Crystals, combinatorics, and $k$-Schur functions

M. Shimozono

Department of Mathematics
Virginia Tech

Berkeley
June 2008
Crystal graphs

Topics

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

General crystal graphs


Crystal graphs of classical Lie algebras


Crystal graphs

\[ \mathfrak{g} \]  simple Lie algebra over \( \mathbb{C} \)

\[ U_q(\mathfrak{g}) \]  quantized universal enveloping algebra of \( \mathfrak{g} \)

\[ U_q(\mathfrak{g})\text{-Mod} \]  category of fin. dim. irreducible integrable \( U_q(\mathfrak{g}) \)-modules

\[ C(\mathfrak{g}) \]  category of crystal graphs of \( M \in U_q(\mathfrak{g})\text{-Mod} \)

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

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

\[
\begin{array}{c}
1 \quad 1 \quad 2
\end{array}
\]
Crystal graphs

\( g \) simple Lie algebra over \( \mathbb{C} \)

\( U_q(g) \) quantized universal enveloping algebra of \( g \)

\( U_q(g) \)-Mod category of fin. dim. irreducible integrable \( U_q(g) \)-modules

\( C(g) \) category of crystal graphs of \( M \in U_q(g) \)-Mod

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

Example for \( g = sl_2 \): \( I = \{1\} \)

\[ \begin{array}{c}
\begin{array}{c}
1
\end{array}
\end{array} \rightarrow \begin{array}{c}
\begin{array}{c}
2
\end{array}
\end{array} \]
Operations on crystal graphs

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

- Disjoint union (direct sum)
- Taking a connected component (summand)
- Cartesian product* (tensor product)
- Reversing all arrows (dual)
- Dynkin automorphisms
$\mathcal{C}(\mathfrak{g})$ 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

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

- Disjoint union (direct sum)
- Taking a connected component (summand)
- Cartesian product* (tensor product)
- Reversing all arrows (dual)
- Dynkin automorphisms
$\mathcal{C}(\mathfrak{g})$ is closed under:

- Disjoint union (direct sum)
- Taking a connected component (summand)
- Cartesian product* (tensor product)
- Reversing all arrows (dual)
- Dynkin automorphisms
$C(g)$ is closed under:

- Disjoint union (direct sum)
- Taking a connected component (summand)
- Cartesian product* (tensor product)
- Reversing all arrows (dual)
- Dynkin automorphisms
Structure of $\mathcal{C}(g)$

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

Representation theory of $U_q(g)$-Mod says:

There is a bijection

\[
P^+ \rightarrow \left\{ \text{iso. classes of connected graphs in } \mathcal{C}(g) \right\} \\
\lambda \mapsto B_\lambda \quad \text{crystal graph of highest weight } \lambda
\]

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

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

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

$\{\omega_i \mid i \in I\}$ fun. wts.

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

Representation theory of $U_q(g)$-Mod says:

There is a bijection

$P^+ \rightarrow \left\{ \text{iso. classes of} \right.$  

$\left\{ \text{connected graphs in } \mathcal{C}(g) \right\}$

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

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

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

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

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

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

There is a bijection

\[ P^+ \to \left\{ \text{iso. classes of connected graphs in } \mathcal{C}(\mathfrak{g}) \right\} \]
\[ \lambda \mapsto B_\lambda \text{ crystal graph of highest weight } \lambda \]

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).
Crystal graphs

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

$$\{\omega_i \mid i \in I\} \text{ fund. wts.}$$

$$P^+ = \bigoplus_{i \in I} \mathbb{Z}_{\geq 0} \omega_i \quad \text{dominant integral weights}$$

Representation theory of $U_q(g)$-Mod says:

There is a bijection

$$P^+ \rightarrow \left\{ \text{iso. classes of connected graphs in } \mathcal{C}(g) \right\}$$

$$\lambda \mapsto B_\lambda \quad \text{crystal graph of highest weight } \lambda$$

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

A morphism $B \rightarrow B'$ between objects in $\mathcal{C}(g)$, 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

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

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

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

A morphism $B \rightarrow B'$ 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 $B_\lambda$ of irr. int. highest weight $U_q(g)$-modules when $g$ is a symm. Kac-Moody algebra.

We consider the special case that $g$ is an affine algebra and focus on:
- Nonhighest weight (Kirillov-Reshetikhin) $U'_q(g)$-modules
- Special properties of affine-to-finite branching
Comment on Littelmann paths

Littelmann paths: construction of crystal graphs $B_{\lambda}$ 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
Crystal graphs

\[ C(\mathfrak{sl}_2) \]

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

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

\[ \bullet \rightarrow e(b) \rightarrow b \rightarrow f(b) \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \]

\[ \left| \begin{align*}
\leftarrow \varepsilon(b) \rightarrow \\
\leftarrow \varphi(b) \rightarrow
\end{align*} \right| \]

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

\[
g = \mathfrak{sl}_2, \quad l = \{1\}, \quad P^+ = \mathbb{Z}_{\geq 0}\omega
\]

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

\[
\bullet \longrightarrow e(b) \longrightarrow b \longrightarrow f(b) \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \bullet
\]

\[
| \quad \leftarrow \varepsilon(b) \quad \longrightarrow | \quad \leftarrow \varphi(b) \quad \longrightarrow |\]

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
\[ C(\mathfrak{sl}_2) \]

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

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

\[ \bullet \rightarrow e(b) \rightarrow b \rightarrow f(b) \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \]

\[ \mid \leftarrow \varepsilon(b) \rightarrow \mid \leftarrow \varphi(b) \rightarrow \mid \]

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\textsuperscript{op})

\[ b \underbrace{\varepsilon(b)}_{\varphi(b)} \underbrace{\varepsilon(b)}_{\varphi(b)} \]

\[ \begin{array}{ccc}
\bullet & ) & 111 \\
\downarrow & \downarrow & \downarrow \\
\bullet & ))( & 112 \\
\downarrow & \downarrow & \downarrow \\
\bullet & )) & 122 \\
\downarrow & \downarrow & \downarrow \\
\bullet & ((( & 222 \\
\end{array} \]

\( f \) changes rightmost ")" to "(" or 1 to 2
\( e \) changes leftmost "(" to "")" or 2 to 1
Signature of elements in a string (Kashiwara\textsuperscript{op})

\[ b \begin{array}{c}
\varphi(b) \\
\epsilon(b) \\
1 \varphi(b) 2 \epsilon(b)
\end{array} \]

\[
\begin{array}{c}
\bullet \\
\downarrow \\
\bullet \\
\downarrow \\
\bullet \\
\downarrow
\end{array} \begin{array}{c}
))) \\
)() \\
)() \\
)( \\
)(
\end{array} \begin{array}{c}
111 \\
112 \\
122 \\
222
\end{array}
\]

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

\( e \) changes leftmost "(" to ")" or 2 to 1
Signature of elements in a string (Kashiwara\textsuperscript{op})

\[
\begin{align*}
\text{\(b\)} & \quad \underbrace{\)\)\cdots\)}_{\varphi(b)} \underbrace{\((\cdots(}_{\varepsilon(b)} 1^{\varphi(b)} 2^{\varepsilon(b)} \\
\bullet & \quad ))) 111 \\
\downarrow & \quad \downarrow \downarrow \\
\bullet & \quad ))) 112 \\
\downarrow & \quad \downarrow \downarrow \\
\bullet & \quad ))(( 122 \\
\downarrow & \quad \downarrow \downarrow \\
\bullet & \quad )))(( 222
\end{align*}
\]

\(f\) changes rightmost ")" to "(" or 1 to 2
\(e\) changes leftmost "(" to ")" or 2 to 1
Tensor product (Kashiwara\textsuperscript{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\textsuperscript{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 \( b_1 \):

\[
\begin{align*}
\varphi(b_2) & \quad \varepsilon(b_2) \\
\varphi(b_1) & \quad \varepsilon(b_1)
\end{align*}
\]

Match parentheses. Unmatched substring has the form
\[
\begin{align*}
\underbrace{\ldots)} \underbrace{\ldots( \\
\varphi(b_2) & \quad \varepsilon(b_2) \\
\varphi(b_1) & \quad \varepsilon(b_1)
\end{align*}
\]

\( \varphi(b) \) 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 \( b_1 \):

\[
\begin{align*}
\underbrace{)} \cdots ) ( & \underbrace{(} \cdots ( & \underbrace{)} \cdots ) ( \\
\varphi(b_2) & \varepsilon(b_2) & \varphi(b_1) & \varepsilon(b_1)
\end{align*}
\]

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 $b_1$:

$$\varphi(b_2) \quad \varepsilon(b_2) \quad \varphi(b_1) \quad \varepsilon(b_1)$$

Match parentheses. Unmatched substring has the form

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

$\varphi(b)$ 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 \( b_1 \):

\[
\varphi(b_2) \varepsilon(b_2) \quad \varphi(b_1) \varepsilon(b_1)
\]

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 \( b_1 \):

\[
\begin{align*}
\varphi(b_2) & \cdots \varepsilon(b_2) \\
\varphi(b_1) & \cdots \varepsilon(b_1)
\end{align*}
\]

Match parentheses. Unmatched substring has the form

\[
)) \cdots ) ((( \cdots ( \varepsilon(b_2)
\]

\( \varphi(b) \) 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 ")".
\[ b = b_1 \otimes b_2 \]
Tensor product is associative

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

\[
\begin{array}{c}
\text{)} \text{)} (\ ( (\ (\otimes) ) \ (\otimes) ) \ (\ \text{)} \text{)} \text{)}
\end{array}
\]

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

\[
\begin{array}{c}
\text{)} \text{)} (\ ( (\ (\otimes) ) \ (\otimes) ) \ (\ \text{)} \text{)} \text{)}
\end{array}
\]

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 \(B_1\).

\[
\begin{array}{c}
\text{)} \text{)} \text{) } ( ( ( (\times) ) (\times) ) )
\end{array}
\]

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

\[
\begin{array}{c}
\text{)} \text{)} \text{) } ( ( ( (\times) ) (\times) ) )
\end{array}
\]

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.
$C(g)$ and $i$-strings

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

$e_i(b)$ 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
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"

$e_i(b)$ 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 $C(g)$

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

- Every connected graph $B \in C(g)$ has a unique highest weight vector $u$ (vertex with no in-edges).
  Let $u_\lambda$ be the h.w.v. of $B_\lambda$.

- 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 $\text{shape}(b) = \lambda$.

- Let $B_1, B_2 \in C(g)$ and $b_2 \otimes b_1 \in B_2 \otimes B_1$. Then

  \[ P(b_2 \otimes b_1) = P(P(b_2) \otimes P(b_1)). \]
For $B \in C(g)$ and $b \in B$, let $C(b) \subset B$ be the component of $b$.

- Every connected graph $B \in C(g)$ has a unique highest weight vector $u$ (vertex with no in-edges). Let $u_\lambda$ be the h.w.v. of $B_\lambda$.

- 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 $\text{shape}(b) = \lambda$.

- Let $B_1, B_2 \in C(g)$ and $b_2 \otimes b_1 \in B_2 \otimes B_1$. Then

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

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

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

- Every connected graph $B \in \mathcal{C}(g)$ has a unique highest weight vector $u$ (vertex with no in-edges). Let $u_\lambda$ be the h.w.v. of $B_\lambda$.
- 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 $\text{shape}(b) = \lambda$.
- Let $B_1, B_2 \in \mathcal{C}(g)$ 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 $b_2 \otimes b_1 \mapsto P(b_2) \otimes P(b_1)$. 

\[
\begin{array}{cccccc}
C(b_2 \otimes b_1) & \longrightarrow & C(P(b_2) \otimes P(b_1)) \\
\downarrow & & \downarrow \\
B_\lambda & \longrightarrow & B_\lambda \\
\downarrow & & \downarrow \\
b_2 \otimes b_1 & \longrightarrow & P(b_2) \otimes P(b_1) \\
\downarrow & & \downarrow \\
P(b_2 \otimes b_1) & \longrightarrow & P(P(b_2) \otimes P(b_1)) \\
\end{array}
\]
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 $b_2 \otimes b_1 \mapsto P(b_2) \otimes P(b_1)$. 

\[
\begin{array}{c}
\begin{array}{c}
C(b_2 \otimes b_1) \\
\downarrow \\
B_\lambda
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
C(P(b_2) \otimes P(b_1)) \\
\downarrow \\
B_\lambda
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
b_2 \otimes b_1 \\
\downarrow \\
P(b_2) \otimes P(b_1)
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
P(b_2 \otimes P(b_1)) \\
\downarrow \\
P(P(b_2) \otimes P(b_1))
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
P(P(b_2 \otimes P(b_1))) \\
\downarrow \\
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\end{array}
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\end{array}
\end{array}
\]
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 $b_2 \otimes b_1 \mapsto P(b_2) \otimes P(b_1)$.

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

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

\[b_2 \otimes b_1 \longrightarrow P(b_2) \otimes P(b_1) \quad \text{id} \]
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 $b_2 \otimes b_1 \mapsto P(b_2) \otimes P(b_1)$. 

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

\[ \begin{array}{ccc}
b_2 \otimes b_1 & \longrightarrow & P(b_2) \otimes P(b_1) \\
& \downarrow & \downarrow \\
P(b_2 \otimes b_1) & \longrightarrow & P(P(b_2) \otimes P(b_1))
\end{array} \]
$P^+ (\mathfrak{sl}_n)$ 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: $n = 4$:

$3\omega_1 + \omega_2 + 2\omega_3 \mapsto \begin{array}{cccc}
\end{array}$

$\omega_1 \mapsto \Box$
$P^+ (\mathfrak{sl}_n)$ 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: $n = 4$:

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

$\omega_1 \mapsto \square$
\( P^+ (\mathfrak{sl}_n) \) and partitions

\[ I = \{1, 2, \ldots, n - 1\} \]

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

\[ \sum_{i=1}^{n-1} a_i \omega_i \text{ goes to the partition with } a_i \text{ columns of size } i. \]

Example: \( n = 4 \):

\[ 3\omega_1 + \omega_2 + 2\omega_3 \leftrightarrow \]

\[ \omega_1 \leftrightarrow \]

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

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

1-strings:
\[ B_{\omega_1} \in C(\mathfrak{g}) \]

\[ B = B_{\omega_1} : \text{Vector rep: vertices 1 through } n. \]

2-strings:
Crystal graphs

\[ B_{\omega_1} \in C(g) \]

\[ B = B_{\omega_1} : \text{Vector rep: vertices } 1 \text{ through } n. \]

3-strings:

[Diagram showing a sequence of vertices 1 through 4 with arrows indicating transitions between them.]
**Words and $\mathcal{C}(s\mathfrak{l}_n)$**

$B^\otimes L$: 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 $C(\mathfrak{sl}_n)$

$B^\otimes L$: 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 $i$.  

Crystal graphs
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 $i$. 

Crystal graphs
**C(\mathfrak{sl}_n) and tableaux**

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

\( B_D \): set of tableaux of shape \( D \)

Row reading word:

\[
\text{word}(t) = 224 \cdot 1334 \cdot 123 \cdot 12
\]

We identify a tableau with its reading word:

\[
B_D \rightarrow B^{|D|}
\]

\[
t \mapsto \text{word}(t)
\]
\( \mathcal{C}(\mathfrak{sl}_n) \) and tableaux

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

\[
\begin{array}{ccc}
2 & 2 & 4 \\
1 & 3 & 3 & 4 \\
1 & 2 & 3 \\
1 & 2 \\
\end{array}
\]

\( B_D \): set of tableaux of shape \( D \)
row reading word:

\[
\text{word}(t) = 224 \cdot 1334 \cdot 123 \cdot 12
\]

We identify a tableau with its reading word:

\[
B_D \rightarrow B^{\otimes |D|}
\]

\[
t \mapsto \text{word}(t)
\]
Crystal graphs

\[ C(\mathfrak{sl}_n) \text{ and tableaux} \]

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

\[
\begin{array}{ccc}
2 & 2 & 4 \\
1 & 3 & 3 & 4 \\
1 & 2 & 3 \\
1 & 2
\end{array}
\]

\( B_D \): set of tableaux of shape \( D \)
row reading word:
\[
\text{word}(t) = 224 \cdot 1334 \cdot 123 \cdot 12
\]

We identify a tableau with its reading word:
\[
B_D \rightarrow B^{\otimes |D|} \\
t \mapsto \text{word}(t)
\]
Crystal graphs

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

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

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

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

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

We identify a tableau with its reading word:

$$B_D \rightarrow B^{\otimes |D|}$$

$$t \mapsto \text{word}(t)$$
$B_D \in C(\mathfrak{sl}_n)$

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 $i+1$.

\[
\begin{array}{cccccc}
1 & 1 & 1 & 2 & 2 & 2 \\
& 1 & 1 & 1 & 1 & 1 & 2 & 2
\end{array}
\]

\[
11122222 \cdot 1111122
\]

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

\[
\begin{array}{c}
2 \\
1
\end{array}
\]

in some row, but this creates a new matching pair, while $e_1$ and $f_1$ preserve the matched letters.
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 $i + 1$.

\[
\begin{array}{cccccccc}
1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 \\
1 & 1 & 1 & 1 & 1 & 2 & 2
\end{array}
\]

\[11122222 \cdot 1111122\]

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

\[
\begin{array}{cc}
2 & 1
\end{array}
\]

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

\[ B_D \in C(\mathfrak{sl}_n) \]

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 \( i+1 \).

\[
\begin{array}{ccccccc}
1 & 1 & 1 & 2 & 2 & 2 & 2 \\
1 & 1 & 1 & 1 & 1 & 2 & 2 \\
\end{array}
\]

\[ 11122222 \cdot 1111122 \]

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

\[
\begin{array}{cc}
2 & 1 \\
\end{array}
\]

in some row, but this creates a new matching pair, while \( e_1 \) and \( f_1 \) preserve the matched letters.
The image of $B_D$ in $B^{|D|}$ is stable under $e_i$ and $f_i$ for all $i \in I$. Enough to check for skew subtableau of letters $i$ and $i + 1$.

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 $B_{\lambda}$

$D = \lambda$ partition shape

Let $u_{\lambda} \in B_{\lambda}$ be the Yamanouchi tableau of shape $\lambda$, the one having only letters $i$ in row $i$ for all $i$.

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

\[
\begin{array}{cccc}
3 \\
2 & 2 \\
1 & 1 & 1 & 1 \\
\end{array}
\]

$u_{\lambda}$ is a highest weight vector: every $i + 1$ is $i$-paired.

$\varphi_1(u_{\lambda}) = 2$, $\varphi_2(u_{\lambda}) = 1$, $\varphi_3(u_{\lambda}) = 1$.

$\lambda = 2\omega_1 + 1\omega_2 + 1\omega_3$. 
Crystal graphs

Yamanouchi tableau in $B_\lambda$

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

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

```
  \[ u_\lambda = \begin{array}{cccc}
  3 \\
  2 & 2 \\
  1 & 1 & 1 & 1 \\
\end{array} \]
```

$u_\lambda$ 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 $B_\lambda$

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

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

$$u_\lambda = \begin{array}{ccc}
3 \\
2 & 2 \\
1 & 1 & 1 & 1
\end{array}$$

$u_\lambda$ 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 $B_\lambda$

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

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

```
+---+---+---+
|   | 3 |   |
+---+---+---+
| 2 | 2 |   |
+---+---+---+
| 1 | 1 | 1 | 1 |
+---+---+---+
```

$u_\lambda$ 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 $t \in B_\lambda$.

\[
t = \begin{array}{c}
\text{w} \\
\text{u} & i+1 & v
\end{array}
\]

\[
\text{word}(t) = \text{word}(w) \text{ } u \text{ } i+1 \text{ } v
\]

$v$ 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 $u_\lambda$ from any $t \in B_\lambda$ by $e$'s.
Consider an $i + 1$ in the first row of a tableau $t \in B_\lambda$.

\[
\begin{array}{c}
t = \\
\begin{array}{c}
w \\
\hline \\
u \quad i + 1 \quad v
\end{array}
\end{array}
\]

$\text{word}(t) = \text{word}(w) \ u \ ii + 1 \ v$

$v$ 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 $u_\lambda$ from any $t \in B_\lambda$ by e's.
Consider an $i + 1$ in the first row of a tableau $t \in B_\lambda$.

\[
t = \begin{array}{c}
\text{w} \\
\text{u} & i+1 & \text{v}
\end{array}
\]

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

$v$ 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 $u_\lambda$ from any $t \in B_\lambda$ by e’s.
Consider an \( i + 1 \) in the first row of a tableau \( t \in B_\lambda \).

\[
\begin{array}{c}
t = \\
\hline \hline \\
\hline w \\
\hline u \quad i+1 \quad v \\
\end{array}
\]

\[
\text{word}(t) = \text{word}(w) \ u \ i+1 \ v
\]

\( v \) 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 \( u_\lambda \) from any \( t \in B_\lambda \) by e’s.
Consider an $i + 1$ in the first row of a tableau $t \in B_\lambda$.

$$t = \begin{array}{cccc}
    & & & w \\
    u & i+1 & v \\
\end{array}$$

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

$v$ 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 $u_\lambda$ from any $t \in B_\lambda$ by e’s.
Consider an $i + 1$ in the first row of a tableau $t \in B_\lambda$.

$$t = \begin{array}{c} w \\ u \quad i+1 \quad v \end{array}$$

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

$v$ 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 $u_\lambda$ from any $t \in B_\lambda$ by $e$'s.
Consider an $i + 1$ in the first row of a tableau $t \in B_\lambda$.

\[
t = \begin{array}{c}
\ & u & i + 1 & v \\
\ & w \\
\end{array}
\]

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

$v$ 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 $u_\lambda$ from any $t \in B_\lambda$ by $e$’s.
$i$-unpaired letters in blue

Crystal graphs
$i$-unpaired letters in blue

\[
\begin{array}{c|c|c|c}
2 & 3 & \text{i} & \text{unpaired letters in blue} \\
1 & 2 & 2 & 3 \\
\end{array}
\]

\[
\begin{array}{c|c|c|c}
2 & 3 & \text{i} & \text{unpaired letters in blue} \\
1 & 1 & 2 & 3 \\
\end{array}
\]

\[
\begin{array}{c|c|c|c}
2 & 3 & \text{i} & \text{unpaired letters in blue} \\
1 & 1 & 1 & 3 \\
\end{array}
\]
$i$-unpaired letters in blue

\begin{align*}
\begin{array}{c|c|c|c|c}
2 & 3 & & & \\
1 & 2 & 2 & 3 \\
\end{array} & \xrightarrow{1} & \begin{array}{c|c|c|c|c}
2 & 3 & & & \\
1 & 1 & 2 & 3 \\
\end{array} & \xrightarrow{1} & \begin{array}{c|c|c|c|c}
2 & 3 & & & \\
1 & 1 & 1 & 3 \\
\end{array} \\
\begin{array}{c|c|c|c|c}
2 & 3 & & & \\
1 & 1 & 1 & 3 \\
\end{array} & \xrightarrow{2} & \begin{array}{c|c|c|c|c}
2 & 2 & & & \\
1 & 1 & 1 & 3 \\
\end{array} & \xrightarrow{2} & \begin{array}{c|c|c|c|c}
2 & 2 & & & \\
1 & 1 & 1 & 2 \\
\end{array} \\
\begin{array}{c|c|c|c|c}
2 & 2 & & & \\
1 & 1 & 1 & 2 \\
\end{array} & \xrightarrow{1} & \begin{array}{c|c|c|c|c}
2 & 2 & & & \\
1 & 1 & 1 & 1 \\
\end{array} & \xrightarrow{1} & \begin{array}{c|c|c|c|c}
2 & 2 & & & \\
1 & 1 & 1 & 1 \\
\end{array} \\
\end{align*}
Crystal graphs

\[ B_2 \cong J \cong B_2 \]
Knuth relations

Exercise: Show that for all $n$ the above isomorphism sends

$\begin{bmatrix} b & c \\ a & \end{bmatrix} \rightarrow \begin{bmatrix} b \\ a & c \end{bmatrix} \quad bca \leftrightarrow bac \quad \text{if } a < b \leq c$

$\begin{bmatrix} a & c \\ b & \end{bmatrix} \rightarrow \begin{bmatrix} c \\ a & b \end{bmatrix} \quad acb \leftrightarrow cab \quad \text{if } a \leq b < c$

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

Exercise: Show that for all $n$ the above isomorphism sends

\[ bca \mapsto bac \quad \text{if} \quad a < b \leq c \]
\[ acb \mapsto cab \quad \text{if} \quad a \leq b < c \]

\[ P(bca) = bac = P(bac) \text{ resp. } P(acb) = cab = P(cab). \]

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

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

$\otimes p \otimes \otimes J \otimes \otimes q \rightarrow \otimes p \otimes \otimes J \otimes \otimes q$

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

$ubcav \equiv ubacv$ if $a < b \leq c$

$uacbv \equiv ucabv$ if $a \leq b < c$. 
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:

$$C(u \otimes bca \otimes v) \rightarrow C(u \otimes bac \otimes v)$$

$$B_\lambda \xrightarrow{id} B_\lambda$$

$$u \otimes bca \otimes v \rightarrow u \otimes bac \otimes v$$

$$P(ubcav) \xrightarrow{id} P(ubacv)$$
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:

\[
C(u \otimes bca \otimes v) \rightarrow C(u \otimes bac \otimes v)
\]
\[
u \otimes bca \otimes v \mapsto u \otimes bac \otimes v
\]
Jeu de taquin

Swap the hole with the entry above or to the right, whichever is smaller.
<table>
<thead>
<tr>
<th>4</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>• 1</td>
<td>3</td>
</tr>
</tbody>
</table>
Crystal graphs

Jeu de taquin

\[
\begin{array}{cccc}
4 & 6 & & \\
3 & 4 & 4 & 6 \\
1 & 3 & 3 & 5 \\
\bullet & 2 & 4 & \\
1 & 1 & 3 & 5 \\
\end{array}
\]
Jeu de taquin
Crystal graphs

Jeu de taquin

<table>
<thead>
<tr>
<th>4</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Crystal graphs

Jeu de taquin

\[
\begin{array}{cccc}
4 & 6 & \\
3 & 4 & \bullet & 6 \\
1 & 3 & 4 & 5 \\
2 & 3 & 4 & \\
1 & 1 & 3 & 5 \\
\end{array}
\]
Jeu de taquin

\[\begin{array}{cccc}
4 & 6 & \cdot & \\
3 & 4 & 6 & \\
1 & 3 & 4 & 5 \\
2 & 3 & 4 & \\
1 & 1 & 3 & 5 \\
\end{array}\]