Symplectic left and right keys
Type C Willis' direct way

João Miguel Santos

CMUC, University of Coimbra

86th Séminaire Lotharingien de Combinatoire
September 07, 2021

Overview

1. Kashiwara-Nakashima tableaux
2. Demazure and opposite Demazure crystals
3. Cocrystals of tableaux
4. Right key map: direct way
5. Left key map: direct way

Notation

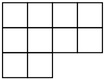
- $n \in \mathbb{N}_{>0}$;
- $[n] := \{1 < \cdots < n\}$
- $[\pm n] := \{1 < \cdots < n < -n < \cdots < -1\}$;

Young diagram

Definition (Partition)

A vector $\lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{Z}^n$ is a partition if $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$.

Young diagram of shape $\lambda = (4, 4, 2, 0)$, $n = 4$:



Semi standard Young tableau

Definition (Semi standard Young tableaux)

A semi standard Young tableau (SSYT) of shape λ is a filling of the boxes of the Young diagram of shape λ with elements from an ordered alphabet such that they are non-decreasing in each row and strictly increasing in each column.

$$T = \begin{array}{|c|c|c|c|}\hline 1 & 1 & 1 & 3 \\ \hline 3 & 3 & 3 & 4 \\ \hline 4 & 4 \\ \cline{1-2}\end{array}$$. T is a SSYT, $\mathrm{sh}(T) = (4,4,2,0)$, $wtT = (3,0,4,3)$.

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 $[\pm n] = \{1 < 2 < \cdots < n < \overline{n} < \cdots < \overline{1}\}$.

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

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.

$$\ell C_1 = \begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline 3 \\ \hline \overline{5} \\ \hline \overline{4} \\ \hline \end{array} \quad \begin{array}{ccccc} 1 & 2 & 3 & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset & \overline{4} & \overline{5} \end{array} \qquad rC_1 = \begin{array}{|c|}\hline 2 \\ \hline 4 \\ \hline 5 \\ \hline \overline{3} \\ \hline \overline{1} \\ \hline \end{array} \quad \begin{array}{ccccc} \emptyset & 2 & \emptyset & 4 & 5 \\ \overline{1} & \emptyset & \overline{3} & \emptyset & \emptyset \end{array} \qquad \ell C_3 = C_3 = rC_3$$

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 $[\pm n] = \{1 < 2 < \cdots < n < \overline{n} < \cdots < \overline{1}\}$.

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

A column is 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**.

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

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

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

$$T_2 = \begin{array}{|c|c|}\hline 2 & 3 \\ \hline 3 & \bar{2} \\ \hline \bar{2} & \\ \cline{1-1} \end{array}\ ,\ spl(T_2) = \begin{array}{|c|c|c|c|}\hline 1 & 2 & 3 & 3 \\ \hline 3 & 3 & \bar{2} & \bar{2} \\ \hline \bar{2} & \bar{1} & & \\ \cline{1-2} \end{array}$$ 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

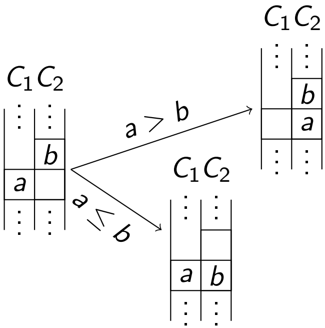
$\lambda = (4, 4, 2, 0)$ and $v = (-4, 0, 2, 4) = [4, \overline{1}, 3, 2]\lambda \in W\lambda$

$$K(\lambda) = \begin{array}{|c|c|c|c|}\hline 1 & 1 & 1 & 1 \\ \hline 2 & 2 & 2 & 2 \\ \hline 3 & 3 \\ \cline{1-2}\end{array} \qquad K(v) = \begin{array}{|c|c|c|c|}\hline 3 & 3 & 4 & 4 \\ \hline 4 & 4 & \overline{1} & \overline{1} \\ \hline \overline{1} & \overline{1} \\ \cline{1-2}\end{array}$$

Jeu de taquin

Jeu de taquin on SSYT

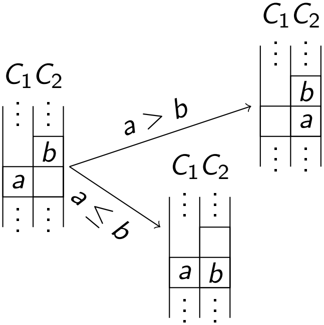
Consider a tableau T with two consecutive columns C_1 and C_2, and we have an empty cell in C_2.



Jeu de taquin

Jeu de taquin on SSYT

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

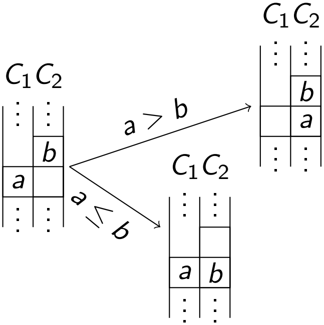


1	1	4
2	2	
3		

Jeu de taquin

Jeu de taquin on SSYT

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

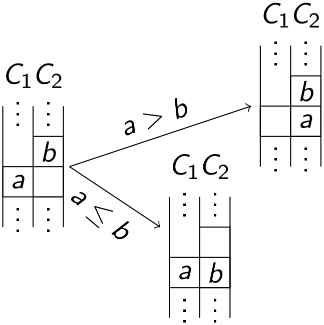


1	1	4
2	2	
	3	

Jeu de taquin

Jeu de taquin on SSYT

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

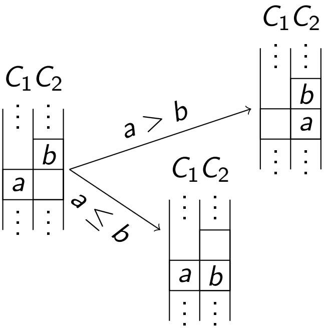


1	1	4
	2	
2	3	

Jeu de taquin

Jeu de taquin on SSYT

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

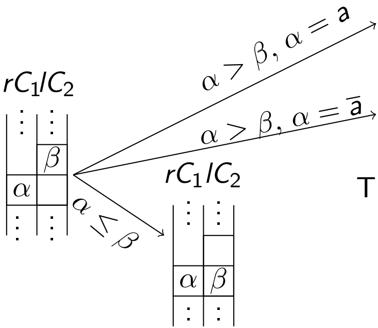


	1	4
1	2	
2	3	

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.



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

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

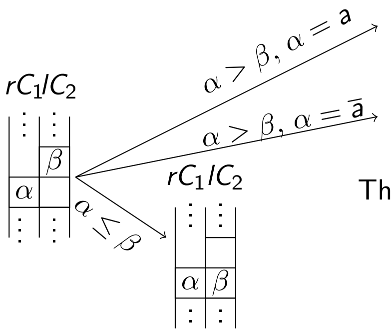
The empty cell moves horizontally.

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

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.

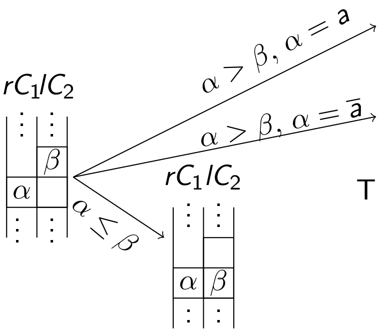
The empty cell moves vertically and the column entries remain the same.

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

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.

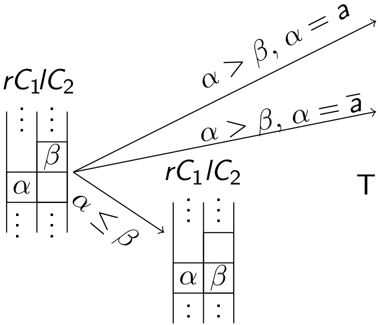
The empty cell moves vertically and the column entries remain the same.

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

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.

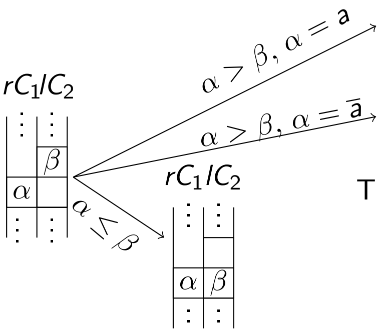
The empty cell moves vertically and the column entries remain the same.

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.



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

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

The empty cell moves horizontally.

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

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

Kashiwara crystal

Type A_{n-1}:
A $\mathfrak{gl}_n$-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 $i \in [n-1]$ 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 $\mathfrak{S}_n$, $i \in [n-1]$.
The crystals that we deal with also allow to define length functions $\varepsilon_i, \varphi_i : B \to \mathbb{Z}$, $i \in [n-1]$,

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

Kashiwara crystal

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

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

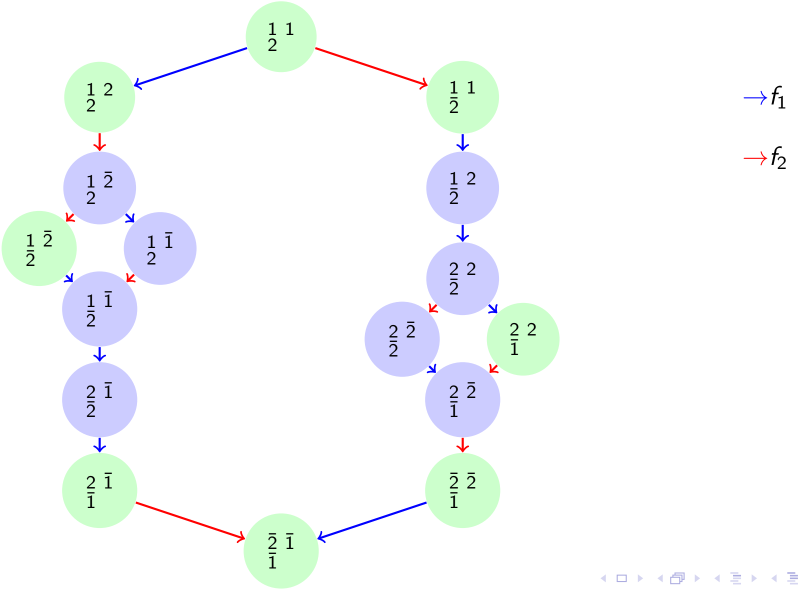
for $i \in [n]$ obeying the following axioms for any $b, b' \in B$,

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 $\mathrm{wt}(\mathrm{b'}) = \mathrm{s_i wt(b)}$,

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 $\varepsilon_i, \varphi_i : B \to \mathbb{Z}$, $i \in [n]$,

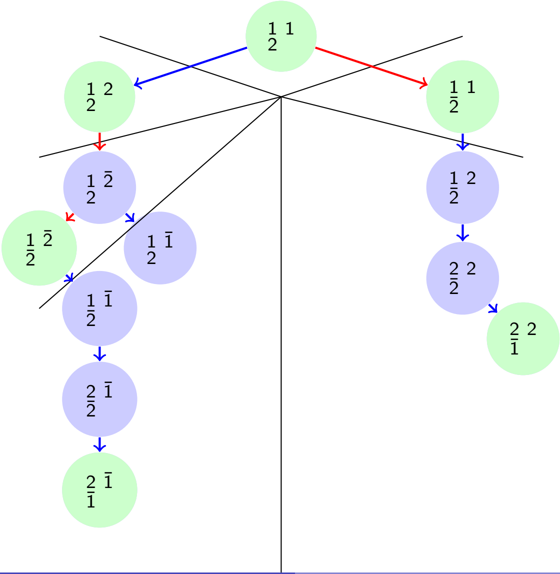
$$\varepsilon_i(b) = \max\{a : e_i^a(b) \neq 0\}, \quad \varphi_i(b) = \max\{a : f_i^a(b) \neq 0\}.$$

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



Demazure crystal - Atom decomposition

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



Demazure character:

$$\kappa_v(x) = \sum_{T \in \mathfrak{B}_v} x^{\mathrm{wt}\, T} = \sum_{K(u) \leq K(v)} \hat{\kappa}_u(x)$$

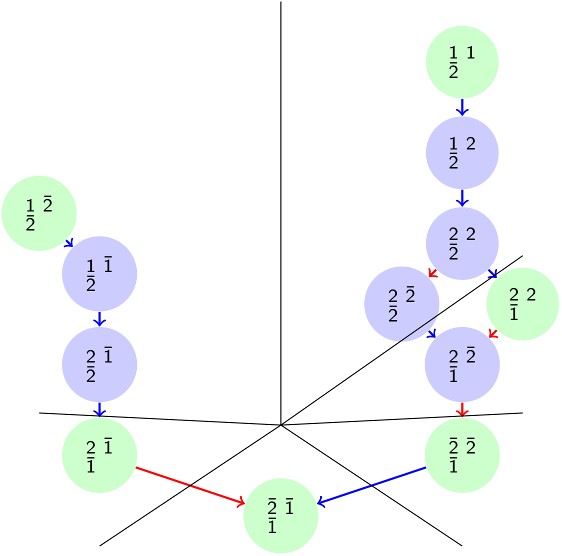
Demazure atom:

$$\hat{\kappa}_v(x) := \sum_{T \in \hat{\mathfrak{B}}_v} x^{\mathrm{wt}\, T}$$

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

Opposite Demazure crystal - Opposite atom decomposition

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



Opp. Demazure character:

$$\kappa^{op}_{-v}(x) = \sum_{T \in \mathfrak{B}^{op}_{-v}} x^{\mathrm{wt}\, T}$$

$$= \sum_{K(-u) \geq K(-v)} \hat{\kappa}^{op}_{-u}(x)$$

Opp. Demazure atom:

$$\hat{\kappa}^{op}_{-v}(x) := \sum_{T \in \hat{\mathfrak{B}}^{op}_{-v}} x^{\mathrm{wt}\, T}$$

The left key map $K_-(T)$ sends a tableau T to the key tableau that identifies the opposite atom.

Type C Lusztig involution

Let $\mathfrak{B}$ be a connected crystal. $L : \mathfrak{B} \to \mathfrak{B}$ is the type C Lusztig involution if the following holds:

1. $wt(Lx) = -wt(x)$
2. $e_i(Lx) = L(f_i(x))$
3. $f_i(Lx) = L(e_i(x))$

Proposition

Let T be a KN tableau:

$$K_+(T) = L(K_-(L(T)))$$

$$K_-(T) = L(K_+(L(T)))$$

Fu-Lascoux non-symmetric versions of Cauchy identities

Type A (Lascoux 2003): $\kappa_v(x_1,\dots,x_n)=\kappa^{op}_{rev(v)}(x_n,\dots,x_1)$, then

$$\frac{1}{\prod_{i+j\le n+1}(1-x_iy_j)}=\sum_{v\in\mathbb{N}^n}\widehat{\kappa}_v(x_1,\dots,x_n)\kappa_{rev(v)}(y_1,\dots,y_n)$$
$$=\sum_{v\in\mathbb{N}^n}\widehat{\kappa}_v(x_1,\dots,x_n)\kappa^{op}_v(y_n,\dots,y_1)$$

Bijective proofs: Lascoux 2003, Azenhas-Emami 2015, Choi-Kwon 2017

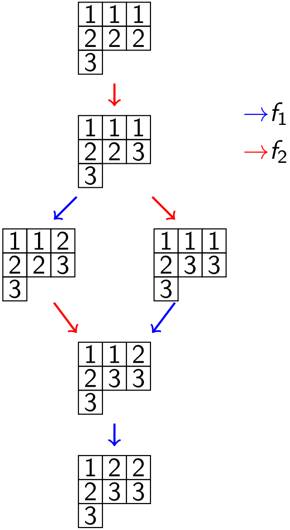
Type C (Fu-Lascoux 2009): $\kappa_v(x_1,\dots,x_n)=\kappa^{op}_{-v}(x_1^{-1},\dots,x_n^{-1})$, then

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

There's no known bijective proof.

Cocrystal for SSYT

Cocrystal: type A crystal isomorphic to $\mathfrak{B}^{\mathrm{sh}(T)'}$, given by the *jeu de taquin* as crystal operators. The weight map is the reversed column lengths.

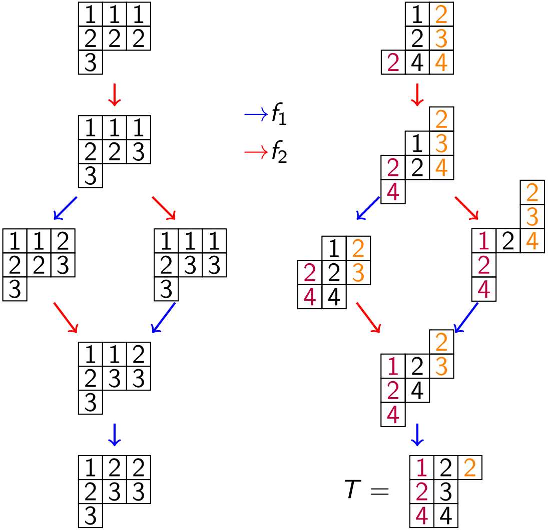


$T =$

1	2	2
2	3	
4	4	

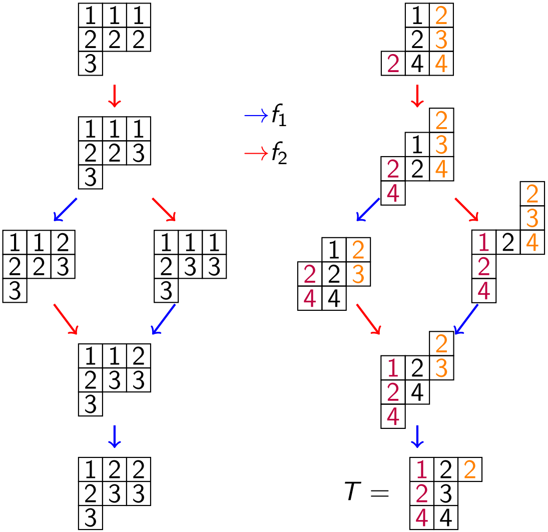
Cocrystal for SSYT

Cocrystal: type A crystal isomorphic to $\mathfrak{B}^{\mathrm{sh}(T)'}$, given by the *jeu de taquin* as crystal operators. The weight map is the reversed column lengths.



Cocrystal for SSYT

Cocrystal: type A crystal isomorphic to $\mathfrak{B}^{\mathrm{sh}(T)'}$, given by the *jeu de taquin* as crystal operators. The weight map is the reversed column lengths.



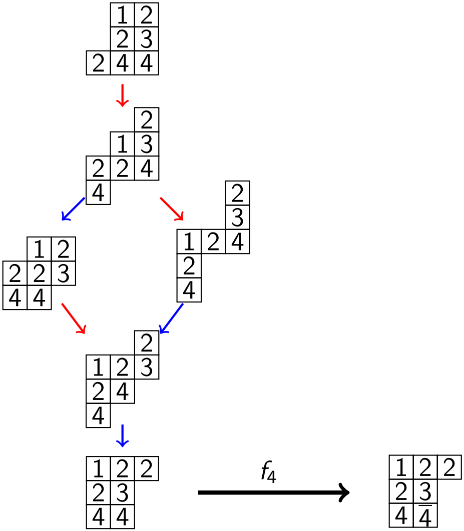
$K_+(T) =$

2	2	2
3	3	
4	4	

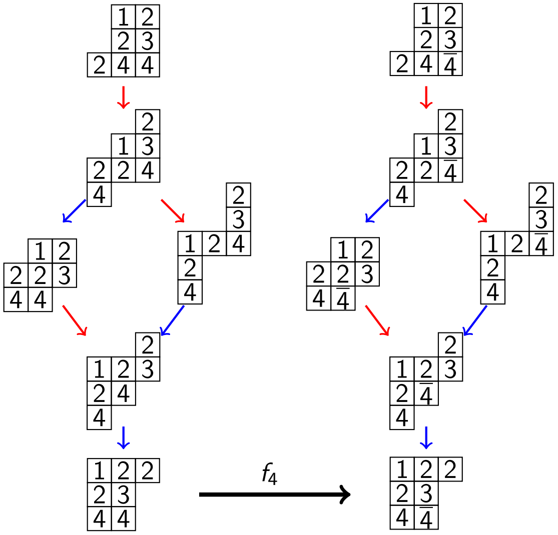
$K_-(T) =$

1	1	2
2	2	
4	4	

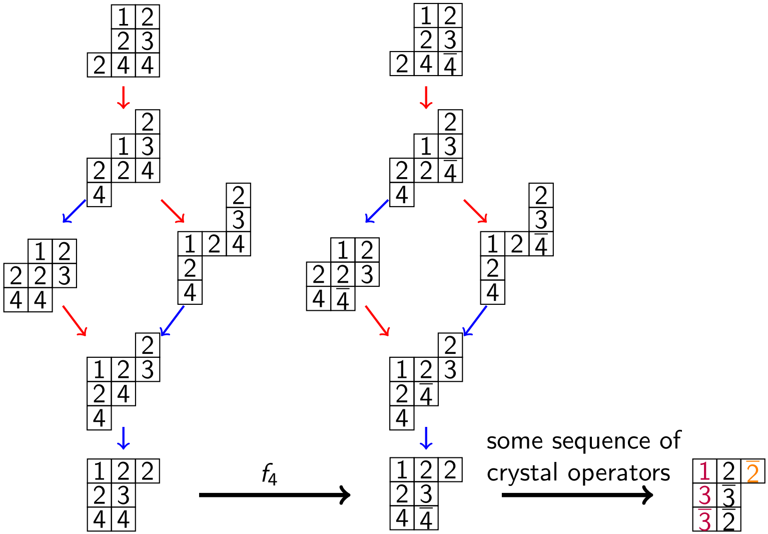
Cocrystal for KN tableaux



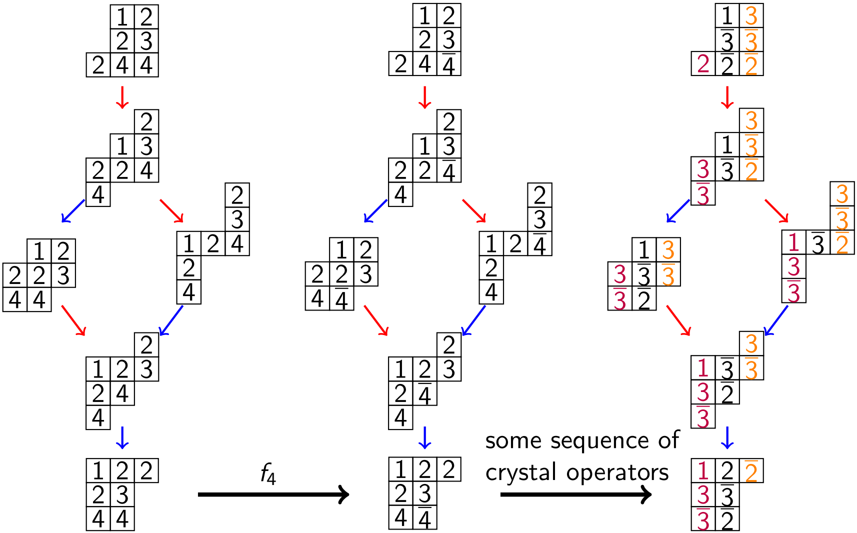
Cocrystal for KN tableaux



Cocrystal for KN tableaux



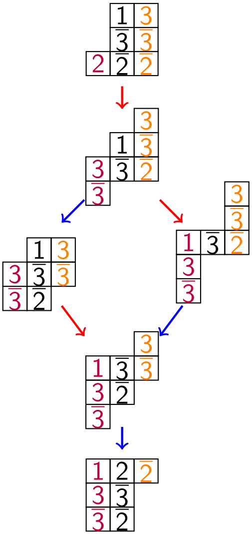
Cocrystal for KN tableaux



Cocrystal for KN tableaux

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

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



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

$$T = \begin{array}{ccc} 1 & 1 & 4 \\ 2 & 2 & \\ 3 & & \end{array} \qquad S = \begin{array}{ccc} 1 & 3 & \bar{1} \\ 3 & \bar{3} & \\ \bar{3} & & \end{array}$$

$$K_+(T) = \begin{array}{ccc} 1 & 1 & 4 \\ 2 & 4 & \\ 4 & & \end{array} = K_+^1\left(\begin{array}{ccc} 1 & 1 & 4 \\ 2 & 2 & \\ 3 & & \end{array} \right) K_+^1\left(\begin{array}{cc} 1 & 4 \\ 2 & \end{array} \right) K_+^1\left(\begin{array}{c} 4 \end{array} \right)$$

$$K_+(S) = \begin{array}{ccc} 3 & 3 & \bar{1} \\ \bar{2} & \bar{1} & \\ \bar{1} & & \end{array} = K_+^1\left(\begin{array}{ccc} 1 & 3 & \bar{1} \\ 3 & \bar{3} & \\ \bar{3} & & \end{array} \right) K_+^1\left(\begin{array}{cc} 3 & \bar{1} \\ \bar{3} & \end{array} \right) K_+^1\left(\begin{array}{c} \bar{1} \end{array} \right)$$

Type A right key map - Willis' direct way (2011)

First we create earliest weakly increasing sequences (EWIS):

- In each EWIS, start in the biggest not used number of the first column.
- Add the biggest not yet used number of the next column if it is bigger or equal than the last number added.
- Repeat the last step until we run out of columns.
  Then start a new sequence.

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

Type A right key map - Willis' direct way (2011)

First we create earliest weakly increasing sequences (EWIS):

- In each EWIS, start in the biggest not used number of the first column.
- Add the biggest not yet used number of the next column if it is bigger or equal than the last number added.
- Repeat the last step until we run out of columns.
  Then start a new sequence.

$$T = \begin{array}{|c|c|c|}\hline 1 & 1 & 4^a \\ \hline 2 & 2 & \\ \cline{1-2} 3^a & & \\ \cline{1-1} \end{array}$$

Type A right key map - Willis' direct way (2011)

First we create earliest weakly increasing sequences (EWIS):

- In each EWIS, start in the biggest not used number of the first column.
- Add the biggest not yet used number of the next column if it is bigger or equal than the last number added.
- Repeat the last step until we run out of columns. Then start a new sequence.

$T =$

1	1	4^a
2^b	2^b	
3^a		

Type A right key map - Willis' direct way (2011)

First we create earliest weakly increasing sequences (EWIS):

- In each EWIS, start in the biggest not used number of the first column.
- Add the biggest not yet used number of the next column if it is bigger or equal than the last number added.
- Repeat the last step until we run out of columns.
  Then start a new sequence.

$$T = \begin{array}{|c|c|c|}\hline 1^c & 1^c & 4^a \\ \hline 2^b & 2^b \\ \cline{1-2} 3^a \\ \cline{1-1}\end{array}$$

Type A right key map - Willis' direct way (2011)

First we create earliest weakly increasing sequences (EWIS):

- In each EWIS, start in the biggest not used number of the first column.
- Add the biggest not yet used number of the next column if it is bigger or equal than the last number added.
- Repeat the last step until we run out of columns.
  Then start a new sequence.

$$T = \begin{array}{|c|c|c|}\hline 1^c & 1^c & 4^a \\ \hline 2^b & 2^b & \\ \cline{1-2} 3^a & & \\ \cline{1-1} \end{array}$$

$K_+^1(T)$ has the biggest number of each sequence.

$$K_+^1(T) = \begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline 4 \\ \hline \end{array}$$

Type C right key map - direct way

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

Type C right key map - direct way

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

Start with $i = 1$.

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

Type C right key map - direct way

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

Start with $i = 1$.

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

Type C right key map - direct way

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

Start with $i = 1$.

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

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

$S =$

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

; $spl(S) =$

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

Start with $i = 1$.

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

1	1^a	2^a	3	$\overline{1}$	$\overline{1}$
2	3^b	$\overline{3}^b$	$\overline{2}$		
$\overline{3}$	$\overline{2}$				

.

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

.

Type C right key map - direct way

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

Start with $i = 1$.

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

$$\begin{array}{|c|c|c|c|c|c|}\hline 1 & 1^a & 2^a & 3 & \bar{1} & \bar{1} \\ \hline 2 & 3^b & \bar{3}^b & \bar{2} \\ \cline{1-4} \bar{3} & \bar{2} \\ \cline{1-2}\end{array}\;.$$

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}:

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

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

Type C right key map - direct way

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

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

Type C right key map - direct way

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

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

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

Type C right key map - direct way

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

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

Type C right key map - direct way

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

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

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

Type C right key map - direct way

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

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

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

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

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

Type C right key map - direct way

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

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

So we add 3 and $\overline{2}$ to rC_3, obtaining: $\begin{array}{|c|c|c|c|c|c|}\hline 1 & 1 & 2 & 3 & \overline{1} & \overline{1} \\ \hline 2 & 3 & \overline{3} & \overline{2} & & 3 \\ \hline \overline{3} & \overline{2} & & \overline{1} & & \overline{2} \\ \hline\end{array}\;.$

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

Type C right key map - direct way

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

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

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

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

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

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

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

The last column of a left 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 last column of $K_-(T)$.
$K_-(T) = K_+^1(C_1) \cdots K_-^1(C_1 \cdots C_{k-1}) K_-^1(C_1 \cdots C_k)$.

Example

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

$$K_-(T) = \begin{array}{|c|c|c|}\hline 1 & 1 & 2 \\ \hline 2 & 2 \\ \cline{1-2} 3 \\ \cline{1-1}\end{array} = K_-^1\left(\begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline 3 \\ \hline\end{array}\right) K_-^1\left(\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 2 & 2 \\ \hline 3 \\ \cline{1-1}\end{array}\right) K_-^1\left(\begin{array}{|c|c|c|}\hline 1 & 1 & 4 \\ \hline 2 & 2 \\ \cline{1-2} 3 \\ \cline{1-1}\end{array}\right)$$

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

Type A left key map - Willis' direct way (2011)

First we create reverse weakly decreasing sequences (RWDS):

- In each RWDS, start in the biggest not used number of the last column.
- Add the biggest not yet used number of the column to its left that is smaller or equal than the last number added.
- Repeat the last step until we run out of columns.
  We will have a sequence for every element of the last column.

$T =$

1	1	4
2	2	
3		

Type A left key map - Willis' direct way (2011)

First we create reverse weakly decreasing sequences (RWDS):

- In each RWDS, start in the biggest not used number of the last column.
- Add the biggest not yet used number of the column to its left that is smaller or equal than the last number added.
- Repeat the last step until we run out of columns.
  We will have a sequence for every element of the last column.

$$T = \begin{array}{|c|c|c|} \hline 1 & 1 & 4^a \\ \hline 2 & 2 \\ \cline{1-2} 3 \\ \cline{1-1} \end{array}$$

Type A left key map - Willis' direct way (2011)

First we create reverse weakly decreasing sequences (RWDS):

- In each RWDS, start in the biggest not used number of the last column.
- Add the biggest not yet used number of the column to its left that is smaller or equal than the last number added.
- Repeat the last step until we run out of columns.
  We will have a sequence for every element of the last column.

$$T = \begin{array}{|c|c|c|}\hline 1 & 1 & 4^a \\ \hline 2 & 2^a \\ \cline{1-2} 3 \\ \cline{1-1}\end{array}$$

Type A left key map - Willis' direct way (2011)

First we create reverse weakly decreasing sequences (RWDS):

- In each RWDS, start in the biggest not used number of the last column.
- Add the biggest not yet used number of the column to its left that is smaller or equal than the last number added.
- Repeat the last step until we run out of columns.
  We will have a sequence for every element of the last column.

$T =$

1	1	4^a
2^a	2^a	
3		

Type A left key map - Willis' direct way (2011)

First we create reverse weakly decreasing sequences (RWDS):

- In each RWDS, start in the biggest not used number of the last column.
- Add the biggest not yet used number of the column to its left that is smaller or equal than the last number added.
- Repeat the last step until we run out of columns.
  We will have a sequence for every element of the last column.

$$T = \begin{array}{|c|c|c|} \hline 1 & 1 & 4^a \\ \hline 2^a & 2^a & \\ \cline{1-2} 3 & & \\ \cline{1-1} \end{array}$$

$K_{-}^{1}(T)$ has the numbers of the first column that belong to a sequence.

$$K_{-}^{1}(T) = \begin{array}{|c|} \hline 2 \\ \hline \end{array}$$

Type C left key map - direct way

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

Type C left key map - direct way

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

Start with $i = 3$.

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

Type C left key map - direct way

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

Start with $i = 3$.

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

Type C left key map - direct way

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

Start with $i = 3$.

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

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

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

Start with $i = 3$.

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

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:

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

Type C left key map - direct way

$S =$

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

; $spl(S) =$

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^a	$\bar{3}^a$	$\bar{3}$
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				
$\bar{3}$	$\bar{2}$				

.

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

Type C left key map - direct way

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

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

Type C left key map - direct way

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

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

Type C left key map - direct way

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

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

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

Type C left key map - direct way

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

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

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:

Type C left key map - direct way

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

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

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

Type C left key map - direct way

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

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

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

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

Type C left key map - direct way

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

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

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

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:

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

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

Hence $K_-^1(S) = \begin{array}{|c|}\hline 2 \\ \hline\end{array}$.

Thank you!