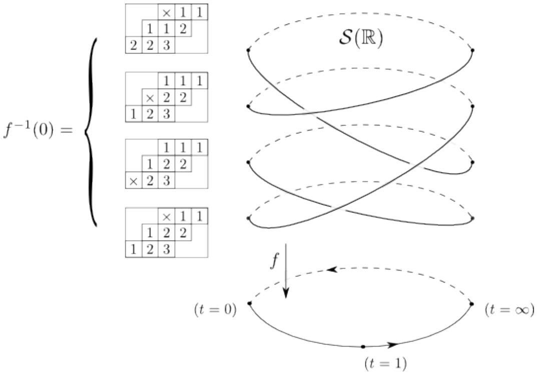
A crystal-like structure on shifted tableaux

Maria Gillespie, UC Davis
Jake Levinson, University of Washington
Kevin Purbhoo, University of Waterloo

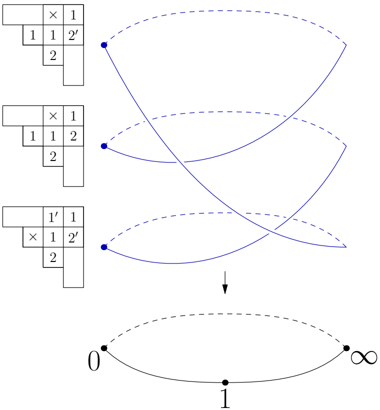
Rocky Mountain Algebraic Combinatorics Seminar
April 20, 2018

Motivation: Schubert curves



- **Grassmannian:** $\mathrm{Gr}(k, n)$ is the set of all k-planes in n-space
- Monodromy of real *Schubert curves* in $\mathrm{Gr}(k, n)$ over $\mathbb{RP}^1$ described by "jeu de taquin" operations (Speyer, 2012)
- Simplified using crystal operators E_i, F_i (G., Levinson, 2015)
- **Key fact:** E_i, F_i commute with jeu de taquin

Motivation: Type B Schubert curves



- Real Schubert curves in *Orthogonal grassmannian* $\mathrm{OG}(n, 2n+1)$ described by shifted Young tableaux (Purbhoo; G., Levinson, Purbhoo)
- **Question:** Are there "crystal" operators on shifted tableaux that commute with shifted jeu de taquin?

Outline

Part 1: Ordinary tableau crystals (known)

Part 2: Combinatorial “Crystals” for shifted tableaux (new!)

Tableaux and Schur functions

- **Skew shape:** λ/μ (below, $\lambda = (5,3,3)$ and $\mu = (2,1)$)
- **Semistandard Young tableau (SSYT):** Entries increasing down columns, weakly increasing across rows

		1	3	3
	2	2		
1	3	4		

Tableaux and Schur functions

- **Skew shape:** λ/μ (below, $\lambda = (5,3,3)$ and $\mu = (2,1)$)
- **Semistandard Young tableau (SSYT):** Entries increasing down columns, weakly increasing across rows

$$\begin{array}{ccccc} & & \boxed{1} & \boxed{3} & \boxed{3} \\ & \boxed{2} & \boxed{2} & & \\ \boxed{1} & \boxed{3} & \boxed{4} & & \end{array}$$

- **Reading word:** concatenate rows, bottom up (13422133)
- **Weight:** $\mathrm{wt}(T) = (m_1, m_2, \ldots)$ where m_i is the number of i's in T. Weight is $(2,2,3,1)$ above.

Tableaux and Schur functions

- **Skew shape:** λ/μ (below, $\lambda = (5,3,3)$ and $\mu = (2,1)$)
- **Semistandard Young tableau (SSYT):** Entries increasing down columns, weakly increasing across rows

		1	3	3
	2	2		
1	3	4		

- **Reading word:** concatenate rows, bottom up (13422133)
- **Weight:** $\mathrm{wt}(T) = (m_1, m_2, \ldots)$ where m_i is the number of i's in T. Weight is $(2,2,3,1)$ above.
- **Character:** $x^T = x_1^{m_1} x_2^{m_2} x_3^{m_3} \cdots$.
- **Schur function:** $s_{\lambda/\mu}(x_1, x_2, \ldots) = \sum_{T \in SSYT(\lambda/\mu)} x^T$

Tableaux and Schur functions

- **Skew shape:** λ/μ (below, $\lambda = (5,3,3)$ and $\mu = (2,1)$)
- **Semistandard Young tableau (SSYT):** Entries increasing down columns, weakly increasing across rows

$$\begin{array}{|c|c|c|c|c|}
\cline{3-5}
\multicolumn{1}{c}{} & \multicolumn{1}{c|}{} & 1 & 3 & 3 \\
\cline{2-5}
\multicolumn{1}{c|}{} & 2 & 2 & \multicolumn{2}{c}{} \\
\cline{1-3}
1 & 3 & 4 & \multicolumn{2}{c}{} \\
\cline{1-3}
\end{array}$$

- **Reading word:** concatenate rows, bottom up (13422133)
- **Weight:** $\mathrm{wt}(T) = (m_1, m_2, \ldots)$ where m_i is the number of i's in T. Weight is $(2,2,3,1)$ above.
- **Character:** $x^T = x_1^{m_1} x_2^{m_2} x_3^{m_3} \cdots$.
- **Schur function:** $s_{\lambda/\mu}(x_1, x_2, \ldots) = \sum_{T \in SSYT(\lambda/\mu)} x^T$
- Schur functions are symmetric in x_i variables.

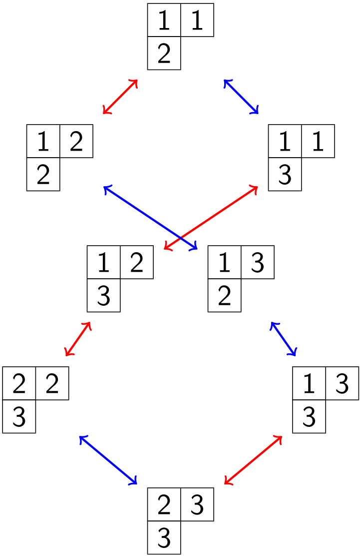
Crystal structure on tableaux

$$s_{(2,1)}(x_1, x_2, x_3) =$$

$$x_1^2 x_2$$
$$+x_1 x_2^2 + x_1^2 x_3$$
$$+2x_1 x_2 x_3$$
$$+x_2^2 x_3 + x_1 x_3^2$$
$$+x_2 x_3^2$$

$F_1 : \swarrow \quad F_2 : \searrow$

$E_1 : \nearrow \quad E_2 : \nwarrow$



Type A Crystals

- Arise from representation theory of Lie algebras.
  Example: $\mathrm{sl}_2 = \{M \in \mathrm{Mat}(2) : \mathrm{tr}(M) = 0\}$ generated by:

$$e = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \ f = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \ h = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Type A Crystals

- Arise from representation theory of Lie algebras.
  Example: $\mathrm{sl}_2 = \{M \in \mathrm{Mat}(2) : \mathrm{tr}(M) = 0\}$ generated by:

$$e = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \; f = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \; h = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

- **Type A_n Crystal:** A set $\mathcal{B}$ along with operators

$$e_i, f_i : \mathcal{B} \to \mathcal{B} \cup \{\varnothing\}, \qquad \varepsilon_i, \varphi_i : \mathcal{B} \to \mathbb{Z}$$

  for $1 \leqslant i \leqslant n$, and a map $\mathrm{wt} : \mathcal{B} \to \mathbb{Z}^{n+1}$ such that:

Type A Crystals

- Arise from representation theory of Lie algebras.
  Example: $\mathrm{sl}_2 = \{M \in \mathrm{Mat}(2) : \mathrm{tr}(M) = 0\}$ generated by:

$$e = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \ f = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \ h = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

- **Type A_n Crystal:** A set $\mathcal{B}$ along with operators

$$e_i, f_i : \mathcal{B} \to \mathcal{B} \cup \{\varnothing\}, \qquad \varepsilon_i, \varphi_i : \mathcal{B} \to \mathbb{Z}$$

  for $1 \leqslant i \leqslant n$, and a map $\mathrm{wt} : \mathcal{B} \to \mathbb{Z}^{n+1}$ such that:
  (1) If $X, Y \in \mathcal{B}$ then $e_i(X) = Y$ iff $f_i(Y) = X$. In this case,

$$\varepsilon_i(Y) = \varepsilon_i(X) - 1, \qquad \varphi_i(Y) = \varphi_i(X) + 1$$

  and $\mathrm{wt}(Y) = \mathrm{wt}(X) + (0^{i-1}, 1, -1, 0^{n-i})$

Type A Crystals

- Arise from representation theory of Lie algebras.
  Example: $\mathrm{sl}_2 = \{M \in \mathrm{Mat}(2) : \mathrm{tr}(M) = 0\}$ generated by:

$$e = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \; f = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \; h = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

- **Type A_n Crystal:** A set $\mathcal{B}$ along with operators

$$e_i, f_i : \mathcal{B} \to \mathcal{B} \cup \{\varnothing\}, \qquad \varepsilon_i, \varphi_i : \mathcal{B} \to \mathbb{Z}$$

  for $1 \leqslant i \leqslant n$, and a map $\mathrm{wt} : \mathcal{B} \to \mathbb{Z}^{n+1}$ such that:

  (1) If $X, Y \in \mathcal{B}$ then $e_i(X) = Y$ iff $f_i(Y) = X$. In this case,

$$\varepsilon_i(Y) = \varepsilon_i(X) - 1, \qquad \varphi_i(Y) = \varphi_i(X) + 1$$

  and $\mathrm{wt}(Y) = \mathrm{wt}(X) + (0^{i-1}, 1, -1, 0^{n-i})$

  (2) For any i and any $X \in \mathcal{B}$, $\varphi_i(X) - \varepsilon_i(X) = \mathrm{wt}(X)_i - \mathrm{wt}(X)_{i+1}$.
  (Often $\varphi_i(X)$ = number of f_i steps that can be applied to X.)

Defining Tableaux crystals: Jeu de Taquin slides

- Inner slide:

$$
\begin{array}{ccccc}
 & \ast & \boxed{1} & \boxed{3} & \boxed{3} \\
 & \boxed{2} & \boxed{2} & & \\
\boxed{1} & \boxed{3} & \boxed{4} & &
\end{array}
$$

Defining Tableaux crystals: Jeu de Taquin slides

- Inner slide:

$$\begin{array}{|c|c|c|c|c|}
\cline{2-2}\cline{4-5}
\multicolumn{1}{c|}{} & 1 & \multicolumn{1}{c|}{} & 3 & 3 \\
\cline{2-5}
\multicolumn{1}{c|}{} & 2 & 2 & \multicolumn{2}{c}{} \\
\hline
1 & 3 & 4 & \multicolumn{2}{c}{} \\
\cline{1-3}
\end{array}$$

Defining Tableaux crystals: Jeu de Taquin slides

- Inner slide:

	1	2	3	3
	2			
1	3	4		

Defining Tableaux crystals: Jeu de Taquin slides

- Inner slide:

$$\begin{array}{ccccc} & \boxed{1} & \boxed{2} & \boxed{3} & \boxed{3} \\ & \boxed{2} & \boxed{4} & & \\ \boxed{1} & \boxed{3} & & & \end{array}$$

Defining Tableaux crystals: Jeu de Taquin slides

- Outer slide:

	1	2	3	3
	2	4		
1	3	*		

Defining Tableaux crystals: Jeu de Taquin slides

- Outer slide:

	1	2	3	3
	2			
1	3	4		

Defining Tableaux crystals: Jeu de Taquin slides

- Outer slide:

	1		3	3
	2	2		
1	3	4		

Defining Tableaux crystals: Jeu de Taquin slides

- Outer slide:

		1	3	3
	2	2		
1	3	4		

Defining Tableaux crystals: Jeu de Taquin slides

- Outer slide:

		1	3	3
	2	2		
1	3	4		

- **Rectification:** Sequence of jeu de taquin slides leading to a straight shape tableau $\mathrm{rect}(T)$.

Defining Tableaux crystals: Jeu de Taquin slides

- Outer slide:

	1		3	3
	2	2		
1	3	4		

- **Rectification:** Sequence of jeu de taquin slides leading to a straight shape tableau $\mathrm{rect}(T)$.

Defining Tableaux crystals: Jeu de Taquin slides

- Outer slide:

	1	2	3	3
	2			
1	3	4		

- **Rectification:** Sequence of jeu de taquin slides leading to a straight shape tableau $\mathrm{rect}(T)$.

Defining Tableaux crystals: Jeu de Taquin slides

- Outer slide:

$$\begin{array}{|c|c|c|c|c|}
\cline{2-5}
\multicolumn{1}{c|}{} & 1 & 2 & 3 & 3 \\
\cline{2-5}
\multicolumn{1}{c|}{} & 2 & 4 & \multicolumn{2}{c}{} \\
\cline{1-3}
1 & 3 & \multicolumn{3}{c}{} \\
\cline{1-2}
\end{array}$$

- **Rectification:** Sequence of jeu de taquin slides leading to a straight shape tableau $\mathrm{rect}(T)$.

Defining Tableaux crystals: Jeu de Taquin slides

- Outer slide:

	1	2	3	3
1	2	4		
	3			

- **Rectification:** Sequence of jeu de taquin slides leading to a straight shape tableau $\mathrm{rect}(T)$.

Defining Tableaux crystals: Jeu de Taquin slides

- Outer slide:

	1	2	3	3
1	2	4		
3				

- **Rectification:** Sequence of jeu de taquin slides leading to a straight shape tableau $\mathrm{rect}(T)$.

Defining Tableaux crystals: Jeu de Taquin slides

- Outer slide:

1	1	2	3	3
	2	4		
3				

- **Rectification:** Sequence of jeu de taquin slides leading to a straight shape tableau $\mathrm{rect}(T)$.

Defining Tableaux crystals: Jeu de Taquin slides

- Outer slide:

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

- **Rectification:** Sequence of jeu de taquin slides leading to a straight shape tableau $\mathrm{rect}(T)$.

Defining Tableaux crystals: Jeu de Taquin slides

- Outer slide:

1	1	2	3	3
2	4			
3				

- **Rectification:** Sequence of jeu de taquin slides leading to a straight shape tableau $\mathrm{rect}(T)$.

Defining Tableaux crystals: Jeu de Taquin slides

- Outer slide:

1	1	2	3	3
2	4			
3				

- **Rectification:** Sequence of jeu de taquin slides leading to a straight shape tableau $\mathrm{rect}(T)$.
- T is **highest weight** if $\mathrm{rect}(T)$ is highest weight for its shape:

1	1	1	1	1
2	2			
3				

Tableaux crystal structure

- Straight shape SSYT's of shape $(5, 3)$ with entries $1, 2$:

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

Tableaux crystal structure

- Straight shape SSYT's of shape $(5,3)$ with entries $1,2$:

$$\emptyset \xleftarrow[E_1]{} \begin{array}{|c|c|c|c|c|}\hline 1&1&1&1&1\\\hline 2&2&2\\\cline{1-3}\end{array} \underset{E_1}{\overset{F_1}{\rightleftarrows}} \begin{array}{|c|c|c|c|c|}\hline 1&1&1&1&2\\\hline 2&2&2\\\cline{1-3}\end{array} \underset{E_1}{\overset{F_1}{\rightleftarrows}} \begin{array}{|c|c|c|c|c|}\hline 1&1&1&2&2\\\hline 2&2&2\\\cline{1-3}\end{array} \xrightarrow{F_1} \emptyset$$

- F_1 changes last 1 to 2 if possible, maps to $\varnothing$ otherwise. E_1 changes first 2 in top row to 1 if possible, maps to $\varnothing$ otherwise.

Tableaux crystal structure

- Straight shape SSYT's of shape $(5,3)$ with entries $1,2$:

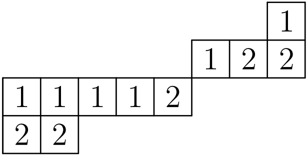
$$\begin{tikzcd} \emptyset & \ytableaushort{11111,222} \arrow[l, "E_1"] \arrow[r, shift left, "F_1"] & \ytableaushort{11112,222} \arrow[l, shift left, "E_1"] \arrow[r, shift left, "F_1"] & \ytableaushort{11122,222} \arrow[l, shift left, "E_1"] \arrow[r, "F_1"] & \emptyset \end{tikzcd}$$

- F_1 changes last 1 to 2 if possible, maps to $\varnothing$ otherwise. E_1 changes first 2 in top row to 1 if possible, maps to $\varnothing$ otherwise.
- Outer slides to get maps F_1, E_1 on skew tableaux:

$$\begin{tikzcd} \emptyset & \ytableaushort{\none1111,1222} \arrow[l, "E_1"] \arrow[r, shift left, "F_1"] & \ytableaushort{\none1112,1222} \arrow[l, shift left, "E_1"] \arrow[r, shift left, "F_1"] & \ytableaushort{\none1112,2222} \arrow[l, shift left, "E_1"] \arrow[r, "F_1"] & \emptyset \end{tikzcd}$$

Tableaux crystal structure

- **Local rule for skew shapes:** In reading word, substitute $2 =$'(' and $1 =$')', match parentheses.

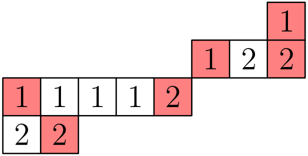


2 2 1 1 1 1 2 1 2 2 1

(( )))) ( ) (( )

Tableaux crystal structure

- **Local rule for skew shapes:** In reading word, substitute 2 ='(' and 1 =')', match parentheses.

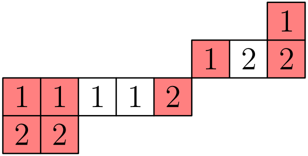


2 2 1 1 1 1 2 1 2 2 1

(( )))) ( ) (( )

Tableaux crystal structure

- **Local rule for skew shapes:** In reading word, substitute 2 =‘(’ and 1 =‘)’, match parentheses.

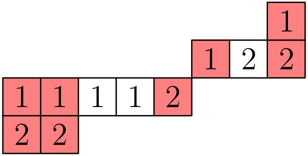


2 2 1 1 1 1 2 1 2 2 1

(( )))) ( ) (( )

Tableaux crystal structure

- **Local rule for skew shapes:** In reading word, substitute 2 =‘(’ and 1 =‘)’, match parentheses.



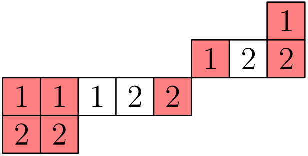
2 2 1 1 1 **1** 2 1 2 2 1

(( )))) ( ) (( )

- F_1 changes the rightmost unmatched 1 to 2 if it exists.

Tableaux crystal structure

- **Local rule for skew shapes:** In reading word, substitute 2 =‘(’ and 1 =‘)’, match parentheses.



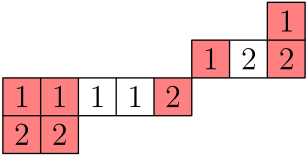
2 2 1 1 1 **2** 2 1 2 2 1

(( )))) ( ) (( )

- F_1 changes the rightmost unmatched 1 to 2 if it exists.
- E_1 changes the leftmost unmatched 2 to 1 if it exists.

Tableaux crystal structure

- **Local rule for skew shapes:** In reading word, substitute $2 =$‘(’ and $1 =$‘)’, match parentheses.



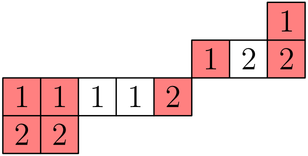
2 2 1 1 1 **1** 2 1 2 2 1

(( )))) ( ) (( )

- F_1 changes the rightmost unmatched 1 to 2 if it exists.
- E_1 changes the leftmost unmatched 2 to 1 if it exists.
- Map to $\varnothing$ otherwise

Tableaux crystal structure

- **Local rule for skew shapes:** In reading word, substitute $2 =$‘(’ and $1 =$‘)’, match parentheses.



2 2 1 1 1 **1** 2 1 2 2 1

(( )))) ( ) (( )

- F_1 changes the rightmost unmatched 1 to 2 if it exists.
- E_1 changes the leftmost unmatched 2 to 1 if it exists.
- Map to $\varnothing$ otherwise
- **Coplactic:** E_1, F_1 commute with all JDT slides

Operators E_i and F_i for general i

- E_i and F_i defined similarly on the subword or subtableau of letters $i, i+1$.

Example

$F_2(1221332) = 1231332$:

$$\begin{array}{ccccccc} 1 & 2 & 2 & 1 & 3 & 3 & 2 \\ &) &) & & (& (&) \end{array}$$

Operators E_i and F_i for general i

- E_i and F_i defined similarly on the subword or subtableau of letters $i, i+1$.

Example

$F_2(1221332) = 1231332$:

$$\begin{array}{ccccccc} 1 & 2 & 2 & 1 & 3 & \color{red}{3} & \color{red}{2} \\ &) &) & & (& \color{red}{(} & \color{red}{)} \end{array}$$

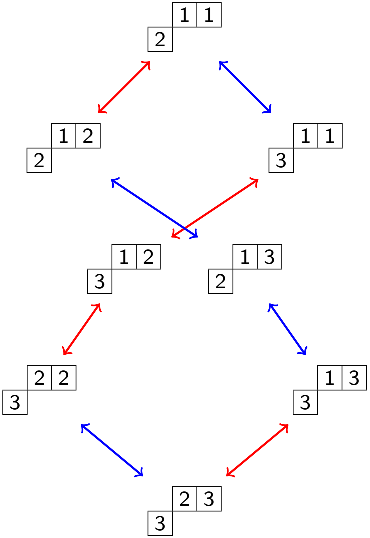
- Highest weight $\iff$ killed by all E_i's (top of crystal graph).
- Skew **Littlewood-Richardson Rule**:

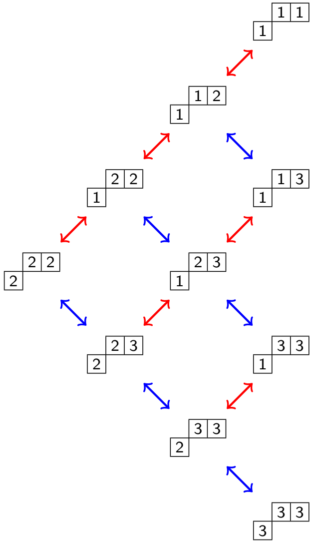
$$s_{\lambda/\mu} = \sum_{\nu} c^{\lambda}_{\mu\nu} s_{\nu}$$

where $c^{\lambda}_{\mu\nu}$ is the number of highest weight tableaux of shape λ/μ and weight ν.

Example

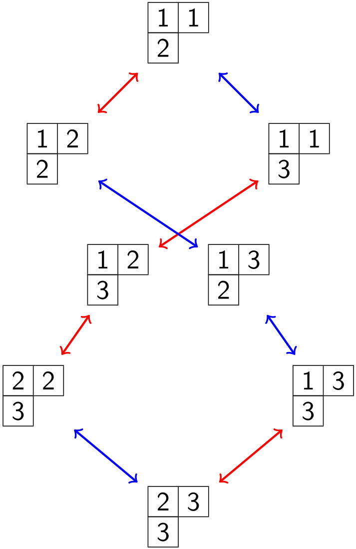
$$s_{(3,1)/(1)} =$$

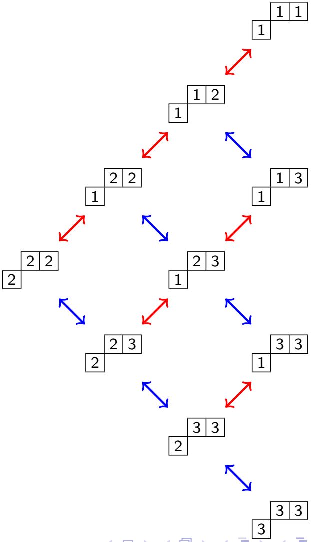




Example

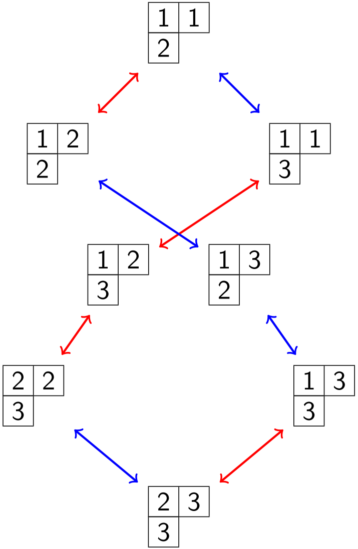
$$s_{(3,1)/(1)} = s_{(2,1)} +$$

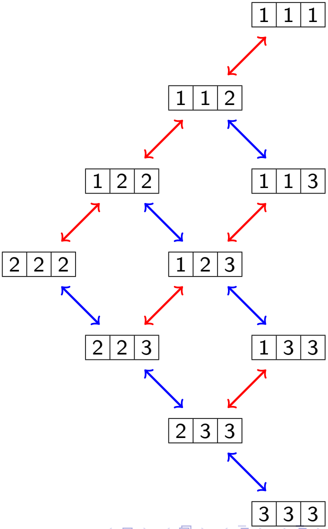




Example

$$s_{(3,1)/(1)} = s_{(2,1)} + s_{(3)}$$





Alternately: Stembridge's axiomatic description

View crystal as a $\mathbb{Z}^n$-weighted, edge-labeled (by $1, \ldots, n-1$) directed graph G. Arrow labeled i is f_i.

Basic structure:

Alternately: Stembridge's axiomatic description

View crystal as a $\mathbb{Z}^n$-weighted, edge-labeled (by $1, \ldots, n-1$) directed graph G. Arrow labeled i is f_i.

Basic structure:

Axiom. Following an edge $w \xrightarrow{i} x$ lowers the weight: $\mathrm{wt}(x) - \mathrm{wt}(w) = \alpha_i = (0 \ldots, -1, 1, \ldots 0)$.

Alternately: Stembridge's axiomatic description

View crystal as a $\mathbb{Z}^n$-weighted, edge-labeled (by $1, \ldots, n-1$) directed graph G. Arrow labeled i is f_i.

Basic structure:

Axiom. Following an edge $w \xrightarrow{i} x$ lowers the weight: $\mathrm{wt}(x) - \mathrm{wt}(w) = \alpha_i = (0 \ldots, -1, 1, \ldots 0)$.

Axiom. For each i, the i-connected components are **strings**:

$$\bullet \xrightarrow{i} \bullet \xrightarrow{i} \bullet \xrightarrow{i} \bullet \xrightarrow{i} \bullet \xrightarrow{i} \bullet$$

Set $\varepsilon_i(w) = \#e_i$-steps available starting from w, $\varphi_i(w) = \#f_i$-steps.

Alternately: Stembridge's axiomatic description

View crystal as a $\mathbb{Z}^n$-weighted, edge-labeled (by $1, \ldots, n-1$) directed graph G. Arrow labeled i is f_i.

Basic structure:

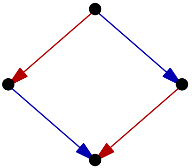
Axiom. Following an edge $w \xrightarrow{i} x$ lowers the weight:
$\mathrm{wt}(x) - \mathrm{wt}(w) = \alpha_i = (0 \ldots, -1, 1, \ldots 0)$.

Axiom. For each i, the i-connected components are **strings**:

$$\bullet \xrightarrow{i} \bullet \xrightarrow{i} \bullet \xrightarrow{i} \bullet \xrightarrow{i} \bullet \xrightarrow{i} \bullet$$

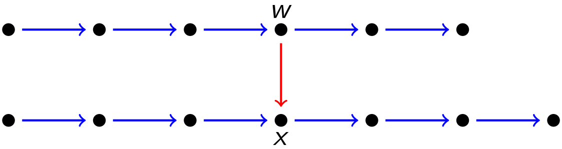
Set $\varepsilon_i(w) = \#e_i$-steps available starting from w,
$\varphi_i(w) = \#f_i$-steps.

Axiom. For $|i - j| > 1$: edges commute:

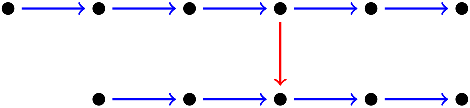


Length axiom

Axiom. Suppose $w \xrightarrow{i\pm1} x$. Then the i-strings passing through w and x are related in one of the following two ways:

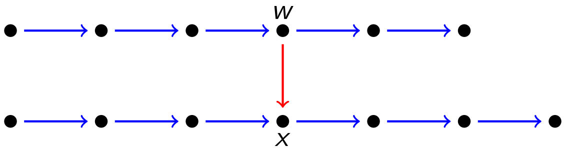


OR

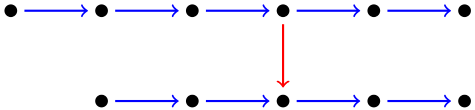


Length axiom

Axiom. Suppose $w \xrightarrow{i\pm1} x$. Then the i-strings passing through w and x are related in one of the following two ways:



OR



$$(\varepsilon_i(w) - \varepsilon_i(x), \varphi_i(w) - \varphi_i(x)) = (0, -1) \text{ or } (1, 0).$$

Axioms relating arrows $\xrightarrow{i}$, $\xrightarrow{i+1}$

Axiom. Suppose $w \xrightarrow{i} x$ and $w \xrightarrow{i+1} y$. Compare ε values:

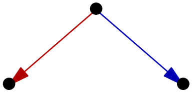
$$\begin{aligned}\Delta :=&(\varepsilon_{i+1}(w) - \varepsilon_{i+1}(x), \varepsilon_i(w) - \varepsilon_i(y))\\ =&(1,1), (1,0), (0,1) \text{ or } (0,0).\end{aligned}$$

Axioms relating arrows $\xrightarrow{i}, \xrightarrow{i+1}$

Axiom. Suppose $w \xrightarrow{i} x$ and $w \xrightarrow{i+1} y$. Compare ε values:

$$\begin{aligned}\Delta :=&(\varepsilon_{i+1}(w) - \varepsilon_{i+1}(x), \varepsilon_i(w) - \varepsilon_i(y))\\ =&(1,1), (1,0), (0,1) \text{ or } (0,0).\end{aligned}$$

Then:

$\Delta \neq (0,0)$	$\Delta = (0,0)$
	

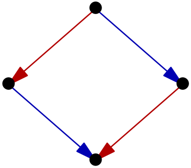
Also: dual versions of same axioms (for going backwards).

Axioms relating arrows $\xrightarrow{i}$, $\xrightarrow{i+1}$

Axiom. Suppose $w \xrightarrow{i} x$ and $w \xrightarrow{i+1} y$. Compare ε values:

$$\begin{aligned}\Delta :=& (\varepsilon_{i+1}(w) - \varepsilon_{i+1}(x), \varepsilon_i(w) - \varepsilon_i(y)) \\ =& (1,1), (1,0), (0,1) \text{ or } (0,0).\end{aligned}$$

Then:

$\Delta \neq (0,0)$	$\Delta = (0,0)$
	

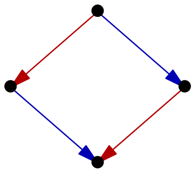
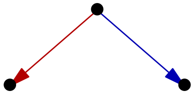
Also: dual versions of same axioms (for going backwards).

Axioms relating arrows $\xrightarrow{i}$, $\xrightarrow{i+1}$

Axiom. Suppose $w \xrightarrow{i} x$ and $w \xrightarrow{i+1} y$. Compare ε values:

$$\begin{aligned}\Delta :=& (\varepsilon_{i+1}(w) - \varepsilon_{i+1}(x), \varepsilon_i(w) - \varepsilon_i(y)) \\ =& (1,1), (1,0), (0,1) \text{ or } (0,0).\end{aligned}$$

Then:

$\Delta \neq (0,0)$	$\Delta = (0,0)$
	

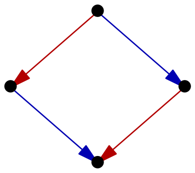
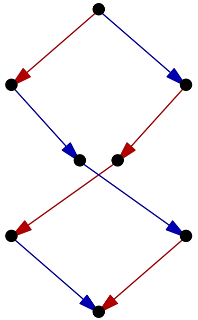
Also: dual versions of same axioms (for going backwards).

Axioms relating arrows $\xrightarrow{i}, \xrightarrow{i+1}$

Axiom. Suppose $w \xrightarrow{i} x$ and $w \xrightarrow{i+1} y$. Compare ε values:

$$\begin{aligned}\Delta :=& (\varepsilon_{i+1}(w) - \varepsilon_{i+1}(x), \varepsilon_i(w) - \varepsilon_i(y)) \\ =& (1,1), (1,0), (0,1) \text{ or } (0,0).\end{aligned}$$

Then:

$\Delta \neq (0,0)$	$\Delta = (0,0)$
	

Also: dual versions of same axioms (for going backwards).

Stembridge axioms

Theorem (Stembridge '03)

Let G be a finite connected graph satisfying the local axioms.

Then G has a unique highest-weight element g^*, with $\mathrm{wt}(g^*) = \lambda$ a partition, and canonically $G \cong SSYT(\lambda, n)$.

Stembridge axioms

Theorem (Stembridge '03)

Let G be a finite connected graph satisfying the local axioms.

Then G has a unique highest-weight element g^*, with $\mathrm{wt}(g^*) = \lambda$ a partition, and canonically $G \cong SSYT(\lambda, n)$.

- **Corollary** (without 'connected' hypothesis):
  Weight generating function of G is Schur-positive!

  (each connected component $\leftrightarrow$ some $SSYT(\lambda, n)$).

Stembridge axioms

Theorem (Stembridge '03)

Let G be a finite connected graph satisfying the local axioms.

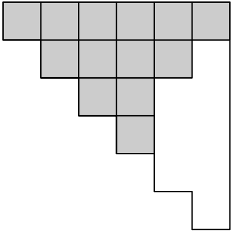
Then G has a unique highest-weight element g^*, with $\mathrm{wt}(g^*) = \lambda$ a partition, and canonically $G \cong SSYT(\lambda, n)$.

- **Corollary** (without 'connected' hypothesis):
  Weight generating function of G is Schur-positive!
  (each connected component $\leftrightarrow$ some $SSYT(\lambda, n)$).

- Morse-Schilling (2015): Crystal-theoretic proof of Schur positivity for Stanley symmetric functions

Part 2: Shifted tableaux

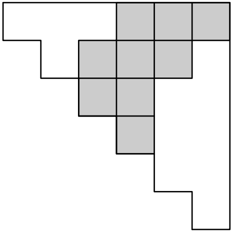
- **Shifted partitions:** Partitions with distinct parts; *i*th row shifted to the right *i* steps.



$$\lambda = (6, 4, 2, 1)$$

Part 2: Shifted tableaux

- **Shifted partitions:** Partitions with distinct parts; *i*th row shifted to the right *i* steps.

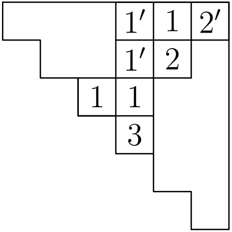


$$\lambda = (6, 4, 2, 1)$$
$$\mu = (3, 1)$$

- **Skew shape:** λ/μ

Part 2: Shifted tableaux

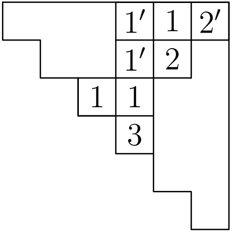
- **Shifted partitions:** Partitions with distinct parts; *i*th row shifted to the right *i* steps.



- **Skew shape:** λ/μ
- **Semistandard tableau:** $1' < 1 < 2' < 2 < 3' < 3 < \cdots$ is alphabet, entries weakly increasing down and right. Primed letters can only repeat in columns and unprimed only in rows.

Part 2: Shifted tableaux

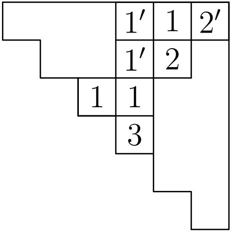
- **Shifted partitions:** Partitions with distinct parts; ith row shifted to the right i steps.



- **Skew shape:** λ/μ
- **Semistandard tableau:** $1' < 1 < 2' < 2 < 3' < 3 < \cdots$ is alphabet, entries weakly increasing down and right. Primed letters can only repeat in columns and unprimed only in rows.
- **Canonical form:** Southwest-most i or i' is always *unprimed*.

Part 2: Shifted tableaux

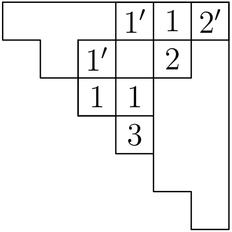
- **Jeu de Taquin:** (Sagan, Worley.) Primed letters lose their primes when sliding into the diagonal.



- **Skew shape:** λ/μ
- **Semistandard tableau:** $1' < 1 < 2' < 2 < 3' < 3 < \cdots$ is alphabet, entries weakly increasing down and right. Primed letters can only repeat in columns and unprimed only in rows.
- **Canonical form:** Southwest-most *i* or *i*′ is always *unprimed*.

Part 2: Shifted tableaux

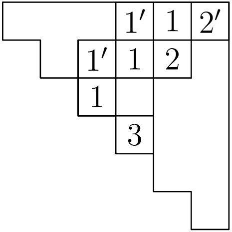
- **Jeu de Taquin:** (Sagan, Worley.) Primed letters lose their primes when sliding into the diagonal.



- **Skew shape:** λ/μ
- **Semistandard tableau:** $1' < 1 < 2' < 2 < 3' < 3 < \cdots$ is alphabet, entries weakly increasing down and right. Primed letters can only repeat in columns and unprimed only in rows.
- **Canonical form:** Southwest-most *i* or *i′* is always *unprimed*.

Part 2: Shifted tableaux

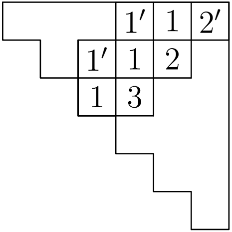
- **Jeu de Taquin:** (Sagan, Worley.) Primed letters lose their primes when sliding into the diagonal.



- **Skew shape:** λ/μ
- **Semistandard tableau:** $1' < 1 < 2' < 2 < 3' < 3 < \cdots$ is alphabet, entries weakly increasing down and right. Primed letters can only repeat in columns and unprimed only in rows.
- **Canonical form:** Southwest-most *i* or *i′* is always *unprimed*.

Part 2: Shifted tableaux

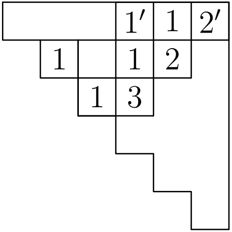
- **Jeu de Taquin:** (Sagan, Worley.) Primed letters lose their primes when sliding into the diagonal.



- **Skew shape:** λ/μ
- **Semistandard tableau:** $1' < 1 < 2' < 2 < 3' < 3 < \cdots$ is alphabet, entries weakly increasing down and right. Primed letters can only repeat in columns and unprimed only in rows.
- **Canonical form:** Southwest-most i or i' is always *unprimed*.

Part 2: Shifted tableaux

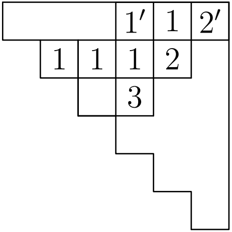
- **Jeu de Taquin:** (Sagan, Worley.) Primed letters lose their primes when sliding into the diagonal.



- **Skew shape:** λ/μ
- **Semistandard tableau:** $1' < 1 < 2' < 2 < 3' < 3 < \cdots$ is alphabet, entries weakly increasing down and right. Primed letters can only repeat in columns and unprimed only in rows.
- **Canonical form:** Southwest-most i or i' is always *unprimed*.

Part 2: Shifted tableaux

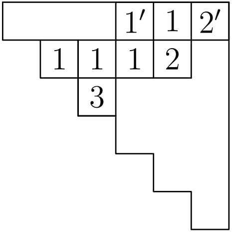
- **Jeu de Taquin:** (Sagan, Worley.) Primed letters lose their primes when sliding into the diagonal.



- **Skew shape:** λ/μ
- **Semistandard tableau:** $1' < 1 < 2' < 2 < 3' < 3 < \cdots$ is alphabet, entries weakly increasing down and right. Primed letters can only repeat in columns and unprimed only in rows.
- **Canonical form:** Southwest-most i or i' is always *unprimed*.

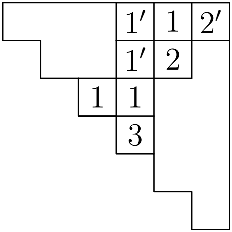
Part 2: Shifted tableaux

- **Jeu de Taquin:** (Sagan, Worley.) Primed letters lose their primes when sliding into the diagonal.



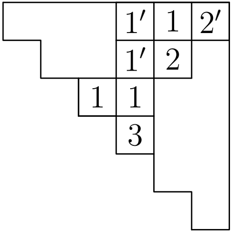
- **Skew shape:** λ/μ
- **Semistandard tableau:** $1' < 1 < 2' < 2 < 3' < 3 < \cdots$ is alphabet, entries weakly increasing down and right. Primed letters can only repeat in columns and unprimed only in rows.
- **Canonical form:** Southwest-most *i* or *i'* is always *unprimed*.

Schur Q-functions (Hall-Littlewood, Stembridge,...)



- **Weight:** $\mathrm{wt}(T) = (m_1, m_2, \ldots)$ where m_i is the number of i and i' entries. Above, $(5, 2, 1)$.
- **Highest weight:** JDT rectification has weight equal to shape

Schur Q-functions (Hall-Littlewood, Stembridge,...)

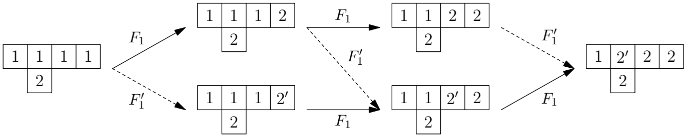


- **Weight:** $\mathrm{wt}(T) = (m_1, m_2, \ldots)$ where m_i is the number of i and i' entries. Above, $(5, 2, 1)$.
- **Highest weight:** JDT rectification has weight equal to shape
- **Character:** $x^T = 2^{\ell(T)} x_1^{m_1} x_2^{m_2} \cdots$ where $\ell(T)$ is the number of nonzero entries in $\mathrm{wt}(T)$. Above, $x^T = 8x_1^5 x_2^2 x_3$.
- **Schur Q-functions:**

$$Q_{\lambda/\mu}(x_1, x_2, \ldots) = \sum_{T \in \mathrm{ShST}(\lambda/\mu)} x^T$$

Operators E_i, F_i, E_i', F_i' (G., Levinson, Purbhoo.)

- Consider $i = 1$ as before. Restrict to alphabet $\{1', 1, 2', 2\}$.
  Shape can have two rows:

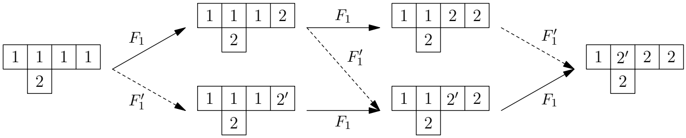


  Or one row:

$$\boxed{1\,|\,1\,|\,1\,|\,1} \underset{F_1'}{\overset{F_1}{\longrightarrow}} \boxed{1\,|\,1\,|\,1\,|\,2} \underset{F_1'}{\overset{F_1}{\longrightarrow}} \boxed{1\,|\,1\,|\,2\,|\,2} \underset{F_1'}{\overset{F_1}{\longrightarrow}} \boxed{1\,|\,2\,|\,2\,|\,2} \underset{F_1'}{\overset{F_1}{\longrightarrow}} \boxed{2\,|\,2\,|\,2\,|\,2} \underset{F_1'}{\overset{F_1}{\longrightarrow}} \emptyset$$

Operators E_i, F_i, E_i', F_i' (G., Levinson, Purbhoo.)

- Consider $i = 1$ as before. Restrict to alphabet $\{1', 1, 2', 2\}$. Shape can have two rows:



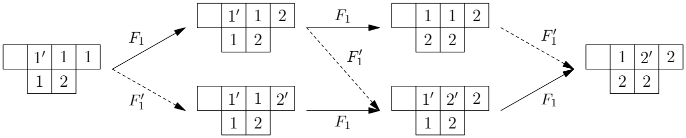
Or one row:

$$\boxed{1\,|\,1\,|\,1\,|\,1} \xrightarrow[F_1']{F_1} \boxed{1\,|\,1\,|\,1\,|\,2} \xrightarrow[F_1']{F_1} \boxed{1\,|\,1\,|\,2\,|\,2} \xrightarrow[F_1']{F_1} \boxed{1\,|\,2\,|\,2\,|\,2} \xrightarrow[F_1']{F_1} \boxed{2\,|\,2\,|\,2\,|\,2} \xrightarrow[F_1']{F_1} \emptyset$$

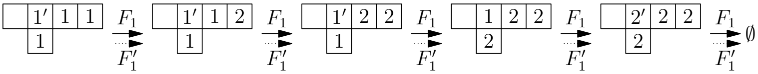
- Extend to skew shapes by applying outer slides.

Operators E_i, F_i, E_i', F_i' (G., Levinson, Purbhoo.)

- Consider $i = 1$ as before. Restrict to alphabet $\{1', 1, 2', 2\}$. Shape can have two rows:



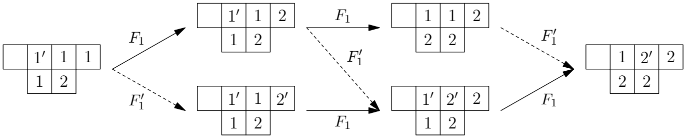
Or one row:



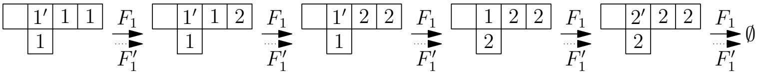
- Extend to skew shapes by applying outer slides.

Operators E_i, F_i, E_i', F_i' (G., Levinson, Purbhoo.)

- Consider $i = 1$ as before. Restrict to alphabet $\{1', 1, 2', 2\}$. Shape can have two rows:



Or one row:

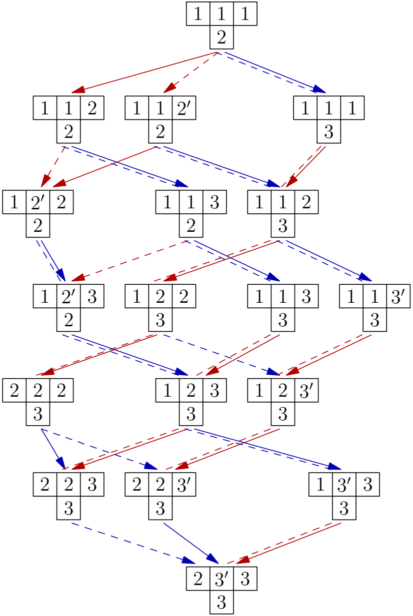


- Extend to skew shapes by applying outer slides.
- **Theorem.** (G., Levinson, Purbhoo, 2016.) There are local, fast ($O(n)$) combinatorial rules for these operators that do not require JDT, similar to parentheses rule for ordinary tableaux. (arxiv:1706.09969)

Crystal-like structure

- "Crystal graph" for $i = 1, 2$:

$$\boxed{F_1 \quad F_1' \qquad F_2 \quad F_2'}$$

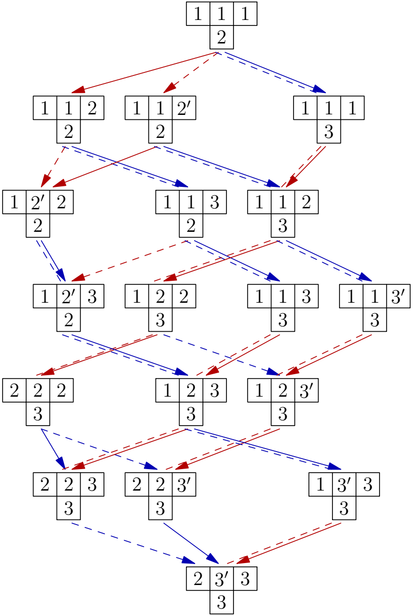


Crystal-like structure

- "Crystal graph" for $i = 1, 2$:

$$\boxed{\color{red}F_1 \quad F_1' \qquad \color{blue}F_2 \quad F_2'}$$

- **Results:**
  - Graph structure implies Q_λ is symmetric.

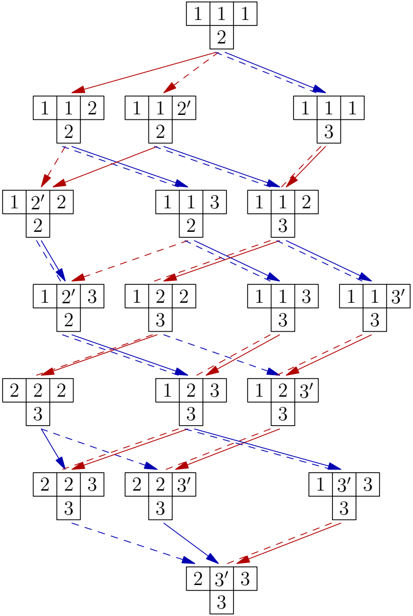


Crystal-like structure

- "Crystal graph" for $i = 1, 2$:

  $\boxed{F_1 \quad F_1' \qquad F_2 \quad F_2'}$

- **Results:**
  - Graph structure implies Q_λ is symmetric.
  - T is highest weight iff $E_i(T) = E_i'(T) = \varnothing$ for all i.

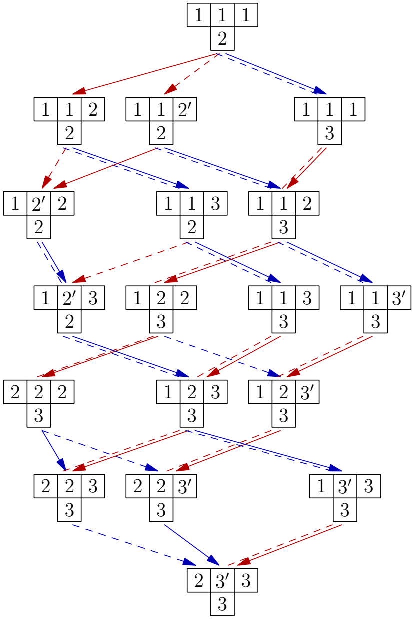


Crystal-like structure

- "Crystal graph" for $i = 1, 2$:

$$F_1 \quad F_1' \qquad F_2 \quad F_2'$$

- **Results:**
  - Graph structure implies Q_λ is symmetric.
  - T is highest weight iff $E_i(T) = E_i'(T) = \varnothing$ for all i.
  - Connected components have unique highest weights

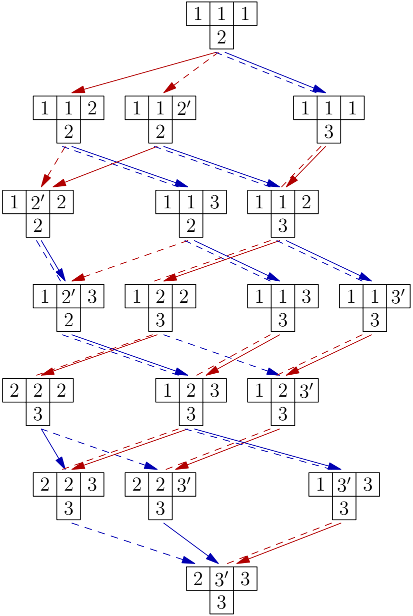


Crystal-like structure

- "Crystal graph" for $i = 1, 2$:

$$\boxed{\color{red}{F_1 \quad F_1'} \qquad \color{blue}{F_2 \quad F_2'}}$$

- **Results:**
  - Graph structure implies Q_λ is symmetric.
  - T is highest weight iff $E_i(T) = E_i'(T) = \varnothing$ for all i.
  - Connected components have unique highest weights
  - Gives LR decomposition $Q_{\lambda/\mu} = \sum_\nu f^\lambda_{\mu\nu} Q_\nu$.



Axioms for shifted tableau crystals

$G = \mathbb{Z}^n$-weighted digraph, edge labels $\{1', 1, \ldots, n{-}1', n{-}1\}$.

Basic structure:

Axioms for shifted tableau crystals

$G = \mathbb{Z}^n$-weighted digraph, edge labels $\{1', 1, \ldots, n{-}1', n{-}1\}$.

Basic structure:

Axiom. Following an edge $w \xrightarrow{i \text{ or } i'} x$ lowers the weight: $\mathrm{wt}(x) - \mathrm{wt}(w) = \alpha_i = (0\ldots, -1, 1\ldots 0)$.

Axiom. For $|i-j| > 1$, all i, j, i', j' edges commute.

Axioms for shifted tableau crystals

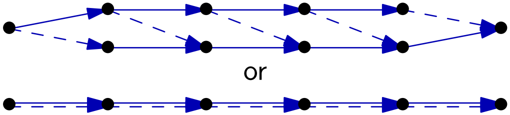
$G = \mathbb{Z}^n$-weighted digraph, edge labels $\{1', 1, \ldots, n{-}1', n{-}1\}$.

Basic structure:

Axiom. Following an edge $w \xrightarrow{i \text{ or } i'} x$ lowers the weight: $\mathrm{wt}(x) - \mathrm{wt}(w) = \alpha_i = (0\ldots, -1, 1\ldots 0)$.

Axiom. For $|i-j| > 1$, all i, j, i', j' edges commute.

Axiom. The $\{i, i'\}$-connected components are **doubled strings**:



(iff $\mathrm{wt}_i = 0$ at bottom)

Axioms for shifted tableau crystals

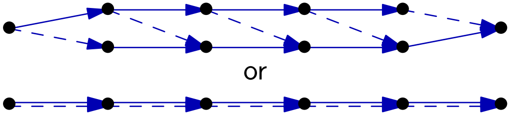
$G = \mathbb{Z}^n$-weighted digraph, edge labels $\{1', 1, \ldots, n{-}1', n{-}1\}$.

Basic structure:

Axiom. Following an edge $w \xrightarrow{i \text{ or } i'} x$ lowers the weight: $\mathrm{wt}(x) - \mathrm{wt}(w) = \alpha_i = (0\ldots, -1, 1\ldots 0)$.

Axiom. For $|i-j| > 1$, all i, j, i', j' edges commute.

Axiom. The $\{i, i'\}$-connected components are **doubled strings**:



(iff $\mathrm{wt}_i = 0$ at bottom)

Set $\varepsilon_i(w), \varphi_i(w)$ = **total** # steps to top, bottom
Also $\varepsilon_i', \varphi_i'$ ($\#\{\dashrightarrow\}$ only) and $\widehat{\varepsilon}_i, \widehat{\varphi}_i$ ($\#\{\longrightarrow\}$ only).

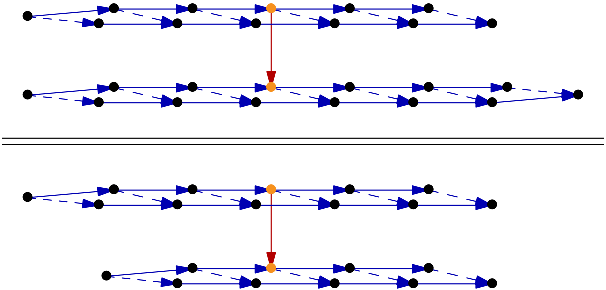
Length Axiom for shifted tableau crystals

Length Axiom for shifted tableau crystals

Axiom. Suppose $w \xrightarrow{i\pm1 \text{ or } i\pm1'} x$. Then

$$(\varepsilon_i(w) - \varepsilon_i(x), \varphi_i(w) - \varphi_i(x)) = (0, -1) \text{ or } (1, 0).$$

Same two possibilities as Stembridge:



Relations between $\xrightarrow{1 \text{ or } 1'}$, $\xrightarrow{2 \text{ or } 2'}$

Suppose we have two edges: $w \xrightarrow{1 \text{ or } 1'} x$, $w \xrightarrow{2 \text{ or } 2'} y$.

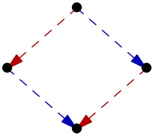
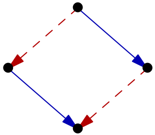
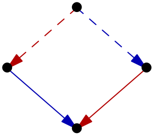
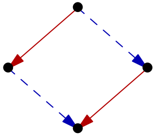
Define:
$$\begin{aligned} \Delta &= (\varepsilon_2(w) - \varepsilon_2(x), \varepsilon_1(w) - \varepsilon_1(y)) \\ &= (1,1), (1,0), (0,1) \text{ or } (0,0). \end{aligned}$$

Relations between $\xrightarrow{1 \text{ or } 1'}$, $\xrightarrow{2 \text{ or } 2'}$

Suppose we have two edges: $w \xrightarrow{1 \text{ or } 1'} x$, $w \xrightarrow{2 \text{ or } 2'} y$.

Define: $$\begin{aligned}\Delta &= (\varepsilon_2(w) - \varepsilon_2(x), \varepsilon_1(w) - \varepsilon_1(y)) \\ &= (1,1), (1,0), (0,1) \text{ or } (0,0).\end{aligned}$$

Axioms. For $\{f_1', f_2'\}$, $\{f_1, f_2'\}$, $\{f_1', f_2\}$ (assume $f_2 \neq f_2'$):

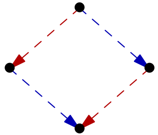
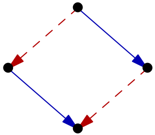
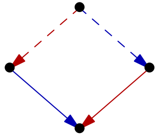
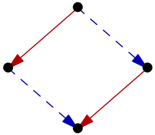
Conditions	Axiom	Conditions	Axiom
–		–	
$\Delta = (0,0)$, $\varphi_2(w) = 1$, $\widehat{\varphi_2}(w) = 0$		$\widehat{\epsilon}_1(w) > 0$	

Relations between $\xrightarrow{1 \text{ or } 1'}$, $\xrightarrow{2 \text{ or } 2'}$

Suppose we have two edges: $w \xrightarrow{1 \text{ or } 1'} x$, $w \xrightarrow{2 \text{ or } 2'} y$.

Define:
$$\begin{aligned} \Delta &= (\varepsilon_2(w) - \varepsilon_2(x), \varepsilon_1(w) - \varepsilon_1(y)) \\ &= (1,1), (1,0), (0,1) \text{ or } (0,0). \end{aligned}$$

Axioms. For $\{f_1', f_2'\}$, $\{f_1, f_2'\}$, $\{f_1', f_2\}$ (assume $f_2 \neq f_2'$):

Conditions	Axiom	Conditions	Axiom
–		–	
$\Delta = (0,0)$, $\varphi_2(w) = 1$, $\widehat{\varphi_2}(w) = 0$		$\hat{\epsilon}_1(w) > 0$	

Note: No axiom for $\{f_1, f_2'\}$ when $\widehat{\varepsilon}_1(w) = 0$!

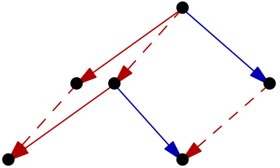
Relations between $\xrightarrow{1}$, $\xrightarrow{2}$

The interesting case: $w \xrightarrow{1} x, w \xrightarrow{2} y$.

Relations between $\xrightarrow{1}$, $\xrightarrow{2}$

The interesting case: $w \xrightarrow{1} x, w \xrightarrow{2} y$.

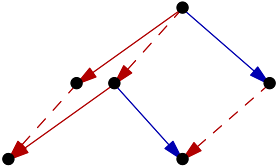
Observation. By applying previous axioms, reduce to case where f_1' ($\dashrightarrow$) **not** defined.



Relations between $\xrightarrow{1}$, $\xrightarrow{2}$

The interesting case: $w \xrightarrow{1} x, w \xrightarrow{2} y$.

Observation. By applying previous axioms, reduce to case where f_1' ($\dashrightarrow$) **not** defined.

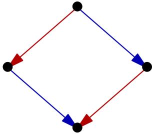
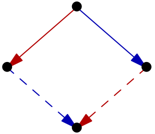
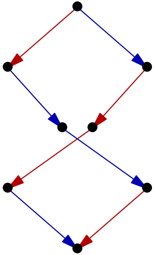
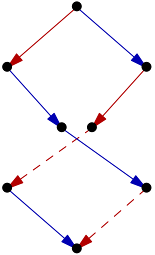


Then, the axioms for $\{f_1, f_2\}$ resemble Stembridge axioms!

Relations between $\xrightarrow{1}$, $\xrightarrow{2}$

Assume f_1' is not defined at w, and $f_2 \neq f_2'$ ('strict' solid edge).

Axioms for $\{f_1, f_2\}$:

Conditions	Axiom	Conditions	Axiom
$\Delta = (0,1),\ (1,0)$		$\Delta = (1,1)$	
$\Delta = (0,0)$, $\widehat{\varphi}_1(f_2(w)) \geqslant 2$		$\Delta = (0,0)$, $\widehat{\varphi}_1(f_2(w)) < 2$	

Uniqueness of shifted tableau crystals

Theorem (G., Levinson)

Let G be a finite connected graph satisfying these local axioms.

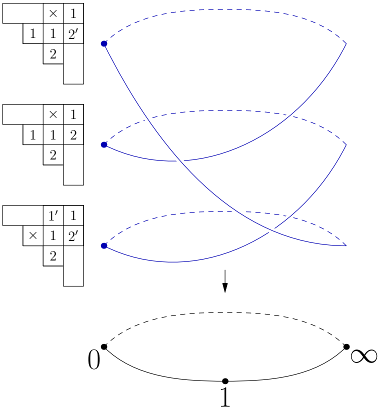
- *G has a unique maximal element g^*, and $\mathrm{wt}(g^*) = \sigma$ is a* **strict** *partition.*
- *There is a canonical isomorphism $ShSYT(\sigma, n) \xrightarrow{\sim} G$.*

In particular, the generating function

$$\sum_{g \in G} 2^{\ell(\mathrm{wt}(g))} x^{\mathrm{wt}(g)}$$

is Schur Q positive.

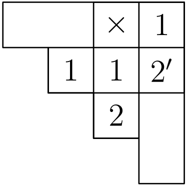
Application: Type B Schubert curves



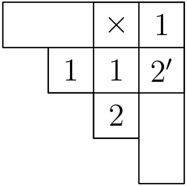
- Zero fiber labeled by tableaux of a skew shifted shape consisting of a marked inner corner ('$\times$') and a highest weight tableau of a fixed weight ν.

The monodromy operator (G., Levinson, Purbhoo)

- Monodromy operator ω is given by commutator of rectification with jeu de taquin (based on work of Levinson, Speyer):
  1. **Rectify**, with $\boxtimes = 0$
  2. **Slide** the $\boxtimes$ to an outer corner with an outer JDT slide
  3. **Unrectify** to the original shape, with $\boxtimes = \infty$
  4. **Slide** the $\boxtimes$ back to an inner corner

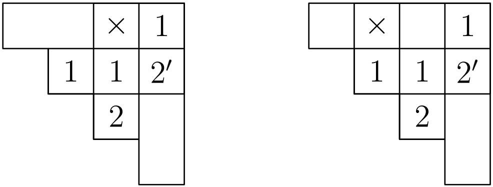


T



The monodromy operator (G., Levinson, Purbhoo)

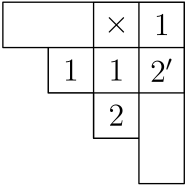
- Monodromy operator ω is given by commutator of rectification with jeu de taquin (based on work of Levinson, Speyer):
    1. **Rectify**, with $\boxtimes = 0$
    2. **Slide** the $\boxtimes$ to an outer corner with an outer JDT slide
    3. **Unrectify** to the original shape, with $\boxtimes = \infty$
    4. **Slide** the $\boxtimes$ back to an inner corner



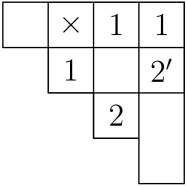
T

The monodromy operator (G., Levinson, Purbhoo)

- Monodromy operator ω is given by commutator of rectification with jeu de taquin (based on work of Levinson, Speyer):
  1. **Rectify**, with $\boxtimes = 0$
  2. **Slide** the $\boxtimes$ to an outer corner with an outer JDT slide
  3. **Unrectify** to the original shape, with $\boxtimes = \infty$
  4. **Slide** the $\boxtimes$ back to an inner corner

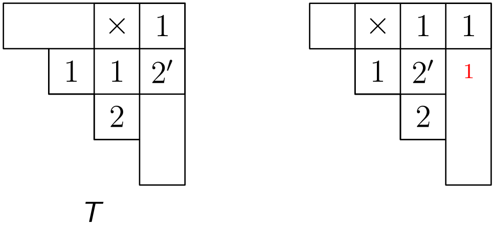


T



The monodromy operator (G., Levinson, Purbhoo)

- Monodromy operator ω is given by commutator of rectification with jeu de taquin (based on work of Levinson, Speyer):
  1. **Rectify**, with $\boxtimes = 0$
  2. **Slide** the $\boxtimes$ to an outer corner with an outer JDT slide
  3. **Unrectify** to the original shape, with $\boxtimes = \infty$
  4. **Slide** the $\boxtimes$ back to an inner corner



The monodromy operator (G., Levinson, Purbhoo)

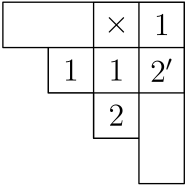
- Monodromy operator ω is given by commutator of rectification with jeu de taquin (based on work of Levinson, Speyer):
  1. **Rectify**, with $\boxtimes = 0$
  2. **Slide** the $\boxtimes$ to an outer corner with an outer JDT slide
  3. **Unrectify** to the original shape, with $\boxtimes = \infty$
  4. **Slide** the $\boxtimes$ back to an inner corner

$$\begin{array}{cccc} & & \times & 1 \\ & 1 & 1 & 2' \\ & & 2 & \\ & & & \end{array} \qquad\qquad \begin{array}{cccc} \times & & 1 & 1 \\ & 1 & 2' & {\color{red}1} \\ & & 2 & \\ & & & \end{array}$$

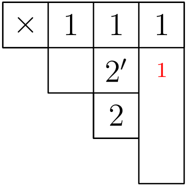
T

The monodromy operator (G., Levinson, Purbhoo)

- Monodromy operator ω is given by commutator of rectification with jeu de taquin (based on work of Levinson, Speyer):
  1. **Rectify**, with $\boxtimes = 0$
  2. **Slide** the $\boxtimes$ to an outer corner with an outer JDT slide
  3. **Unrectify** to the original shape, with $\boxtimes = \infty$
  4. **Slide** the $\boxtimes$ back to an inner corner



T



The monodromy operator (G., Levinson, Purbhoo)

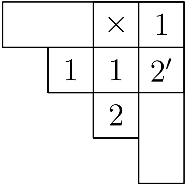
- Monodromy operator ω is given by commutator of rectification with jeu de taquin (based on work of Levinson, Speyer):
  1. **Rectify**, with $\boxtimes = 0$
  2. **Slide** the $\boxtimes$ to an outer corner with an outer JDT slide
  3. **Unrectify** to the original shape, with $\boxtimes = \infty$
  4. **Slide** the $\boxtimes$ back to an inner corner

$$\begin{array}{|c|c|c|c|}
\hline
 & & \times & 1 \\
\hline
 & 1 & 1 & 2' \\
\hline
 & & 2 & \\
\hline
 & & & \\
\hline
\end{array}
\qquad\qquad
\begin{array}{|c|c|c|c|}
\hline
\times & 1 & 1 & 1 \\
\hline
 & 2 & & \color{red}{1} \\
\hline
 & & 2 & \\
\hline
 & & & \\
\hline
\end{array}$$

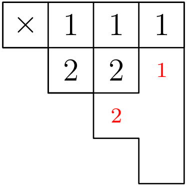
T

The monodromy operator (G., Levinson, Purbhoo)

- Monodromy operator ω is given by commutator of rectification with jeu de taquin (based on work of Levinson, Speyer):
  1. **Rectify**, with $\boxtimes = 0$
  2. **Slide** the $\boxtimes$ to an outer corner with an outer JDT slide
  3. **Unrectify** to the original shape, with $\boxtimes = \infty$
  4. **Slide** the $\boxtimes$ back to an inner corner



T



The monodromy operator (G., Levinson, Purbhoo)

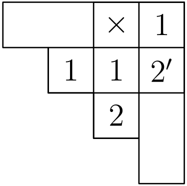
- Monodromy operator ω is given by commutator of rectification with jeu de taquin (based on work of Levinson, Speyer):
  1. **Rectify**, with $\boxtimes = 0$
  2. **Slide** the $\boxtimes$ to an outer corner with an outer JDT slide
  3. **Unrectify** to the original shape, with $\boxtimes = \infty$
  4. **Slide** the $\boxtimes$ back to an inner corner

$$\begin{array}{cccc} & & \times & 1 \\ & 1 & 1 & 2' \\ & & 2 & \\ & & & \end{array} \qquad\qquad \begin{array}{cccc} 1 & \times & 1 & 1 \\ & 2 & 2 & {\color{red}1} \\ & & {\color{red}2} & \\ & & & \end{array}$$

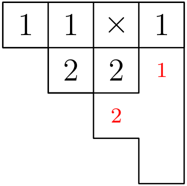
T

The monodromy operator (G., Levinson, Purbhoo)

- Monodromy operator ω is given by commutator of rectification with jeu de taquin (based on work of Levinson, Speyer):
  1. **Rectify**, with $\boxtimes = 0$
  2. **Slide** the $\boxtimes$ to an outer corner with an outer JDT slide
  3. **Unrectify** to the original shape, with $\boxtimes = \infty$
  4. **Slide** the $\boxtimes$ back to an inner corner

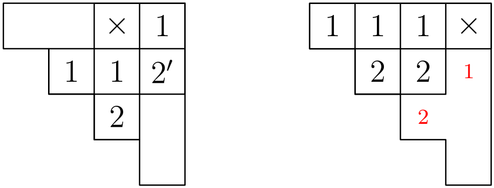


T



The monodromy operator (G., Levinson, Purbhoo)

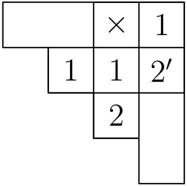
- Monodromy operator ω is given by commutator of rectification with jeu de taquin (based on work of Levinson, Speyer):
  1. **Rectify**, with $\boxtimes = 0$
  2. **Slide** the $\boxtimes$ to an outer corner with an outer JDT slide
  3. **Unrectify** to the original shape, with $\boxtimes = \infty$
  4. **Slide** the $\boxtimes$ back to an inner corner



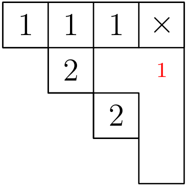
T

The monodromy operator (G., Levinson, Purbhoo)

- Monodromy operator ω is given by commutator of rectification with jeu de taquin (based on work of Levinson, Speyer):
  1. **Rectify**, with $\boxtimes = 0$
  2. **Slide** the $\boxtimes$ to an outer corner with an outer JDT slide
  3. **Unrectify** to the original shape, with $\boxtimes = \infty$
  4. **Slide** the $\boxtimes$ back to an inner corner

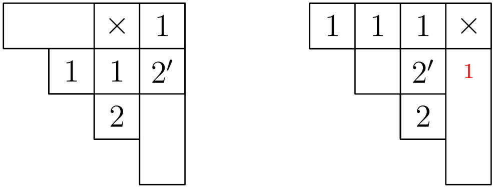


T



The monodromy operator (G., Levinson, Purbhoo)

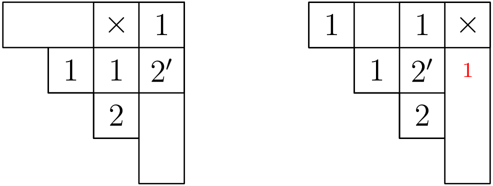
- Monodromy operator ω is given by commutator of rectification with jeu de taquin (based on work of Levinson, Speyer):
  1. **Rectify**, with $\boxtimes = 0$
  2. **Slide** the $\boxtimes$ to an outer corner with an outer JDT slide
  3. **Unrectify** to the original shape, with $\boxtimes = \infty$
  4. **Slide** the $\boxtimes$ back to an inner corner



T

The monodromy operator (G., Levinson, Purbhoo)

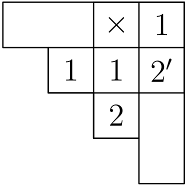
- Monodromy operator ω is given by commutator of rectification with jeu de taquin (based on work of Levinson, Speyer):
  1. **Rectify**, with $\boxtimes = 0$
  2. **Slide** the $\boxtimes$ to an outer corner with an outer JDT slide
  3. **Unrectify** to the original shape, with $\boxtimes = \infty$
  4. **Slide** the $\boxtimes$ back to an inner corner



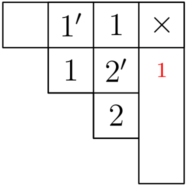
T

The monodromy operator (G., Levinson, Purbhoo)

- Monodromy operator ω is given by commutator of rectification with jeu de taquin (based on work of Levinson, Speyer):
  1. **Rectify**, with $\boxtimes = 0$
  2. **Slide** the $\boxtimes$ to an outer corner with an outer JDT slide
  3. **Unrectify** to the original shape, with $\boxtimes = \infty$
  4. **Slide** the $\boxtimes$ back to an inner corner

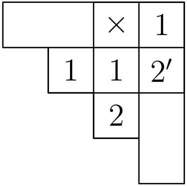


T

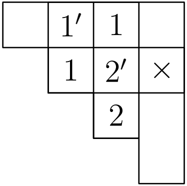


The monodromy operator (G., Levinson, Purbhoo)

- Monodromy operator ω is given by commutator of rectification with jeu de taquin (based on work of Levinson, Speyer):
  1. **Rectify**, with $\boxtimes = 0$
  2. **Slide** the $\boxtimes$ to an outer corner with an outer JDT slide
  3. **Unrectify** to the original shape, with $\boxtimes = \infty$
  4. **Slide** the $\boxtimes$ back to an inner corner



T



The monodromy operator (G., Levinson, Purbhoo)

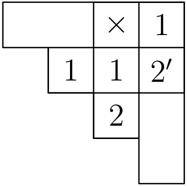
- Monodromy operator ω is given by commutator of rectification with jeu de taquin (based on work of Levinson, Speyer):
    1. **Rectify**, with $\boxtimes = 0$
    2. **Slide** the $\boxtimes$ to an outer corner with an outer JDT slide
    3. **Unrectify** to the original shape, with $\boxtimes = \infty$
    4. **Slide** the $\boxtimes$ back to an inner corner

$$\begin{array}{cccc} \cdot & \cdot & \times & 1 \\ & 1 & 1 & 2' \\ & & 2 & \; \\ & & & \; \end{array} \qquad\qquad \begin{array}{cccc} \cdot & 1' & \; & 1 \\ & 1 & 2' & \times \\ & & 2 & \; \\ & & & \; \end{array}$$

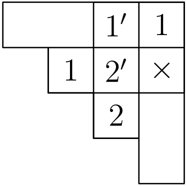
T

The monodromy operator (G., Levinson, Purbhoo)

- Monodromy operator ω is given by commutator of rectification with jeu de taquin (based on work of Levinson, Speyer):
  1. **Rectify**, with $\boxtimes = 0$
  2. **Slide** the $\boxtimes$ to an outer corner with an outer JDT slide
  3. **Unrectify** to the original shape, with $\boxtimes = \infty$
  4. **Slide** the $\boxtimes$ back to an inner corner

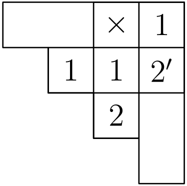


T

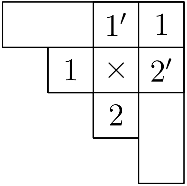


The monodromy operator (G., Levinson, Purbhoo)

- Monodromy operator ω is given by commutator of rectification with jeu de taquin (based on work of Levinson, Speyer):
    1. **Rectify**, with $\boxtimes = 0$
    2. **Slide** the $\boxtimes$ to an outer corner with an outer JDT slide
    3. **Unrectify** to the original shape, with $\boxtimes = \infty$
    4. **Slide** the $\boxtimes$ back to an inner corner

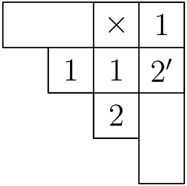


T

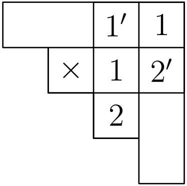


The monodromy operator (G., Levinson, Purbhoo)

- Monodromy operator ω is given by commutator of rectification with jeu de taquin (based on work of Levinson, Speyer):
  1. **Rectify**, with $\boxtimes = 0$
  2. **Slide** the $\boxtimes$ to an outer corner with an outer JDT slide
  3. **Unrectify** to the original shape, with $\boxtimes = \infty$
  4. **Slide** the $\boxtimes$ back to an inner corner

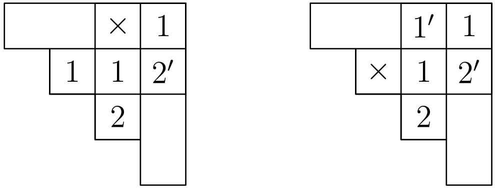


T



The monodromy operator (G., Levinson, Purbhoo)

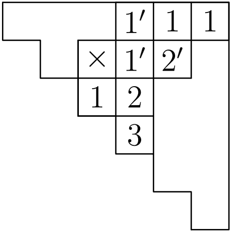
- Monodromy operator ω is given by commutator of rectification with jeu de taquin (based on work of Levinson, Speyer):
  1. **Rectify**, with $\boxtimes = 0$
  2. **Slide** the $\boxtimes$ to an outer corner with an outer JDT slide
  3. **Unrectify** to the original shape, with $\boxtimes = \infty$
  4. **Slide** the $\boxtimes$ back to an inner corner



T $\omega(T)$

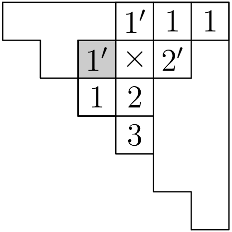
A new local algorithm (G., Levinson, Purbhoo)

- Local rule for steps 1 – 3, without rectifying:
  - **Phase 1.** Switch $\boxtimes$ with the nearest $1'$ *after* it in reading order, if one exists, and then with the nearest 1 *before* it in reading order. Do the same for $2'$ and 2, and so on until there is no i' after it for some i.



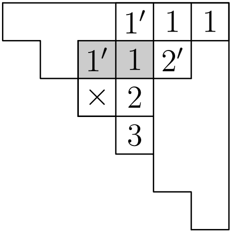
A new local algorithm (G., Levinson, Purbhoo)

- Local rule for steps $1-3$, without rectifying:
  - **Phase 1.** Switch $\boxtimes$ with the nearest $1'$ *after* it in reading order, if one exists, and then with the nearest 1 *before* it in reading order. Do the same for $2'$ and 2, and so on until there is no i' after it for some i.



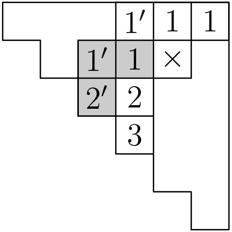
A new local algorithm (G., Levinson, Purbhoo)

- Local rule for steps $1-3$, without rectifying:
  - **Phase 1.** Switch $\boxtimes$ with the nearest $1'$ *after* it in reading order, if one exists, and then with the nearest 1 *before* it in reading order. Do the same for $2'$ and 2, and so on until there is no i' after it for some i.



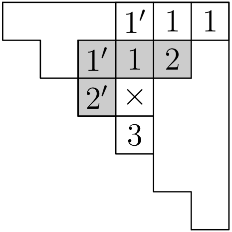
A new local algorithm (G., Levinson, Purbhoo)

- Local rule for steps $1-3$, without rectifying:
  - **Phase 1.** Switch $\boxtimes$ with the nearest $1'$ *after* it in reading order, if one exists, and then with the nearest 1 *before* it in reading order. Do the same for $2'$ and 2, and so on until there is no i' after it for some i.



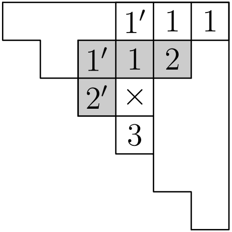
A new local algorithm (G., Levinson, Purbhoo)

- Local rule for steps $1-3$, without rectifying:
  - **Phase 1.** Switch $\boxtimes$ with the nearest $1'$ *after* it in reading order, if one exists, and then with the nearest 1 *before* it in reading order. Do the same for $2'$ and 2, and so on until there is no i' after it for some i. At this point go to Phase 2.



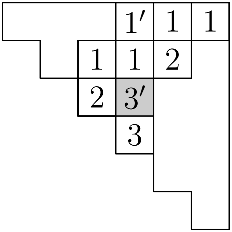
A new local algorithm (G., Levinson, Purbhoo)

- Local rule for steps $1-3$, without rectifying:
  - **Phase 1.** Switch $\boxtimes$ with the nearest $1'$ *after* it in reading order, if one exists, and then with the nearest 1 *before* it in reading order. Do the same for $2'$ and 2, and so on until there is no i' after it for some i. At this point go to Phase 2.
  - **Phase 2.** Replace the $\boxtimes$ with i' and apply $F_i, F_{i+1}, \ldots$ in that order until only one entry is changing. Then replace that entry with $\boxtimes$.



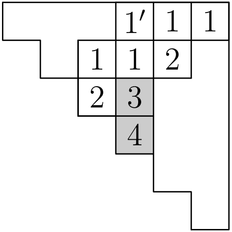
A new local algorithm (G., Levinson, Purbhoo)

- Local rule for steps $1-3$, without rectifying:
  - **Phase 1.** Switch ☒ with the nearest $1'$ *after* it in reading order, if one exists, and then with the nearest 1 *before* it in reading order. Do the same for $2'$ and 2, and so on until there is no i' after it for some i. At this point go to Phase 2.
  - **Phase 2.** Replace the ☒ with i' and apply $F_i, F_{i+1}, \ldots$ in that order until only one entry is changing. Then replace that entry with ☒.



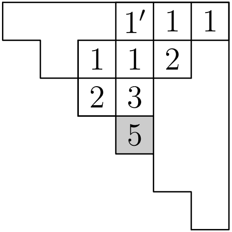
A new local algorithm (G., Levinson, Purbhoo)

- Local rule for steps $1-3$, without rectifying:
  - **Phase 1.** Switch $\boxtimes$ with the nearest $1'$ *after* it in reading order, if one exists, and then with the nearest 1 *before* it in reading order. Do the same for $2'$ and 2, and so on until there is no i' after it for some i. At this point go to Phase 2.
  - **Phase 2.** Replace the $\boxtimes$ with i' and apply $F_i, F_{i+1}, \ldots$ in that order until only one entry is changing. Then replace that entry with $\boxtimes$.



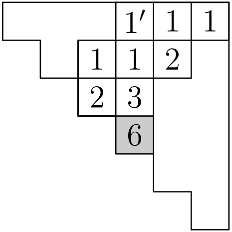
A new local algorithm (G., Levinson, Purbhoo)

- Local rule for steps $1-3$, without rectifying:
  - **Phase 1.** Switch $\boxtimes$ with the nearest $1'$ *after* it in reading order, if one exists, and then with the nearest 1 *before* it in reading order. Do the same for $2'$ and 2, and so on until there is no i' after it for some i. At this point go to Phase 2.
  - **Phase 2.** Replace the $\boxtimes$ with i' and apply $F_i, F_{i+1}, \ldots$ in that order until only one entry is changing. Then replace that entry with $\boxtimes$.



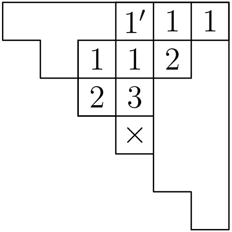
A new local algorithm (G., Levinson, Purbhoo)

- Local rule for steps $1-3$, without rectifying:
  - **Phase 1.** Switch $\boxtimes$ with the nearest $1'$ *after* it in reading order, if one exists, and then with the nearest 1 *before* it in reading order. Do the same for $2'$ and 2, and so on until there is no i' after it for some i. At this point go to Phase 2.
  - **Phase 2.** Replace the $\boxtimes$ with i' and apply $F_i, F_{i+1}, \ldots$ in that order until only one entry is changing. Then replace that entry with $\boxtimes$.



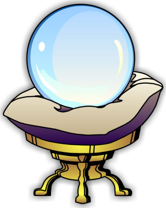
A new local algorithm (G., Levinson, Purbhoo)

- Local rule for steps $1-3$, without rectifying:
  - **Phase 1.** Switch $\boxtimes$ with the nearest $1'$ *after* it in reading order, if one exists, and then with the nearest 1 *before* it in reading order. Do the same for $2'$ and 2, and so on until there is no i' after it for some i. At this point go to Phase 2.
  - **Phase 2.** Replace the $\boxtimes$ with i' and apply $F_i, F_{i+1}, \ldots$ in that order until only one entry is changing. Then replace that entry with $\boxtimes$.



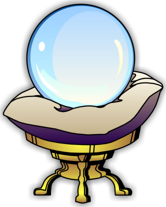
Future work

- Morse-Schilling approach to Schur Q-positivity of type B Stanley symmetric functions using the local axioms?



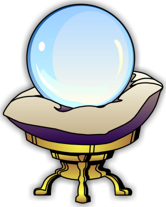
Future work

- Morse-Schilling approach to Schur Q-positivity of type B Stanley symmetric functions using the local axioms?
- Type A crystals correspond to representations of sl_n. Representation theory for these crystals? Relation to quantum queer superalgebra crystals of Grantcharov et. al?



Future work

- Morse-Schilling approach to Schur Q-positivity of type B Stanley symmetric functions using the local axioms?
- Type A crystals correspond to representations of sl_n. Representation theory for these crystals? Relation to quantum queer superalgebra crystals of Grantcharov et. al?
- Geometry of type B Schubert curves (monodromy operator now understood in terms of crystal-like operators).



Thank You!