column 1. The operator $b_e(\cdot)$ acts as if acting on $(K, \emptyset)$.

Now take $D = (L_1, E_1, t) \in RSVT(\alpha)$. If $D$ is the empty diagram pair, we have $\Phi_\alpha(D) = D$. Otherwise, assume $E_1 = \{e_1 > \cdots > e_{|E_1|}\}$. Let $D^0$ be the diagram pair $(L_1, \emptyset, t)$. Then compute $b_e(D_i^{i-1}) = (D_i^i, g_i)$ and write $D_i^i$ as $(L_i^i, E_i^i, t)$. Notice that $t \in RSVT(M(\alpha, L_1^i))$ and its Kohnert cells is in $KD(M(\alpha, L_1^{|E_1|}))$. Thus, by Lemma 49, we may view $t$ as an element of $RSVT(\gamma)$, where $\gamma = M(\alpha, L_1^{|E_1|})$. Let $d = \Phi_\gamma(t)$. Finally,  

```
Example 73. Consider $\alpha = (0,0,2,0,3,1,2)$. Let $D = (L, E)$ be the following element in $RSVT(\alpha)$.
```

![Diagram](image)

If we want to compute $\Phi_\alpha(D)$ using the description in subsection 3.2, we would go through the following iterations.

```
L =
```

![Diagram](image)

Thus, $\Phi_\alpha(D)$ is:

```
(K, (L \cup E) - K) =
```

![Diagram](image)

Now we try our new recursive description. We may write $D$ as $(L_1, E_1, t)$, where $L_1 = \{3, 5, 6, 7\}$, $E_1 = \{1, 2, 4\}$ and $t$ is illustrated below. Our new description would first perform $b_4, b_2$ and $b_1$ on $D^0$:
Finally, $\Phi_d$ that $(d,D)$

To prove Lemma 23, we need to show $e_i \leq D$ no ghost cells in column 1. Next, we need to show map from $\text{RSVT}$

Proof of Lemma 23. Prove by induction on max($\alpha$). We may assume $\Phi_\gamma$ a well-defined map from $\text{RSVT}(\gamma)$ to $\text{KKD}(\gamma)$ for any $\gamma$ with max($\gamma$) < max($\alpha$).

Clearly $D^0$ is a diagram pair whose Kohnert cells form a diagram in $\text{KD}(\alpha)$ and has no ghost cells in column 1. Next, we need to show $D^i$ is well-defined for an arbitrary $1 \leq i \leq |E_1|$. Notice that the first $i - 1$ iterations will not move any cells weakly below row $e_{i-1}$. Let $L'_1$ consists of row indices of Kohnert cells in column 1 of $t$. Thus,

$$|(e_{i-1}, n) \cap L^{-1}_1| = |(e_{i-1}, n) \cap L_1| > |(e_{i-1}, n) \cap L'_1|,$$

where the inductive hypothesis follows from Theorem 47. By Lemma 63, $D^i$ exists. Finally, by the inductive hypothesis, $d \in \text{KKD}(M(\alpha, L^{E_1}_1))$.

Next, we need to check the final image is in $\text{KKD}(\alpha)$. In other words, we need to check that $(L^{E_1}_1, \{g_1, \ldots, g_{|E_1|}\}, d)$ satisfies all four conditions in Theorem 44.

1. The first condition is immediate.

2. Since Kohnert cells of $D^{E_1} = (L^{E_1}_1, \emptyset, t)$ is in $\text{KD}(\alpha)$, we have $L^{E_1}_1 \subseteq \text{supp}(\alpha)$.

3. For each $g_i$, we show $|\{g_i, n\} \cap \text{supp}(\alpha)| > |\{g_i, n\} \cap L^{E_1}_1|$. Since $(L_1, E_1, t) \in \text{RSVT}(\alpha)$, we have $L_1 \subseteq \text{supp}(\alpha)$. Since $L^{E_1}_1$ is obtained from $L^{-1}_1$ by replacing $g_j$ with a smaller number, $L^{E_1}_1 \leq \cdots \leq L^0_1 \leq \text{supp}(\alpha)$. By Lemma 29,

$$|\{g_i, n\} \cap L^{E_1}_1| \leq \cdots \leq |\{g_i, n\} \cap L^0_1| \leq |\{g_i, n\} \cap \text{supp}(\alpha)|.$$

Notice that $|\{g_i, n\} \cap L^0_1| = |\{g_i, n\} \cap L^{-1}_1| - 1$. Thus, $|\{g_i, n\} \cap L^{E_1}_1| < |\{g_i, n\} \cap \text{supp}(\alpha)|$. 

Now, view $t$ as an element of $\text{KKD}(M(\alpha, L^0_1)) = \text{KKD}((0, 1, 0, 2, 0, 0, 1))$ and send it to $d$:

$$t = \begin{pmatrix} 6 & \cdot \\ 5 & \cdot \\ 4 & \cdot \\ 3 & \cdot \\ 2 & \cdot \\ 1 & \cdot \end{pmatrix} \rightarrow \begin{pmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ X \end{pmatrix} = d.$$

Finally, $\Phi_\alpha(D) = (\{1, 2, 4, 7\}, \{3, 5, 6\}, d)$, which agrees with the computation above.
4. This is checked above.

7.3 Proof of Theorem 3

In this subsection, we prove Theorem 3.

**Proof of Theorem 3.** The maps clearly preserve $\text{wt}(\cdot)$ and $\text{ex}(\cdot)$. To show they are mutually inverses, we only need to check the following two statements.

1. Take $D \in \text{KKD}(\alpha)$. Let $T = \Psi_\alpha(D) \in \text{RSVT}(\alpha)$. Then $\Phi_\alpha(T) = D$.

2. Take $T \in \text{RSVT}(\alpha)$. Let $D = \Phi_\alpha(D) \in \text{KKD}(\alpha)$. Then $\Psi_\alpha(D) = T$.

We only establish the first statement using Lemma 67. The second statement can be proved similarly using Lemma 69 instead.

We prove by induction on $\max(\alpha)$. When $\max(\alpha) = 0$, $D = (\emptyset, \emptyset)$ and our claim is immediate.

Now assume $\max(\alpha) > 0$. Let $D = (K_1, G_1, d)$. First, we compute $\Psi_\alpha(D)$ using our recursive description. Let $t = \Psi_\gamma(d)$ where $\gamma = M(\alpha, K_1)$. Assume $G_1 = \{g_1 < g_2 < \cdots < g_{|G_1|}\}$. Let $D^0 = (K_1, \emptyset, t)$ and $\sharp_i(D^{i-1}) = (D^i, k_i) = ((K_1^i, \emptyset, t), k_i)$ for $i = 1, \ldots, |G_1|$. Then we know $D$ is sent to $T = (K_1^{[G_1]}, E_1, t) \in \text{RSVT}(\alpha)$, where $E_1 = \{k_1, \ldots, k_{|G_1|}\}$.

Now we compute $\Phi_\alpha(T)$ using our recursive description. Now write $E_1$ as $\{e_1 > \cdots > e_{|G_1|}\}$. After applying $\flat_{e_1}, \ldots, \flat_{e_{|G_1|}}$ on $(K_1^{[G_1]}, \emptyset, t)$, by Lemma 67, the resulting diagram pair is $(K_1, \emptyset, t)$ and $G_1$ consists of the output numbers. Finally, by the inductive hypothesis, $\Phi_\gamma(t) = d$. Thus, $\Phi_\alpha(T) = (K_1, G_1, d) = D$. □

Now we have the desired weight-preserving bijection between $\text{KKD}(\alpha)$ and $\text{RSVT}(\alpha)$. We can claim the Ross-Yong conjecture is correct.

**Corollary 74.** The Lascoux polynomial indexed by $\alpha$, has a combinatorial formula with $\text{KKD}(\alpha)$, i.e.,

$$L_\alpha^{(\beta)} = \sum_{D \in \text{KKD}(\alpha)} \beta^{\text{ex}(D)} x^{\text{wt}(D)}.$$  

Acknowledgements

The authors thank Sami Assaf, Darij Grinberg, Brendon Rhoades and Mark Shimozono for helpful conversations.

References


