Crystals, combinatorics, and k-Schur functions

M. Shimozono

Department of Mathematics
Virginia Tech

Berkeley
June 2008

Topics

- Crystal graphs for affine type A
- Combinatorics of certain graded GL_n-modules supported in the nullcone
- k-Schur functions

Topics

- Crystal graphs for affine type A
- Combinatorics of certain graded GL_n-modules supported in the nullcone
- k-Schur functions

Topics

- Crystal graphs for affine type A
- Combinatorics of certain graded GL_n-modules supported in the nullcone
- k-Schur functions

General crystal graphs

- M. Kashiwara, On crystal bases, Representations of Groups, Proceedings of the 1994 Annual Seminar of the Canadian Math. Soc. Ban 16 (1995) 155–197, Amer. Math. Soc., Providence, RI.
- P. Littelmann, Paths and root operators in representation theory. Ann. of Math. (2) 142 (1995), no. 3, 499–525.

Crystal graphs of classical Lie algebras

- M. Kashiwara and T. Nakashima, Crystal graphs for representations of the q-analogue of classical Lie algebras. J. Algebra 165 (1994), no. 2, 295–345.
- C. Lecouvey, Schensted-type correspondence, plactic monoid, and jeu de taquin for type C_n. J. Algebra 247 (2002), no. 2, 295–331. Schensted-type correspondences and plactic monoids for types B_n and D_n. J. Algebraic Combin. 18 (2003), no. 2, 99–133.

Crystal graphs

$\mathfrak{g}$	simple Lie algebra over $\mathbb{C}$
$U_q(\mathfrak{g})$	quantized universal enveloping algebra of $\mathfrak{g}$
$U_q(\mathfrak{g})$-Mod	category of fin. dim. irreducible integrable $U_q(\mathfrak{g})$-modules
$\mathcal{C}(\mathfrak{g})$	category of crystal graphs of $M \in U_q(\mathfrak{g})$-Mod

The crystal basis B of $M \in U_q(\mathfrak{g})$-Mod, is the vertex set of a directed graph with edges labeled by the Dynkin nodes I of $\mathfrak{g}$.

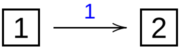
Example for $\mathfrak{g} = \mathfrak{sl}_2$: $I = \{1\}$

Crystal graphs

$\mathfrak{g}$	simple Lie algebra over $\mathbb{C}$
$U_q(\mathfrak{g})$	quantized universal enveloping algebra of $\mathfrak{g}$
$U_q(\mathfrak{g})$-Mod	category of fin. dim. irreducible integrable $U_q(\mathfrak{g})$-modules
$\mathcal{C}(\mathfrak{g})$	category of crystal graphs of $M \in U_q(\mathfrak{g})$-Mod

The crystal basis B of $M \in U_q(\mathfrak{g})$-Mod, is the vertex set of a directed graph with edges labeled by the Dynkin nodes I of $\mathfrak{g}$.

Example for $\mathfrak{g} = \mathfrak{sl}_2$: $I = \{1\}$



Operations on crystal graphs

$\mathcal{C}(\mathfrak{g})$ is closed under:

- Disjoint union (direct sum)
- Taking a connected component (summand)
- Cartesian product* (tensor product)
- Reversing all arrows (dual)
- Dynkin automorphisms

Operations on crystal graphs

$\mathcal{C}(\mathfrak{g})$ is closed under:

- Disjoint union (direct sum)
- Taking a connected component (summand)
- Cartesian product* (tensor product)
- Reversing all arrows (dual)
- Dynkin automorphisms

Operations on crystal graphs

$\mathcal{C}(\mathfrak{g})$ is closed under:

- Disjoint union (direct sum)
- Taking a connected component (summand)
- Cartesian product* (tensor product)
- Reversing all arrows (dual)
- Dynkin automorphisms

Operations on crystal graphs

$\mathcal{C}(\mathfrak{g})$ is closed under:

- Disjoint union (direct sum)
- Taking a connected component (summand)
- Cartesian product* (tensor product)
- Reversing all arrows (dual)
- Dynkin automorphisms

Operations on crystal graphs

$\mathcal{C}(\mathfrak{g})$ is closed under:

- Disjoint union (direct sum)
- Taking a connected component (summand)
- Cartesian product* (tensor product)
- Reversing all arrows (dual)
- Dynkin automorphisms

Structure of $\mathcal{C}(\mathfrak{g})$

$\{\omega_i \mid i \in I\}$ fund. wts.
$P^+ = \bigoplus_{i\in I} \mathbb{Z}_{\geq 0}\omega_i$ dominant integral weights

Representation theory of $U_q(\mathfrak{g})$-Mod says:

There is a bijection

$$P^+ \to \left\{ \begin{array}{c} \text{iso. classes of} \\ \text{connected graphs in } \mathcal{C}(\mathfrak{g}) \end{array} \right\}$$

$\lambda \mapsto B_\lambda$ crystal graph of highest weight λ

The identity map is the unique morphism $B_\lambda \to B_\lambda$.

A morphism $B \to B'$ between objects in $\mathcal{C}(\mathfrak{g})$, sends each component of B isomorphically to a component of B' or “makes it disappear" (sends the corresponding summand to zero).

Structure of $\mathcal{C}(\mathfrak{g})$

$\{\omega_i \mid i \in I\}$ fund. wts.
$P^+ = \bigoplus_{i \in I} \mathbb{Z}_{\geq 0}\omega_i$ dominant integral weights

Representation theory of $U_q(\mathfrak{g})$-Mod says:

There is a bijection

$$P^+ \to \left\{ \begin{array}{c} \text{iso. classes of} \\ \text{connected graphs in } \mathcal{C}(\mathfrak{g}) \end{array} \right\}$$

$\lambda \mapsto B_\lambda$ crystal graph of highest weight λ

The identity map is the unique morphism $B_\lambda \to B_\lambda$.

A morphism $B \to B'$ between objects in $\mathcal{C}(\mathfrak{g})$, sends each component of B isomorphically to a component of B' or “makes it disappear" (sends the corresponding summand to zero).

Structure of $\mathcal{C}(\mathfrak{g})$

$\{\omega_i \mid i \in I\}$ fund. wts.
$P^+ = \bigoplus_{i \in I} \mathbb{Z}_{\geq 0}\omega_i$ dominant integral weights

Representation theory of $U_q(\mathfrak{g})$-Mod says:

There is a bijection

$$P^+ \to \left\{ \begin{matrix} \text{iso. classes of} \\ \text{connected graphs in } \mathcal{C}(\mathfrak{g}) \end{matrix} \right\}$$

$\lambda \mapsto B_\lambda$ crystal graph of highest weight λ

The identity map is the unique morphism $B_\lambda \to B_\lambda$.

A morphism $B \to B'$ between objects in $\mathcal{C}(\mathfrak{g})$, sends each component of B isomorphically to a component of B' or “makes it disappear" (sends the corresponding summand to zero).

Structure of $\mathcal{C}(\mathfrak{g})$

$\{\omega_i \mid i \in I\}$ fund. wts.
$P^+ = \bigoplus_{i \in I} \mathbb{Z}_{\geq 0}\omega_i$ dominant integral weights

Representation theory of $U_q(\mathfrak{g})$-Mod says:

There is a bijection

$$P^+ \to \left\{ \begin{array}{c} \text{iso. classes of} \\ \text{connected graphs in } \mathcal{C}(\mathfrak{g}) \end{array} \right\}$$

$\lambda \mapsto B_\lambda$ crystal graph of highest weight λ

The identity map is the unique morphism $B_\lambda \to B_\lambda$.

A morphism $B \to B'$ between objects in $\mathcal{C}(\mathfrak{g})$, sends each component of B isomorphically to a component of B' or “makes it disappear" (sends the corresponding summand to zero).

Structure of $\mathcal{C}(\mathfrak{g})$

$\{\omega_i \mid i \in I\}$ fund. wts.
$P^+ = \bigoplus_{i \in I} \mathbb{Z}_{\geq 0}\omega_i$ dominant integral weights

Representation theory of $U_q(\mathfrak{g})$-Mod says:

There is a bijection

$$P^+ \to \left\{ \begin{matrix} \text{iso. classes of} \\ \text{connected graphs in } \mathcal{C}(\mathfrak{g}) \end{matrix} \right\}$$

$\lambda \mapsto B_\lambda$ crystal graph of highest weight λ

The identity map is the unique morphism $B_\lambda \to B_\lambda$.

A morphism $B \to B'$ between objects in $\mathcal{C}(\mathfrak{g})$, sends each component of B isomorphically to a component of B' or "makes it disappear" (sends the corresponding summand to zero).

Comment on Littelmann paths

Littelmann paths: construction of crystal graphs B_λ of irr. int. highest weight $U_q(\mathfrak{g})$-modules when $\mathfrak{g}$ is a symm. Kac-Moody algebra.

We consider the special case that $\mathfrak{g}$ is an affine algebra and focus on:

- Nonhighest weight (Kirillov-Reshetikhin) $U_q'(\mathfrak{g})$-modules
- Special properties of affine-to-finite branching

Comment on Littelmann paths

Littelmann paths: construction of crystal graphs B_λ of irr. int. highest weight $U_q(\mathfrak{g})$-modules when $\mathfrak{g}$ is a symm. Kac-Moody algebra.

We consider the special case that $\mathfrak{g}$ is an affine algebra and focus on:

- Nonhighest weight (Kirillov-Reshetikhin) $U_q'(\mathfrak{g})$-modules
- Special properties of affine-to-finite branching

$\mathcal{C}(\mathfrak{sl}_2)$

$\mathfrak{g} = \mathfrak{sl}_2$, $I = \{1\}$, $P^+ = \mathbb{Z}_{\geq 0}\omega$

$B_{r\omega}$ is a directed path of length r; it has $r+1$ vertices. "string"

$$\bullet \longrightarrow e(b) \longrightarrow b \longrightarrow f(b) \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \bullet$$

$$| \longleftarrow \varepsilon(b) \longrightarrow | \longleftarrow \varphi(b) \longrightarrow |$$

Notation:

$e(b)$	vertex before b on its string
$f(b)$	vertex after b on its string
$\varepsilon(b)$	distance (number of edges) to beginning of string
$\varphi(b)$	distance to end of string

$\mathcal{C}(\mathfrak{sl}_2)$

$\mathfrak{g} = \mathfrak{sl}_2$, $I = \{1\}$, $P^+ = \mathbb{Z}_{\geq 0}\omega$

$B_{r\omega}$ is a directed path of length r; it has $r + 1$ vertices. “string"

$$\bullet \longrightarrow e(b) \longrightarrow b \longrightarrow f(b) \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \bullet$$

$$| \longleftarrow \varepsilon(b) \longrightarrow | \longleftarrow \varphi(b) \longrightarrow |$$

Notation:

$e(b)$	vertex before b on its string
$f(b)$	vertex after b on its string
$\varepsilon(b)$	distance (number of edges) to beginning of string
$\varphi(b)$	distance to end of string

$\mathcal{C}(\mathfrak{sl}_2)$

$\mathfrak{g} = \mathfrak{sl}_2$, $I = \{1\}$, $P^+ = \mathbb{Z}_{\geq 0}\omega$

$B_{r\omega}$ is a directed path of length r; it has $r+1$ vertices. “string"

$$\bullet \longrightarrow e(b) \longrightarrow b \longrightarrow f(b) \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \bullet$$

$$| \longleftarrow \varepsilon(b) \longrightarrow | \longleftarrow \varphi(b) \longrightarrow |$$

Notation:

$e(b)$	vertex before b on its string
$f(b)$	vertex after b on its string
$\varepsilon(b)$	distance (number of edges) to beginning of string
$\varphi(b)$	distance to end of string

Signature of elements in a string (Kashiwara^{op})

b	$\underbrace{))\cdots)}_{\varphi(b)}\underbrace{((\cdots(}_{\varepsilon(b)}$	$1^{\varphi(b)}2^{\varepsilon(b)}$
$\bullet$	)))	111
$\downarrow$	$\downarrow$	$\downarrow$
$\bullet$	))(	112
$\downarrow$	$\downarrow$	$\downarrow$
$\bullet$	)((	122
$\downarrow$	$\downarrow$	$\downarrow$
$\bullet$	(((	222

f changes rightmost ")" to "(" or 1 to 2

e changes leftmost "(" to ")" or 2 to 1

Signature of elements in a string (Kashiwara$^{\text{op}}$)

b	$\underbrace{))\cdots)}_{\varphi(b)}\underbrace{((\cdots(}_{\varepsilon(b)}$	$1^{\varphi(b)}2^{\varepsilon(b)}$
•	)))	111
↓	↓	↓
•	))(	112
↓	↓	↓
•	)((	122
↓	↓	↓
•	(((	222

f changes rightmost ")" to "(" or 1 to 2

e changes leftmost "(" to ")" or 2 to 1

Signature of elements in a string (Kashiwara^{op})

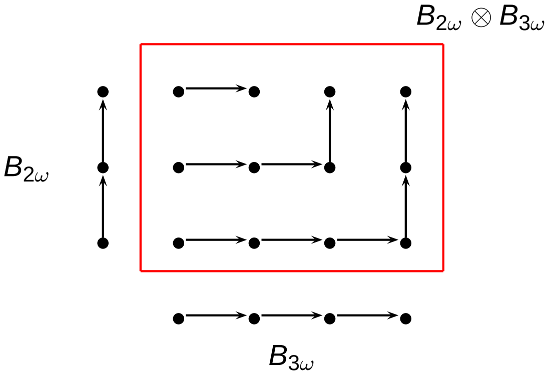
b	$\underbrace{))\cdots)}_{\varphi(b)}\underbrace{((\cdots(}_{\varepsilon(b)}$	$1^{\varphi(b)}2^{\varepsilon(b)}$
•	)))	111
↓	↓	↓
•	))(	112
↓	↓	↓
•	)((	122
↓	↓	↓
•	(((	222

f changes rightmost ")" to "(" or 1 to 2
e changes leftmost "(" to ")" or 2 to 1

Tensor product (Kashiwara$^{\mathrm{op}}$)

$B_{r\omega} \otimes B_{s\omega}$ has vertex set $B_{r\omega} \times B_{s\omega}$.

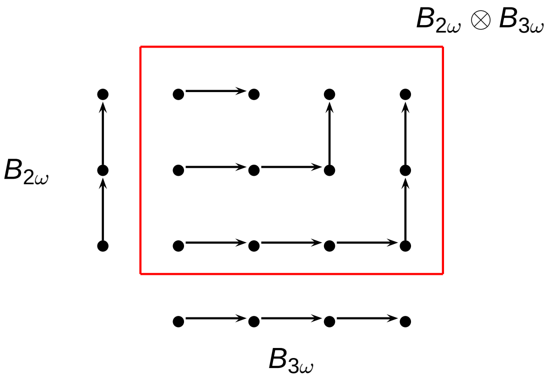
Example for $r = 2$, $s = 3$



$$B_{2\omega} \otimes B_{3\omega} \cong B_{5\omega} \sqcup B_{3\omega} \sqcup B_{\omega}$$

Tensor product (Kashiwara$^{\text{op}}$)

$B_{r\omega} \otimes B_{s\omega}$ has vertex set $B_{r\omega} \times B_{s\omega}$.
Example for $r = 2$, $s = 3$



$$B_{2\omega} \otimes B_{3\omega} \cong B_{5\omega} \sqcup B_{3\omega} \sqcup B_{\omega}$$

Signature rule for tensor product

Let $b = b_2 \otimes b_1 \in B_2 \otimes B_1$.
Write signatures of b_2 and b_1:

$$\underbrace{))\cdots)}_{\varphi(b_2)}\underbrace{((\cdots(}_{\varepsilon(b_2)} \quad \underbrace{))\cdots)}_{\varphi(b_1)}\underbrace{((\cdots(}_{\varepsilon(b_1)}$$

Match parentheses. Unmatched substring has the form

$$))\cdots)\ ((\cdots($$

$\varphi(b)$ is the number of unmatched ")".
$\varepsilon(b)$ is the number of unmatched "(".

If $\varphi(b) > 0$ then $f(b)$ is defined and

$$f(b) = \begin{cases} b_2 \otimes f(b_1) & \text{if } \varepsilon(b_2) < \varphi(b_1) \\ f(b_2) \otimes b_1 & \text{if } \varepsilon(b_2) \geq \varphi(b_1) \end{cases}$$

effect on signature: change rightmost unmatched ")" to "(".

Signature rule for tensor product

Let $b = b_2 \otimes b_1 \in B_2 \otimes B_1$.
Write signatures of b_2 and b_1:

$$\underbrace{))\cdots)}_{\varphi(b_2)}\underbrace{((\cdots(}_{\varepsilon(b_2)} \quad \underbrace{))\cdots)}_{\varphi(b_1)}\underbrace{((\cdots(}_{\varepsilon(b_1)}$$

Match parentheses. Unmatched substring has the form

$$))\cdots)\ ((\cdots($$

$\varphi(b)$ is the number of unmatched ")".
$\varepsilon(b)$ is the number of unmatched "(".

If $\varphi(b) > 0$ then $f(b)$ is defined and

$$f(b) = \begin{cases} b_2 \otimes f(b_1) & \text{if } \varepsilon(b_2) < \varphi(b_1) \\ f(b_2) \otimes b_1 & \text{if } \varepsilon(b_2) \geq \varphi(b_1) \end{cases}$$

effect on signature: change rightmost unmatched ")" to "(".

Signature rule for tensor product

Let $b = b_2 \otimes b_1 \in B_2 \otimes B_1$.
Write signatures of b_2 and b_1:

$$\underbrace{))\cdots)}_{\varphi(b_2)}\underbrace{((\cdots(}_{\varepsilon(b_2)} \quad \underbrace{))\cdots)}_{\varphi(b_1)}\underbrace{((\cdots(}_{\varepsilon(b_1)}$$

Match parentheses. Unmatched substring has the form

$$))\cdots)\ ((\cdots($$

$\varphi(b)$ is the number of unmatched ")".
$\varepsilon(b)$ is the number of unmatched "(".

If $\varphi(b) > 0$ then $f(b)$ is defined and

$$f(b) = \begin{cases} b_2 \otimes f(b_1) & \text{if } \varepsilon(b_2) < \varphi(b_1) \\ f(b_2) \otimes b_1 & \text{if } \varepsilon(b_2) \geq \varphi(b_1) \end{cases}$$

effect on signature: change rightmost unmatched ")" to "(".

Signature rule for tensor product

Let $b = b_2 \otimes b_1 \in B_2 \otimes B_1$.
Write signatures of b_2 and b_1:

$$\underbrace{))\cdots)}_{\varphi(b_2)}\underbrace{((\cdots(}_{\varepsilon(b_2)} \quad \underbrace{))\cdots)}_{\varphi(b_1)}\underbrace{((\cdots(}_{\varepsilon(b_1)}$$

Match parentheses. Unmatched substring has the form

$$))\cdots)\ ((\cdots($$

$\varphi(b)$ is the number of unmatched ")".
$\varepsilon(b)$ is the number of unmatched "(".

If $\varphi(b) > 0$ then $f(b)$ is defined and

$$f(b) = \begin{cases} b_2 \otimes f(b_1) & \text{if } \varepsilon(b_2) < \varphi(b_1) \\ f(b_2) \otimes b_1 & \text{if } \varepsilon(b_2) \geq \varphi(b_1) \end{cases}$$

effect on signature: change rightmost unmatched ")" to "(".

Signature rule for tensor product

Let $b = b_2 \otimes b_1 \in B_2 \otimes B_1$.
Write signatures of b_2 and b_1:

$$\underbrace{))\cdots)}_{\varphi(b_2)}\underbrace{((\cdots(}_{\varepsilon(b_2)} \quad \underbrace{))\cdots)}_{\varphi(b_1)}\underbrace{((\cdots(}_{\varepsilon(b_1)}$$

Match parentheses. Unmatched substring has the form

$$))\cdots)\ ((\cdots($$

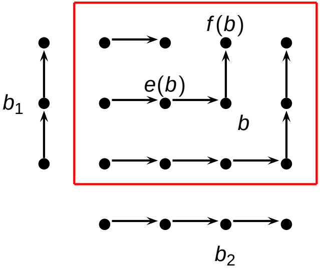
$\varphi(b)$ is the number of unmatched ")".
$\varepsilon(b)$ is the number of unmatched "(".

If $\varepsilon(b) > 0$ then $e(b)$ is defined and

$$e(b) = \begin{cases} e(b_2) \otimes b_1 & \text{if } \varepsilon(b_2) > \varphi(b_1) \\ b_2 \otimes e(b_1) & \text{if } \varepsilon(b_2) \leq \varphi(b_1) \end{cases}$$

effect on signature: change leftmost unmatched “(" to “)".

$$b = b_1 \otimes b_2$$



$e(b)$	)	($\otimes$)	)	(
b	)	($\otimes$)	(	(
$f(b)$	(	($\otimes$)	(	(

Tensor product is associative

$(B_3 \otimes B_2) \otimes B_1$:
match parens in B_3 and B_2, then match with B_1.

$$) \;) \; (\; (\; (\otimes) \;) \; (\otimes) \;) \; ($$

$B_3 \otimes (B_2 \otimes B_1)$:
match parens in B_2 and B_1 then match with B_3.

$$) \;) \; (\; (\; (\otimes) \;) \; (\otimes) \;) \; ($$

The matched parentheses are the same in either case.

For any number of tensor factors: write signatures, pair parens, see which parenthesis is turned around and apply e or f in that tensor factor.

Tensor product is associative

$(B_3 \otimes B_2) \otimes B_1$:
match parens in B_3 and B_2, then match with B_1.

$$) \;) \; (\; (\; (\otimes) \;) \; (\otimes) \;) \; ($$

$B_3 \otimes (B_2 \otimes B_1)$:
match parens in B_2 and B_1 then match with B_3.

$$) \;) \; (\; (\; (\otimes) \;) \; (\otimes) \;) \; ($$

The matched parentheses are the same in either case.

For any number of tensor factors: write signatures, pair parens, see which parenthesis is turned around and apply e or f in that tensor factor.

Tensor product is associative

$(B_3 \otimes B_2) \otimes B_1$:
match parens in B_3 and B_2, then match with B_1.

$$) \;) \; (\; (\; (\otimes) \;) \; (\otimes) \;) \; ($$

$B_3 \otimes (B_2 \otimes B_1)$:
match parens in B_2 and B_1 then match with B_3.

$$) \;) \; (\; (\; (\otimes) \;) \; (\otimes) \;) \; ($$

The matched parentheses are the same in either case.

For any number of tensor factors: write signatures, pair parens, see which parenthesis is turned around and apply e or f in that tensor factor.

$\mathcal{C}(\mathfrak{g})$ and i-strings

For each Dynkin node $i \in I$ there is a copy $U_q(\mathfrak{sl}_2) \subset U_q(\mathfrak{g})$ which makes B into an $\mathfrak{sl}_2$ crystal graph.
Label these directed edges with i. "i-strings"

$$\bullet \xrightarrow{i} e_i(b) \xrightarrow{i} b \xrightarrow{i} f_i(b) \xrightarrow{i} \bullet \xrightarrow{i} \bullet \xrightarrow{i} \bullet$$

$$| \longleftarrow \varepsilon_i(b) \longrightarrow | \longleftarrow \varphi_i(b) \longrightarrow |$$

$e_i(b)$ vertex before b on its i-string
$f_i(b)$ vertex after b on its i-string
$\varepsilon_i(b)$ distance to beginning of i-string
$\varphi_i(b)$ distance to end of i-string

$\mathcal{C}(\mathfrak{g})$ and i-strings

For each Dynkin node $i \in I$ there is a copy $U_q(\mathfrak{sl}_2) \subset U_q(\mathfrak{g})$ which makes B into an $\mathfrak{sl}_2$ crystal graph.
Label these directed edges with i. "i-strings"

$$\bullet \xrightarrow{i} e_i(b) \xrightarrow{i} b \xrightarrow{i} f_i(b) \xrightarrow{i} \bullet \xrightarrow{i} \bullet \xrightarrow{i} \bullet$$

$$| \longleftarrow \varepsilon_i(b) \longrightarrow | \longleftarrow\!\!\!-\!\!\!-\!\!\!- \varphi_i(b) -\!\!\!-\!\!\!-\!\!\!\longrightarrow |$$

$e_i(b)$ vertex before b on its i-string
$f_i(b)$ vertex after b on its i-string
$\varepsilon_i(b)$ distance to beginning of i-string
$\varphi_i(b)$ distance to end of i-string

Connected graphs in $\mathcal{C}(\mathfrak{g})$

For $B \in \mathcal{C}(\mathfrak{g})$ and $b \in B$, let $C(b) \subset B$ be the component of b.

- Every connected graph $B \in \mathcal{C}(\mathfrak{g})$ has a unique highest weight vector u (vertex with no in-edges).
  Let u_λ be the h.w.v. of B_λ.
- Let $\lambda = \sum_{i \in I} \varphi_i(u)\omega_i$. Then there is a unique isomorphism $B \cong B_\lambda$ denoted $b \mapsto P(b)$. Write $\mathrm{shape}(b) = \lambda$.
- Let $B_1, B_2 \in \mathcal{C}(\mathfrak{g})$ and $b_2 \otimes b_1 \in B_2 \otimes B_1$. Then

$$P(b_2 \otimes b_1) = P(P(b_2) \otimes P(b_1)).$$

Connected graphs in $\mathcal{C}(\mathfrak{g})$

For $B \in \mathcal{C}(\mathfrak{g})$ and $b \in B$, let $C(b) \subset B$ be the component of b.

- Every connected graph $B \in \mathcal{C}(\mathfrak{g})$ has a unique highest weight vector u (vertex with no in-edges).
  Let u_λ be the h.w.v. of B_λ.
- Let $\lambda = \sum_{i \in I} \varphi_i(u)\omega_i$. Then there is a unique isomorphism $B \cong B_\lambda$ denoted $b \mapsto P(b)$. Write $\mathrm{shape}(b) = \lambda$.
- Let $B_1, B_2 \in \mathcal{C}(\mathfrak{g})$ and $b_2 \otimes b_1 \in B_2 \otimes B_1$. Then

$$P(b_2 \otimes b_1) = P(P(b_2) \otimes P(b_1)).$$

Connected graphs in $\mathcal{C}(\mathfrak{g})$

For $B \in \mathcal{C}(\mathfrak{g})$ and $b \in B$, let $C(b) \subset B$ be the component of b.

- Every connected graph $B \in \mathcal{C}(\mathfrak{g})$ has a unique highest weight vector u (vertex with no in-edges).
  Let u_λ be the h.w.v. of B_λ.
- Let $\lambda = \sum_{i \in I} \varphi_i(u)\omega_i$. Then there is a unique isomorphism $B \cong B_\lambda$ denoted $b \mapsto P(b)$. Write $\mathrm{shape}(b) = \lambda$.
- Let $B_1, B_2 \in \mathcal{C}(\mathfrak{g})$ and $b_2 \otimes b_1 \in B_2 \otimes B_1$. Then

$$P(b_2 \otimes b_1) = P(P(b_2) \otimes P(b_1)).$$

Say $C(b_1) \cong B_\mu$, $C(b_2) \cong B_\nu$, $C(b_2 \otimes b_1) \cong B_\lambda$.

Then $C(b_2) \otimes C(b_1) \cong B_\nu \otimes B_\mu$ with $b_2 \otimes b_1 \mapsto P(b_2) \otimes P(b_1)$.

$$\begin{array}{ccc}
C(b_2 \otimes b_1) & \longrightarrow & C(P(b_2) \otimes P(b_1)) \\
\downarrow & & \downarrow \\
B_\lambda & \xrightarrow[\text{id}]{} & B_\lambda
\end{array}$$

$$\begin{array}{ccc}
b_2 \otimes b_1 & \longrightarrow & P(b_2) \otimes P(b_1) \\
\downarrow & & \downarrow \\
P(b_2 \otimes b_1) & \xrightarrow[id]{} & P(P(b_2) \otimes P(b_1))
\end{array}$$

Say $C(b_1) \cong B_\mu$, $C(b_2) \cong B_\nu$, $C(b_2 \otimes b_1) \cong B_\lambda$.
Then $C(b_2) \otimes C(b_1) \cong B_\nu \otimes B_\mu$ with $b_2 \otimes b_1 \mapsto P(b_2) \otimes P(b_1)$.

$$\begin{array}{ccc} C(b_2 \otimes b_1) & \longrightarrow & C(P(b_2) \otimes P(b_1)) \\ \downarrow & & \downarrow \\ B_\lambda & \xrightarrow[\text{id}]{} & B_\lambda \end{array}$$

$$\begin{array}{ccc} b_2 \otimes b_1 & \longrightarrow & P(b_2) \otimes P(b_1) \\ \downarrow & & \downarrow \\ P(b_2 \otimes b_1) & \xrightarrow[id]{} & P(P(b_2) \otimes P(b_1)) \end{array}$$

Say $C(b_1) \cong B_\mu$, $C(b_2) \cong B_\nu$, $C(b_2 \otimes b_1) \cong B_\lambda$.
Then $C(b_2) \otimes C(b_1) \cong B_\nu \otimes B_\mu$ with $b_2 \otimes b_1 \mapsto P(b_2) \otimes P(b_1)$.

$$\begin{array}{ccc} C(b_2 \otimes b_1) & \longrightarrow & C(P(b_2) \otimes P(b_1)) \\ \downarrow & & \downarrow \\ B_\lambda & \xrightarrow[\text{id}]{} & B_\lambda \end{array}$$

$$\begin{array}{ccc} b_2 \otimes b_1 & \longrightarrow & P(b_2) \otimes P(b_1) \\ \downarrow & & \downarrow \\ P(b_2 \otimes b_1) & \xrightarrow[id]{} & P(P(b_2) \otimes P(b_1)) \end{array}$$

Say $C(b_1) \cong B_\mu$, $C(b_2) \cong B_\nu$, $C(b_2 \otimes b_1) \cong B_\lambda$.
Then $C(b_2) \otimes C(b_1) \cong B_\nu \otimes B_\mu$ with $b_2 \otimes b_1 \mapsto P(b_2) \otimes P(b_1)$.

$$\begin{array}{ccc}
C(b_2 \otimes b_1) & \longrightarrow & C(P(b_2) \otimes P(b_1)) \\
\downarrow & & \downarrow \\
B_\lambda & \xrightarrow[\text{id}]{} & B_\lambda
\end{array}$$

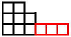
$$\begin{array}{ccc}
b_2 \otimes b_1 & \longrightarrow & P(b_2) \otimes P(b_1) \\
\downarrow & & \downarrow \\
P(b_2 \otimes b_1) & \xrightarrow[id]{} & P(P(b_2) \otimes P(b_1))
\end{array}$$

$P^+(\mathfrak{sl}_n)$ and partitions

$I = \{1, 2, \ldots, n-1\}$

$P^+ \leftrightarrow \{\text{partitions with} < n \text{ parts}\}$

$\sum_{i=1}^{n-1} a_i \omega_i$ goes to the partition with a_i columns of size i.
Example: $n = 4$:

$$3\omega_1 + \omega_2 + 2\omega_3 \mapsto$$ 

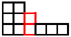
$$\omega_1 \mapsto$$ 

$P^+(\mathfrak{sl}_n)$ and partitions

$I = \{1, 2, \ldots, n-1\}$

$P^+ \leftrightarrow \{\text{partitions with} < n \text{ parts}\}$

$\sum_{i=1}^{n-1} a_i \omega_i$ goes to the partition with a_i columns of size i.
Example: $n = 4$:

$$3\omega_1 + \omega_2 + 2\omega_3 \mapsto$$ 

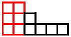
$$\omega_1 \mapsto$$ 

$P^+(\mathfrak{sl}_n)$ and partitions

$I = \{1, 2, \ldots, n-1\}$

$P^+ \leftrightarrow \{\text{partitions with} < n \text{ parts}\}$

$\sum_{i=1}^{n-1} a_i\omega_i$ goes to the partition with a_i columns of size i.
Example: $n = 4$:

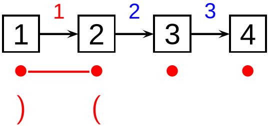
$3\omega_1 + \omega_2 + 2\omega_3 \mapsto$ 

$\omega_1 \mapsto$ 

$B_{\omega_1} \in \mathcal{C}(\mathfrak{g})$

$B = B_{\omega_1}$: Vector rep: vertices $\boxed{1}$ through $\boxed{n}$.

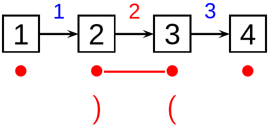
1-strings:



$B_{\omega_1} \in \mathcal{C}(\mathfrak{g})$

$B = B_{\omega_1}$: Vector rep: vertices $\boxed{1}$ through $\boxed{n}$.

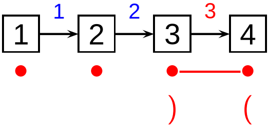
2-strings:



$B_{\omega_1} \in \mathcal{C}(\mathfrak{g})$

$B = B_{\omega_1}$: Vector rep: vertices $\boxed{1}$ through $\boxed{n}$.

3-strings:



Words and $\mathcal{C}(\mathfrak{sl}_n)$

$B^{\otimes L}$: words of length L in alphabet $B = B_{\omega_1} = \{1, 2, \ldots, n\}$.

Fix $i \in I = \{1, \ldots, n-1\}$.

To get i-string of $u \in B^{\otimes L}$:

- Ignore letters not in $\{i, i+1\}$; their i-signature is empty.
- i-signature of each i is).
- i-signature of each $i+1$ is (.
- Match parens.
- To get $f_i(u)$ change rightmost unmatched i to $i+1$.
- To get $e_i(u)$ change leftmost unmatched $i+1$ to i.

Words and $\mathcal{C}(\mathfrak{sl}_n)$

$B^{\otimes L}$: words of length L in alphabet $B = B_{\omega_1} = \{1, 2, \dots, n\}$.

Fix $i \in I = \{1, \dots, n-1\}$.

To get i-string of $u \in B^{\otimes L}$:

- Ignore letters not in $\{i, i+1\}$; their i-signature is empty.
- i-signature of each i is).
- i-signature of each $i+1$ is (.
- Match parens.
- To get $f_i(u)$ change rightmost unmatched i to $i+1$.
- To get $e_i(u)$ change leftmost unmatched $i+1$ to i.

Words and $\mathcal{C}(\mathfrak{sl}_n)$

$B^{\otimes L}$: words of length L in alphabet $B = B_{\omega_1} = \{1, 2, \ldots, n\}$.

Fix $i \in I = \{1, \ldots, n-1\}$.

To get i-string of $u \in B^{\otimes L}$:

- Ignore letters not in $\{i, i+1\}$; their i-signature is empty.
- i-signature of each i is).
- i-signature of each $i+1$ is (.
- Match parens.
- To get $f_i(u)$ change rightmost unmatched i to $i+1$.
- To get $e_i(u)$ change leftmost unmatched $i+1$ to i.

Words and $\mathcal{C}(\mathfrak{sl}_n)$

$B^{\otimes L}$: words of length L in alphabet $B = B_{\omega_1} = \{1, 2, \ldots, n\}$.

Fix $i \in I = \{1, \ldots, n-1\}$.

To get i-string of $u \in B^{\otimes L}$:

- Ignore letters not in $\{i, i+1\}$; their i-signature is empty.
- i-signature of each i is).
- i-signature of each $i+1$ is (.
- Match parens.
- To get $f_i(u)$ change rightmost unmatched i to $i+1$.
- To get $e_i(u)$ change leftmost unmatched $i+1$ to i.

Words and $\mathcal{C}(\mathfrak{sl}_n)$

$B^{\otimes L}$: words of length L in alphabet $B = B_{\omega_1} = \{1, 2, \ldots, n\}$.

Fix $i \in I = \{1, \ldots, n-1\}$.

To get i-string of $u \in B^{\otimes L}$:

- Ignore letters not in $\{i, i+1\}$; their i-signature is empty.
- i-signature of each i is).
- i-signature of each $i+1$ is (.
- Match parens.
- To get $f_i(u)$ change rightmost unmatched i to $i+1$.
- To get $e_i(u)$ change leftmost unmatched $i+1$ to i.

Words and $\mathcal{C}(\mathfrak{sl}_n)$

$B^{\otimes L}$: words of length L in alphabet $B = B_{\omega_1} = \{1, 2, \ldots, n\}$.

Fix $i \in I = \{1, \ldots, n-1\}$.

To get i-string of $u \in B^{\otimes L}$:

- Ignore letters not in $\{i, i+1\}$; their i-signature is empty.
- i-signature of each i is).
- i-signature of each $i+1$ is (.
- Match parens.
- To get $f_i(u)$ change rightmost unmatched i to $i+1$.
- To get $e_i(u)$ change leftmost unmatched $i+1$ to i.

Words and $\mathcal{C}(\mathfrak{sl}_n)$

$B^{\otimes L}$: words of length L in alphabet $B = B_{\omega_1} = \{1, 2, \ldots, n\}$.

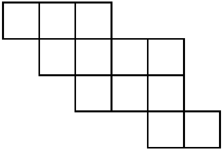
Fix $i \in I = \{1, \ldots, n-1\}$.

To get i-string of $u \in B^{\otimes L}$:

- Ignore letters not in $\{i, i+1\}$; their i-signature is empty.
- i-signature of each i is).
- i-signature of each $i+1$ is (.
- Match parens.
- To get $f_i(u)$ change rightmost unmatched i to $i+1$.
- To get $e_i(u)$ change leftmost unmatched $i+1$ to i.

$\mathcal{C}(\mathfrak{sl}_n)$ and tableaux

Skew partition diagram: $D = (6,5,5,3)/(4,2,1)$ $|D| = 12$



B_D: set of tableaux of shape D

row reading word:

$$\mathrm{word}(t) = 224 \cdot 1334 \cdot 123 \cdot 12$$

We identify a tableau with its reading word:

$$B_D \rightarrow B^{\otimes |D|}$$
$$t \mapsto \mathrm{word}(t)$$

$\mathcal{C}(\mathfrak{sl}_n)$ and tableaux

Tableau t of shape D:
filling of D with entries in $B_{\omega_1} = \{1, 2, \ldots, n\}$
$\leq$ in rows $\vee$ in columns

2	2	4			
	1	3	3	4	
		1	2	3	
				1	2

B_D: set of tableaux of shape D
row reading word:

$$\mathrm{word}(t) = 224 \cdot 1334 \cdot 123 \cdot 12$$

We identify a tableau with its reading word:

$$B_D \to B^{\otimes |D|}$$
$$t \mapsto \mathrm{word}(t)$$

$\mathcal{C}(\mathfrak{sl}_n)$ and tableaux

Tableau t of shape D:
filling of D with entries in $B_{\omega_1} = \{1, 2, \ldots, n\}$
$\leq$ in rows $\quad\quad$ $\vee$ in columns

2	2	4			
	1	3	3	4	
		1	2	3	
				1	2

B_D: set of tableaux of shape D
row reading word:

$$\mathrm{word}(t) = 224 \cdot 1334 \cdot 123 \cdot 12$$

We identify a tableau with its reading word:

$$B_D \longrightarrow B^{\otimes |D|}$$
$$t \longmapsto \mathrm{word}(t)$$

$\mathcal{C}(\mathfrak{sl}_n)$ and tableaux

Tableau t of shape D:
filling of D with entries in $B_{\omega_1} = \{1, 2, \ldots, n\}$
$\leq$ in rows $\quad$ $\vee$ in columns

2	2	4			
	1	3	3	4	
		1	2	3	
				1	2

B_D: set of tableaux of shape D
row reading word:

$$\mathrm{word}(t) = 224 \cdot 1334 \cdot 123 \cdot 12$$

We identify a tableau with its reading word:

$$B_D \longrightarrow B^{\otimes |D|}$$
$$t \longmapsto \mathrm{word}(t)$$

$B_D \in \mathcal{C}(\mathfrak{sl}_n)$

The image of B_D in $B^{\otimes |D|}$ is stable under e_i and f_i for all $i \in I$.
Enough to check for skew subtableau of letters i and $i+1$.

1	1	1	2	2	2	2	2			
				1	1	1	1	1	2	2

$$11122222 \cdot 1111122$$

No column violation: Letters in columns of size two are always matched, and so not changed by e_1 or f_1.
Only possible violation is to create

2	1

in some row, but this creates a new matching pair, while e_1 and f_1 preserve the matched letters.

$B_D \in \mathcal{C}(\mathfrak{sl}_n)$

The image of B_D in $B^{\otimes |D|}$ is stable under e_i and f_i for all $i \in I$.
Enough to check for skew subtableau of letters i and $i+1$.

1	1	1	2	2	2	2	2			
				1	1	1	1	1	2	2

$$11122222 \cdot 1111122$$

No column violation: Letters in columns of size two are always matched, and so not changed by e_1 or f_1.
Only possible violation is to create

2	1

in some row, but this creates a new matching pair, while e_1 and f_1 preserve the matched letters.

$B_D \in \mathcal{C}(\mathfrak{sl}_n)$

The image of B_D in $B^{\otimes|D|}$ is stable under e_i and f_i for all $i \in I$.
Enough to check for skew subtableau of letters i and $i+1$.

1	1	1	2	2	2	2	2			
				1	1	1	1	1	2	2

$$11122222 \cdot 1111122$$

No column violation: Letters in columns of size two are always matched, and so not changed by e_1 or f_1.

Only possible violation is to create

2	1

in some row, but this creates a new matching pair, while e_1 and f_1 preserve the matched letters.

$B_D \in \mathcal{C}(\mathfrak{sl}_n)$

The image of B_D in $B^{\otimes |D|}$ is stable under e_i and f_i for all $i \in I$.
Enough to check for skew subtableau of letters i and $i+1$.

1	1	1	2	2	2	2	2			
				1	1	1	1	1	2	2

$$11122222 \cdot 1111122$$

No column violation: Letters in columns of size two are always matched, and so not changed by e_1 or f_1.
Only possible violation is to create

2	1

in some row, but this creates a new matching pair, while e_1 and f_1 preserve the matched letters.

Yamanouchi tableau in B_λ

$D = \lambda$ partition shape
Let $u_\lambda \in B_\lambda$ be the Yamanouchi tableau of shape λ, the one having only letters i in row i for all i.

Example: $n = 4$, $\lambda = (4, 2, 1)$

u_λ

3			
2	2		
1	1	1	1

u_λ is a highest weight vector: every $i + 1$ is i-paired.

$\varphi_1(u_\lambda) = 2$, $\varphi_2(u_\lambda) = 1$, $\varphi_3(u_\lambda) = 1$.
$\lambda = 2\omega_1 + 1\omega_2 + 1\omega_3$.

Yamanouchi tableau in B_λ

$D = \lambda$ partition shape
Let $u_\lambda \in B_\lambda$ be the Yamanouchi tableau of shape λ, the one having only letters i in row i for all i.

Example: $n = 4$, $\lambda = (4, 2, 1)$

u_λ

3			
2	2		
1	1	1	1

u_λ is a highest weight vector: every $i + 1$ is i-paired.

$\varphi_1(u_\lambda) = 2$, $\varphi_2(u_\lambda) = 1$, $\varphi_3(u_\lambda) = 1$.
$\lambda = 2\omega_1 + 1\omega_2 + 1\omega_3$.

Yamanouchi tableau in B_λ

$D = \lambda$ partition shape
Let $u_\lambda \in B_\lambda$ be the Yamanouchi tableau of shape λ, the one having only letters i in row i for all i.

Example: $n = 4$, $\lambda = (4, 2, 1)$

u_λ

3			
2	2		
1	1	1	1

u_λ is a highest weight vector: every $i + 1$ is i-paired.

$\varphi_1(u_\lambda) = 2$, $\varphi_2(u_\lambda) = 1$, $\varphi_3(u_\lambda) = 1$.
$\lambda = 2\omega_1 + 1\omega_2 + 1\omega_3$.

Yamanouchi tableau in B_λ

$D = \lambda$ partition shape
Let $u_\lambda \in B_\lambda$ be the Yamanouchi tableau of shape λ, the one having only letters i in row i for all i.

Example: $n = 4$, $\lambda = (4, 2, 1)$

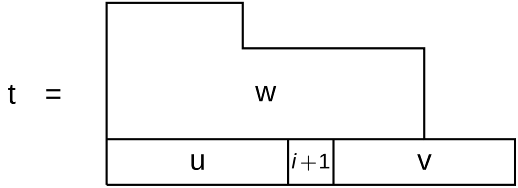
u_λ

3			
2	2		
1	1	1	1

u_λ is a highest weight vector: every $i + 1$ is i-paired.

$\varphi_1(u_\lambda) = 2$, $\varphi_2(u_\lambda) = 1$, $\varphi_3(u_\lambda) = 1$.
$\lambda = 2\omega_1 + 1\omega_2 + 1\omega_3$.

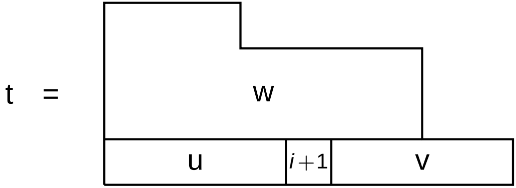
Consider an $i+1$ in the first row of a tableau $t \in B_\lambda$.



$$\mathrm{word}(t) = \mathrm{word}(w)\, u \;\; i{+}1 \;\; v$$

v has no letters i. Thus the $i+1$ is i-unpaired.
After applying e_i several times, this $i+1$ is changed to i.
Repeating this process, using various e_i we may reach a tableau whose first row consists of only 1s.
Similarly using various e_i the second row can be made into only 2s, ... Can reach u_λ from any $t \in B_\lambda$ by e's.

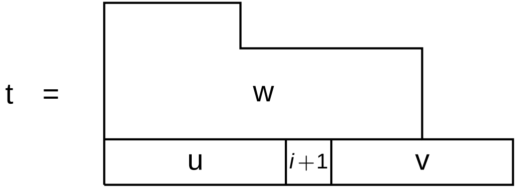
Consider an $i+1$ in the first row of a tableau $t \in B_\lambda$.



$$\text{word}(t) = \text{word}(w)\, u \;\; i{+}1 \;\; v$$

v has no letters i. Thus the $i+1$ is i-unpaired.
After applying e_i several times, this $i+1$ is changed to i.
Repeating this process, using various e_i we may reach a tableau whose first row consists of only 1s.
Similarly using various e_i the second row can be made into only 2s, ... Can reach u_λ from any $t \in B_\lambda$ by e's.

Consider an $i+1$ in the first row of a tableau $t \in B_\lambda$.

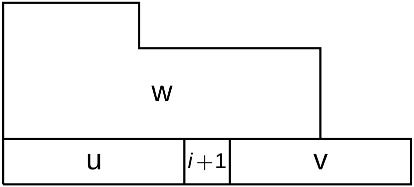


$$\mathrm{word}(t) = \mathrm{word}(w)\, u \;\; i{+}1 \;\; v$$

v has no letters i. Thus the $i+1$ is i-unpaired.
After applying e_i several times, this $i+1$ is changed to i.
Repeating this process, using various e_i we may reach a tableau whose first row consists of only 1s.
Similarly using various e_i the second row can be made into only 2s, ... Can reach u_λ from any $t \in B_\lambda$ by e's.

Consider an $i+1$ in the first row of a tableau $t \in B_\lambda$.

t =

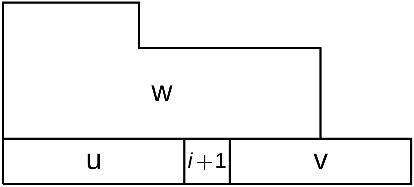


$$\mathrm{word}(t) = \mathrm{word}(w)\, u \;\; i{+}1 \;\; v$$

v has no letters i. Thus the $i+1$ is i-unpaired.
After applying e_i several times, this $i+1$ is changed to i.
Repeating this process, using various e_i we may reach a tableau whose first row consists of only 1s.
Similarly using various e_i the second row can be made into only 2s, ... Can reach u_λ from any $t \in B_\lambda$ by e's.

Consider an $i+1$ in the first row of a tableau $t \in B_\lambda$.

t =

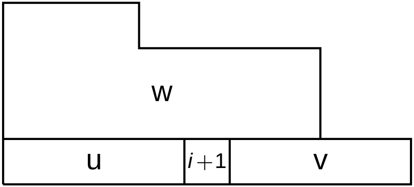


$$\mathrm{word}(t) = \mathrm{word}(w)\, u \;\; i{+}1 \;\; v$$

v has no letters i. Thus the $i+1$ is i-unpaired.
After applying e_i several times, this $i+1$ is changed to i.
Repeating this process, using various e_i we may reach a tableau whose first row consists of only 1s.
Similarly using various e_i the second row can be made into only 2s, ... Can reach u_λ from any $t \in B_\lambda$ by e's.

Consider an $i+1$ in the first row of a tableau $t \in B_\lambda$.

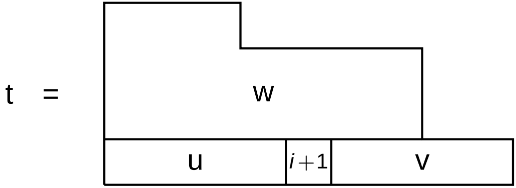
t =



$$\mathrm{word}(t) = \mathrm{word}(w)\, u \;\; i{+}1 \;\; v$$

v has no letters i. Thus the $i+1$ is i-unpaired.
After applying e_i several times, this $i+1$ is changed to i.
Repeating this process, using various e_i we may reach a tableau whose first row consists of only 1s.
Similarly using various e_i the second row can be made into only 2s, ... Can reach u_λ from any $t \in B_\lambda$ by e's.

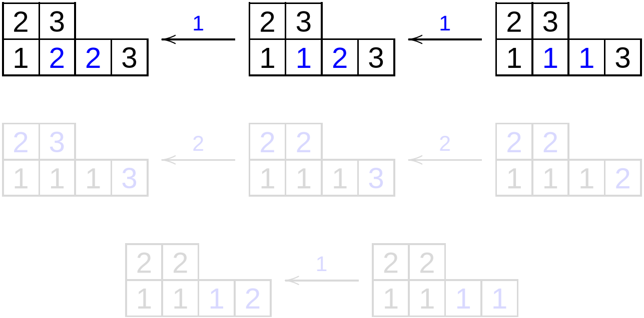
Consider an $i+1$ in the first row of a tableau $t \in B_\lambda$.



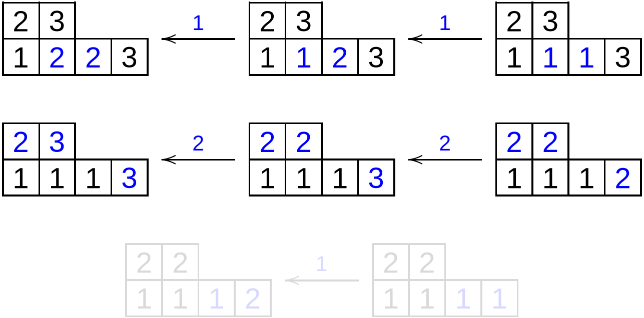
$$\mathrm{word}(t) = \mathrm{word}(w)\, u \;\; i+1 \;\; v$$

v has no letters i. Thus the $i+1$ is i-unpaired.
After applying e_i several times, this $i+1$ is changed to i.
Repeating this process, using various e_i we may reach a tableau whose first row consists of only 1s.
Similarly using various e_i the second row can be made into only 2s, ... Can reach u_λ from any $t \in B_\lambda$ by e's.

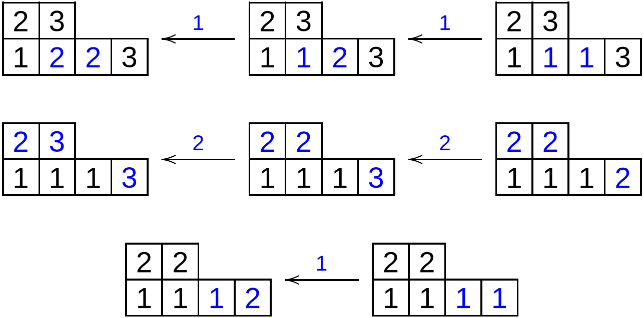
i-unpaired letters in blue



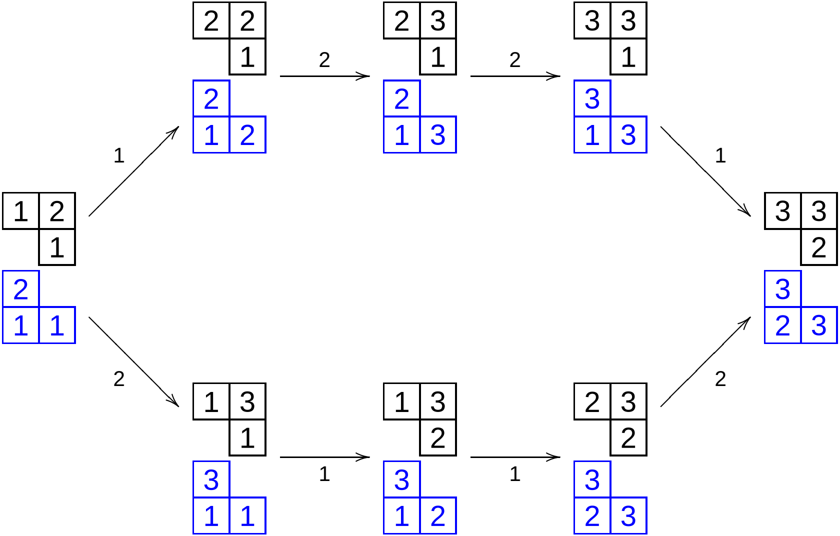
i-unpaired letters in blue



i-unpaired letters in blue

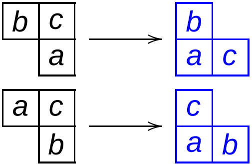


$$B_{(2,2)/(1)} \overset{J}{\cong} B_{(2,1)}$$



Knuth relations

Exercise: Show that for all n the above isomorphism sends



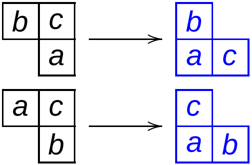
$bca \mapsto bac$ if $a < b \leq c$

$acb \mapsto cab$ if $a \leq b < c$

$P(bca) = bac = P(bac)$ resp. $P(acb) = cab = P(cab)$. Jeu:

Knuth relations

Exercise: Show that for all n the above isomorphism sends



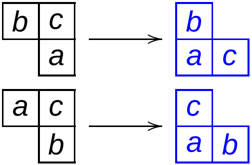
$bca \mapsto bac$ if $a < b \leq c$

$acb \mapsto cab$ if $a \leq b < c$

$P(bca) = bac = P(bac)$ resp. $P(acb) = cab = P(cab)$. Jeu:

Knuth relations

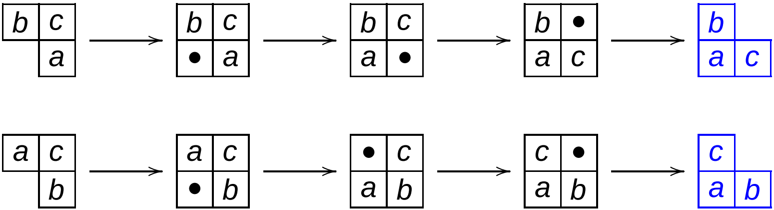
Exercise: Show that for all n the above isomorphism sends



$bca \mapsto bac \quad \text{if } a < b \leq c$

$acb \mapsto cab \quad \text{if } a \leq b < c$

$P(bca) = bac = P(bac)$ resp. $P(acb) = cab = P(cab)$. Jeu:



Knuth relations

There are isomorphisms $\mathrm{id} \otimes J \otimes \mathrm{id}$

$$\square^{\otimes p} \otimes \begin{smallmatrix}\square & \square \\ & \square\end{smallmatrix} \otimes \square^{\otimes q} \longrightarrow \square^{\otimes p} \otimes \begin{smallmatrix}\square & \\ \square & \square\end{smallmatrix} \otimes \square^{\otimes q}$$

Define equivalence relation $\equiv$ on $B^{\otimes L}$ by

$$\begin{aligned} ubcav &\equiv ubacv && \text{if } a < b \leq c \\ uacbv &\equiv ucabv && \text{if } a \leq b < c. \end{aligned}$$

Lemma. $w \equiv w'$ implies $C(w) \cong C(w')$ with $w \mapsto w'$ and $P(w) = P(w')$.
Proof. $w = ubcav$, $w' = ubacv$.

$$B^{\otimes p} \otimes C(bca) \otimes B^{\otimes q} \cong B^{\otimes p} \otimes C(bac) \otimes B^{\otimes q}$$
$$u \otimes bca \otimes v \mapsto u \otimes bac \otimes v$$

Restrict:

$$\begin{array}{ccc} C(u \otimes bca \otimes v) & \rightarrow & C(u \otimes bac \otimes v) \\ \downarrow & & \downarrow \\ B_\lambda & \xrightarrow[\text{id}]{} & B_\lambda \end{array}$$

$$\begin{array}{ccc} u \otimes bca \otimes v & \rightarrow & u \otimes bac \otimes v \\ \downarrow & & \downarrow \\ P(ubcav) & \xrightarrow[\text{id}]{} & P(ubacv) \end{array}$$

Lemma. $w \equiv w'$ implies $C(w) \cong C(w')$ with $w \mapsto w'$ and $P(w) = P(w')$.
Proof. $w = ubcav$, $w' = ubacv$.

$$B^{\otimes p} \otimes C(bca) \otimes B^{\otimes q} \cong B^{\otimes p} \otimes C(bac) \otimes B^{\otimes q}$$
$$u \otimes bca \otimes v \mapsto u \otimes bac \otimes v$$

Restrict:

$$\begin{array}{ccc} C(u \otimes bca \otimes v) & \rightarrow & C(u \otimes bac \otimes v) \\ \downarrow & & \downarrow \\ B_\lambda & \xrightarrow[\text{id}]{} & B_\lambda \end{array}$$

$$\begin{array}{ccc} u \otimes bca \otimes v & \rightarrow & u \otimes bac \otimes v \\ \downarrow & & \downarrow \\ P(ubcav) & \xrightarrow[\text{id}]{} & P(ubacv) \end{array}$$

Jeu de taquin

Swap the hole with the entry above or to the right, whichever is smaller.

Jeu de taquin

4	6			
3	4	4	6	
1	3	3	5	
	1	2	4	
	•	1	3	5

Jeu de taquin

4	6			
3	4	4	6	
1	3	3	5	
	•	2	4	
	1	1	3	5

Jeu de taquin

4	6			
3	4	4	6	
1	3	3	5	
	2	•	4	
	1	1	3	5

Jeu de taquin

4	6			
3	4	4	6	
1	3	•	5	
	2	3	4	
	1	1	3	5

Jeu de taquin

4	6			
3	4	•	6	
1	3	4	5	
	2	3	4	
	1	1	3	5

Jeu de taquin

4	6			
3	4	6	•	
1	3	4	5	
	2	3	4	
	1	1	3	5