Symplectic right keys
Type C Willis' direct way

João Miguel Santos

CMUC, University of Coimbra
Full version at: arXiv:2111.11160

FPSAC 2021
January 19, 2022

Symplectic tableaux: Kashiwara-Nakashima tableaux

Admissible columns - De Concini 1979, Lakshmibai, Musili, Seshadri 1979

A column is a word whose letters are strictly increasing according to the alphabet $[\pm n] = \{1 < 2 < \cdots < n < \overline{n} < \cdots < \overline{1}\}$.

$$C_1 = \begin{array}{|c|}\hline 2 \\ \hline 4 \\ \hline 5 \\ \hline \overline{5} \\ \hline \overline{4} \\ \hline \end{array} \quad \begin{array}{ccccc} \emptyset & 2 & \emptyset & 4 & 5 \\ \emptyset & \emptyset & \emptyset & \overline{4} & \overline{5} \end{array} \qquad C_2 = \begin{array}{|c|}\hline 2 \\ \hline 3 \\ \hline 4 \\ \hline \overline{4} \\ \hline \overline{3} \\ \hline \end{array} \quad \begin{array}{ccccc} \emptyset & 2 & 3 & 4 & \emptyset \\ \emptyset & \emptyset & \overline{3} & \overline{4} & \emptyset \end{array} \qquad C_3 = \begin{array}{|c|}\hline 2 \\ \hline 3 \\ \hline 4 \\ \hline \overline{5} \\ \hline \overline{1} \\ \hline \end{array} \quad \begin{array}{ccccc} \emptyset & 2 & 3 & 4 & \emptyset \\ \overline{1} & \emptyset & \emptyset & \emptyset & \overline{5} \end{array}$$

A column is an admissible column if the diagram is such that there is a matching which sends each full slot to an empty slot to its left.

$$\ell C_1 = \begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline 3 \\ \hline \overline{5} \\ \hline \overline{4} \\ \hline \end{array} \quad \begin{array}{ccccc} 1 & 2 & 3 & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset & \overline{4} & \overline{5} \end{array} \qquad r C_1 = \begin{array}{|c|}\hline 2 \\ \hline 4 \\ \hline 5 \\ \hline \overline{3} \\ \hline \overline{1} \\ \hline \end{array} \quad \begin{array}{ccccc} \emptyset & 2 & \emptyset & 4 & 5 \\ \overline{1} & \emptyset & \overline{3} & \emptyset & \emptyset \end{array} \qquad \ell C_3 = C_3 = r C_3$$

Symplectic tableaux: Kashiwara-Nakashima tableaux

Admissible columns - De Concini 1979, Lakshmibai, Musili, Seshadri 1979

A column is a word whose letters are strictly increasing according to the alphabet $[\pm n] = \{1 < 2 < \cdots < n < \overline{n} < \cdots < \overline{1}\}$.

$$C_1 = \begin{array}{|c|}\hline 2 \\ \hline 4 \\ \hline 5 \\ \hline \overline{5} \\ \hline \overline{4} \\ \hline \end{array} \quad \begin{array}{ccccc} \emptyset & 2 & \emptyset & \frac{4}{4} & \frac{5}{5} \\ \emptyset & \emptyset & \emptyset & & \end{array} \qquad C_2 = \begin{array}{|c|}\hline 2 \\ \hline 3 \\ \hline 4 \\ \hline \overline{4} \\ \hline \overline{3} \\ \hline \end{array} \quad \begin{array}{ccccc} \emptyset & 2 & \frac{3}{3} & \frac{4}{4} & \emptyset \\ \emptyset & \emptyset & & & \emptyset \end{array} \qquad C_3 = \begin{array}{|c|}\hline 2 \\ \hline 3 \\ \hline 4 \\ \hline \overline{5} \\ \hline \overline{1} \\ \hline \end{array} \quad \begin{array}{ccccc} \frac{\emptyset}{1} & 2 & 3 & 4 & \frac{\emptyset}{5} \\ & \emptyset & \emptyset & \emptyset & \end{array}$$

A column is a **coadmissible** column if the diagram is such that there is a matching which sends each full slot to an empty slot to its **right**.

$$\Phi(C_1) = \begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline 3 \\ \hline \overline{3} \\ \hline \overline{1} \\ \hline \end{array} \quad \begin{array}{ccccc} \frac{1}{1} & 2 & \frac{3}{3} & \emptyset & \emptyset \\ & \emptyset & & \emptyset & \emptyset \end{array}$$

Φ is a bijection between admissible and coadmissible columns

Symplectic tableaux: Kashiwara-Nakashima tableaux

KN tableaux

Let T be a tableau with all columns admissible. $spl(T)$ is the tableau obtained after replacing each column C by the pair of columns ℓC, rC. T is a Kashiwara-Nakashima (KN) tableau if spl(T) is a SSYT.

Example

$$T_1 = \begin{array}{|c|c|} \hline 3 & 3 \\ \hline \overline{3} & \overline{3} \\ \hline \overline{1} & \\ \cline{1-1} \end{array}, \; spl(T_1) = \begin{array}{|c|c|c|c|} \hline 2 & 3 & 2 & 3 \\ \hline \overline{3} & \overline{2} & \overline{3} & \overline{2} \\ \hline \overline{1} & \overline{1} & & \\ \cline{1-2} \end{array} \text{ is not a KN tableau.}$$

$$T_2 = \begin{array}{|c|c|} \hline 2 & 3 \\ \hline 3 & \overline{2} \\ \hline \overline{2} & \\ \cline{1-1} \end{array}, \; spl(T_2) = \begin{array}{|c|c|c|c|} \hline 1 & 2 & 3 & 3 \\ \hline 3 & 3 & \overline{2} & \overline{2} \\ \hline \overline{2} & \overline{1} & & \\ \cline{1-2} \end{array} \text{ is a KN tableau; } wt(T_2) = (0, -1, 2)$$

Symplectic key tableaux

Definition (Key tableau)

A key tableau on the alphabet $[\pm n]$ is a KN tableau with nested columns and with no symmetric entries, or equivalently, it is a KN tableau of shape λ whose weight is in $W\lambda$.

There is a bijection between symplectic key tableaux of shape λ and vectors in the orbit $W\lambda$, where W is the type C_n Weyl group.

Example

$\lambda = (4,4,2,0)$ and $v = (-4,0,2,4) = [4,\overline{1},3,2]\lambda \in W\lambda$

$K(\lambda) =$

1	1	1	1
2	2	2	2
3	3		

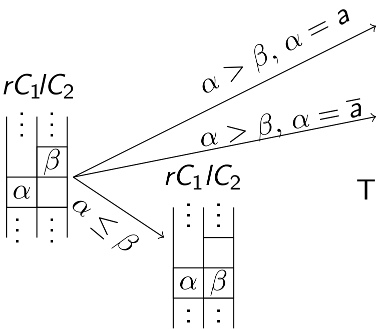
$K(v) =$

3	3	4	4
4	4	$\overline{1}$	$\overline{1}$
$\overline{1}$	$\overline{1}$		

Symplectic jeu de taquin

Lecouvey-Sheats Jeu de taquin on a KN tableau

Consider a tableau T with two consecutive admissible columns C_1 and C_2 and we have an empty cell in C_2. Split both columns.



$$\begin{cases} C_1 := C_1 \setminus \{a\} \\ C_2 := \Phi^{-1}(\Phi(C_2) \cup \{a\}) \end{cases}$$

$$\begin{cases} C_1 := \Phi^{-1}(\Phi(C_1) \setminus \{\overline{a}\}) \\ C_2 := C_2 \cup \{\overline{a}\} \end{cases}$$

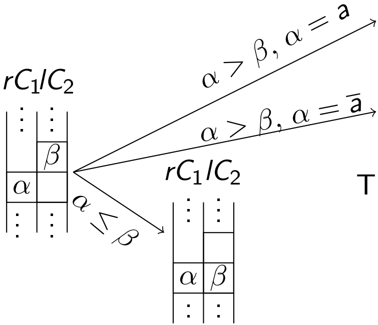
The empty cell moves horizontally.

The empty cell moves vertically and the column entries remain the same.

Symplectic jeu de taquin

Lecouvey-Sheats Jeu de taquin on a KN tableau

Consider a tableau T with two consecutive admissible columns C_1 and C_2 and we have an empty cell in C_2. Split both columns.



$$\begin{cases} C_1 := C_1 \setminus \{a\} \\ C_2 := \Phi^{-1}(\Phi(C_2) \cup \{a\}) \end{cases}$$

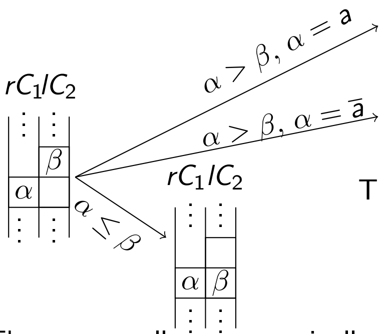
$$\begin{cases} C_1 := \Phi^{-1}(\Phi(C_1) \setminus \{\bar{a}\}) \\ C_2 := C_2 \cup \{\bar{a}\} \end{cases}$$

The empty cell moves horizontally.

The empty cell moves vertically and the column entries remain the same.

1	3	$\bar{1}$
3	$\bar{3}$	
$\bar{3}$		

Consider a tableau T with two consecutive admissible columns C_1 and C_2 and we have an empty cell in C_2. Split both columns.



$$\begin{cases} C_1 := C_1 \setminus \{a\} \\ C_2 := \Phi^{-1}(\Phi(C_2) \cup \{a\}) \end{cases}$$

$$\begin{cases} C_1 := \Phi^{-1}(\Phi(C_1) \setminus \{\overline{a}\}) \\ C_2 := C_2 \cup \{\overline{a}\} \end{cases}$$

The empty cell moves horizontally.

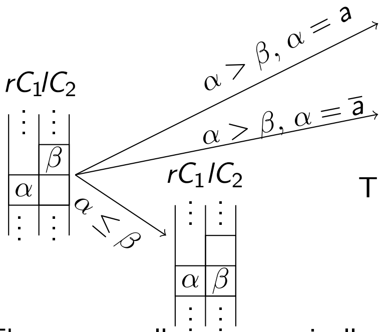
The empty cell moves vertically and the column entries remain the same.

1	1	2	3	$\overline{1}$	$\overline{1}$
2	3	$\overline{3}$	$\overline{2}$		
$\overline{3}$	$\overline{2}$				

Symplectic jeu de taquin

Lecouvey-Sheats Jeu de taquin on a KN tableau

Consider a tableau T with two consecutive admissible columns C_1 and C_2 and we have an empty cell in C_2. Split both columns.



$$\begin{cases} C_1 := C_1 \setminus \{a\} \\ C_2 := \Phi^{-1}(\Phi(C_2) \cup \{a\}) \end{cases}$$

$$\begin{cases} C_1 := \Phi^{-1}(\Phi(C_1) \setminus \{\bar{a}\}) \\ C_2 := C_2 \cup \{\bar{a}\} \end{cases}$$

The empty cell moves horizontally.

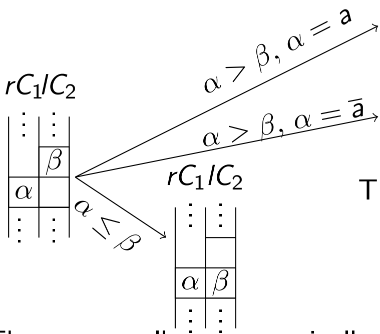
The empty cell moves vertically and the column entries remain the same.

1	3	$\bar{1}$
2	$\bar{3}$	
	$\bar{2}$	

Symplectic jeu de taquin

Lecouvey-Sheats Jeu de taquin on a KN tableau

Consider a tableau T with two consecutive admissible columns C_1 and C_2 and we have an empty cell in C_2. Split both columns.



$$\begin{cases} C_1 := C_1 \setminus \{a\} \\ C_2 := \Phi^{-1}(\Phi(C_2) \cup \{a\}) \end{cases}$$

$$\begin{cases} C_1 := \Phi^{-1}(\Phi(C_1) \setminus \{\bar{a}\}) \\ C_2 := C_2 \cup \{\bar{a}\} \end{cases}$$

The empty cell moves horizontally.

The empty cell moves vertically and the column entries remain the same.

	3	$\bar{1}$
1	$\bar{3}$	
2	$\bar{2}$	

Kashiwara crystal

Type C_n:
A $\mathfrak{sp}_{2n}$-crystal is a finite set B along with maps

$$\mathrm{wt} : \mathrm{B} \to \mathbb{Z}^{\mathrm{n}} \quad \mathrm{e_i}, \mathrm{f_i} : \mathrm{B} \to \mathrm{B} \cup \{0\}$$

for $i \in [n]$ obeying the following axioms for any $b, b' \in B$,

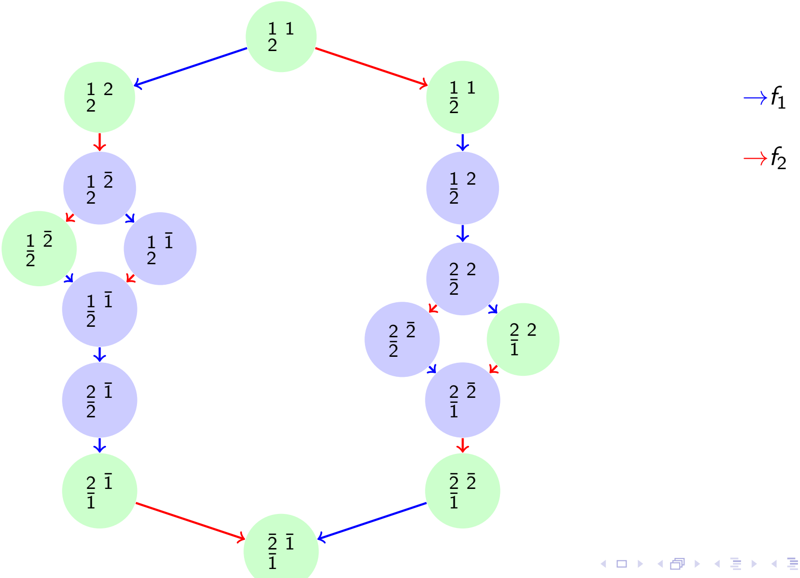
1. $b' = e_i(b)$ if and only if $b = f_i(b')$,
2. if $f_i(b) \neq 0$ then $\mathrm{wt}(\mathrm{f_i}(\mathrm{b})) = \mathrm{wt}(\mathrm{b}) - \alpha_{\mathrm{i}}$,
3. if $b, b' \in B$ such that $e_i(b) = f_i(b') = 0$ and $f_i^k(b) = b'$ for some $k \geq 0$, then $\mathrm{wt}(\mathrm{b}') = \mathrm{s_i}\mathrm{wt}(\mathrm{b})$,

where $\alpha_i = e_i - e_{i+1}$, and s_i is the simple transposition of $\mathfrak{S}_n \subseteq B_n$, $i \in [n-1]$, $\alpha_n = 2e_n$ and s_n changes last entry's sign. The crystals that we deal with also allow to define length functions $\varepsilon_i, \varphi_i : B \to \mathbb{Z}$, $i \in [n]$,

$$\varepsilon_i(b) = \max\{a : e_i^a(b) \neq 0\}, \quad \varphi_i(b) = \max\{a : f_i^a(b) \neq 0\}.$$

Kashiwara crystal

Example of type C_2 crystal: $\mathfrak{B}^{(2,1)}$



Demazure crystal

Definition (Demazure crystal)

Let $v = \sigma\lambda$, where $\sigma \in B_n$ has reduced word $s_{i_\ell} \dots s_{i_2} s_{i_1}$. The Demazure crystal $\mathfrak{B}_v$ is the set of all elements of the form

$$f_{i_\ell}^{\alpha_{i_\ell}}(\dots f_{i_2}^{\alpha_2}(f_{i_1}^{\alpha_1}(K(\lambda)))\dots),$$

where $\alpha_i \geq 0$.

Definition (Demazure crystal atom)

Let $v = \sigma\lambda$. The Demazure crystal atom $\widehat{\mathfrak{B}}_v$ is defined in the following way:

$$\widehat{\mathfrak{B}}_v := \mathfrak{B}_v \setminus \bigcup_{u <_1 v} \mathfrak{B}_u = \mathfrak{B}_v \setminus \bigcup_{K(u) <_2 K(v)} \mathfrak{B}_u,$$

where $<_1$ ir the order induced in the B_n-orbit of λ by the Bruhat order, and $<_2$ is entrywise comparison.

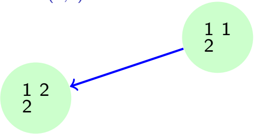
Demazure crystal

Example $\mathfrak{B}_{(\bar{2},1)} = \mathfrak{B}_{s_1 s_2 s_1 \lambda}$

1 1
2

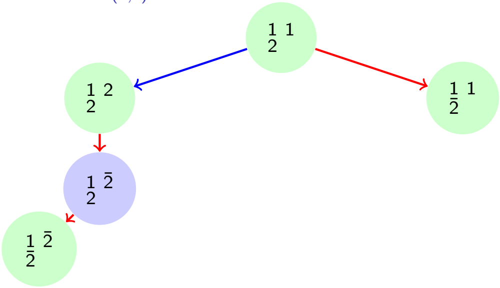
Demazure crystal

Example $\mathfrak{B}_{(\bar{2},1)} = \mathfrak{B}_{s_1 s_2 s_1 \lambda}$



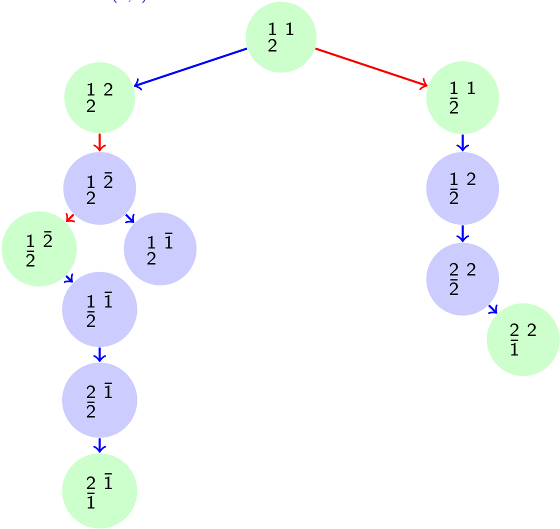
Demazure crystal

Example $\mathfrak{B}_{(\bar{2},1)} = \mathfrak{B}_{s_1 s_2 s_1 \lambda}$



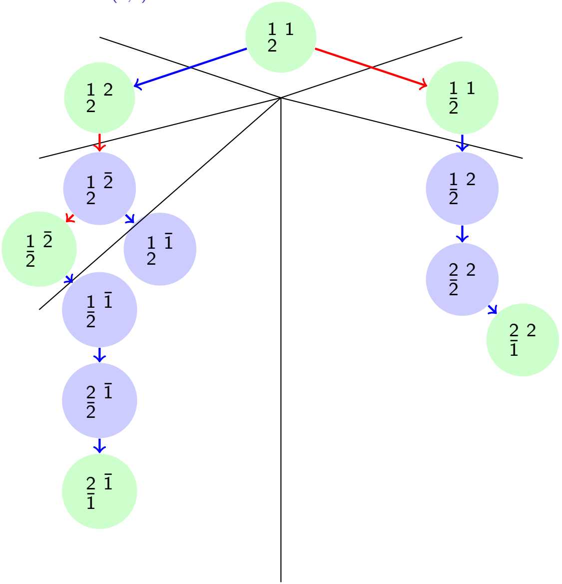
Demazure crystal

Example $\mathfrak{B}_{(\bar{2},1)} = \mathfrak{B}_{s_1 s_2 s_1 \lambda}$



Demazure crystal

Example $\mathfrak{B}_{(\bar{2},1)} = \mathfrak{B}_{s_1 s_2 s_1 \lambda}$



The right key map $K_+(T)$ sends a tableau T to the key tableau that detects the Demazure atom that contains it.

The first column of a right key tableau

Let $T = C_1 C_2 \cdots C_k$ be a KN tableau with columns $C_1, C_2, \ldots, C_k$.
Let $K_+^1(T)$ be the map that returns the first column of $K_+(T)$.
$K_+(T) = K_+^1(C_1 \cdots C_k) K_+^1(C_2 \cdots C_k) \cdots K_+^1(C_k)$.

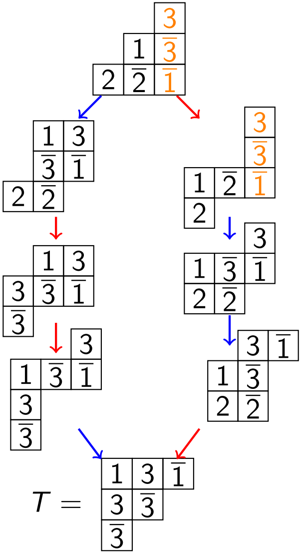
Example

$$T = \begin{array}{|c|c|c|}\hline 1 & 3 & \bar{1} \\ \hline 3 & \bar{3} \\ \cline{1-2} \bar{3} \\ \cline{1-1}\end{array}$$

$$K_+(T) = \begin{array}{|c|c|c|}\hline 3 & 3 & \bar{1} \\ \hline \bar{2} & \bar{1} \\ \cline{1-2} \bar{1} \\ \cline{1-1}\end{array} = K_+^1\left(\begin{array}{|c|c|c|}\hline 1 & 3 & \bar{1} \\ \hline 3 & \bar{3} \\ \cline{1-2} \bar{3} \\ \cline{1-1}\end{array}\right) K_+^1\left(\begin{array}{|c|c|}\hline 3 & \bar{1} \\ \hline \bar{3} \\ \cline{1-1}\end{array}\right) K_+^1\left(\begin{array}{|c|}\hline \bar{1} \\ \hline\end{array}\right)$$

Type C right key map - *symplectic jeu de taquin*

Cocrystal



If we consider as weight map the reverse column lengths, the cocrystal is crystal isomorphic to the type A crystal of tableaux with conjugated shape λ'.
$K_+^1(T)$ is the right column of right-most columns with maximal length.

$$K_+^1(T) = r\ \begin{array}{|c|}\hline 3 \\ \hline \bar{3} \\ \hline \bar{1} \\ \hline \end{array} = \begin{array}{|c|}\hline 3 \\ \hline \bar{2} \\ \hline \bar{1} \\ \hline \end{array}$$

$$\begin{array}{|c|}\hline 3 \\ \hline \bar{3} \\ \hline \bar{1} \\ \hline \end{array} \quad \begin{array}{ccc} \emptyset & \emptyset & 3 \\ \bar{1} & \emptyset & \bar{3} \end{array} \xrightarrow{r} \begin{array}{|c|}\hline 3 \\ \hline \bar{2} \\ \hline \bar{1} \\ \hline \end{array} \quad \begin{array}{ccc} \emptyset & \emptyset & 3 \\ \bar{1} & \bar{2} & \emptyset \end{array}$$

Type C right key map - direct way

$$T = \begin{array}{|c|c|c|}\hline 1 & 3 & \bar{1} \\ \hline 3 & \bar{3} \\ \cline{1-2} \bar{3} \\ \cline{1-1}\end{array} \; ; \; spl(T) = \begin{array}{|c|c|c|c|c|c|}\hline 1 & 1 & 2 & 3 & \bar{1} & \bar{1} \\ \hline 2 & 3 & \bar{3} & \bar{2} \\ \cline{1-4} \bar{3} & \bar{2} \\ \cline{1-2}\end{array}$$

Type C right key map - direct way

$$T = \begin{array}{|c|c|c|}\hline 1 & 3 & \bar{1} \\ \hline 3 & \bar{3} & \\ \cline{1-2} \bar{3} & & \\ \cline{1-1} \end{array} \; ; \; spl(T) = \begin{array}{|c|c|c|c|c|c|}\hline 1 & 1 & 2 & 3 & \bar{1} & \bar{1} \\ \hline 2 & 3 & \bar{3} & \bar{2} & & \\ \cline{1-4} \bar{3} & \bar{2} & & & & \\ \cline{1-2} \end{array}$$

Start with $i = 1$.

Create the *matching* between rC_i and ℓC_{i+1}:

Type C right key map - direct way

$$T = \begin{array}{|c|c|c|} \hline 1 & 3 & \bar{1} \\ \hline 3 & \bar{3} & \\ \cline{1-2} \bar{3} & & \\ \cline{1-1} \end{array} \; ; \; spl(T) = \begin{array}{|c|c|c|c|c|c|} \hline 1 & 1 & 2 & 3 & \bar{1} & \bar{1} \\ \hline 2 & 3 & \bar{3} & \bar{2} & & \\ \cline{1-4} \bar{3} & \bar{2} & & & & \\ \cline{1-2} \end{array}$$

Start with $i = 1$.

Create the *matching* between rC_i and ℓC_{i+1}: $\begin{array}{|c|c|c|c|c|c|} \hline 1 & {\color{red}1} & {\color{red}2} & 3 & \bar{1} & \bar{1} \\ \hline 2 & {\color{blue}3} & {\color{blue}\bar{3}} & \bar{2} & & \\ \cline{1-4} \bar{3} & \bar{2} & & & & \\ \cline{1-2} \end{array}$.

Type C right key map - direct way

$$T = \begin{array}{|c|c|c|}\hline 1 & 3 & \bar{1} \\ \hline 3 & \bar{3} \\ \cline{1-2} \bar{3} \\ \cline{1-1}\end{array} \; ; \; spl(T) = \begin{array}{|c|c|c|c|c|c|}\hline 1 & 1 & 2 & 3 & \bar{1} & \bar{1} \\ \hline 2 & 3 & \bar{3} & \bar{2} \\ \cline{1-4} \bar{3} & \bar{2} \\ \cline{1-2}\end{array}$$

Start with $i = 1$.

Create the *matching* between rC_i and ℓC_{i+1}: $\begin{array}{|c|c|c|c|c|c|}\hline 1 & 1 & 2 & 3 & \bar{1} & \bar{1} \\ \hline 2 & 3 & \bar{3} & \bar{2} \\ \cline{1-4} \bar{3} & \bar{2} \\ \cline{1-2}\end{array}$.

From smallest to biggest, for each not used number in rC_i, add in rC_{i+1} the smallest number bigger or equal than it such that neither it nor its symmetric appear in rC_{i+1}:

Type C right key map - direct way

$$T = \begin{array}{|c|c|c|}\hline 1 & 3 & \bar{1} \\ \hline 3 & \bar{3} & \\ \cline{1-2} \bar{3} & & \\ \cline{1-1} \end{array} \; ; \; spl(T) = \begin{array}{|c|c|c|c|c|c|}\hline 1 & 1 & 2 & 3 & \bar{1} & \bar{1} \\ \hline 2 & 3 & \bar{3} & \bar{2} & & \\ \cline{1-4} \bar{3} & \bar{2} & & & & \\ \cline{1-2} \end{array}$$

Start with $i = 1$.

Create the *matching* between rC_i and ℓC_{i+1}: $\begin{array}{|c|c|c|c|c|c|}\hline 1 & {\color{red}1} & {\color{red}2} & 3 & \bar{1} & \bar{1} \\ \hline 2 & {\color{blue}3} & {\color{blue}\bar{3}} & \bar{2} & & \\ \cline{1-4} \bar{3} & \bar{2} & & & & \\ \cline{1-2} \end{array}$.

From smallest to biggest, for each not used number in rC_i, add in rC_{i+1} the smallest number bigger or equal than it such that neither it nor its symmetric appear in rC_{i+1}:

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

Type C right key map - direct way

$T =$

1	3	$\bar{1}$
3	$\bar{3}$	
$\bar{3}$		

; $spl(T) =$

1	1	2	3	$\bar{1}$	$\bar{1}$
2	3	$\bar{3}$	$\bar{2}$		
$\bar{3}$	$\bar{2}$				

Start with $i = 1$.

Create the *matching* between rC_i and ℓC_{i+1}:

1	1	2	3	$\bar{1}$	$\bar{1}$
2	3	$\bar{3}$	$\bar{2}$		
$\bar{3}$	$\bar{2}$				

.

From smallest to biggest, for each not used number in rC_i, add in rC_{i+1} the smallest number bigger or equal than it such that neither it nor its symmetric appear in rC_{i+1}:

1	1	2	3	$\bar{1}$	$\bar{1}$
2	3	$\bar{3}$	$\bar{2}$		
$\bar{3}$	$\bar{2}$		$\mathbf{\bar{1}}$		

.

$i := i + 1$ and repeat until we run out of columns.

Type C right key map - direct way

$$T = \begin{array}{|c|c|c|}\hline 1 & 3 & \bar{1} \\ \hline 3 & \bar{3} \\ \cline{1-2} \bar{3} \\ \cline{1-1}\end{array} \; ; \; spl(T) = \begin{array}{|c|c|c|c|c|c|}\hline 1 & 1 & 2 & 3 & \bar{1} & \bar{1} \\ \hline 2 & 3 & \bar{3} & \bar{2} \\ \cline{1-4} \bar{3} & \bar{2} \\ \cline{1-2}\end{array}$$

Now $i = 2$. Create a *matching* between rC_2 and ℓC_3: $\begin{array}{|c|c|c|c|c|c|}\hline 1 & 1 & 2 & 3 & \color{blue}{\bar{1}} & \bar{1} \\ \hline 2 & 3 & \bar{3} & \bar{2} \\ \cline{1-4} \bar{3} & \bar{2} & & \color{blue}{\boldsymbol{\bar{1}}} \\ \cline{1-4}\end{array}$

Type C right key map - direct way

$$T = \begin{array}{|c|c|c|}\hline 1 & 3 & \overline{1} \\ \hline 3 & \overline{3} & \\ \cline{1-2} \overline{3} & & \\ \cline{1-1} \end{array} \; ; \; spl(T) = \begin{array}{|c|c|c|c|c|c|}\hline 1 & 1 & 2 & 3 & \overline{1} & \overline{1} \\ \hline 2 & 3 & \overline{3} & \overline{2} & & \\ \cline{1-4} \overline{3} & \overline{2} & & & & \\ \cline{1-2} \end{array}$$

Now $i = 2$. Create a *matching* between rC_2 and ℓC_3:

$$\begin{array}{|c|c|c|c|c|c|}\hline 1 & 1 & 2 & 3 & \overline{1} & \overline{1} \\ \hline 2 & 3 & \overline{3} & \overline{2} & & \\ \cline{1-4} \overline{3} & \overline{2} & & \mathbf{\overline{1}} & & \\ \cline{1-4} \end{array}$$

So we add 3 and $\overline{2}$ to rC_3, obtaining:

Type C right key map - direct way

$T =$

1	3	$\bar{1}$
3	$\bar{3}$	
$\bar{3}$		

; $spl(T) =$

1	1	2	3	$\bar{1}$	$\bar{1}$
2	3	$\bar{3}$	$\bar{2}$		
$\bar{3}$	$\bar{2}$				

Now $i = 2$. Create a *matching* between rC_2 and ℓC_3:

1	1	2	3	$\bar{1}$	$\bar{1}$
2	3	$\bar{3}$	$\bar{2}$		
$\bar{3}$	$\bar{2}$		$\boldsymbol{\bar{1}}$		

So we add 3 and $\bar{2}$ to rC_3, obtaining:

1	1	2	3	$\bar{1}$	$\bar{1}$
2	3	$\bar{3}$	$\bar{2}$		$\mathbf{3}$
$\bar{3}$	$\bar{2}$		$\boldsymbol{\bar{1}}$		$\boldsymbol{\bar{2}}$

.

Type C right key map - direct way

$$T = \begin{array}{|c|c|c|}\hline 1 & 3 & \bar{1} \\ \hline 3 & \bar{3} & \\ \cline{1-2} \bar{3} & & \\ \cline{1-1}\end{array} \; ; \; spl(T) = \begin{array}{|c|c|c|c|c|c|}\hline 1 & 1 & 2 & 3 & \bar{1} & \bar{1} \\ \hline 2 & 3 & \bar{3} & \bar{2} & & \\ \cline{1-4} \bar{3} & \bar{2} & & & & \\ \cline{1-2}\end{array}$$

Now $i = 2$. Create a *matching* between rC_2 and ℓC_3:

$$\begin{array}{|c|c|c|c|c|c|}\hline 1 & 1 & 2 & 3 & \mathbf{\bar{1}} & \bar{1} \\ \hline 2 & 3 & \bar{3} & \bar{2} & & \\ \cline{1-4} \bar{3} & \bar{2} & & \mathbf{\bar{1}} & & \\ \cline{1-4}\end{array}$$

So we add 3 and $\bar{2}$ to rC_3, obtaining:

$$\begin{array}{|c|c|c|c|c|c|}\hline 1 & 1 & 2 & 3 & \bar{1} & \bar{1} \\ \hline 2 & 3 & \bar{3} & \bar{2} & & \mathbf{3} \\ \hline \bar{3} & \bar{2} & & \mathbf{\bar{1}} & & \mathbf{\bar{2}} \\ \hline\end{array} .$$

$K_+^1(T)$ will be the rightmost column that we obtain, after ordering its entries.

Type C right key map - direct way

$$T = \begin{array}{|c|c|c|}\hline 1 & 3 & \bar{1} \\ \hline 3 & \bar{3} & \\ \cline{1-2} \bar{3} & & \\ \cline{1-1}\end{array} \; ; \; spl(T) = \begin{array}{|c|c|c|c|c|c|}\hline 1 & 1 & 2 & 3 & \bar{1} & \bar{1} \\ \hline 2 & 3 & \bar{3} & \bar{2} & & \\ \cline{1-4} \bar{3} & \bar{2} & & & & \\ \cline{1-2}\end{array}$$

Now $i = 2$. Create a *matching* between rC_2 and ℓC_3:

$$\begin{array}{|c|c|c|c|c|c|}\hline 1 & 1 & 2 & 3 & \bar{1} & \bar{1} \\ \hline 2 & 3 & \bar{3} & \bar{2} & & \\ \hline \bar{3} & \bar{2} & & \mathbf{\bar{1}} & & \\ \cline{1-4}\end{array}$$

So we add 3 and $\bar{2}$ to rC_3, obtaining:

$$\begin{array}{|c|c|c|c|c|c|}\hline 1 & 1 & 2 & 3 & \bar{1} & \bar{1} \\ \hline 2 & 3 & \bar{3} & \bar{2} & & \mathbf{3} \\ \hline \bar{3} & \bar{2} & & \mathbf{\bar{1}} & & \mathbf{\bar{2}} \\ \hline\end{array} \; .$$

$K^1_+(T)$ will be the rightmost column that we obtain, after ordering its entries.

Hence $K^1_+(T) = \begin{array}{|c|}\hline 3 \\ \hline \bar{2} \\ \hline \bar{1} \\ \hline\end{array}$.

Type C left key map - direct way

The left key of T, $K_-(T)$, is the tableau formed by left columns of first columns from the cocrystal. $K_-^1(T)$ returns the last column of $K_-(T)$.

$$T = \begin{array}{|c|c|c|}\hline 2 & 3 & \overline{3} \\ \hline 3 & \overline{3} \\ \cline{1-2} \overline{3} \\ \cline{1-1}\end{array} \; ; \; spl(T) = \begin{array}{|c|c|c|c|c|c|}\hline 1 & 2 & 2 & 3 & \overline{3} & \overline{3} \\ \hline 2 & 3 & \overline{3} & \overline{2} \\ \cline{1-4} \overline{3} & \overline{1} \\ \cline{1-2}\end{array}$$

Type C left key map - direct way

The left key of T, $K_-(T)$, is the tableau formed by left columns of first columns from the cocrystal. $K_-^1(T)$ returns the last column of $K_-(T)$.

$$T = \begin{array}{|c|c|c|}\hline 2 & 3 & \overline{3} \\ \hline 3 & \overline{3} & \\ \cline{1-2} \overline{3} & & \\ \cline{1-1}\end{array} \; ; \; spl(T) = \begin{array}{|c|c|c|c|c|c|}\hline 1 & 2 & 2 & 3 & \overline{3} & \overline{3} \\ \hline 2 & 3 & \overline{3} & \overline{2} & & \\ \cline{1-4} \overline{3} & \overline{1} & & & & \\ \cline{1-2}\end{array}$$

Start with $i = 3$.

Create a *matching* between rC_{i-1} and ℓC_i:

Type C left key map - direct way

The left key of T, $K_-(T)$, is the tableau formed by left columns of first columns from the cocrystal. $K_-^1(T)$ returns the last column of $K_-(T)$.

$T =$

2	3	$\overline{3}$
3	$\overline{3}$	
$\overline{3}$		

; $spl(T) =$

1	2	2	3	$\overline{3}$	$\overline{3}$
2	3	$\overline{3}$	$\overline{2}$		
$\overline{3}$	$\overline{1}$				

Start with $i = 3$.

Create a *matching* between rC_{i-1} and ℓC_i:

1	2	2	3	$\overline{3}$	$\overline{3}$
2	3	$\overline{3}$	$\overline{2}$		
$\overline{3}$	$\overline{1}$				

.

Type C left key map - direct way

The left key of T, $K_-(T)$, is the tableau formed by left columns of first columns from the cocrystal. $K_-^1(T)$ returns the last column of $K_-(T)$.

$$T = \begin{array}{|c|c|c|}\hline 2 & 3 & \bar{3} \\ \hline 3 & \bar{3} & \\ \cline{1-2} \bar{3} & & \\ \cline{1-1}\end{array} \;;\; spl(T) = \begin{array}{|c|c|c|c|c|c|}\hline 1 & 2 & 2 & 3 & \bar{3} & \bar{3} \\ \hline 2 & 3 & \bar{3} & \bar{2} & & \\ \cline{1-4} \bar{3} & \bar{1} & & & & \\ \cline{1-2}\end{array}$$

Start with $i = 3$.

Create a *matching* between rC_{i-1} and ℓC_i: $\begin{array}{|c|c|c|c|c|c|}\hline 1 & 2 & 2 & \color{blue}{3} & \color{blue}{\bar{3}} & \bar{3} \\ \hline 2 & 3 & \bar{3} & \bar{2} & & \\ \cline{1-4} \bar{3} & \bar{1} & & & & \\ \cline{1-2}\end{array}$.

From smallest to biggest, for each not matched α in rC_{i-1}, remove it and remove from ℓC_{i-1} the biggest entry not below α bigger than the entry Northeast of it:

Type C left key map - direct way

The left key of T, $K_-(T)$, is the tableau formed by left columns of first columns from the cocrystal. $K_-^1(T)$ returns the last column of $K_-(T)$.

$T =$

2	3	$\bar{3}$
3	$\bar{3}$	
$\bar{3}$		

; $spl(T) =$

1	2	2	3	$\bar{3}$	$\bar{3}$
2	3	$\bar{3}$	$\bar{2}$		
$\bar{3}$	$\bar{1}$				

Start with $i = 3$.

Create a *matching* between rC_{i-1} and ℓC_i:

1	2	2	3	$\bar{3}$	$\bar{3}$
2	3	$\bar{3}$	$\bar{2}$		
$\bar{3}$	$\bar{1}$				

.

From smallest to biggest, for each not matched α in rC_{i-1}, remove it and remove from ℓC_{i-1} the biggest entry not below α bigger than the entry Northeast of it:

1	1	2	3	$\bar{1}$	$\bar{1}$
2	3				
$\bar{3}$	$\bar{2}$				

.

Type C left key map - direct way

The left key of T, $K_-(T)$, is the tableau formed by left columns of first columns from the cocrystal. $K_-^1(T)$ returns the last column of $K_-(T)$.

$T =$

2	3	$\overline{3}$
3	$\overline{3}$	
$\overline{3}$		

; $spl(T) =$

1	2	2	3	$\overline{3}$	$\overline{3}$
2	3	$\overline{3}$	$\overline{2}$		
$\overline{3}$	$\overline{1}$				

Start with $i = 3$.

Create a *matching* between rC_{i-1} and ℓC_i:

1	2	2	3	$\overline{3}$	$\overline{3}$
2	3	$\overline{3}$	$\overline{2}$		
$\overline{3}$	$\overline{1}$				

.

From smallest to biggest, for each not matched α in rC_{i-1}, remove it and remove from ℓC_{i-1} the biggest entry not below α bigger than the entry Northeast of it:

1	1	2	3	$\overline{1}$	$\overline{1}$
2	3				
$\overline{3}$	$\overline{2}$				

.

$i := i - 1$ and repeat until we run out of columns.

Type C left key map - direct way

The left key of T, $K_-(T)$, is the tableau formed by left columns of first columns from the cocrystal. $K_-^1(T)$ returns the last column of $K_-(T)$.

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

Now $i = 2$. Create a *matching* between rC_1 and ℓC_2:.
$$\begin{array}{|c|c|c|c|c|c|} \hline 1 & 2 & 2 & 3 & \bar{3} & \bar{3} \\ \hline 2 & 3 & & & \multicolumn{2}{c}{} \\ \cline{1-4} \bar{3} & \bar{1} & \multicolumn{4}{c}{} \\ \cline{1-2} \end{array}$$

Type C left key map - direct way

The left key of T, $K_-(T)$, is the tableau formed by left columns of first columns from the cocrystal. $K_-^1(T)$ returns the last column of $K_-(T)$.

$T =$

2	3	$\overline{3}$
3	$\overline{3}$	
$\overline{3}$		

; $spl(T) =$

1	2	2	3	$\overline{3}$	$\overline{3}$
2	3	$\overline{3}$	$\overline{2}$		
$\overline{3}$	$\overline{1}$				

Now $i = 2$. Create a *matching* between rC_1 and ℓC_2:.

1	2	2	3	$\overline{3}$	$\overline{3}$
2	3				
$\overline{3}$	$\overline{1}$				

Type C left key map - direct way

The left key of T, $K_-(T)$, is the tableau formed by left columns of first columns from the cocrystal. $K^1_-(T)$ returns the last column of $K_-(T)$.

$$T = \begin{array}{|c|c|c|}\hline 2 & 3 & \bar{3} \\ \hline 3 & \bar{3} \\ \cline{1-2} \bar{3} \\ \cline{1-1}\end{array} \; ; \; spl(T) = \begin{array}{|c|c|c|c|c|c|}\hline 1 & 2 & 2 & 3 & \bar{3} & \bar{3} \\ \hline 2 & 3 & \bar{3} & \bar{2} \\ \cline{1-4} \bar{3} & \bar{1} \\ \cline{1-2}\end{array}$$

Now $i = 2$. Create a *matching* between rC_1 and ℓC_2:.

$$\begin{array}{|c|c|c|c|c|c|}\hline 1 & \color{blue}{2} & \color{blue}{2} & 3 & \bar{3} & \bar{3} \\ \hline 2 & 3 & & \\ \cline{1-4} \bar{3} & \bar{1} \\ \cline{1-2}\end{array}$$

So we remove 3 from rC_1 and 1 from ℓC_1; and remove $\bar{1}$ from rC_1 and $\bar{3}$ from ℓC_1, obtaining:

Type C left key map - direct way

The left key of T, $K_-(T)$, is the tableau formed by left columns of first columns from the cocrystal. $K_-^1(T)$ returns the last column of $K_-(T)$.

$$T = \begin{array}{|c|c|c|} \hline 2 & 3 & \overline{3} \\ \hline 3 & \overline{3} & \\ \cline{1-2} \overline{3} & & \\ \cline{1-1} \end{array} \; ; \; spl(T) = \begin{array}{|c|c|c|c|c|c|} \hline 1 & 2 & 2 & 3 & \overline{3} & \overline{3} \\ \hline 2 & 3 & \overline{3} & \overline{2} & & \\ \cline{1-4} \overline{3} & \overline{1} & & & & \\ \cline{1-2} \end{array}$$

Now $i = 2$. Create a *matching* between rC_1 and ℓC_2:.

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

So we remove 3 from rC_1 and 1 from ℓC_1; and remove $\overline{1}$ from rC_1 and $\overline{3}$ from ℓC_1, obtaining:

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

Type C left key map - direct way

The left key of T, $K_-(T)$, is the tableau formed by left columns of first columns from the cocrystal. $K_-^1(T)$ returns the last column of $K_-(T)$.

$$T = \begin{array}{|c|c|c|}\hline 2 & 3 & \overline{3} \\ \hline 3 & \overline{3} \\ \cline{1-2} \overline{3} \\ \cline{1-1}\end{array} \; ; \; spl(T) = \begin{array}{|c|c|c|c|c|c|}\hline 1 & 2 & 2 & 3 & \overline{3} & \overline{3} \\ \hline 2 & 3 & \overline{3} & \overline{2} \\ \cline{1-4} \overline{3} & \overline{1} \\ \cline{1-2}\end{array}$$

Now $i = 2$. Create a *matching* between rC_1 and ℓC_2:.

$$\begin{array}{|c|c|c|c|c|c|}\hline 1 & {\color{blue}2} & {\color{blue}2} & 3 & \overline{3} & \overline{3} \\ \hline 2 & 3 & & \\ \cline{1-4} \overline{3} & \overline{1} \\ \cline{1-2}\end{array}$$

So we remove 3 from rC_1 and 1 from ℓC_1; and remove $\overline{1}$ from rC_1 and $\overline{3}$ from ℓC_1, obtaining:

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

$K_-^1(T)$ will be the leftmost column that we obtain.

Type C left key map - direct way

The left key of T, $K_-(T)$, is the tableau formed by left columns of first columns from the cocrystal. $K_-^1(T)$ returns the last column of $K_-(T)$.

$$T = \begin{array}{|c|c|c|}\hline 2 & 3 & \overline{3} \\ \hline 3 & \overline{3} \\ \cline{1-2} \overline{3} \\ \cline{1-1}\end{array} \; ; \; spl(T) = \begin{array}{|c|c|c|c|c|c|}\hline 1 & 2 & 2 & 3 & \overline{3} & \overline{3} \\ \hline 2 & 3 & \overline{3} & \overline{2} \\ \cline{1-4} \overline{3} & \overline{1} \\ \cline{1-2}\end{array}$$

Now $i = 2$. Create a *matching* between rC_1 and ℓC_2:.
$$\begin{array}{|c|c|c|c|c|c|}\hline 1 & \color{blue}{2} & \color{blue}{2} & 3 & \overline{3} & \overline{3} \\ \hline 2 & 3 & & \\ \cline{1-4} \overline{3} & \overline{1} \\ \cline{1-2}\end{array}$$

So we remove 3 from rC_1 and 1 from ℓC_1; and remove $\overline{1}$ from rC_1 and $\overline{3}$ from ℓC_1, obtaining:
$$\begin{array}{|c|c|c|c|c|c|}\hline & 2 & 2 & 3 & \overline{3} & \overline{3} \\ \hline 2 & & & \\ \cline{1-4} & \\ \cline{1-2}\end{array}.$$

$K_-^1(T)$ will be the leftmost column that we obtain.

Hence $K_-^1(T) = \boxed{2}$.