Pré-Publicações do Departamento de Matemática
Universidade de Coimbra
Preprint Number 21–56

SYMPLECTIC KEYS - TYPE C WILLIS' DIRECT WAY

JOÃO MIGUEL SANTOS

Abstract: It is known that the right and the left keys of a Kashiwara-Nakashima tableau in type C can be computed using the Lecouvey-Sheats symplectic *jeu de taquin*. Motivated by Willis' direct way of computing type A right and left keys, we also give a way of computing symplectic, right and left, keys without the use of symplectic *jeu de taquin*.

Keywords: Direct way, Symplectic keys, Right and left key maps, Symplectic *jeu de taquin*.
Math. Subject Classification (2020): 05E05, 05E10, 17B37.

1. Introduction

Symplectic tableaux [15, 8, 14] provide the monomial weight generators for the characters of the symplectic Lie algebra $sp(2n, \mathbb{C})$. Given a partition $\lambda \in \mathbb{Z}^n$, symplectic Kashiwara-Nakashima tableaux [14] of shape λ, on the alphabet $[\pm n]$ [14], a variation of De Concini tableaux [8], are endowed with a type C_n crystal structure $\mathfrak{B}^\lambda$ compatible with a plactic monoid and sliding algorithms, studied by Lecouvey in terms of crystal isomorphisms [18]. Let $W\lambda$ be the orbit of λ, where W is the type C_n Weyl group. Type C_n Demazure characters, κ_v, are indexed by vectors v in $W\lambda$ and can be seen as "partial" characters. Kashiwara [13] and Littelmann [20] have shown that they can be obtained by summing the monomial weights over certain subsets $\mathfrak{B}_v$ in the crystal $\mathfrak{B}^\lambda$, called Demazure crystals. Demazure crystals $\mathfrak{B}_v$ can be partitioned into Demazure atom crystals, $\widehat{\mathfrak{B}}_u$, where $u \in W\lambda$ runs in the Bruhat interval $\lambda \leq u \leq v$.

In type A_{n-1}, Lascoux and Schützenberger characterized key tableaux as semistandard Young tableaux (SSYT) with nested columns [17], and have used the *jeu de taquin* to define the *right key map* which sends a SSYT to a key tableau, called the *right key* of that SSYT. In each Demazure atom crystal there exists exactly one key tableau and the right key map detects the Demazure atom crystal that contains a given SSYT [17, Theorem 3.8]. By direct inspection of a Young tableau, Willis [26] has given an alternative

Received December 20, 2021.

algorithm to compute the right key tableau that does not require the use of *jeu de taquin.* Other methods to compute the type A right key map includes the alcove path model [19], semi skyline augmented fillings [21], and coloured vertex models [4]. For a complete overview in type A, see [4] and the references therein.

In type C_n, the symplectic key tableaux are characterized in [2, 23, 22, 12]. They are the unique tableaux in $\mathfrak{B}^\lambda$ whose weight is in $W\lambda$, and, for each one, there is exactly one Demazure atom crystal indexed by the corresponding weight. Using the Lecouvey-Sheats symplectic *jeu de taquin*, a right key map is given, in [23, 22, 12], to send a Kashiwara-Nakashima tableau T to its right key tableau $K_+(T)$, that detects the Demazure atom crystal which contains T. They are also computed in the type C_n alcove path model [19], and in the coloured five vertex model [5]. In [5], it is also computed the right key for reverse King tableaux [15]. Henceforth, the symplectic Demazure character $\kappa_v(x)$ is expressed in terms of right keys [23]

$$\kappa_v(x) = \sum_{\substack{T\in\mathfrak{B}^\lambda \\ K_+(T)\leq K(v)}} x^{\mathrm{wt}T},$$

where $K(v)$ is the key tableau of shape λ and weight v, $x^{\mathrm{wt}T}$ is the weight monomial corresponding to T, with $\mathrm{wt}T \in \mathbb{Z}^n$ the weight of T, and $K_+(T) \leq K(v)$ means entrywise comparison.

There is also a notion of *left key map*, which is a dual version of the aforementioned right key map. It can be used to identify which tableaux go in each *opposite* Demazure crystal [7]. Given $v \in W\lambda$, the opposite Demazure crystal $\mathfrak{B}^{op}_{-v}$ is the image of Demazure crystal $\mathfrak{B}_v \subseteq \mathfrak{B}^\lambda$ by the the Schützenberger-Lusztig involution on $\mathfrak{B}^\lambda$ [23, Proposition 64]. It can be partitioned into opposite Demazure atom crystals, $\widehat{\mathfrak{B}}^{op}_u$, where $u \in W\lambda$ runs in the Bruhat interval $-v \leq u \leq -\lambda$. Given a Demazure crystal and its opposite, for each tableau weight in the Demazure crystal there is a symmetric tableau weight in the opposite Demazure crystal.

Jacon and Lecouvey have suggested, in [12], that Willis' method [26] to compute right and left keys in type A_{n-1} should be adaptable to type C_n. Motivated by Willis' direct inspection [26], we create an alternative algorithm, based on a Kashiwara-Nakashima tableau, for the symplectic right key map, and for the symplectic left key map, that does not use the symplectic *jeu de taquin.* Due to the added technicality of the symplectic *jeu de taquin* compared to the one for SSYT, Willis' *earliest weakly increasing* *subsequence* will fail to predict what gets slid during the Lecouvey-Sheats symplectic *jeu de taquin*. Instead we need a way to calculate, without the use of *jeu de taquin*, what would appear in each column if we were to swap its length with the previous column length via *jeu de taquin*. The role of Willis' sequences will be replaced by our matchings (see Section 5). In type A, these kind of matches were used earlier [1, 19].

The paper is organized in eight sections as follows. In Section 2, we discuss the type C Kashiwara-Nakashima tableaux and the symplectic *jeu de taquin*. Section 3 recalls the symplectic key tableaux, and define right and left key maps in terms of Demazure crystal and opposite Demazure crystal. We conclude this section by attaching a type A *cocrystal* to each vertex of the type C_n crystal of Kashiwara-Nakashima tableaux $\mathfrak{B}^\lambda$. Given a Kashiwara-Nakashima tableau in $\mathfrak{B}^\lambda$, the vertices of its cocrystal are Kashiwara-Nakashima skew tableaux obtained by symplectic *jeu de taquin* on consecutive columns. This cocrystal is motivated by Lascoux' double crystal graph in type A [16], and by Heo-Kwon work in [11], where Schützenberger *jeu de taquin* slides are used as crystal operators for $\mathfrak{sl}_2$. Our construction builds on Heo-Kwon work [11, Lemma 2.3, Lemma 2.4] and uses the dual RSK correspondence [10]. More precisely, to each vertex of a straight shaped Kashiwara-Nakashima tableau in type C_n, it is attached a type A cocrystal, where, in our case, symplectic *jeu de taquin* slides on consecutive columns are used as crystal operators. These cocrystals are type A crystals and contain all the needed information to compute the right and left key maps, as shown in the next sections. Actually, their key skew tableaux (Definition 3.11) provide the information to compute left and right key maps of a Kashiwrara-Nakashima tableau. Section 4 recalls the right key map via symplectic *jeu de taquin* in [23], as a preparation for the alternative method. In Section 5, we give an algorithm for computing the symplectic right key map that does not require the *jeu de taquin*, and prove that it returns the same object as the previous method. Sections 6 and 7, in parallel to sections 4 and 5, recall the left key map via symplectic *jeu de taquin* and show a way of computing them that does not require the *jeu de taquin*. Finally, in Section 8, we end with an illustrative example of our new algorithm and the one based on the Lecouvey-Sheats *jeu de taquin*.

An extended abstract [24] of this paper was accepted in the Proceedings of the 33rd Conference on Formal Power Series and Algebraic Combinatorics, 2021.

2. Type C Kashiwara-Nakashima tableaux and *jeu de taquin*

We recall the symplectic tableaux introduced by Kashiwara and Nakashima to label the vertices of the type C_n crystal graphs [14]. Fix $n \in \mathbb{N}_{>0}$. Define the sets $[n] = \{1, \dots, n\}$ and $[\pm n] = \{1, \dots, n, \overline{n}, \dots, \overline{1}\}$ where $\overline{i}$ is just another way of writing $-i$, hence $\overline{\overline{i}} = i$. In the second set we will consider the following order of its elements: $1 < \cdots < n < \overline{n} < \cdots < \overline{1}$ instead of the usual order. A vector $\lambda = (\lambda_1, \dots, \lambda_n) \in \mathbb{Z}^n$ is a partition of $|\lambda| = \sum_{i=1}^{n} \lambda_i$ if $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$. The *Young diagram* of shape λ, in English notation, is an array of boxes (or cells), left justified, in which the i-th row, from top to bottom, has λ_i boxes. We identify a partition with its Young diagram. For example, the Young diagram of shape $\lambda = (2, 2, 1)$ is $\ydiagram{2,2,1}$. Given μ and ν two partitions with $\nu \leq \mu$ entrywise, we write $\nu \subseteq \mu$. The Young diagram of shape μ/ν is obtained after removing the boxes of the Young diagram of ν from the Young diagram of μ. For example, the Young diagram of shape $\mu/\nu = (2, 2, 1)/(1, 0, 0)$ is $\ydiagram{1+1,2,1}$,.

Let $\nu \subseteq \mu$ be two partitions and A a completely ordered alphabet. A *semistandard Young tableau* (SSYT) of skew shape μ/ν, on the alphabet A, is a filling of the diagram μ/ν with letters from A, such that the entries are strictly increasing, from top to bottom, in each column and weakly increasing, from left to right, in each row. When $|\nu| = 0$ then we obtain a semistandard Young tableau of straight shape μ. Denote by $\mathcal{SSYT}(\mu/\nu, A)$ the set of all skew SSYT T of shape μ/ν, with entries in A. In particular, when $|v| = 0$ we write $\mathcal{SSYT}(\mu, A)$ and when $A = [n]$ we write $\mathcal{SSYT}(\mu/\nu, n)$.

When considering tableaux with entries in $[\pm n]$, it is usual to have some extra conditions besides being semistandard. We will use a family of tableaux known as *Kashiwara-Nakashima* tableaux. From now on we consider tableaux on the alphabet $[\pm n]$.

A *column* is a strictly increasing sequence of numbers (or letters) in $[\pm n]$ and it is usually displayed vertically. The height of a column is the number of letters in it. A column is said to be *admissible* if the following *one column condition* (1CC) holds for that column:

Definition 2.1 (1CC)**.** Let C be a column. The $1CC$ holds for C if for all pairs i and $\overline{i}$ in C, where i is in the a-th row counting from the top of the column, and $\bar{\imath}$ in the b-th row counting from the bottom, we have $a+b \leq i$. Equivalently, for all pairs i and $\bar{\imath}$ in C, the number $N(i)$ of letters x in C such that $x \leq i$ or $x \geq \bar{\imath}$ satisfies $N(i) \leq i$.

If a column C satisfies the $1CC$ then C has at most n letters. If $1CC$ doesn't hold for C we say that C *breaks the* $1CC$ *at* z, where z is the minimal positive integer such that z and $\bar{z}$ exist in C and there are more than z numbers in C with absolute value less or equal than z.

Example 2.2. The column $\begin{array}{|c|}\hline 1\\\hline 2\\\hline \bar{1}\\\hline\end{array}$ breaks the $1CC$ at 1, and $\begin{array}{|c|}\hline 2\\\hline 3\\\hline \bar{3}\\\hline\end{array}$ is an admissible column.

The following definition states conditions to when C can be *split*:

Definition 2.3. Let C be a column and let $I = \{z_1 > \cdots > z_r\}$ be the set of unbarred letters z such that the pair $(z, \bar{z})$ occurs in C. The column C can be split when there exists a set of r unbarred letters $J = \{t_1 > \cdots > t_r\} \subseteq [n]$ such that:

1. t_1 is the greatest letter of $[n]$ satisfying $t_1 < z_1$, $t_1 \notin C$, and $\overline{t_1} \notin C$,
2. for $i = 2, \ldots, r$, we have that t_i is the greatest letter of $[n]$ satisfying $t_i < \min(t_{i-1}, z_i)$, $t_i \notin C$, and $\overline{t_i} \notin C$.

The $1CC$ holds for a column C (or C is admissible) if and only if C can be split [25, Lemma 3.1]. If C can be split then we define *right column* of C, rC, and the *left column* of C, ℓC, as follows:

1. rC is the column obtained by changing in C, $\overline{z_i}$ into $\overline{t_i}$ for each letter $z_i \in I$ and by reordering if necessary,
2. ℓC is the column obtained by changing in C, z_i into t_i for each letter $z_i \in I$ and by reordering if necessary.

If C is admissible then $\ell C \leq C \leq rC$ by entrywise comparison, where ℓC has the same barred part as C and rC the same unbarred part. If C doesn't have symmetric entries, then C is admissible and $\ell C = C = rC$. In the next definition we give conditions for a column C to be *coadmissible.*

Definition 2.4. We say that a column C is coadmissible if for every pair i and $\bar{\imath}$ on C, where i is on the a-th row counting from the top of the column, and $\bar{\imath}$ on the b-th row counting from the top, then $b - a \leq n - i$. Equivalently, for every pair i and $\bar{\imath}$ on C, the number $N^*(i)$ of letters x in C such that $i \leq x \leq \bar{\imath}$ satisfies $N^*(i) \leq n - i + 1$.

Unlike in Definition 2.1, in the last definition $\boldsymbol{b}$ is counted from the top of the column.

Definition 2.5. Let $\boldsymbol{C}$ be a column and let $\boldsymbol{I} = \{z_1 > \cdots > z_r\}$ be the set of unbarred letters $\boldsymbol{z}$ such that the pair $(\boldsymbol{z}, \overline{\boldsymbol{z}})$ occurs in $\boldsymbol{C}$. The column $\boldsymbol{C}$ is coadmissible if and only if there exists a set of $\boldsymbol{r}$ unbarred letters $\boldsymbol{H} = \{h_1 > \cdots > h_r\} \subseteq [\boldsymbol{n}]$ such that:

1. h_r is the smallest letter of $[\boldsymbol{n}]$ satisfying $h_r > z_r$, $h_r \notin \boldsymbol{C}$, and $\overline{h_r} \notin \boldsymbol{C}$,
2. for $i = r - 1, \ldots, 1$, we have that h_i is the smallest letter of $[\boldsymbol{n}]$ satisfying $h_i > \max(h_{i+1}, z_i)$, $h_i \notin \boldsymbol{C}$, and $\overline{h_i} \notin \boldsymbol{C}$.

Given an admissible column $\boldsymbol{C}$, consider the map

$$\Phi : C \mapsto C^*$$

that sends $\boldsymbol{C}$ to the column $\boldsymbol{C}^*$ of the same size in which the unbarred entries are taken from $\ell \boldsymbol{C}$ and the barred entries are taken from $r\boldsymbol{C}$.

Lemma 2.6. *Let $\boldsymbol{C}$ be an admissible column on the alphabet $[\pm \boldsymbol{n}]$, and $\boldsymbol{I}$ and $\boldsymbol{J}$ the sets in Definition 2.3. The entries $\boldsymbol{x}$ (barred or unbarred) of $\Phi(\boldsymbol{C})$ are such that*

(1) $x \in \Phi(C)$ *and* $\overline{x} \notin \Phi(C)$ *if and only if* $x \in C$ *and* $\overline{x} \notin C$.
(2) $x, \overline{x} \in \Phi(C)$ *if and only if* $x \in J$ *or* $\overline{x} \in J$.

Equivalently, the set of entries in $\Phi(\boldsymbol{C})$ is $(J \cup \overline{J} \cup C) \setminus (I \cup \overline{I})$.

Henceforth, $\Phi(C) = C$ if and only if $I = \emptyset$ (hence $J = \emptyset$), that is, $\boldsymbol{C}$ does not have symmetric entries.

The *column* $\Phi(\boldsymbol{C})$ is a coadmissible column and the algorithm to form $\Phi(\boldsymbol{C})$ from $\boldsymbol{C}$ is reversible [18, Section 2.2]. In particular, every column on the alphabet $[\boldsymbol{n}]$ is simultaneously admissible and coadmissible. The map Φ is a bijection between admissible and coadmissible columns of the same height on the alphabet $[\pm \boldsymbol{n}]$.

Example 2.7. Let $C = \begin{array}{|c|}\hline 2 \\ \hline 4 \\ \hline \overline{2} \\ \hline\end{array}$ be an admissible column, so it can be split. Then $\ell C = \begin{array}{|c|}\hline 1 \\ \hline 4 \\ \hline \overline{2} \\ \hline\end{array}$ and $rC = \begin{array}{|c|}\hline 2 \\ \hline 4 \\ \hline \overline{1} \\ \hline\end{array}$. So $\Phi(C) = \begin{array}{|c|}\hline 1 \\ \hline 4 \\ \hline \overline{1} \\ \hline\end{array}$ is coadmissible. $\boldsymbol{C}$ is also coadmissible and $\Phi^{-1}(C) = \begin{array}{|c|}\hline 3 \\ \hline 4 \\ \hline \overline{3} \\ \hline\end{array}$.

Let T be a skew tableau with all of its columns admissible. The *split form* of a skew tableau T, $spl(T)$, is the skew tableau obtained after replacing each column C of T by the two columns $\ell C\, rC$. The tableau $spl(T)$ has double the amount of columns of T.

A semistandard skew tableau T is a *Kashiwara-Nakashima (KN) skew tableau* if its split form is a semistandard skew tableau. We define $\mathcal{KN}(\mu/\nu, n)$ to be the set of all KN tableaux of shape μ/ν in the alphabet $[\pm n]$. When $\nu = 0$, we obtain $\mathcal{KN}(\mu, n)$. The weight of T is a vector whose i-th entry is the number of i's minus the number of $\bar{i}$.

Example 2.8. The split of the tableau $T = \begin{array}{|c|c|}\hline 2 & 2 \\ \hline 3 & 3 \\ \hline \bar{3} & \multicolumn{1}{c}{} \\ \cline{1-1} \end{array}$ is the tableau $spl(T) = \begin{array}{|c|c|c|c|}\hline 1 & 2 & 2 & 2 \\ \hline 2 & 3 & 3 & 3 \\ \hline \bar{3} & \bar{1} & \multicolumn{2}{c}{} \\ \cline{1-2} \end{array}$. Hence $T \in \mathcal{KN}((2,2,1),3)$ and weight $\mathrm{wt}T = (0,2,1)$.

If T is a tableau without symmetric entries in any of its columns, i.e., for all $i \in [n]$ and for all columns C in T, i and $\bar{i}$ do not appear simultaneously in the entries of C, then in order to check whether T is a KN tableau it is enough to check whether T is semistandard in the alphabet $[\pm n]$. In particular $\mathcal{SSYT}(\mu/\nu, n) \subseteq \mathcal{KN}(\mu/\nu, n)$.

2.1. Symplectic *jeu de taquin*. Lecouvey-Sheats symplectic *jeu de taquin* (SJDT) [18, 25] is a procedure on KN skew tableaux, compatible with *Knuth equivalence* (or plactic equivalence on words over the alphabet $[\pm n]$) [18], that allows us to change the shape of a tableau and to rectify it. To explain how the SJDT behaves, we need to look how it works on 2-column $C_1 C_2$ KN skew tableaux. A skew tableau is *punctured* if one of its box contains the symbol $*$ called the *puncture*. A punctured column is admissible if the column is admissible when ignoring the puncture. A punctured skew tableau is admissible if its columns are admissible and the rows of its split form are weakly increasing ignoring the puncture. Let T be a punctured skew tableau with two columns C_1 and C_2 with the puncture in C_1. In that case, the puncture splits into two punctures in $spl(T)$, and ignoring the punctures, $spl(T)$ must be semistandard. Let α be the entry under the puncture of

rC_1, and β the entry to the right of the puncture of rC_1.

$$spl(T) = \ell C_1 r C_1 \ell C_2 r C_2 = \begin{array}{|c|c|c|c|}\hline \cdots & \cdots & \cdots & \cdots \\ \hline * & * & \beta & \cdots \\ \hline \cdots & \alpha & \cdots & \cdots \\ \hline \cdots & \cdots \\ \cline{1-2} \end{array},$$

where α or β may not exist. The elementary steps of SJDT are the following:

A. If $\alpha \leq \beta$ or β does not exist, then the puncture of T will change its position with the cell beneath it. This is a vertical slide.

B. If the slide is not vertical, then it is horizontal. So we have $\alpha > \beta$ or α does not exist. Let C_1' and C_2' be the columns obtained after the slide. We have two subcases, depending on the sign of β:

1. If β is barred, we are moving a barred letter, β, from ℓC_2 to the punctured box of rC_1, and the puncture will occupy β's place in ℓC_2. Note that ℓC_2 has the same barred part as C_2 and that rC_1 has the same barred part as $\Phi(C_1)$. Looking at T, we will have an horizontal slide of the puncture, getting $C_2' = C_2 \setminus \{\beta\} \sqcup \{*\}$ and $C_1' = \Phi^{-1}(\Phi(C_1) \setminus * \sqcup \{\beta\})$. In a sense, β went from C_2 to $\Phi(C_1)$.

2. If β is unbarred, we have a similar story, but this time β will go from $\Phi(C_2)$ to C_1, hence $C_1' = C_1 \setminus * \cup \{\beta\}$ and $C_2' = \Phi^{-1}(\Phi(C_2) \setminus \{\beta\} \sqcup *)$. Although in this case it may happen that C_1' is no longer admissible. In this situation, if the 1CC breaks at i, we erase both i and $\bar{\imath}$ from the column and remove a cell from the bottom and from the top column, and place all the remaining cells orderly with respect to their entries.

Applying successively elementary SJDT slides, eventually, the puncture will be a cell such that α and β do not exist. In this case we redefine the shape to not include this cell and the *jeu de taquin* ends.

Given an admissible tableau T of shape μ/ν, a box of the diagram of shape ν such that boxes under it and to the right are not in that shape is called an inner corner of μ/ν. An outside corner is a box of μ such that boxes under it and to the right are not in the shape μ. The rectification of T consists in playing the SJDT until we get a tableau of shape λ, for some partition λ. More precisely, apply successively elementary SJDT steps to T until each cell of ν becomes an outside corner. At the end, we obtain a KN tableau for some shape λ. The rectification is independent of the order in which the inner corners of ν are filled [18, Corollary 6.3.9].

Example 2.9. Consider the KN skew tableau $T = \begin{array}{cc} & 2 \\ 1 & 3 \\ 2 & \bar{1} \end{array}$. Let C_1 and C_2 be the first and second columns of T. To rectify T via symplectic *jeu taquin*, one creates a puncture in the inner corner of T and, by splitting, one obtains $\begin{array}{cccc} * & * & 2 & 2 \\ 1 & 1 & 3 & 3 \\ 2 & 2 & \bar{1} & \bar{1} \end{array}$. So, the first two slides are vertical, obtaining $\begin{array}{cccc} 1 & 1 & 2 & 2 \\ 2 & 2 & 3 & 3 \\ * & * & \bar{1} & \bar{1} \end{array}$. Finally, we do an horizontal slide, of type **B.1**, in which we take $\bar{1}$ from C_2, and add it to the coadmissible column $\Phi(C_1)$. That is, $C_2' = (C_2 \cup \{*\}) \setminus \bar{1}$ and $C_1' = \Phi^{-1}((\Phi(C_1) \setminus \{*\}) \cup \bar{1})$, obtaining the tableau $\begin{array}{cc} 2 & 2 \\ 3 & 3 \\ \bar{3} & \end{array}$.

Let T be a skew tableau of shape μ/ν. Consider a punctured box that can be added to μ, so that $\mu \cup \{*\}$ is a valid shape. The SJDT is reversible, meaning that we can move $*$, the empty cell outside of μ, to the inner shape ν of the skew tableau T, simultaneously increasing both the inner and outer shapes of T by one cell. The slides work similarly to the previous case: the vertical slide means that an empty cell is going up and an horizontal slide means that an entry goes from $\Phi(C_1)$ to C_2 or from C_1 to $\Phi(C_2)$, depending on whether the slid entry is barred or not, respectively. We will also call the *reverse jeu de taquin* as SJDT. In the next sections we will be mostly dealing with the *reverse jeu de taquin*. Consider the following examples, each containing a tableau and a punctured box that will be slid to its inner shape: $\begin{array}{ccc} \square & \square & * \\ 1 & \bar{1} & \\ 2 & & \end{array} \mapsto \begin{array}{ccc} \square & \square & \square \\ 1 & \bar{1} & \\ 2 & & \end{array}$; $\begin{array}{cc} \square & \square \\ 1 & \bar{1} \\ 2 & * \end{array} \mapsto \begin{array}{cc} \square & \square \\ \square & 2 \\ 2 & \bar{2} \end{array}$.

If a tableau with columns C_1 and C_2 does not have symmetric entries then the SJDT applied to C_1C_2 coincides with the *jeu de taquin* known for SSYT. In sections 4 and 6, we use SJDT to swap lengths of consecutive columns in a skew tableau, to obtain skew tableaux Knuth related to a straight tableau, which is minimal for the number of cells within its Knuth class. Recall that in the elementary step $B.2$ it is possible to lose cells. If we do a reverse elementary step $B.2$ that results in having two more cells in the skew tableau, we would have to start by adding two symmetric entries to an admissible column, making it non admissible [18, Lemma 3.2.3], and then slide an unbarred cell to the column to its right. For instance, consider the following reverse elementary step $B.2$ ($\equiv$ denotes type C_n Knuth equivalence [18, Definition

3.2.1]):

$$\begin{array}{|c|c|}\hline 1 & * \\ \hline 2 & \multicolumn{1}{c}{} \\ \cline{1-1}\end{array} \equiv \begin{array}{|c|c|}\hline 1 & * \\ \hline 2 & \multicolumn{1}{c}{} \\ \cline{1-1} 3 & \multicolumn{1}{c}{} \\ \cline{1-1} \bar{3} & \multicolumn{1}{c}{} \\ \cline{1-1}\end{array} \equiv \begin{array}{|c|c|}\hline * & 1 \\ \hline 2 & \multicolumn{1}{c}{} \\ \cline{1-1} 3 & \multicolumn{1}{c}{} \\ \cline{1-1} \bar{3} & \multicolumn{1}{c}{} \\ \cline{1-1}\end{array}$$

.

The first and last skew tableaux are Knuth equivalent, but the middle tableau is not a KN skew tableau. The three semistandard tableaux are Knuth equivalent column words, via te contractor/dilator Knuth relation [18, Definition **3.2.1**].

Hence, a reverse elementary step $B.2$ that results in having more cells in the skew tableau has to be forced, since we have to start by forcing the existence of a non admissible column. This means that if we start with a minimal skew tableau, that is, a skew-tableau with the number of cells of its rectification, we can play SJDT, or its reverse, without ever incur in a loss/gain of boxes.

3. Type C_n crystals, Demazure crystals, their opposite and cocrystals

In this section we review Demazure and opposite Demazure crystals of a type C_n crystal $\mathfrak{B}^\lambda$ of KN tableaux, where λ is a partition with at most n parts, and the corresponding relation with right and left key maps. Motivated by Lascoux's double crystal graph construction in type A [16], and by Heo-Kwon work in [11] where Schützenberger *jeu de taquin* slides are used as crystal operators for $\mathfrak{sl}_2$, the cocrystal of each KN tableau in the type C_n crystal $\mathfrak{B}^\lambda$ is introduced. These cocrystals contain all the needed information to compute right and left keys of a tableau in the type C_n crystal $\mathfrak{B}^\lambda$ and refine our previous construction of te symplectic key maps in [23] based on the symplectic *jeu de taquin.*

3.1. Kashiwara crystal. Let V be an Euclidean space with inner product $\langle\cdot,\cdot\rangle$. Fix a root system Φ with simple roots $\{\alpha_i \mid i \in I\}$ where I is an indexing set and a weight lattice $\Lambda \supseteq \mathbb{Z}\text{-span}\{\alpha_i \mid i \in I\}$. A *Kashiwara crystal* of type Φ is a nonempty set $\mathfrak{B}$ together with maps [6]:

$$e_i, f_i : \mathfrak{B} \to \mathfrak{B} \sqcup \{0\} \quad \varepsilon_i, \varphi_i : \mathfrak{B} \to \mathbb{Z} \sqcup \{-\infty\} \quad \mathrm{wt} : \mathfrak{B} \to \Lambda$$

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

(1) if $a, b \in \mathfrak{B}$ then $e_i(a) = b \Leftrightarrow f_i(b) = a$. In this case, we also have $\mathrm{wt}(b) = \mathrm{wt}(a) + \alpha_i$, $\varepsilon_i(b) = \varepsilon_i(a) - 1$ and $\varphi_i(b) = \varphi_i(a) + 1$;
(2) for all $a \in \mathfrak{B}$, we have $\varphi_i(a) = \langle \mathrm{wt}(a), \frac{2\alpha_i}{\langle \alpha_i, \alpha_i \rangle} \rangle + \varepsilon_i(a)$.

The crystals we deal with are seminormal [6], i.e., $\varphi_i(a) = \max\{k \in \mathbb{Z} \geq 0 \mid f_i^k(a) \neq 0\}$ and $\varepsilon_i(a) = \max\{k \in \mathbb{Z} \geq 0 \mid e_i^k(a) \neq 0\}$. An element $u \in \mathfrak{B}$ such that $e_i(u) = 0$ for all $i \in I$ is called a *highest weight element*. A *lowest weight element* is an element $u \in \mathfrak{B}$ such that $f_i(u) = 0$ for all $i \in I$. We associate with $\mathfrak{B}$ a coloured oriented graph with vertices in $\mathfrak{B}$ and edges labeled by $i \in I$: $b \xrightarrow{i} b'$ iff $b' = f_i(b)$, $i \in I$, $b, b' \in \mathfrak{B}$. This is the *crystal graph* of $\mathfrak{B}$.

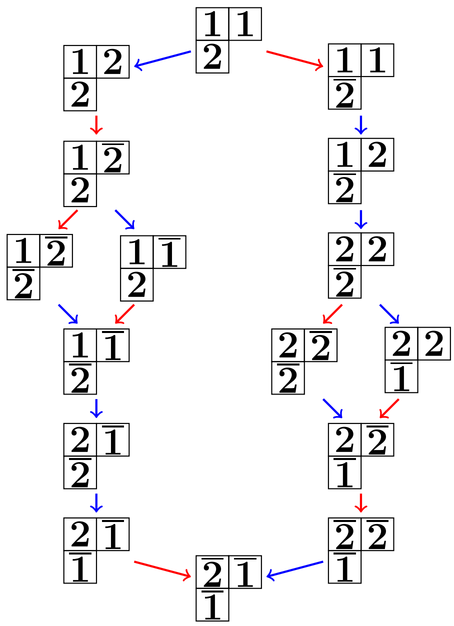
The set $\mathcal{KN}(\lambda, n)$ (resp. $\mathcal{SSYT}(\lambda, n)$) is endowed with a Kashiwara crystal structure of type C_n (resp. A_{n-1}) [13, 18]. The crystal operators are fully characterized on words on the alphabet $[\pm n]$ (resp. $[n]$) or on KN skew tableaux (resp. skew SSYT) via a signature rule [6]. Observe that $\mathcal{SSYT}(\lambda, n)$ is a subcrystal of $\mathfrak{B}^\lambda = \mathcal{KN}(\lambda, n)$.

The Weyl group W of type C_n, known as hyperocthaedral group, is the Coxeter group B_n ($2^n n!$ elements) generated by the involutions $s_1, \dots, s_n$ subject to relations $(s_i s_{i+1})^3 = 1, 1 \leq i \leq n-2$; $(s_{n-1}s_n)^4 = 1$; $(s_i s_j)^2 = 1$, $1 \leq i < j \leq n$, $|i - j| > 1$. The elements of W act on $z = (z_1, \dots, z_n) \in \mathbb{Z}^n$ by $s_i z := (z_1, \dots, z_{i+1}, z_i, \dots, z_n)$, $1 \leq i \leq n-1$ and $s_n z = (z_1, \dots, \overline{z_n})$.

A *key tableau* of shape λ, in type C_n is a KN tableau in $\mathcal{KN}(\lambda, n)$, in which the set of elements of each column, left to right, is contained in the set of elements of the previous column, if any, and the letters i and $\bar{i}$ do not appear simultaneously as entries, for any $i \in [n]$.

The key tableaux in type C_n are also the KN tableaux of shape λ whose weight is in $B_n\lambda$, the B_n-orbit of λ. For each element of $B_n\lambda$ there is exactly one key tableau of shape λ with that weight. The key tableaux in $\mathcal{KN}(\lambda, n)$ are thus characterized by their weight $\alpha\lambda$, for all $\alpha \in W$, and thereby denoted $K(\alpha\lambda)$. The orbit of $K(\lambda)$, the highest weight element of $\mathfrak{B}^\lambda$, under the action of the Weyl group W, is defined to be $O(\lambda) = \{K(\alpha\lambda) : \alpha \in W\}$. In particular, $K(w_0\lambda) = K(-\lambda)$, with w_0 the longest element of B_n, is the lowest weight element of the type C_n crystal $\mathfrak{B}^\lambda$.

Example 3.1. Here we have the type C_2 crystal graph $\mathcal{KN}((2,1), 2)$ containing the A_1 crystal $\mathcal{SSYT}((2,1), 2)$ as a subcrystal:



3.2. Demazure crystal. Let $\boldsymbol{\lambda}$ be a partition.

Given a subset $\boldsymbol{X}$ of $\mathfrak{B}^{\boldsymbol{\lambda}}$, consider the operator $\mathfrak{D}_{\boldsymbol{i}}$ on $\boldsymbol{X}$, with $\boldsymbol{i} \in [\boldsymbol{n}]$ defined by $\mathfrak{D}_{\boldsymbol{i}}\boldsymbol{X} = \{\boldsymbol{x} \in \mathfrak{B}^{\boldsymbol{\lambda}} \mid \boldsymbol{e}_{\boldsymbol{i}}^{\boldsymbol{k}}(\boldsymbol{x}) \in \boldsymbol{X}$ for some $\boldsymbol{k} \geq \mathbf{0}\}$ [6]. If $\boldsymbol{v} = \boldsymbol{\sigma}\boldsymbol{\lambda}$ where $\boldsymbol{\sigma} = \boldsymbol{s}_{\boldsymbol{i}_{\boldsymbol{\ell}}} \cdots \boldsymbol{s}_{\boldsymbol{i}_1} \in \boldsymbol{B}_{\boldsymbol{n}}$ is a reduced word, we define the *Demazure crystal* $\mathfrak{B}_{\boldsymbol{v}}$ to be

$$\mathfrak{B}_{\boldsymbol{v}} = \mathfrak{D}_{\boldsymbol{i}_{\boldsymbol{\ell}}} \cdots \mathfrak{D}_{\boldsymbol{i}_1}\{\boldsymbol{K}(\boldsymbol{\lambda})\}.$$

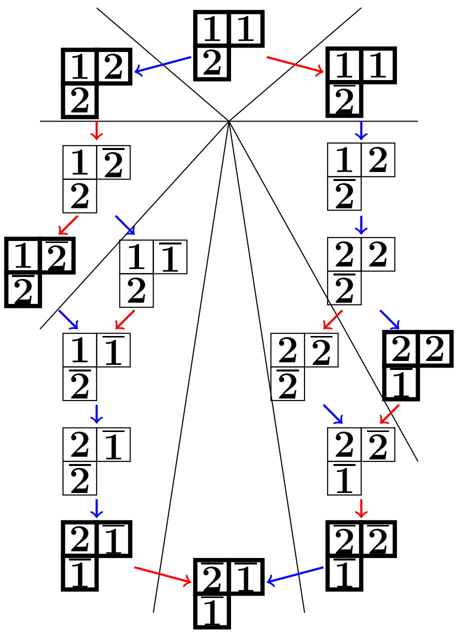
This definition is independent of the reduced word for $\boldsymbol{\sigma}$ [6, Theorem 13.5].

From [3, Proposition 2.5.1], if $\boldsymbol{\rho} \leq \boldsymbol{\sigma}$, for the Bruhat order of $\boldsymbol{B}_{\boldsymbol{n}}$, then $\boldsymbol{u} = \boldsymbol{\rho}\boldsymbol{\lambda} \leq \boldsymbol{v}$. Since $\boldsymbol{e}_{\boldsymbol{i}}^{\mathbf{0}}(\boldsymbol{x}) = \boldsymbol{x}$, if $\boldsymbol{\rho} \leq \boldsymbol{\sigma}$ then $\mathfrak{B}_{\boldsymbol{u}} \subseteq \mathfrak{B}_{\boldsymbol{v}}$. Thus we define the *Demazure atom crystal* $\widehat{\mathfrak{B}}_{\boldsymbol{v}}$ to be

$$\widehat{\mathfrak{B}}_{\boldsymbol{v}} = \mathfrak{B}_{\boldsymbol{v}} \setminus \bigcup_{\boldsymbol{u}<\boldsymbol{v}} \mathfrak{B}_{\boldsymbol{u}} = \mathfrak{B}_{\boldsymbol{v}} \setminus \bigcup_{\boldsymbol{K}(\boldsymbol{u})<\boldsymbol{K}(\boldsymbol{v})} \mathfrak{B}_{\boldsymbol{u}}.$$

In the next example, every Demazure atom crystal contains exactly one key tableau. The right key map sends each tableau of $\mathfrak{B}^{\boldsymbol{\lambda}}$ to the unique key tableau living in the Demazure atom crystal that contains the given tableau. The right key of a tableau $\boldsymbol{T}$ is a key tableau of the same shape as $\boldsymbol{T}$, entrywise "slightly" bigger than $\boldsymbol{T}$.

Example 3.2. Recall the type $\boldsymbol{C}_{\mathbf{2}}$ crystal graph associated to the partition $\boldsymbol{\lambda} = (\mathbf{2}, \mathbf{1})$:



The crystal is split into $|B_2(2,1)| = 8$ parts, the number of elements of the B_2-orbit of $(2,1)$. Each part is a Demazure atom crystal and contains exactly one symplectic key tableau in $O(\lambda)$, drawn with a thick line, so we can identify each part with the weight of that key tableau, which is a vector in the B_2-orbit of $(2,1)$.

3.3. Opposite Demazure crystal. Let λ be a partition. Analogously to the previous case, we start by creating an opposing operator $\mathfrak{D}_i^{op}$ on X, with $i \in [n]$ defined by $\mathfrak{D}_i^{op} X = \{x \in \mathfrak{B}^\lambda \mid f_i^k(x) \in X \text{ for some } k \geq 0\}$. If $v = \sigma\lambda$ where $\sigma = s_{i_\ell} \cdots s_{i_1} \in B_n$ is a reduced word, we define the *opposite Demazure crystal* $\mathfrak{B}_{-v}^{op}$ to be

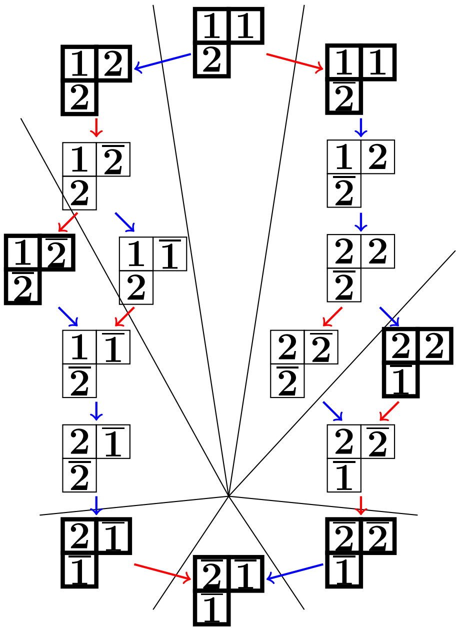
$$\mathfrak{B}_{-v}^{op} = \mathfrak{D}_{i_\ell}^{op} \cdots \mathfrak{D}_{i_1}^{op} \{K(-\lambda)\}.$$

We define the *opposite Demazure atom crystal* $\widehat{\mathfrak{B}}_v$ to be

$$\widehat{\mathfrak{B}}_{-v}^{op} = \mathfrak{B}_{-v}^{op} \setminus \bigcup_{-u > -v} \mathfrak{B}_{-u}^{op} = \mathfrak{B}_{-v}^{op} \setminus \bigcup_{K(-u) > K(-v)} \mathfrak{B}_{-u}^{op}.$$

In other words, the opposite Demazure crystal $\mathfrak{B}_{-v}^{op}$ is the image of $\mathfrak{B}_v$ by the Schützenberger-Lusztig involution [23, Section 5]. The evacuation algorithm, is a realization of the Schützenberger-Lusztig involution on $\mathcal{KN}(\lambda, n)$ [23, Algorithm 59], and the right key of a tableau is the left key of the evacuation of the same tableau [23, Proposition 64]. In particular, the tableau weights in $\mathfrak{B}_v$ and in $\mathfrak{B}_{-v}^{op}$ are symmetric.

Example 3.3. The C_2 crystal graph $\mathfrak{B}^{(2,1)}$ can also be split into opposite Demazure atom crystals:



Every opposite Demazure atom crystal contains exactly one key tableau. The left key map sends each tableau to the key tableau present in the opposite Demazure atom crystal that contains the tableau. The left key of a tableau $\boldsymbol{T}$ is a key tableau of the same shape as $\boldsymbol{T}$, entrywise "slightly" smaller than $\boldsymbol{T}$. In [23, 22] left and right keys are computed using the aforementioned *SJDT*.

3.4. Demazure characters and opposite Demazure characters. The Demazure characters, or key polinomials, and Demazure atoms can be seen as generating functions of the tableaux in Demazure crystals. Given $\boldsymbol{v} \in \boldsymbol{B_n \lambda}$:

$$\boldsymbol{\kappa}_v(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) = \sum_{\boldsymbol{T} \in \mathfrak{B}_v} \boldsymbol{x}^{\mathrm{wt}\boldsymbol{T}}; \ \widehat{\boldsymbol{\kappa}}_v(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) = \sum_{\boldsymbol{T} \in \widehat{\mathfrak{B}}_v} \boldsymbol{x}^{\mathrm{wt}\boldsymbol{T}},$$

and we can define, analogously, opposite Demazure characters and opposite Demazure atoms:

$$\boldsymbol{\kappa}^{op}_{-v}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) = \sum_{\boldsymbol{T} \in \mathfrak{B}^{op}_{-v}} \boldsymbol{x}^{\mathrm{wt}\boldsymbol{T}}; \ \widehat{\boldsymbol{\kappa}}^{op}_{-v}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) = \sum_{\boldsymbol{T} \in \widehat{\mathfrak{B}}^{op}_{-v}} \boldsymbol{x}^{\mathrm{wt}\boldsymbol{T}}.$$

Since the tableau weights in $\mathfrak{B}_v$ and in $\mathfrak{B}^{op}_{-v}$ are symmetric, we have the following result:

Corollary 3.4.

$$\boldsymbol{\kappa}_v(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) = \boldsymbol{\kappa}^{op}_{-v}(\boldsymbol{x}_1^{-1}, \dots, \boldsymbol{x}_n^{-1})$$

As a consequence, for instance, the type C_n Fu-Lascoux non-symmetric Cauchy kernel, given in [9], can be written as:

$$\begin{aligned}\frac{\prod_{1\le i<j\le n}(1-x_ix_j)}{\prod_{i,j=1}^{n}(1-x_iy_j)\prod_{i,j=1}^{n}(1-x_i/y_j)} &= \sum_{v\in\mathbb{N}^n}\hat{\kappa}_v(x_1,\dots,x_n)\kappa_{-v}(y_1,\dots,y_n)\\ &= \sum_{v\in\mathbb{N}^n}\hat{\kappa}_v(x_1,\dots,x_n)\kappa_v^{op}(y_1^{-1},\dots,y_n^{-1})\end{aligned}$$

In [7], Choi and Kwon, working in the Lakshmibai-Seshadri paths, prove the type A non-symmetric Cauchy kernel using a similar approach.

3.5. Dual RSK correspondence. In this subsection we work with SSYT. Given a tableau $T \in \mathcal{SSYT}(\lambda, n)$, with column decomposition $T = C_1C_2\cdots C_k$, its column reading word, $\mathrm{cr}(T)$, is the word obtained after concatenating all of its columns from right to left. Note that $\mathrm{cr}(T) = \mathrm{cr}(C_k)\cdots\mathrm{cr}(C_2)\mathrm{cr}(C_1)$.

Example 3.5. Given $T = \begin{array}{|c|c|c|}\hline 1&2&2\\\hline 2&3\\\cline{1-2} 4&4\\\cline{1-2}\end{array} \in \mathcal{SSYT}((3,2,2,0),4)$, the column reading of T is $\mathrm{cr}(T) = 2\,234\,124$.

Given a column reading word, we can recover the original tableau via *column insertion*: Let $w = w_1\cdots w_\ell$. We start with $i := 1$, $T = \emptyset$, the empty tableau, and $p = 1$.

(1) If w_i is bigger than all entries of C_p, Just add a cell to the column C_p with entry w_i. Else find $\alpha \in C_p$ the smallest entry of C_p bigger or equal than w_i. Then replace α by w_i in C_p and redefine $w_i := \alpha$, p:=p+1 and go to **(1)** (this is called a *bumping*).
(2) If $i \neq \ell$, then $i := i+1$, $p := 1$ and go to **(1)**. Else the algorithm ends.

Given $r \geq 1$, let E_n^r be the set of biwords without repeated biletters, in lexicographic order, $\begin{pmatrix}u\\v\end{pmatrix} \leq \begin{pmatrix}u'\\v'\end{pmatrix}$ if $u < u'$ or if $u = u'$ and $v \leq v'$, with the bottom word on the alphabet $[n]$, and the top word on the alphabet $[r]$. The set E_n^r can also be thought as the set of sequences of r columns, possibly some of them empty, on the alphabet $[n]$, where each pair of consecutive columns has maximum overlapping, and, in the case of two non-empty columns whose intermediate columns are empty, the top edge of the left column and the bottom edge of the right column are aligned. In particular, E_n^r has a subset identified with $\mathcal{SSYT}(\lambda, n)$, such that $\ell(\lambda') \leq r$, where $\ell(\lambda')$ is the length of λ', the conjugate partition of λ'. Given a tableau $T \in \mathcal{SSYT}(\lambda, n)$, we create a biword, without repeated biletters, whose bottom word is $cr(T)$ and in the top word we register in which column of T, counted from the right, was each letter of $\mathrm{cr}(T)$ read. Each biword will be an element of E_n^r, where $\ell(\lambda') \leq r$. In Example 3.5, the biword of T is

$$w = \begin{pmatrix} 1 & 2 & 2 & 2 & 3 & 3 & 3 \\ 2 & 2 & 3 & 4 & 1 & 2 & 4 \end{pmatrix} \in E_4^3, \text{ with } \ell(\lambda') = 3$$

.

The dual RSK, RSK^*, is a bijection [10, Section A.4.3] between E_n^r and pairs of SSYT's of conjugate shapes and lengths $\leq n$ and $\leq r$, respectively:

$$\begin{aligned} RSK^* : E_n^r &\to \bigsqcup_{\substack{\ell(\lambda) \leq n \\ \ell(\lambda') \leq r}} \mathcal{SSYT}(\lambda, n) \times \mathcal{SSYT}(\lambda', r) = \bigsqcup_{\substack{\ell(\lambda) \leq n \\ \ell(\lambda') \leq r \\ P \in \mathcal{SSYT}(\lambda, n)}} \{P\} \times \mathcal{SSYT}(\lambda', r) \\ w &\mapsto (P, Q). \end{aligned}$$

The bijection can be calculated in the following way:

Let $w = \begin{pmatrix} x_1 & x_2 & \ldots & x_m \\ y_1 & y_2 & \ldots & y_m \end{pmatrix}$. Then start with $i = 1$, $P = Q$ are empty tableaux.

(1) Column insert y_i into P.
(2) Add one cell to Q whose entry is x_i, in a position such that P and Q, with this new cell, have conjugate shapes.
(3) If $i \neq m$, then $i := i + 1$ and return to (1). Else the algorithm is finished.

Given a biword w, the first and second components of $RSK^*(w)$ are the P- symbol and the Q-symbol of w.

For instance, the biword $w = \begin{pmatrix} 1 & 2 & 2 & 2 & 3 & 3 & 3 \\ 2 & 2 & 3 & 4 & 1 & 2 & 4 \end{pmatrix}$ of T, in Example 3.5, is mapped to the pair

$$\left(T = \begin{array}{|c|c|c|} \hline 1 & 2 & 2 \\ \hline 2 & 3 \\ \cline{1-2} 4 & 4 \\ \cline{1-2} \end{array}, K(rev((3,2,2)')) = \begin{array}{|c|c|c|} \hline 1 & 2 & 2 \\ \hline 2 & 3 & 3 \\ \hline 3 \\ \cline{1-1} \end{array} \right).$$

More generally, given $T \in \mathcal{SSYT}(\lambda, n)$ with $\ell(\lambda') \leq r$, the dual RSK maps its biword, w, to the pair $(T, K(rev(\lambda')))$, where $rev(\lambda')$ is the vector λ' written backwards. Note that the weight of $K(\lambda')$ registers the column lengths of T, from right to left.

We also can compute RSK^* of a biword obtained from a skew SSYT. For instance, let $\tilde{T}$ be the skew SSYT $\begin{array}{|c|c|c|}\hline \multicolumn{2}{c|}{} & 2 \\ \hline 1 & 2 & 3 \\ \hline 2 & 4 & \multicolumn{1}{c}{} \\ \cline{1-2} 4 & \multicolumn{2}{c}{} \\ \cline{1-1} \end{array}$. Its biword is

$$\tilde{w} = \begin{pmatrix} 1 & 1 & 2 & 2 & 3 & 3 & 3 \\ 2 & 3 & 2 & 4 & 1 & 2 & 4 \end{pmatrix}.$$

Finally,

$$RSK^*(\tilde{w}) = \left(T = \mathrm{rect}(\tilde{T}), \tilde{Q} = \begin{array}{|c|c|c|}\hline 1 & 1 & 2 \\ \hline 2 & 3 & 3 \\ \hline 3 & \multicolumn{2}{c}{} \\ \cline{1-1} \end{array} \right),$$

where $\mathrm{rect}(\tilde{T})$ is the rectification of $\tilde{T}$ via SJDT. The weight of $\tilde{Q}$ records the column lengths of $\tilde{T}$ from right to left.

3.6. Cocrystal of SSYT's. Given $T \in \mathcal{SSYT}(\lambda, n)$ with $\ell(\lambda') \leq r$, we define the *cocrystal* of T, $\mathfrak{CB}^{\lambda'}(T)$, to be the $\mathfrak{gl}_r$-crystal,

$$\mathfrak{CB}^{\lambda'}(T) = (\mathrm{RSK}^*)^{-1}(\{T\}, \mathcal{SSYT}(\lambda', r)), \tag{1}$$

whose crystal operators, lowering $\mathcal{F}_i$ and raising $\mathcal{E}_i$, are SJDT slides on consecutive columns i, $i+1$ of T, for $i = 1, \dots, r-1$. More precisely, $\mathcal{F}_i$ sends a cell from the i-th column to the $i+1$-th column, counting from right to left. The lowest weight element of $\mathfrak{CB}^{\lambda'}(T)$ is T, and the highest weight element is the anti rectification of T, that is, the rectification is performed south-eastward. The type A_{r-1} crystals $\mathcal{SSYT}(\lambda', r)$ and $\mathfrak{CB}^{\lambda'}(T)$ are isomorphic. This isomorphism relies on the following proposition, a consequence of [11, Lemma 2.3, Lemma 2.4] by Heo-Kwon:

Proposition 3.6. *Let T be a skew SSYT. The Q-symbol of $\mathcal{F}_i(T)$ is the same as f_i applied to the Q-symbol of T, and the weight of the Q-symbol of T records the column lengths of T from right to left.*

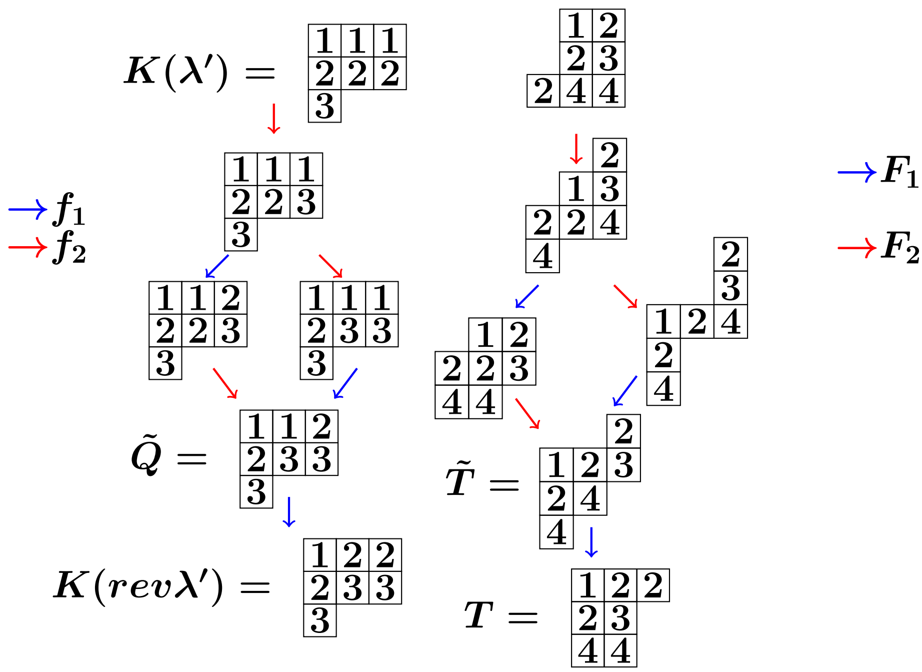
Example 3.7. Recall T and $\tilde{T}$ from the previous subsection. Note that $T = \mathcal{F}_i(\tilde{T})$ and that the Q-symbols obtained from both tableaux are connected via f_i, that is, $\tilde{Q} = f_1(K(rev((3,2,2)')))$.

This can be easily seen in the next crystal graphs. On the right, we have the cocrystal $\mathfrak{CB}^{\lambda'}(T)$, whose vertices are obtained by applying the elementary

SJDT slides $\mathcal{E}_i$, for $i = 1, 2$, on T, the lowest weight element of the cocrystal $\mathfrak{CB}^{\lambda'}(T)$. Namely, $\mathfrak{E}_1$ sends an entry from the second column to the first column, and $\mathfrak{E}_2$ sends an entry from the third column to the second column, where we count columns starting from the right. $\mathfrak{F}_1$ and $\mathfrak{F}_2$ are the inverse operations.

On the left, we have the type A_2 crystal $\mathcal{SSYT}((3,3,1),3)$, formed by the Q-symbols of every skew tableau that exists in the type A_2 crystal $\mathfrak{CB}^{\lambda'}(T)$ on the right. The type A_2 crystal operators on the left are defined by the signature rule on the alphabet $[3]$, whereas, on the right, $\mathcal{F}_1$ and $\mathcal{F}_2$ are type A_2 crystal operators defined by SJDT.

Type A_2 crystal $\mathcal{SSYT}((3,3,1),3)$ $\mathfrak{CB}^{\lambda'}(T)$



The type A_2 crystal operators f_1 and f_2 are given by the signature rule on the alphabet $[3]$, whereas $\mathcal{F}_1$ and $\mathcal{F}_2$, even though they are also type A_2 crystal operators, are defined by SJDT.

3.7. Cocrystal of KN tableaux. Let $T \in \mathcal{SSYT}(\lambda, n)$. Note that T, the lowest weight of the cocrystal $\mathfrak{CB}^{\lambda'}(T)$, is also in the type C_n crystal $\mathfrak{B}^{\lambda}$ (recall that $\mathcal{SSYT}(\lambda, n)$ is a subcrystal of $\mathfrak{B}^{\lambda}$). Fixed an arbitrary tableau Y in the crystal $\mathfrak{B}^{\lambda}$, there is a sequence S, of type C_n crystal operators of $\mathfrak{B}^{\lambda}$, such that $S(T) = Y$. All elements of the cocrystal $\mathfrak{CB}^{\lambda'}(T)$ are

SJDT related and we can apply this sequence S to all skew tableaux on the cocrystal, obtaining, for each skew tableau, a new skew tableau of the same shape. All these skew tableaux, obtained by application of the sequence S to each element of $\mathfrak{CB}^{\lambda'}(T)$, will be connected via SJDT, because the SJDT and the crystal operators of $\mathfrak{B}^{\lambda}$ commute [18, Theorem 6.3.8], hence they are the elements of a new cocrystal $\mathfrak{CB}^{\lambda'}(S(T))$ of type A_{r-1}, despite the possibility that its vertices are type C_n objects (i.e. KN skew tableaux). Recalling that the weight function of $\mathfrak{CB}^{\lambda'}(T)$ is given by the column lengths of each vertex, from right to left, which is preserved by any sequence S of crystal operators given by the C_n signature rule in $\mathfrak{B}^{\lambda}$, the following is a consequence of Proposition 3.6.

Proposition 3.8. *Given $T \in \mathcal{KN}(\lambda, n)$, with $\ell(\lambda') \geq r$, the cocrystal $\mathfrak{CB}^{\lambda'}(T)$ with lowest weight element T, obtained from T by successive application of elementary SJDT moves, is crystal isomorphic to the $\mathfrak{gl}_r$-crystal $\mathcal{SSYT}(\lambda, r)$.*

Fulton [10] has proved the following result for semistandard tableaux.

Proposition 3.9. *[10, Proposition 7, Corollary 1, Appendix A.5] Given $T \in \mathcal{SSYT}(\lambda, n)$ and a skew shape whose column lengths are a permutation of λ, the column lengths of T, there is exactly one skew tableau with that shape that rectifies to T. Furthermore, the last and first columns only depend on their lengths.*

This means that given $T \in \mathcal{SSYT}(\lambda, n)$, the cocrystal $\mathfrak{CB}^{\lambda'}(T)$ attached to $T \in \mathcal{SSYT}(\lambda, n)$ has a distinguished set of skew tableaux whose column lengths are a permutation of λ', the column lengths of T. The skew shapes of these distinguished vertices are preserved by any sequence S of type C_n crystal operators of the crystal $\mathfrak{B}^{\lambda}$. Thus we obtain another proof of our Proposition 40 and Corollary 41 in our previous work [23] which is an extension of Proposition 3.9 to KN tableaux.

Proposition 3.10. *[23, 22, Proposition 40, Corollary 41] Given $T \in \mathcal{KN}(\lambda, n)$ and a skew shape whose column lengths are a permutation of the column lengths of T, there is exactly one skew tableau with that shape that rectifies to T. Furthermore, the last and first columns only depend on their lengths.*

A key tableau in the type A_{r-1} crystal $\mathcal{SSYT}(\lambda', r)$ is a tableau of shape λ' whose weight is in $\mathfrak{S}_n\lambda'$, the $\mathfrak{S}_r$-orbit of λ'. For each element of $\mathfrak{S}_r\lambda'$

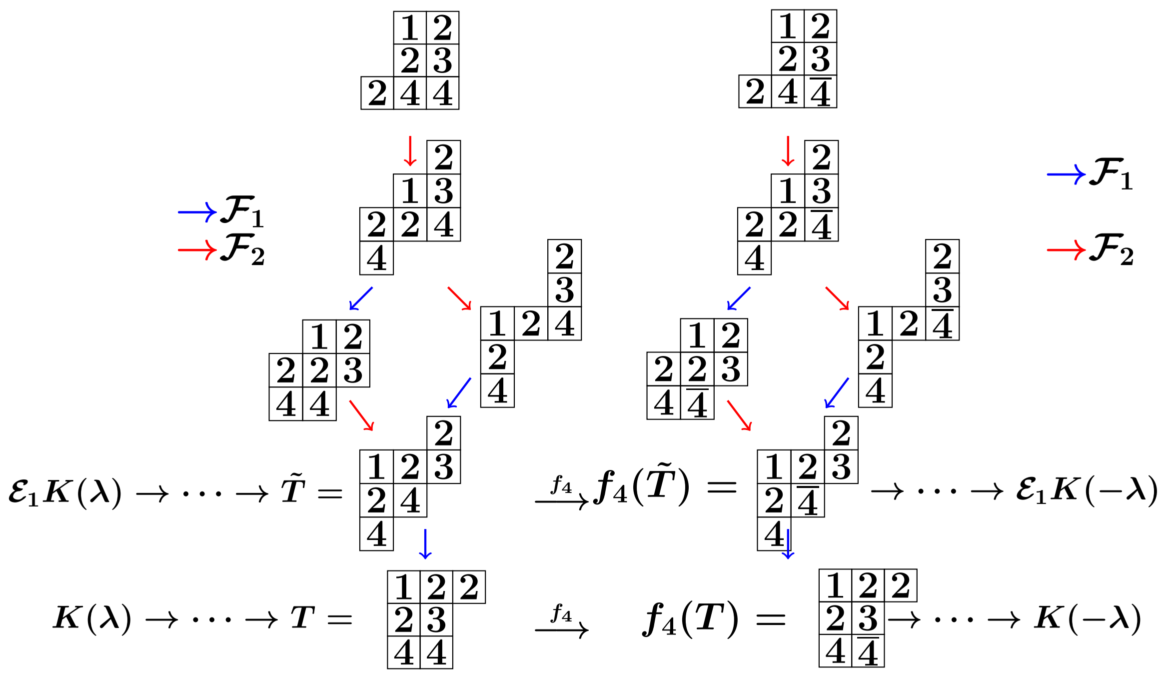
there is exactly one key tableau of shape $\boldsymbol{\lambda'}$ with that weight. More precisely the key tableaux in $\mathcal{SSYT}(\boldsymbol{\lambda'}, \boldsymbol{r})$ are distinguished vertices and define the set $\mathfrak{S}_r \boldsymbol{K}(\boldsymbol{\lambda'})$ where $\boldsymbol{s_i K(\lambda') = K(s_i \lambda')}$ and $\boldsymbol{s_i}$, for $\boldsymbol{i = 1, \dots, r-1}$, are the simple transpositions of $\mathfrak{S}_r$. Thereby it is natural to define keys in a cocrystal.

Definition 3.11. Given $\boldsymbol{T} \in \mathcal{KN}(\boldsymbol{\lambda}, \boldsymbol{n})$, with $\boldsymbol{\ell(\lambda')} \leq \boldsymbol{r}$, and $\boldsymbol{X} \in \mathfrak{CB}^{\lambda'}(\boldsymbol{T})$, $\boldsymbol{X}$ is said to be a key (skew tableau) of $\mathfrak{CB}^{\lambda'}(\boldsymbol{T})$ if its weight as an element of the said cocrystal, the sequence column lengths of $\boldsymbol{X}$, from right to left, is a permutation of the weight of $\boldsymbol{T}$ as an element of the same cocrystal.

in other words, the keys of $\mathfrak{CB}^{\lambda'}(\boldsymbol{T})$ are the image of the keys in $\mathcal{SSYT}(\boldsymbol{\lambda'}, \boldsymbol{r})$ via the crystal isomorphism (1). We then have an action of $\mathfrak{S}_r$ on set of keys of $\mathfrak{CB}^{\lambda'}(\boldsymbol{T})$.

Example 3.12. Recall the right hand side crystal from Example 3.7. $\boldsymbol{T}$ is in the type $\boldsymbol{C_4}$ crystal $\mathfrak{B}^{(3,2,2,0)}$. Hence we can apply to each vertex of $\mathfrak{CB}^{\lambda'}(\boldsymbol{T})$ the sequence of crystal operators $\boldsymbol{S} = \boldsymbol{f_4}$, obtaining a new cocrystal, on the right, whose vertices are KN skew tableaux connected via SJDT. This cocrystal $\mathfrak{CB}^{\lambda'}(\boldsymbol{f_4(T)})$ is a type $\boldsymbol{A_2}$ crystal.

$$\mathfrak{CB}^{\lambda'}(\boldsymbol{T}) \qquad \mathfrak{CB}^{\lambda'}(\boldsymbol{f_4(T)}) = \boldsymbol{f_4}\mathfrak{CB}^{\lambda'}(\boldsymbol{T})$$



The KN tableaux T and $f_4(T)$ are contained in a type C_4 crystal with highest weight element $K(\lambda)$ and lowest weight element $K(-\lambda)$. The KN skew tableaux in a same position of the cocrystal define a type C_4 crystal isomorphic to the crystal $\mathfrak{B}^{\lambda}$. In fact, their highest weight are the Littlewood-Richardson tableaux [10] of weight λ, defining the cocrystal attached to $K(\lambda)$, the Yamanouchi tableau of weight and shape λ. For instance, the type C_4 crystal containing $\tilde{T}$ and $f_4(\tilde{T})$ has highest weight element the Littlewood-Richardson tableau $\mathcal{E}_1(K(\lambda)) = \begin{array}{|c|c|c|} \hline \multicolumn{2}{c|}{} & 1 \\ \hline 1 & 1 & 2 \\ \hline 2 & 3 \\ \cline{1-2} 3 \\ \cline{1-1} \end{array}$ and lowest weight element its reversal (in the sense of Lusztig involution [23]) $\mathcal{E}_1(K(-\lambda)) = \begin{array}{|c|c|c|} \hline \multicolumn{2}{c|}{} & \bar{3} \\ \hline \bar{3} & \bar{2} & \bar{1} \\ \hline \bar{2} & \bar{1} \\ \cline{1-2} \bar{1} \\ \cline{1-1} \end{array}$.

Given a tableau T, to determine the content of a given column of its right key, $K_+(T)$, we need to compute the right column of a last column, with the same length, of a skew tableau in the cocrystal $\mathfrak{CB}^{\lambda'}(T)$. Analogously, to determine the content of a given column of its left key, $K_-(T)$, we need to compute the left column of a first column, with the same length, of a skew tableau in the cocrystal $\mathfrak{CB}^{\lambda'}(T)$. Proposition 3.10 ensures that such computation of the right, or left, key of a KN tableau via SJDT is well-defined:

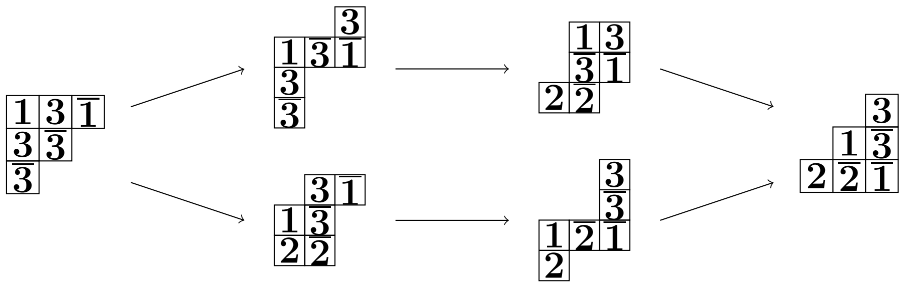
Theorem 3.13. *[23],[22, Theorem 43] Given a KN tableau T, we can replace each column with a column of the same size taken from the right columns of the last columns of all skew tableaux associated to it. We call this tableau the right key tableau of T and denote it by $K_+(T)$. If we replace each column of T with a column of the same size taken from the left columns of the first columns of all skew tableaux associated to it we obtain the left key $K_-(T)$.*

Hence, the cocrystal $\mathfrak{CB}^{\lambda'}(T)$ contains all the needed information to compute right and left keys. This is explored again in the next section.

4. The right key of a tableau - *Jeu de taquin* approach

Given $T \in \mathcal{KN}(\lambda, n)$, we apply the SJDT on consecutive columns to compute the keys of $\mathfrak{CB}^{\lambda'}(T)$, and, henceforth, all skew tableaux in the conditions of Proposition 3.10.

Example 4.1. The tableau $\boldsymbol{T} = \begin{array}{|c|c|c|}\hline 1 & 3 & \bar{1} \\ \hline 3 & \bar{3} & \multicolumn{1}{c}{} \\ \cline{1-2} \bar{3} & \multicolumn{2}{c}{} \\ \cline{1-1}\end{array}$ gives rise to the cocrystal $\mathfrak{CB}^{\lambda'}(\boldsymbol{T})$, with $\boldsymbol{\lambda} = (3, 2, 1)$. The following are the vertices of $\mathfrak{CB}^{\lambda'}(\boldsymbol{T})$ consisting of the six KN skew tableaux with the same number of columns of each length as $\boldsymbol{T}$, each one corresponding to a permutation of its column lengths.



The right key tableau of $\boldsymbol{T}$ has columns $\boldsymbol{r}\begin{array}{|c|}\hline 3 \\ \hline \bar{3} \\ \hline \bar{1} \\ \hline\end{array}$, $\boldsymbol{r}\begin{array}{|c|}\hline 3 \\ \hline \bar{1} \\ \hline\end{array}$ and $\boldsymbol{r}\begin{array}{|c|}\hline \bar{1} \\ \hline\end{array}$. Hence

$$\boldsymbol{K}_+(\boldsymbol{T}) = \begin{array}{|c|c|c|}\hline 3 & 3 & \bar{1} \\ \hline \bar{2} & \bar{1} & \multicolumn{1}{c}{} \\ \cline{1-2} \bar{1} & \multicolumn{2}{c}{} \\ \cline{1-1}\end{array}.$$

Proposition 3.10 shows that the column commutation action defined by the SJDT on two consecutive columns of a straight KN tableau $\boldsymbol{T}$ of shape $\boldsymbol{\lambda}$ gives rise to a permutohedron where the vertices are all the KN skew tableaux in the Knuth class of $\boldsymbol{T}$ whose column length sequence is a permutation of the column length sequence of $\boldsymbol{T}$ [17]. For instance, (4.1) is a permutohedron (hexagon) for $\mathfrak{S}_3$. These vertices are the key (skew) tableaux of the cocrystal $\mathfrak{CB}^{(3,2,1)'}(\boldsymbol{T})$.

Let $\boldsymbol{T} = \boldsymbol{C_1 C_2 \cdots C_k}$ be a straight KN tableau with columns $\boldsymbol{C_1, C_2, \ldots, C_k}$. Note that, to compute which entries appear in the $\boldsymbol{i}$-th column of $\boldsymbol{K}_+(\boldsymbol{T})$ we do not need to look to the first $\boldsymbol{i - 1}$ columns of $\boldsymbol{T}$. We only need the last column of a skew tableau obtained by applying the SJDT to the columns $\boldsymbol{C_i \cdots C_k}$ of $\boldsymbol{T}$, so that the last column has the length of $\boldsymbol{C_i}$, because, by Proposition 3.10, all last columns of skew tableaux associated to $\boldsymbol{T}$ with the same length are equal. Let $\boldsymbol{K}^1_+(\boldsymbol{T})$ be the map that given a tableau returns the first column of $\boldsymbol{K}_+(\boldsymbol{T})$. This is noticeable in Example 4.1 where $\boldsymbol{K}_+(\boldsymbol{T}) = \boldsymbol{K}^1_+(\boldsymbol{C1C2C3})\boldsymbol{K}^1_+(\boldsymbol{C2C3})\boldsymbol{K}^1_+(\boldsymbol{C3})$. In general,

$K_+(T) = K_+^1(C_1 \cdots C_k)K_+^1(C_2 \cdots C_k) \cdots K_+^1(C_k)$. Based on this observation and Proposition 3.10, next algorithm refines our way to compute $K_1^+(T)$ using SJDT:

Algorithm 4.2. Let T be a straight KN tableau:

(1) Let $i = 2$.
(2) If T has exactly one column, return the right column of T. Otherwise, let $T_i := T_2$ be the tableau formed by the first two columns of T.
(3) If the length of the two columns of T_i is the same, put $T_i' := T_i$. Else, play the SJDT on T_i until both column lengths are swapped, obtaining T_i'.
(4) If T has more than i columns, redefine $i := i + 1$, and define T_i to be the two-columned tableau formed with the rightmost column of T_{i-1}' and the i-th column of T, and go back to **2.**. Else, return the right column of the rightmost column of T_i'.

This algorithm is illustrated on the bottom path of (4.1).

Corollary 4.3. *If T is a rectangular tableau, $K_+(T) = rC_k rC_k \cdots rC_k$ (k times).*

Next, we present a way of computing $K_+^1(T)$ that does not require the SJDT. Willis has done this when T is a SSYT [26]. It is a simplified version of the algorithm presented here.

5. Right key - a direct way

Let $T = C_1C_2$ be a straight KN two column tableau and $spl(T) = \ell C_1 rC_1 \ell C_2 rC_2$ a straight semistandard tableau. In particular, $rC_1 \ell C_2$ is a semistandard tableau. The *matching between rC_1 and ℓC_2* is defined as follows:

- Let $\beta_1 < \cdots < \beta_{m'}$ be the elements of ℓC_2. Let i go from m' to 1, match β_i with the biggest, not yet matched, element of rC_1 smaller or equal than β_i.

Theorem 5.1 (The direct way algorithm for the right key)**.** *Let T be a straight KN tableau with columns $C_1, C_2, \ldots, C_k$, and consider its split form $spl(T)$. For every right column $rC_2, \ldots, rC_k$, add empty cells to the bottom in order to have all columns with the same length as rC_1. We will fill all of these empty cells recursively, proceeding from left to right. The extra numbers that are written in the column rC_2 are found in the following way:*

- *match* $\boldsymbol{rC_1}$ *and* $\boldsymbol{\ell C_2}$.
- *Let* $\boldsymbol{\alpha_1 < \cdots < \alpha_m}$ *be the elements of* $\boldsymbol{rC_1}$. *Let* $\boldsymbol{i}$ *go from* $\mathbf{1}$ *to* $\boldsymbol{m}$. *If* $\boldsymbol{\alpha_i}$ *is not matched with any entry of* $\boldsymbol{\ell C_2}$, *write in the new empty cells of* $\boldsymbol{rC_2}$ *the smallest element bigger or equal than* $\boldsymbol{\alpha_i}$ *such that neither it or its symmetric exist in* $\boldsymbol{rC_2}$ *or in its new cells. Let* $\boldsymbol{C'_2}$ *be the column defined by* $\boldsymbol{rC_2}$ *together with the filled extra cells, after ordering.*

To compute the filling of the extra cells of $\boldsymbol{rC_3}$, *we do the same thing, with* $\boldsymbol{C'_2}$ *and* $\boldsymbol{C_3}$. *If we do this for all pairs of consecutive columns, we eventually obtain a column* $\boldsymbol{C'_k}$, *consisting of* $\boldsymbol{rC_k}$ *together with extra cells, with the same length as* $\boldsymbol{rC_1}$. *We claim that* $\boldsymbol{C'_k = K^1_+(T)}$.

Example 5.2. Let $\boldsymbol{T = C_1C_2C_3 =}$ $\begin{array}{ccc} \mathbf{1} & \mathbf{3} & \overline{\mathbf{1}} \\ \mathbf{3} & \overline{\mathbf{3}} & \\ \overline{\mathbf{3}} & & \end{array}$, with split form $\boldsymbol{spl(T) =}$ $\begin{array}{cccccc} \mathbf{1} & \mathbf{1} & \mathbf{2} & \mathbf{3} & \overline{\mathbf{1}} & \overline{\mathbf{1}} \\ \mathbf{2} & \mathbf{3} & \overline{\mathbf{3}} & \overline{\mathbf{2}} & & \\ \overline{\mathbf{3}} & \overline{\mathbf{2}} & & & & \end{array}$. We match $\boldsymbol{rC_1}$ and $\boldsymbol{\ell C_2}$, as indicated by the letters $\boldsymbol{a}$ and $\boldsymbol{b}$: $\begin{array}{cccccc} \mathbf{1} & \mathbf{1}^{a} & \mathbf{2}^{a} & \mathbf{3} & \overline{\mathbf{1}} & \overline{\mathbf{1}} \\ \mathbf{2} & \mathbf{3}^{b} & \overline{\mathbf{3}}^{b} & \overline{\mathbf{2}} & & \\ \overline{\mathbf{3}} & \overline{\mathbf{2}} & & & & \end{array}$. Hence $\overline{\mathbf{2}}$ creates a $\overline{\mathbf{1}}$ in $\boldsymbol{rC_2}$, completing the right column $\boldsymbol{rC_2}$: $\begin{array}{cccccc} \mathbf{1} & \mathbf{1} & \mathbf{2} & \mathbf{3} & \overline{\mathbf{1}}^{a} & \overline{\mathbf{1}} \\ \mathbf{2} & \mathbf{3} & \overline{\mathbf{3}} & \overline{\mathbf{2}} & & \\ \overline{\mathbf{3}} & \overline{\mathbf{2}} & & \overline{\mathbf{1}}^{a} & & \end{array}$. Now we match $\boldsymbol{C'_2}$ and $\boldsymbol{\ell C_3}$, which is already done, and see what new cells $\mathbf{3}$ and $\overline{\mathbf{2}}$ create in $\boldsymbol{rC_3}$, obtaining $\begin{array}{cccccc} \mathbf{1} & \mathbf{1} & \mathbf{2} & \mathbf{3} & \overline{\mathbf{1}} & \overline{\mathbf{1}} \\ \mathbf{2} & \mathbf{3} & \overline{\mathbf{3}} & \overline{\mathbf{2}} & & \mathbf{3} \\ \overline{\mathbf{3}} & \overline{\mathbf{2}} & & \overline{\mathbf{1}} & & \overline{\mathbf{2}} \end{array}$. Hence $\boldsymbol{K^1_+(T) =}$ $\begin{array}{c} \mathbf{3} \\ \overline{\mathbf{2}} \\ \overline{\mathbf{1}} \end{array}$ is obtained from $\boldsymbol{C'_3}$ after reordering its entries.

5.1. The proof of Theorem 5.1. It is enough to prove that by the end of this algorithm, the entries in $\boldsymbol{C'_k}$ are the entries on the right column of the rightmost column of $\boldsymbol{T'_k}$ from Algorithm 4.2. In fact, it is enough to do this for $\boldsymbol{k = 2}$. For bigger $\boldsymbol{k}$ note that the entries that are "slid" into $\boldsymbol{C_k}$ come from $\boldsymbol{rC_{k-1}}$, so, to go to the next step on the SJDT algorithm we only need to know the previous right column, which is exactly what we claim to compute this way. The next lemma determines which number is added to $\boldsymbol{rC_2}$ given that we know $\boldsymbol{\alpha}$, the entry that is horizontally slid:

Lemma 5.3. *Suppose that $\boldsymbol{T} = \boldsymbol{C_1C_2}$ is a non-rectangular two-column tableau (if the tableau is rectangular then we have nothing to do). Play the SJDT on this tableau which ends up moving one cell from the first column to the second (some entries may change their values). Then,*

- *Immediately before the horizontal slide of the SJDT, the entry $\boldsymbol{\alpha}$, on the left of the puncture, is an unmatched cell of $\boldsymbol{rC_1}$.*
- *Call $\boldsymbol{C_1'}$ and $\boldsymbol{C_2'}$ to both columns after the horizontal slide on $\boldsymbol{T}$. The new entry in $\boldsymbol{rC_2'}$, compared to $\boldsymbol{rC_2}$, is the smallest element bigger or equal than $\boldsymbol{\alpha}$ such that neither it or its symmetric exist in $\boldsymbol{rC_2}$.*

Example 5.4. Let $\boldsymbol{T} = \begin{array}{|c|c|} \hline \mathbf{2} & \mathbf{3} \\ \hline \mathbf{3} & \mathbf{4} \\ \hline \mathbf{5} & \overline{\mathbf{5}} \\ \hline \overline{\mathbf{5}} & \multicolumn{1}{c}{} \\ \cline{1-1} \overline{\mathbf{2}} & \multicolumn{1}{c}{} \\ \cline{1-1} \end{array}$. After splitting, and just before the first horizontal slide, we have $\boldsymbol{T} = \begin{array}{|c|c|c|c|} \hline \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{3} \\ \hline \mathbf{3} & \mathbf{3} & \mathbf{4} & \mathbf{4} \\ \hline \mathbf{4} & \mathbf{5} & \overline{\mathbf{5}} & \overline{\mathbf{5}} \\ \hline \overline{\mathbf{5}} & \overline{\mathbf{4}} & * & * \\ \hline \overline{\mathbf{2}} & \overline{\mathbf{1}} & \multicolumn{2}{c}{} \\ \cline{1-2} \end{array}$. The new entry in $\boldsymbol{rC_2}$ is $\overline{\mathbf{2}}$, as predicted by the lemma: $\begin{array}{|c|c|c|c|} \hline \mathbf{1} & \mathbf{2} & \mathbf{2} & \mathbf{3} \\ \hline \mathbf{3} & \mathbf{3} & \mathbf{3} & \mathbf{4} \\ \hline \mathbf{4} & \mathbf{4} & \overline{\mathbf{5}} & \overline{\mathbf{5}} \\ \hline * & * & \overline{\mathbf{4}} & \overline{\mathbf{2}} \\ \hline \overline{\mathbf{2}} & \overline{\mathbf{1}} & \multicolumn{2}{c}{} \\ \cline{1-2} \end{array}$.

Proof: **Case 1:** $\boldsymbol{\alpha}$ is barred. Then $\boldsymbol{C_2'} = \boldsymbol{C_2} \cup \{\boldsymbol{\alpha}\}$. If $\overline{\boldsymbol{\alpha}}$ does not exist neither in $\boldsymbol{C_2}$ nor in $\boldsymbol{\Phi(C_2)}$, then $\boldsymbol{\alpha}$ will exist in both $\boldsymbol{C_2'}$ and $\boldsymbol{\Phi(C_2')}$. If $\overline{\boldsymbol{\alpha}}$ does exist in $\boldsymbol{C_2}$, and consequently in $\boldsymbol{\Phi(C_2)}$ (but $\boldsymbol{\alpha} \notin \boldsymbol{\Phi(C_2)}$), then $\boldsymbol{\alpha}$ and $\overline{\boldsymbol{\alpha}}$ will both exist in $\boldsymbol{C_2'}$. Hence, in the construction of the barred part of $\boldsymbol{\Phi(C_2')}$, compared to $\boldsymbol{\Phi(C_2)}$, there will be a new barred number which is the smallest number bigger (or equal, but the equality can not happen) than $\boldsymbol{\alpha}$ such that neither it nor its symmetric exist in the barred part of $\boldsymbol{\Phi(C_2)}$ or the unbarred part of $\boldsymbol{C_2}$ (i.e., $\boldsymbol{rC_2}$). If $\boldsymbol{\alpha}$ existed in $\boldsymbol{\Phi(C_2)}$, then $\overline{\boldsymbol{\alpha}}$ existed in $\boldsymbol{\Phi(C_2)}$. That means that whatever number got sent to $\boldsymbol{\alpha}$ in the construction of $\boldsymbol{\Phi(C_2)}$ will be sent to the next available number, meaning that in $\boldsymbol{rC_2}$ will appear a new number, the smallest number bigger (or equal, but the equality can not happen because $\boldsymbol{\alpha}$ is already there) than $\boldsymbol{\alpha}$ such that neither it nor its symmetric exist in $\boldsymbol{rC_2}$.

Case 2: $\boldsymbol{\alpha}$ is unbarred. Then $\boldsymbol{C_2'} = \boldsymbol{\Phi^{-1}(\Phi(C_2) \cup \{\alpha\})}$. If $\overline{\boldsymbol{\alpha}}$ does not exist in $\boldsymbol{C_2}$ nor in $\boldsymbol{\Phi(C_2)}$, then $\boldsymbol{\alpha}$ will exist in both $\boldsymbol{C_2'}$ and $\boldsymbol{\Phi(C_2')}$. If $\overline{\boldsymbol{\alpha}}$

existed in $\boldsymbol{\Phi(C_2)}$, and consequently in $\boldsymbol{C_2}$, then both $\boldsymbol{\alpha}$ and $\boldsymbol{\overline{\alpha}}$ will exist in $\boldsymbol{\Phi(C_2')}$, hence, if we start in the coadmissible column, in the construction of the unbarred part of $\boldsymbol{C_2'}$, compared to $\boldsymbol{C_2}$, there will be a new unbarred number which is the smallest number bigger than $\boldsymbol{\alpha}$ such that neither it nor its symmetric exist in $\boldsymbol{rC_2}$. Finally, if $\boldsymbol{\alpha}$ existed in $\boldsymbol{C_2}$, then $\boldsymbol{\overline{\alpha}}$ also existed in $\boldsymbol{C_2}$. That means that whatever number got sent to $\boldsymbol{\alpha}$ in the construction of $\boldsymbol{C_2}$, from $\boldsymbol{\Phi(C_2)}$, will be sent to the next available number, meaning that in $\boldsymbol{rC_2}$ will appear a new number, the smallest number bigger than $\boldsymbol{\alpha}$ such that neither it nor its symmetric exist in $\boldsymbol{rC_2}$. ■

Proof of Theorem 5.1: Each SJDT in $\boldsymbol{T}$, a two-column skew tableau, moves a cell from the first to the second column. We will prove that if we apply the direct way algorithm after each SJDT, the output $\boldsymbol{C_2'}$ does not change. The cells on $\boldsymbol{\ell C_2}$ without cells to its left do not get to be matched. When we slide horizontally, the columns $\boldsymbol{rC_1}$ and $\boldsymbol{\ell C_2}$ may change more than the adding/removal of $\boldsymbol{\alpha}$, the horizontally slid entry. Since the horizontal slides happen from top to bottom, we only need to see what changes happen to bigger entries than the one slid. All entries above $\boldsymbol{\alpha}$ are matched to the entry in the same row in $\boldsymbol{\ell C_2}$.

If $\boldsymbol{\alpha}$ is barred then, the remaining barred entries of $\boldsymbol{rC_1}$ and $\boldsymbol{\ell C_2}$ remain unchanged, and since all entries above $\boldsymbol{\alpha}$, including the unbarred ones, are matched to the entry directly on their right, there is no noteworthy change and everything runs as expected.

If $\boldsymbol{\alpha}$ is unbarred then, the remaining unbarred entries of $\boldsymbol{rC_1}$ and $\boldsymbol{\ell C_2}$ remain unchanged. In the barred part of $\boldsymbol{rC_1}$ either nothing happens, or there is an entry bigger than $\boldsymbol{\overline{\alpha}}$, $\boldsymbol{\overline{x}}$, that gets replaced by $\boldsymbol{\overline{\alpha}}$. Note that $\boldsymbol{\overline{x}}$ must be such that for every number between $\boldsymbol{\overline{x}}$ and $\boldsymbol{\overline{\alpha}}$, either it or its symmetric existed in $\boldsymbol{rC_1}$. In the barred part of $\boldsymbol{\ell C_2}$, if $\boldsymbol{\overline{\alpha} \in \ell C_2}$, then $\boldsymbol{\overline{\alpha}}$ gets replaced by $\boldsymbol{\overline{y}}$, smaller than $\boldsymbol{\overline{\alpha}}$, such that for every number between $\boldsymbol{\overline{y}}$ and $\boldsymbol{\overline{\alpha}}$, either it or its symmetric existed in $\boldsymbol{\ell C_2}$, and both $\boldsymbol{y}$ and $\boldsymbol{\overline{y}}$ do not exist in $\boldsymbol{\ell C_2}$.

Let's look to $\boldsymbol{\ell C_2}$. Let $\boldsymbol{\alpha < p_1 < p_2 < \cdots < p_m = y}$ be the numbers between $\boldsymbol{\alpha}$ and $\boldsymbol{y}$ that does not exist in $\boldsymbol{\ell C_2}$, right before the horizontal slide. Then, their symmetric exist in $\boldsymbol{\ell C_2}$. For all numbers in $\boldsymbol{rC_2}$ between $\boldsymbol{\alpha}$ and $\boldsymbol{y}$, there exists, in the same row in $\boldsymbol{rC_1}$, a number between $\boldsymbol{\alpha}$ and $\boldsymbol{y}$. Let $\boldsymbol{\alpha < p_1' < p_2' < \cdots < p_m' = y}$ be the missing numbers between $\boldsymbol{\alpha}$ and $\boldsymbol{y}$ in $\boldsymbol{rC_1}$, then $\boldsymbol{p_i \leq p_i'}$. Note that $\boldsymbol{\overline{p_1} > \overline{p_2} > \cdots > \overline{p_m} = \overline{y}}$ exist in

ℓC_2 after the horizontal slide and that the biggest numbers between $\overline{\alpha}$ and $\overline{y}$ (not including $\overline{\alpha}$) that can exist in rC_1 are $\overline{p'_1} > \overline{p'_2} > \cdots > \overline{p'_m}$, and since $\overline{p_i} \geq \overline{p'_i}$, the matching holds for this interval after swapping $\overline{\alpha}$ by $\overline{y}$ in ℓC_2.

Now let's look to rC_1. Before the slide, call $\overline{x'}$ to the biggest unmatched number of rC_1 smaller or equal than $\overline{x}$. If there is no such $\overline{x'}$, then everything in rC_1 between $\overline{\alpha}$ and $\overline{x}$ is matched, hence swapping $\overline{x}$ by $\overline{\alpha}$ will keep all of them matched, meaning that the algorithm works in this scenario. Let $x' < q_1 < q_2 < \cdots < q_m < \alpha$ be the numbers between x' and α that does not exist in rC_1, right before the horizontal slide. Then, their symmetric exist in rC_1. For all numbers in rC_1 between x' and α, there exists, in the same row in ℓC_2, a number between x' and α, because α is unmatched. Let $x' < q'_1 < q'_2 < \cdots < q'_m < \alpha$ be the missing numbers between x' and α in ℓC_2, then $q_i \geq q'_i$. Note that $\overline{q_1} > \overline{q_2} > \cdots > \overline{q_m} > \overline{\alpha}$ exist in rC_1 after the horizontal slide and the numbers between $\overline{x'}$ and $\overline{\alpha}$ that can exist in ℓC_2 are $\overline{q'_1} > \overline{q'_2} > \cdots > \overline{q'_m}$, and since $\overline{q_i} \leq \overline{q'_i}$, these numbers are matching a number bigger or equal then q_i in rC_1, meaning that α is unmatched in rC_1. Ignoring signs, the numbers that appear in either rC_1 or ℓC_2 are the same. So before playing the SJDT, applying the direct way algorithm we have that the unmatched numbers in rC_1 are sent to the not used numbers of $\overline{q'_1} > \overline{q'_2} > \cdots > \overline{q'_m}$ in ℓC_2 (this is a bijection), and $\overline{x'}$ is sent to the smallest available number, bigger or equal than $\overline{x'}$. Now consider rC_1 and ℓC_2 after the slide. In rC_1 we replace x' by $\overline{\alpha}$ and remove α and in ℓC_2 there is α or $\overline{\alpha}$. In the direct algorithm, all unmatched numbers of $\overline{q_1} > \overline{q_2} > \cdots > \overline{q_m} > \overline{\alpha}$ are sent to the not used numbers of $\overline{q'_1} > \overline{q'_2} > \cdots > \overline{q'_m}$ in ℓC_2, but now we have more numbers in the first set than in the second, meaning that $\overline{\alpha}$ will bump the image of the least unmatched number, which will bump the image of the second least unmatched number, and so on, meaning that the image of biggest unmatched will be out of this set. This image will be the smallest number available, which was the image of x' before the horizontal slide.

Hence, the outcome of the direct way does not change due to the changes to the columns when we play the SJDT, meaning that the outcome is what we intend. ■

6. The left key of a tableau - *Jeu de taquin* approach

Now we recall the definition of the left key map via SJDT in [23], and in Section 7 we present an alternative way which does not require the use of SJDT.

Definition 6.1 (Left key map)**.** [23, 22] Given a KN tableau T, we consider the KN skew tableaux with same number of columns of each length as T, each one corresponding to a permutation of its column lengths. Then we replace each column of T with a column of the same size taken from the left columns of the first columns of all those skew tableaux associated to T. We call this tableau the left key tableau of T and denote it by $K_-(T)$.

Example 6.2. Recall Example 4.1. The left key tableau of the tableau $T = \begin{array}{lll} \boxed{1} & \boxed{3} & \boxed{\bar{1}} \\ \boxed{3} & \boxed{\bar{3}} & \\ \boxed{\bar{3}} & & \end{array}$ has columns $\ell \begin{array}{l} \boxed{1} \\ \boxed{3} \\ \boxed{\bar{3}} \end{array}$, $\ell \begin{array}{l} \boxed{1} \\ \boxed{2} \end{array}$ and $\ell \boxed{2}$. Hence $K_-(T) = \begin{array}{lll} \boxed{1} & \boxed{1} & \boxed{2} \\ \boxed{2} & \boxed{2} & \\ \boxed{\bar{3}} & & \end{array}$.

Let $T = C_1C_2 \cdots C_k$ be a KN tableau with columns $C_1, C_2, \ldots, C_k$. We note that given a tableau T, to compute which entries appear in the i-th column of $K_-(T)$ we only need to look to the first i columns of T. We need the first column of a skew tableau obtained by applying the SJDT to the columns $C_1 \cdots C_i$ of T, so that the first column has the length of C_i. Let $K^1_-(T)$ be the map that given a tableau returns the last column of $K_-(T)$.

In Example 6.2 we have $K_-(T) = K^1_-(C1)K^1_-(C1C2)K^1_-(C1C2C3)$. In general, $K_-(T) = K^1_-(C_1) \cdots K^1_-(C_1 \cdots C_{k-1})K^1_-(C_1 \cdots C_k)$. Next we present how we compute $K^1_-(T)$ using SJDT:

Algorithm 6.3. Let k be the number of columns of T and $i = k - 1$.

(1) If T has exactly one column, return the left column of T. Otherwise, let $T_i := T_{k-1}$ be the tableau formed by the last two columns of T.
(2) If the length of the two columns of T_i is the same, put $T'_i := T_i$. Else, play the SJDT on T_i until both column lengths are swapped, obtaining T'_i.
(3) If $i \neq 1$, redefine $i := i - 1$, and define T_i as the two-columned tableau formed with the leftmost column of T'_{i+1} and the i-th column of T, and go back to (1). Else, return the left column of the leftmost column of T'_i.

This algorithm is exemplified on the top path of Example 4.1.

Corollary 6.4. *If T is a rectangular tableau, $K_-(T) = \ell C_1 \ell C_1 \cdots \ell C_1$ (k times).*

Next, we present a way of computing $K^1_-(T)$ that does not require the use of SJDT. In [26], this is done when T is a SSYT. It is simplified version of the algorithm presented here.

7. Left key - a direct way

Theorem 7.1. *Let T be a KN tableau with columns $C_1, C_2, \dots, C_k$, and consider its split form $spl(T)$.*

We will now delete entries from the left columns, proceeding from right to left, in such a way that in the end every left column has as many entries as C_k. The entries deleted from ℓC_{k-1} are found in the following way:

We start by creating a matching between rC_{k-1} and ℓC_k. Let $\beta_1 < \cdots < \beta_m$ be the unmatched elements of rC_{k-1}. For i between 1 and m, let α_i be the entry on ℓC_{k-1} next to β_i. Let i go from 1 to m. Starting at α_i and going up, delete the first entry of ℓC_{k-1} bigger than the entry directly Northeast of it. If there is no entry in this conditions, delete the top entry of ℓC_{k-1}. Also delete b_i from rC_{k-1}. By the end of this procedure we obtain $\ell C'_{k-1}$ with the same number of cells as C_k.

To continue the algorithm, we do the same thing with C_{k-2} and $\ell C'_{k-1}$. If we do this for all pairs of consecutive columns, we eventually obtain a column $\ell C'_1$, consisting of ℓC_1 with some entries deleted, with the same length as C_k. We claim that $\ell C'_1 = K^1_-(T)$.

Example 7.2. Consider $T = \begin{array}{|c|c|c|}\hline 2 & 3 & \bar{3} \\ \hline 3 & \bar{3} & \\ \cline{1-2} \bar{3} & & \\ \cline{1-1} \end{array}$, whose split form is $spl(T) = \begin{array}{|c|c|c|c|c|c|}\hline 1 & 2 & 2 & 3 & \bar{3} & \bar{3} \\ \hline 2 & 3 & \bar{3} & \bar{2} & & \\ \cline{1-4} \bar{3} & \bar{1} & & & & \\ \cline{1-2} \end{array}$.

We match rC_2 and ℓC_1, obtaining: $\begin{array}{|c|c|c|c|c|c|}\hline 1 & 2 & 2 & 3^a & \bar{3}^a & \bar{3} \\ \hline 2 & 3 & \bar{3} & \bar{2} & & \\ \cline{1-4} \bar{3} & \bar{1} & & & & \\ \cline{1-2} \end{array}$. Hence $\bar{2}$ is unmatched in rC_2. So it will get deleted, alongside the $\bar{3}$ in ℓC_2. Thus we have $\begin{array}{|c|c|c|c|c|c|}\hline 1 & 2^a & 2^a & 3 & \bar{3} & \bar{3} \\ \hline 2 & 3 & {\color{gray}\bar{3}} & {\color{gray}\bar{2}} & & \\ \cline{1-4} \bar{3} & \bar{1} & & & & \\ \cline{1-2} \end{array}$ (the deleted entries are greyed out).

Now we have to create the match between $\ell C'_2$ and rC_1, which is already done. The entries 3 and $\bar{1}$ are unmatched in rC_1, hence they will be removed alongside the entries $\mathbf{1}$ and $\overline{\mathbf{3}}$ in $\boldsymbol{\ell C_1}$, obtaining $\begin{array}{|c|c|c|c|c|c|}\hline \color{gray}1 & \mathbf{2} & \mathbf{2} & \mathbf{3} & \overline{\mathbf{3}} & \overline{\mathbf{3}} \\ \hline \mathbf{2} & \color{gray}3 & \color{gray}\overline{3} & \color{gray}\overline{2} \\ \cline{1-4} \color{gray}\overline{3} & \color{gray}\overline{1} \\ \cline{1-2} \end{array}$.

Hence $\boldsymbol{K^1_-(T)} = \boxed{\overline{\mathbf{2}}}$.

7.1. Proof of Theorem 7.1. It is enough to prove that by the end of this algorithm, the entries in $\boldsymbol{\ell C'_j}$ are the entries on the left column of the leftmost column of $\boldsymbol{T'_j}$ from Algorithm 6.3. Just like in the right key case, it is enough to do this for $\boldsymbol{j = k - 1}$. For smaller $\boldsymbol{j}$ note that we only need to know what remains in the left column $\boldsymbol{\ell C'_j}$, which is exactly what we claim to compute this way.

So only need to prove this when $\boldsymbol{T}$ is a two-column tableaux.

Lemma 7.3. *Suppose that $\boldsymbol{T}$ is a non-rectangular two-column tableau (if the tableau is rectangular then we have nothing to do). Play the SJDT on this tableau, which ends up moving one cell from the first column to the second (some entries may change its value). Immediately before the horizontal slide of the SJDT, the entry $\boldsymbol{\beta}$, on the left of the puncture, is an unmatched cell of $\boldsymbol{rC_1}$. Call $\boldsymbol{C'_1}$ and $\boldsymbol{C'_2}$ to both columns after the slide.*

Then $\boldsymbol{\ell C'_1}$ will lose an entry, compared to $\boldsymbol{\ell C_1}$, which is the biggest entry of $\boldsymbol{\ell C_1}$, in a row not under the row that contains $\boldsymbol{\beta}$, bigger than the entry directly Northeast of it.

Example 7.4. Consider the tableau $\boldsymbol{T} = \begin{array}{|c|c|}\hline \mathbf{2} & \mathbf{3} \\ \hline \mathbf{4} & \mathbf{4} \\ \hline \mathbf{5} & \overline{\mathbf{2}} \\ \hline \overline{\mathbf{5}} \\ \cline{1-1} \overline{\mathbf{2}} \\ \cline{1-1} \end{array}$. After split, and just before the horizontal slide, we have $\boldsymbol{T} = \begin{array}{|c|c|c|c|}\hline \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{3} \\ \hline \mathbf{3} & \mathbf{4} & \mathbf{4} & \mathbf{4} \\ \hline \mathbf{4} & \mathbf{5} & * & * \\ \hline \overline{\mathbf{5}} & \overline{\mathbf{3}} & \overline{\mathbf{2}} & \overline{\mathbf{2}} \\ \hline \overline{\mathbf{2}} & \overline{\mathbf{1}} \\ \cline{1-2} \end{array}$. So $\mathbf{5}$ slides from $\boldsymbol{rC_1}$ to $\boldsymbol{\ell C_2}$, obtaining the tableau $\begin{array}{|c|c|}\hline \mathbf{2} & \mathbf{3} \\ \hline \mathbf{4} & \mathbf{4} \\ \hline * & \mathbf{5} \\ \hline \overline{\mathbf{5}} & \overline{\mathbf{2}} \\ \hline \overline{\mathbf{2}} \\ \cline{1-1} \end{array}$, whose split is $\begin{array}{|c|c|c|c|}\hline \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{3} \\ \hline \mathbf{4} & \mathbf{4} & \mathbf{4} & \mathbf{4} \\ \hline * & * & \mathbf{5} & \mathbf{5} \\ \hline \overline{\mathbf{5}} & \overline{\mathbf{5}} & \overline{\mathbf{2}} & \overline{\mathbf{2}} \\ \hline \overline{\mathbf{2}} & \overline{\mathbf{1}} \\ \cline{1-2} \end{array}$. The entry removed from $\boldsymbol{\ell C_1}$ is $\mathbf{3}$, as predicted by the lemma.

Proof: If $\boldsymbol{\beta}$ is unbarred then look at all numbers $\boldsymbol{\beta \leq i \leq n}$, and count, in $\boldsymbol{C_1}$, count how many of them exist together with its symmetric and it is not matched to a number with bigger than β in the coadmissible column. Let k be that count. Now let i go from $\beta - 1$ to 1. If i and $\bar{i}$ exist in C_1 then $k := k + 1$, and if neither exist then $k := k - 1$. Since C_1 is admissible, eventually $k = 0$ and this is the i removed from ℓC_1. So, the columns ℓC_1 and rC_1 have same number of entries with absolute value bigger or equal than i, hence the entry i of ℓC_1 is bigger than the entry directly Northeast of it.

If β is barred then look at all numbers $\beta \leq i \leq \bar{1}$, and count, in C_1, count how many of them exist together with its symmetric and it is not matched to a number bigger than β in the coadmissible column. Let k be that count. Now let i go from $\beta - 1$ to $\bar{n}$. If i and $\bar{i}$ exist in C_1 then $k := k + 1$, and if neither exist then $k := k - 1$. Since $\Phi(C_1)$ is coadmissible, eventually $k = 0$ and this is the i removed from ℓC_1. The columns ℓC_1 and rC_1 have same number of entries with absolute value smaller or equal than $\bar{i}$, hence the entry i of ℓC_1 is bigger than the entry directly Northeast of it (remember that i is negative). ■

Proof of Theorem 7.1: Hence we have determined which entry is removed from ℓC_1 given that we know β, the entry of the cell that is horizontally slid. The SJDT on T may change the entries or the matching in rC_1. We need to prove that, even with these eventual changes, the entries removed from ℓC_1 are the ones that we calculated in the beginning, before doing any SJDT slide.

If β is barred, since we run the unmatched entries of rC_1 from smallest to biggest, when removing β from rC_1 the unbarred part of rC_1 remains the same, hence, the remaining entries and matched entries do not change, hence the outcome will be the one predicted.

If β is unbarred then the remaining unbarred entries of rC_1 remain unchanged. In the barred part of rC_1 either nothing happens, or there is an entry bigger than $\overline{\beta}$, $\overline{x}$, that gets replaced by $\overline{\beta}$. Note that $\overline{x}$ must be such that for every number between $\overline{x}$ and $\overline{\beta}$, either it or its symmetric existed in rC_1. This can only happen if k, from the proof of Lemma 7.3 starts being bigger than 0.

Since for all numbers between $\overline{x}$ and $\overline{\beta}$ either it or its symmetric exist in rC_1, all unmatched entries here will remove from ℓC_1 an entry smaller or equal than $\overline{x}$. In fact, the way of constructing $\overline{x}$ and i, from the proof of Lemma 7.3, is effectively the same. Since, after the slide of β, we may have different matches in the numbers between $\overline{x}$ and $\overline{\beta}$, and the number of unmatched entries remains the same after the slide. Since all unmatched entries in here will remove something smaller or equal than $\overline{\beta}$ from ℓC_1, the outcome of the algorithm is the same as if we apply it to $\ell C_1, rC_1$ before or after the horizontal slide. Hence we do not need to do any SJDT in order to know the entries of ℓC_1 after the SJDT. ∎

8. Example

In this section we present a KN tableau and compute its right and left keys via SJDT and using the direct way.

Let T be the KN tableau $\begin{array}{cccc} 2 & 3 & 3 & 4 \\ 4 & \overline{4} & \overline{4} & \overline{4} \\ \overline{5} & \overline{3} & \overline{2} & \\ \overline{4} & & & \\ \overline{3} & & & \end{array}$ with split form $\begin{array}{cccccccc} 1 & 2 & 2 & 3 & 3 & 3 & 3 & 4 \\ 3 & 4 & \overline{4} & \overline{4} & \overline{4} & \overline{4} & \overline{4} & \overline{3} \\ \overline{5} & \overline{5} & \overline{3} & \overline{2} & \overline{2} & \overline{2} & & \\ \overline{4} & \overline{3} & & & & & & \\ \overline{3} & \overline{1} & & & & & & \end{array}$.

In order to find the right (resp. left) key of T, we play the SJDT to swap heights of consecutive columns, and find skew tableaux, Knuth related to T, such that for every column height there is a skew tableau whose last column (resp. first) has that height.

$$\begin{array}{cccc} 2 & 3 & 3 & 4 \\ 4 & \overline{4} & \overline{4} & \overline{4} \\ \overline{5} & \overline{3} & \overline{2} & \\ \overline{4} & & & \\ \overline{3} & & & \end{array} \quad \begin{array}{cccc} & 3 & 3 & 4 \\ & 5 & \overline{4} & \overline{4} \\ 2 & \overline{5} & \overline{2} & \\ \overline{5} & \overline{4} & & \\ \overline{3} & \overline{3} & & \end{array} \quad \begin{array}{cccc} & & & 4 \\ & 3 & 3 & \overline{4} \\ & 5 & \overline{4} & \overline{2} \\ 2 & \overline{5} & & \\ \overline{5} & \overline{4} & & \\ \overline{3} & \overline{3} & & \end{array} \quad \begin{array}{cccc} & & 3 & 4 \\ & & 5 & \overline{4} \\ & & \overline{5} & \overline{2} \\ 2 & 2 & \overline{4} & \\ \overline{5} & \overline{4} & \overline{2} & \\ \overline{3} & & & \end{array} \quad \begin{array}{cccc} & & 3 & 4 \\ & & 5 & \overline{4} \\ & 2 & \overline{5} & \overline{2} \\ 2 & \overline{4} & \overline{4} & \\ \overline{5} & \overline{3} & \overline{2} & \end{array} \quad \begin{array}{cccc} & & & 4 \\ & & & 5 \\ & 2 & 3 & \overline{5} \\ 2 & \overline{4} & \overline{4} & \overline{4} \\ \overline{5} & \overline{3} & \overline{2} & \overline{2} \end{array}$$

Each tableau is obtained from the previous after playing SJDT in two consecutive columns, swapping their heights.

If we compute the right (resp. left) columns of all last (resp. first) columns of these tableaux, we find the columns of the right (resp. left) key associated to T:

$$K_+(T) = \begin{array}{cccc} 4 & 4 & 4 & 4 \\ 5 & \overline{3} & \overline{3} & \overline{3} \\ \overline{3} & \overline{2} & \overline{2} & \\ \overline{2} & & & \\ \overline{1} & & & \end{array} \quad \text{and } K_-(T) = \begin{array}{cccc} 1 & 2 & 2 & 2 \\ 2 & \overline{5} & \overline{5} & \overline{5} \\ \overline{5} & \overline{3} & \overline{3} & \\ \overline{4} & & & \\ \overline{3} & & & \end{array}.$$

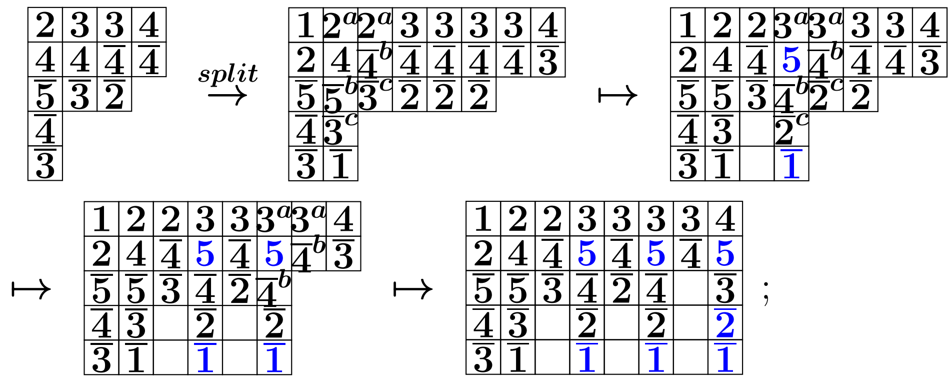
Note that we have **9** horizontal slides in our sequence of tableaux, and for each horizontal slide we have to apply the map Φ, or its inverse, two times. This means that we are effectively computing the split form of **9** skew tableaux, even though we only need **3** tableaux (the first, the third and the last one) to have all column heights in each end of the tableau.

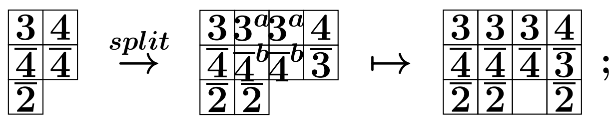
Now we compute both keys using the direct way. In here we only need to compute one split form, and make some calculations on it, and on sub-tableaux of the split form.

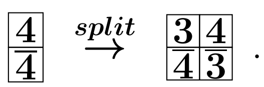
To compute the right key, via direct way, we need to compute the columns

$$\boldsymbol{K}_+^{\mathbf{1}}\left(\begin{array}{|c|c|c|c|}\hline 2&3&3&4\\\hline 4&\bar 4&\bar 4&\bar 4\\\hline \bar 5&\bar 3&\bar 2\\\cline{1-3} \bar 4\\\cline{1-1} \bar 3\\\cline{1-1}\end{array}\right),\ \boldsymbol{K}_+^{\mathbf{1}}\left(\begin{array}{|c|c|c|}\hline 3&3&4\\\hline \bar 4&\bar 4&\bar 4\\\hline \bar 3&\bar 2\\\cline{1-2}\end{array}\right)=\boldsymbol{K}_+^{\mathbf{1}}\left(\begin{array}{|c|c|}\hline 3&4\\\hline \bar 4&\bar 4\\\hline \bar 2\\\cline{1-1}\end{array}\right)\ \text{and}\ \boldsymbol{K}_+^{\mathbf{1}}\left(\begin{array}{|c|}\hline 4\\\hline \bar 4\\\hline\end{array}\right).$$

We start by splitting and matching, and every $\mapsto$ marks when new entries, written in blue, are added to a right column, and we do these until there are no columns left.







To compute the left key, via direct way, we need to compute the columns

$$\boldsymbol{K}_-^{\mathbf{1}}\left(\begin{array}{|c|c|c|c|}\hline 2&3&3&4\\\hline 4&\bar 4&\bar 4&\bar 4\\\hline \bar 5&\bar 3&\bar 2\\\cline{1-3} \bar 4\\\cline{1-1} \bar 3\\\cline{1-1}\end{array}\right),\ \boldsymbol{K}_-^{\mathbf{1}}\left(\begin{array}{|c|c|}\hline 2&3\\\hline 4&\bar 4\\\hline \bar 5&\bar 3\\\hline \bar 4\\\cline{1-1} \bar 3\\\cline{1-1}\end{array}\right)=\boldsymbol{K}_-^{\mathbf{1}}\left(\begin{array}{|c|c|c|}\hline 2&3&3\\\hline 4&\bar 4&\bar 4\\\hline \bar 5&\bar 3&\bar 2\\\hline \bar 4\\\cline{1-1} \bar 3\\\cline{1-1}\end{array}\right)\ \text{and}\ \boldsymbol{K}_-^{\mathbf{1}}\left(\begin{array}{|c|}\hline 2\\\hline 4\\\hline \bar 5\\\hline \bar 4\\\hline \bar 3\\\hline\end{array}\right).$$

We start by splitting and matching, and every $\mapsto$ marks when entries are removed from a left column, and we do these until there are no columns left. Recall that this algorithm goes from right to left.

$$\begin{array}{cccc} 2 & 3 & 3 & 4 \\ 4 & \bar{4} & \bar{4} & \bar{4} \\ \bar{5} & \bar{3} & \bar{2} & \\ \bar{4} & & & \\ \bar{3} & & & \end{array} \xrightarrow{split} \begin{array}{cccccccc} 1 & 2 & 2 & 3 & 3 & 3^a & 3^a & 4 \\ 2 & 4 & \bar{4} & \bar{4} & \bar{4} & \bar{4}^b & \bar{4}^b & \bar{3} \\ \bar{5} & \bar{5} & \bar{3} & \bar{2} & \bar{2} & \bar{2} & & \\ \bar{4} & \bar{3} & & & & & & \\ \bar{3} & \bar{1} & & & & & & \end{array} \mapsto \begin{array}{cccccccc} 1 & 2 & 2 & 3^a & 3^a & 3 & 3 & 4 \\ 2 & 4 & \bar{4} & \bar{4}^b & \bar{4}^b & \bar{4} & \bar{4} & \bar{3} \\ \bar{5} & \bar{5} & \bar{3} & \bar{2} & & & & \\ \bar{4} & \bar{3} & & & & & & \\ \bar{3} & \bar{1} & & & & & & \end{array}$$

$$\mapsto \begin{array}{cccccccc} 1 & 2^a & 2^a & 3 & 3 & 3 & 3 & 4 \\ 2 & 4 & \bar{4}^b & \bar{4} & \bar{4} & \bar{4} & \bar{4} & \bar{3} \\ \bar{5} & \bar{5}^b & & & & & & \\ \bar{4} & \bar{3} & & & & & & \\ \bar{3} & \bar{1} & & & & & & \end{array} \mapsto \begin{array}{cccccccc} & 2 & 2 & 3 & 3 & 3 & 3 & 4 \\ 2 & & \bar{4} & \bar{4} & \bar{4} & \bar{4} & \bar{4} & \bar{3} \\ \bar{5} & \bar{5} & & \bar{2} & & \bar{2} & & \\ & & & & & & & \\ & & & & & & & \end{array}$$

In the final step, we are removing $\bar{\mathbf{3}}$ from $\boldsymbol{\ell C_1}$, because the entry directly Northeast of it is $\bar{\mathbf{5}}$, because the $\bar{\mathbf{3}}$ of $\boldsymbol{rC_1}$ has already been slid out.

$$\begin{array}{cc} 2 & 3 \\ 4 & \bar{4} \\ \bar{5} & \bar{3} \\ \bar{4} & \\ \bar{3} & \end{array} \xrightarrow{split} \begin{array}{cccc} 1 & 2^a & 2^a & 3 \\ 2 & 4 & \bar{4}^b & \bar{4} \\ \bar{5} & \bar{5}^b & \bar{3}^c & \bar{2} \\ \bar{4} & \bar{3}^c & & \\ \bar{3} & \bar{1} & & \end{array} \mapsto \begin{array}{cccc} & 2 & 2 & 3 \\ 2 & & \bar{4} & \bar{4} \\ \bar{5} & \bar{5} & \bar{3} & \bar{2} \\ & \bar{3} & & \\ \bar{3} & & & \end{array}$$

$$\begin{array}{c} 2 \\ 4 \\ \bar{5} \\ \bar{4} \\ \bar{3} \end{array} \xrightarrow{split} \begin{array}{cc} 1 & 2 \\ 2 & 4 \\ \bar{5} & \bar{5} \\ \bar{4} & \bar{3} \\ \bar{3} & \bar{1} \end{array}$$

9. Acknowledgements

This work was partially supported by the Centre for Mathematics of the University of Coimbra- UIDB/00324/2020, funded by the Portuguese Government through FCT/MCTES. It was also supported by FCT, through the grant PD/BD/142954/2018, under POCH funds, co-financed by the European Social Fund and Portuguese National Funds from MCTES.

I am grateful to O. Azenhas, my Ph.D. advisor, for her help on the preparation of this paper.

References

[1] O. Azenhas. “Schur functions, pairing of parentheses, jeu de taquin and invariant factors”. In: *Mathematical papers in honour of Eduardo Marques de Sá* Textos Mat. Sér. B (2006), pp. 7–24.

[2] O. Azenhas and R. Mamede. “Key polynomials of type C”. In: *Slides from the presention in Séminaire Lotharingien de Combinatoire 69, Strobl.* 2012.

[3] A. Bjorner and F. Brenti. *Combinatorics of Coxeter groups*. Vol. 231. Springer Science & Business Media, 2006.

[4] B. Brubaker et al. "Colored five-vertex models and Demazure atoms". In: *Journal of Combinatorial Theory, Series A* 178 (2021), pp. 105–354.

[5] V. Buciumas and T. Scrimshaw. "Quasi-solvable lattice models for $\mathbf{Sp_{2n}}$ and $\mathbf{SO_{2n+1}}$ Demazure atoms and characters". In: *arXiv preprint arXiv:2101.08907* (2021).

[6] D. Bump and A. Schilling. *Crystal bases: representations and combinatorics*. World Scientific Publishing Company, 2017.

[7] S.-I. Choi and J.-H. Kwon. "Lakshmibai-Seshadri paths and non-symmetric Cauchy identity". In: *Algebras and Representation Theory* 21.6 (2018), pp. 1381–1394.

[8] C. De Concini. "Symplectic standard tableaux". In: *Advances in Mathematics* 34.1 (1979), pp. 1–27.

[9] A. M Fu and A. Lascoux. "Non-symmetric Cauchy kernels for the classical groups". In: *Journal of Combinatorial Theory, Series A* 116.4 (2009), pp. 903–917.

[10] W. Fulton. *Young tableaux: with applications to representation theory and geometry*. Vol. 35. Cambridge University Press, 1997.

[11] T.-H. Heo and J.-H. Kwon. "Combinatoria Howe duality of symplectic type". In: *arXiv:2008.05093* (2020), pp. 1–36.

[12] N. Jacon and C. Lecouvey. "Keys and Demazure crystals for Kac–Moody algebras". In: *Journal of Combinatorial Algebra* 4.4 (2020), pp. 325–358.

[13] M Kashiwara. "The crystal base and Littelmann's refined Demazure character formula". In: *Duke Mathematical Journal* 71.3 (1993), pp. 839 –858.

[14] M. Kashiwara and T. Nakashima. "Crystal Graphs for Representations of the q-Analogue of Classical Lie Algebras". In: *Journal of Algebra* 165 (1994), pp. 295–345.

[15] R. C. King. "Weight multiplicities for the classical groups". In: *Group theoretical methods in physics*. Springer, 1976, pp. 490–499.

[16] A. Lascoux. "Double crystal graphs". In: *Studies in memory of Issai Schur*. Springer, 2003, pp. 95–114.

[17] A. Lascoux and M.-P. Schützenberger. "Keys and standard bases". In: *Invariant theory and tableaux* 19 (1990), pp. 125–144.

[18] C. Lecouvey. "Schensted-type correspondence, plactic monoid, and jeu de taquin for type Cn". In: *Journal of Algebra* 247.2 (2002), pp. 295–331.

[19] C. Lenart and A. Lubovsky. "A generalization of the alcove model and its applications". In: *Journal of Algebraic Combinatorics* 41.3 (2015), pp. 751–783.

[20] P. Littelmann. "Crystal graphs and Young tableaux". In: *Journal of Algebra* 175.1 (1995), pp. 65–87.

[21] S. Mason. "An explicit construction of type A Demazure atoms". In: *Journal of Algebraic Combinatorics* 29 (2007), pp. 295–313.

[22] J. M. Santos. "Symplectic keys and Demazure atoms in type C". In: *Séminaire Lotharingien de Combinatoire - FPSAC 2020* 84B (2020).

[23] J. M. Santos. "Symplectic keys and Demazure atoms in type C". In: *Electronic Journal of Combinatorics* 28(2) (2021).

[24] J. M. Santos. "Symplectic Right Keys - Type $\boldsymbol{C}$ Willis' direct way". In: *Séminaire Lotharingien de Combinatoire - FPSAC 2021* 85B (2021).

[25] J. Sheats. "A symplectic jeu de taquin bijection between the tableaux of King and of De Concini". In: *Transactions of the American Mathematical Society* 351.9 (1999), pp. 3569–3607.

[26] M. Willis. "A Direct Way to Find the Right Key of a Semistandard Young Tableau". In: *Annals of Combinatorics* 17 (2011), pp. 393–400.

João Miguel Santos
CMUC, Department of Mathematics, University of Coimbra, Apartado 3008, 3001–454 Coimbra, Portugal

E-mail address: jmsantos@mat.uc.pt