Combinatorial Models in the Representation Theory of Affine Lie Algebras

Permutation Patterns 2022

Adam Schultze

Saint Olaf College

Joint work with Carly Briggs and Cristian Lenart

The Story

The Story

- Crystal Graphs provide nice combinatorial models for representations of Lie algebras.

The Story

- Crystal Graphs provide nice combinatorial models for representations of Lie algebras.
- The Tableaux model is simpler and has less structure.

The Story

- Crystal Graphs provide nice combinatorial models for representations of Lie algebras.
- The Tableaux model is simpler and has less structure.
- The Quantum Alcove model has extra structure which makes it easier to do several computations (energy function, combinatorial R-Matrix, ...).

The Story

- Crystal Graphs provide nice combinatorial models for representations of Lie algebras.
- The Tableaux model is simpler and has less structure.
- The Quantum Alcove model has extra structure which makes it easier to do several computations (energy function, combinatorial R-Matrix, ...).
- It is therefore beneficial to have an explicit isomorphism between the two models.

Crystal Graphs

Crystal Graphs

Main idea: use colored directed graphs to encode certain representations of $\mathfrak{g}$ complex semisimple or affine Lie algebras.

Crystal Graphs

Main idea: use colored directed graphs to encode certain representations of $\mathfrak{g}$ complex semisimple or affine Lie algebras.

As a combinatorial object, a *Kashiwara crystal* of a given type, rank n, and shape $\lambda = (\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n)$ is

Crystal Graphs

Main idea: use colored directed graphs to encode certain representations of $\mathfrak{g}$ complex semisimple or affine Lie algebras.

As a combinatorial object, a *Kashiwara crystal* of a given type, rank n, and shape $\lambda = (\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n)$ is

1. a set B

Crystal Graphs

Main idea: use colored directed graphs to encode certain representations of $\mathfrak{g}$ complex semisimple or affine Lie algebras.

As a combinatorial object, a *Kashiwara crystal* of a given type, rank n, and shape $\lambda = (\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n)$ is

1. a set B
2. along with the maps $e_i, f_i : B \to B \cup \{0\}$ (for $1 \leq i \leq n$).

Crystal Graphs

Main idea: use colored directed graphs to encode certain representations of $\mathfrak{g}$ complex semisimple or affine Lie algebras.

As a combinatorial object, a *Kashiwara crystal* of a given type, rank n, and shape $\lambda = (\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n)$ is

1. a set B
2. along with the maps $e_i, f_i : B \to B \cup \{0\}$ (for $1 \leq i \leq n$).

Crystal graph: directed graph on B with edges colored $b \xrightarrow{i} b'$ exactly for $f_i(b) = b'$.

Kirillov-Reshetikhin (KR) crystals

Kirillov-Reshetikhin (KR) crystals

Crystals for certain affine Lie algebras $\hat{\mathfrak{g}}$.

Kirillov-Reshetikhin (KR) crystals

Crystals for certain affine Lie algebras $\hat{\mathfrak{g}}$.

The corresponding crystals have operators $f_0, f_1, \ldots f_n$.

Kirillov-Reshetikhin (KR) crystals

Crystals for certain affine Lie algebras $\hat{\mathfrak{g}}$.

The corresponding crystals have operators $f_0, f_1, \dots f_n$.

Labeled by $p \times q$ rectangles, and denoted $\mathbf{B}^{p,q}$.

Kirillov-Reshetikhin (KR) crystals

Crystals for certain affine Lie algebras $\hat{\mathfrak{g}}$.

The corresponding crystals have operators $f_0, f_1, \ldots f_n$.

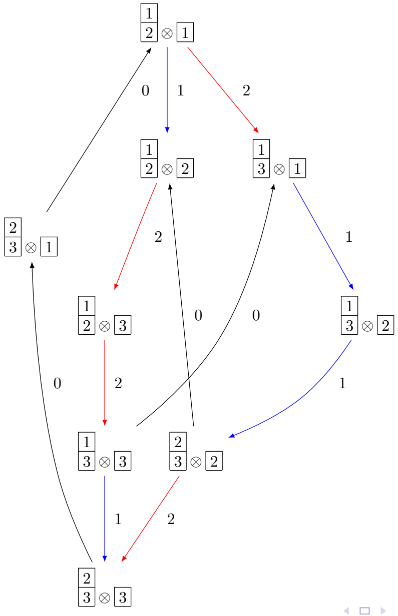
Labeled by $p \times q$ rectangles, and denoted $\mathbf{B}^{p,q}$.

Definition. Given a partition $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n)$, let

$$\mathbf{B}^{\lambda} = \mathbf{B}^{\lambda_1^{'},1} \otimes \mathbf{B}^{\lambda_m^{'},1} \otimes \ldots .$$

The crystal operators are defined on $\mathbf{B}^{\lambda}$ by a tensor product rule.

Type A KR crystal graph with $n = 3$ and shape $\lambda = (2, 1)$



The Quantum Alcove Model for $\mathbf{B}^\lambda$

The Quantum Alcove Model for $\mathbf{B}^\lambda$

The main ingredient is the Weyl group $\mathbf{W}$.

The Quantum Alcove Model for $\mathbf{B}^\lambda$

The main ingredient is the Weyl group $\mathbf{W}$.

For type A_{n-1} this is the Symmetric group S_n.

The Quantum Alcove Model for $\mathbf{B}^\lambda$

The main ingredient is the Weyl group $\mathbf{W}$.

For type A_{n-1} this is the Symmetric group S_n.

The *quantum Bruhat graph* on $\mathbf{W}$ is the directed graph with labeled edges $w \to wt_{i,j}$, where

The Quantum Alcove Model for $\mathbf{B}^\lambda$

The main ingredient is the Weyl group $\mathbf{W}$.

For type A_{n-1} this is the Symmetric group S_n.

The *quantum Bruhat graph* on $\mathbf{W}$ is the directed graph with labeled edges $w \to wt_{i,j}$, where

$l(wt_{i,j}) = l(w) + 1$ (Bruhat graph on black edges), or

The Quantum Alcove Model for $\mathbf{B}^\lambda$

The main ingredient is the Weyl group $\mathbf{W}$.

For type A_{n-1} this is the Symmetric group S_n.

The *quantum Bruhat graph* on $\mathbf{W}$ is the directed graph with labeled edges $w \to wt_{i,j}$, where

$l(wt_{i,j}) = l(w) + 1$ (Bruhat graph on black edges), or
$l(wt_{i,j}) = l(w) + 1 - 2\langle \rho, \alpha_{i,j}^\vee \rangle$ (on red edges).

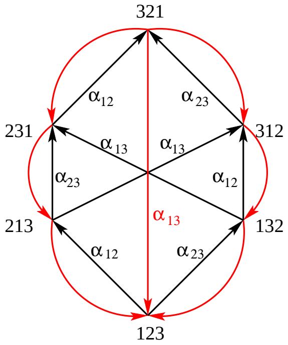
The Quantum Alcove Model for $\mathbf{B}^\lambda$

The main ingredient is the Weyl group **W**.

For type A_{n-1} this is the Symmetric group S_n.

The *quantum Bruhat graph* on **W** is the directed graph with labeled edges $w \to wt_{i,j}$, where

$l(wt_{i,j}) = l(w) + 1$ (Bruhat graph on black edges), or
$l(wt_{i,j}) = l(w) + 1 - 2\langle \rho, \alpha^\vee_{i,j} \rangle$ (on red edges).



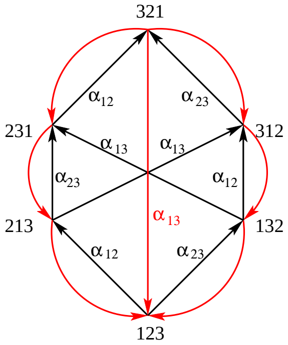
The Quantum Alcove Model for $\mathbf{B}^\lambda$

The main ingredient is the Weyl group $\mathbf{W}$.

For type A_{n-1} this is the Symmetric group S_n.

The *quantum Bruhat graph* on $\mathbf{W}$ is the directed graph with labeled edges $w \to wt_{i,j}$, where

$$l(wt_{i,j}) = l(w) + 1 \text{ (Bruhat graph on black edges), or}$$
$$l(wt_{i,j}) = l(w) + 1 - 2\langle \rho, \alpha^\vee_{i,j} \rangle \text{ (on red edges).}$$



The Quantum Alcove Model for $\mathbf{B}^\lambda$

The Quantum Alcove Model for $\mathbf{B}^\lambda$

Definition. Given a partition $\lambda = (\lambda_1, \ldots, \lambda_n)$, we associate with it a sequence of transpositions, called a λ-*chain*:

$$\Gamma(\lambda) = (t_1, t_2, \ldots, t_m).$$

The Quantum Alcove Model for $\mathbf{B}^\lambda$

Definition. Given a partition $\lambda = (\lambda_1, \ldots, \lambda_n)$, we associate with it a sequence of transpositions, called a λ-*chain*:

$$\Gamma(\lambda) = (t_1, t_2, \ldots, t_m).$$

We consider subsets of positions in Γ,

$$J = (j_1 < j_2 < \ldots < j_s) \subseteq \{1, \ldots, m\}.$$

The Quantum Alcove Model for $\mathbf{B}^\lambda$

Definition. Given a partition $\lambda = (\lambda_1, \ldots, \lambda_n)$, we associate with it a sequence of transpositions, called a λ-*chain*:

$$\Gamma(\lambda) = (t_1, t_2, \ldots, t_m).$$

We consider subsets of positions in Γ,

$$J = (j_1 < j_2 < \ldots < j_s) \subseteq \{1, \ldots, m\}.$$

Definition. A subset J is *admissible* if we have a path in the quantum Bruhat graph

$$Id = w \xrightarrow{t_{j_1}} w_{t_{j_1}} \xrightarrow{t_{j_2}} w_{t_{j_1} t_{j_2}} \ldots \xrightarrow{t_{j_s}} w_{t_{j_1} t_{j_2} t_{j_s}}.$$

The Quantum Alcove Model for $\mathbf{B}^\lambda$

Definition. Given a partition $\lambda = (\lambda_1, \ldots, \lambda_n)$, we associate with it a sequence of transpositions, called a λ-*chain*:

$$\Gamma(\lambda) = (t_1, t_2, \ldots, t_m).$$

We consider subsets of positions in Γ,

$$J = (j_1 < j_2 < \ldots < j_s) \subseteq \{1, \ldots, m\}.$$

Definition. A subset J is *admissible* if we have a path in the quantum Bruhat graph

$$Id = w \xrightarrow{t_{j_1}} w_{t_{j_1}} \xrightarrow{t_{j_2}} w_{t_{j_1} t_{j_2}} \ldots \xrightarrow{t_{j_s}} w_{t_{j_1} t_{j_2} t_{j_s}}.$$

Theorem [Lenart , Naito , Sagaki , Schilling , Shimozono , 2017]
The collection of all admissible subsets, $\mathcal{A}(\Gamma)$,is a combinatorial model for $\mathbf{B}^\lambda$.

Relating the Two Models

Relating the Two Models

We build a forgetful map *fill* : $\mathcal{A}(\Gamma(\lambda)) \to \mathit{Tableau}(\lambda)$.

Relating the Two Models

We build a forgetful map $\mathit{fill} : \mathcal{A}(\Gamma(\lambda)) \to \mathit{Tableau}(\lambda)$.

Example Consider type A with $n = 4$ and $\lambda = (3, 2, 1, 0)$.

Relating the Two Models

We build a forgetful map $\mathit{fill} : \mathcal{A}(\Gamma(\lambda)) \to \mathit{Tableau}(\lambda)$.

Example Consider type A with $n = 4$ and $\lambda = (3, 2, 1, 0)$. Then the associated λ-chain is

$$\Gamma(\lambda) = ((3,4),(2,4),(1,4)|(2,3),(2,4),(1,3),(1,4)|(1,2),(1,3),(1,4)).$$

Relating the Two Models

Relating the Two Models

Example $J = \{1, 2, 4, 5, 8\} \in \mathcal{A}(\Gamma)$ and $\lambda = (3, 2, 1, 0)$.

Relating the Two Models

Example $J = \{1,2,4,5,8\} \in \mathcal{A}(\Gamma)$ and $\lambda = (3,2,1,0)$.

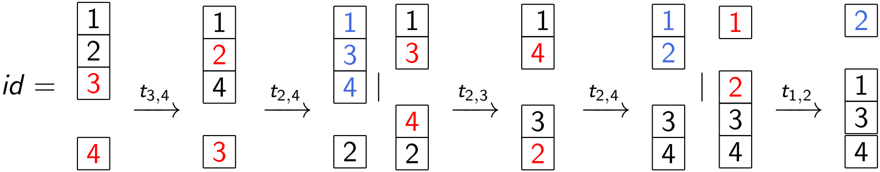
$$(\underline{(3,4)}, \underline{(2,4)}, (1,4) | \underline{(2,3)}, \underline{(2,4)}, (1,3), (1,4) | \underline{(1,2)}, (1,3), (1,4))$$

Relating the Two Models

Example $J = \{1, 2, 4, 5, 8\} \in \mathcal{A}(\Gamma)$ and $\lambda = (3, 2, 1, 0)$.

$$(\underline{(3,4)}, \underline{(2,4)}, (1,4) | \underline{(2,3)}, \underline{(2,4)}, (1,3), (1,4) | \underline{(1,2)}, (1,3), (1,4))$$

We get the corresponding path in the quantum Bruhat graph

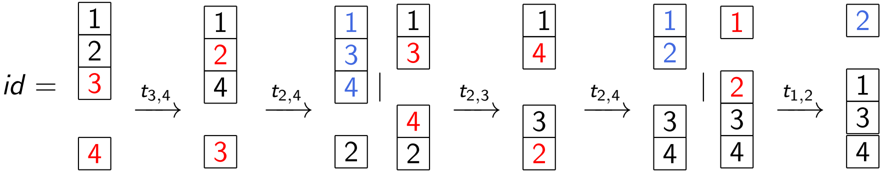


Relating the Two Models

Example $J = \{1, 2, 4, 5, 8\} \in \mathcal{A}(\Gamma)$ and $\lambda = (3, 2, 1, 0)$.

$$(\underline{(3,4)}, \underline{(2,4)}, (1,4) | \underline{(2,3)}, \underline{(2,4)}, (1,3), (1,4) | \underline{(1,2)}, (1,3), (1,4))$$

We get the corresponding path in the quantum Bruhat graph



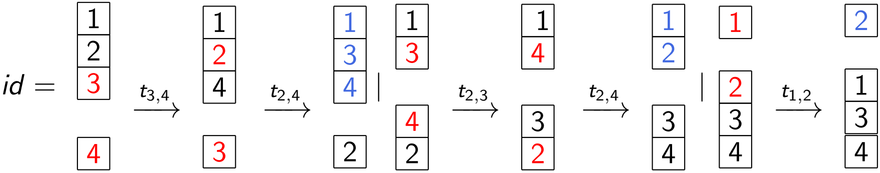
The blue columns (with entries sorted increasingly)

Relating the Two Models

Example $J = \{1, 2, 4, 5, 8\} \in \mathcal{A}(\Gamma)$ and $\lambda = (3, 2, 1, 0)$.

$$(\underline{(3,4)}, \underline{(2,4)}, (1,4) | \underline{(2,3)}, \underline{(2,4)}, (1,3), (1,4) | \underline{(1,2)}, (1,3), (1,4))$$

We get the corresponding path in the quantum Bruhat graph



The blue columns (with entries sorted increasingly) then give us $fill(J) =$

1	1	2
3	2	
4		

.

The Reverse Map in Type A_{n-1}

The Reverse Map in Type A_{n-1}

This is done with two algorithms: Reorder and Path

The Reverse Map in Type A_{n-1}

This is done with two algorithms: Reorder and Path

The Reorder algorithm undoes the "increasingly sorted" part of the fill map.

The Reverse Map in Type A_{n-1}

This is done with two algorithms: Reorder and Path

The Reorder algorithm undoes the "increasingly sorted" part of the fill map.

The Path algorithm then parses through the transpositions of Γ to select the correct path through the quantum Bruhat graph.

The Reverse Map in Type A_{n-1}

This is done with two algorithms: Reorder and Path

The Reorder algorithm undoes the "increasingly sorted" part of the fill map.

The Path algorithm then parses through the transpositions of Γ to select the correct path through the quantum Bruhat graph.

The resulting bijection is a crystal isomorphism [Lenart, Lubovsky, 2015].

The Type C_n Map

- The filling map is similar.
- The inverse map has one major change. Many KN columns have both i and $\bar{\imath}$ in them, so we use the splitting algorithm [Lecouvey] to bijectively make two columns with no $i, \bar{\imath}$ pairs in either.
- Example:

$$\begin{array}{|c|}\hline 4 \\ \hline 5 \\ \hline \bar{5} \\ \hline \bar{4} \\ \hline \bar{3} \\ \hline \end{array} \xrightarrow{\textit{split}} \begin{array}{|c|c|}\hline 4 & 1 \\ \hline 5 & 2 \\ \hline \bar{3} & \bar{5} \\ \hline \bar{2} & \bar{4} \\ \hline \bar{1} & \bar{3} \\ \hline \end{array}$$

  The $\Gamma(k)$ in type C_n comes in two parts.
  One traverses the split columns and the other moves to the next column.
- Then similar Reorder and Path algorithms work.
- So now the reverse map is made up of a process of Split, Reorder, and Path.

The Type B_n Map

- There is a similar filling map
- For the reverse, similar to C_n, we need a splitting map.
- Recall that we now have columns of length $k - 2l$, so we need to Extend back to length k [Briggs].
- Further, the Path algorithm and Reorder algorithm no longer work.
- There is a configuration of two columns CC' that we call being blocked-off.
- Modify Path and Modify Reorder to avoid block-off pattern.

Blocked-Off Pattern

Definition: We say that columns $L = (l_1, l_2, ..., l_k), R' = (r_1, r_2, ..., r_k)$ are *blocked off at* i *by* $b = r_i$ iff $0 < b \geq |l_i|$ and

$$\{1, 2, ..., b\} \subset \{|l_1|, |l_2|, ..., |l_i|\}$$

and

$$\{1, 2, ..., b\} \subset \{|r_1|, |r_2|, ..., |r_i|\}$$

and $|\{j : 1 \leq j \leq i, l_j < 0, r_j > 0\}|$ is odd.

Example: The following columns CC' of height 5 with entries from $[\bar{8}]$ are blocked-off at 4 by 3:

1	1
4	5
$\bar{2}$	$\bar{2}$
$\bar{3}$	3
5	8

The Type D_n Map

- There is a similar filling map
- The splitting and Extend maps extend naturally.
- There is a Type D blocked-off pattern.
- Modify Path and Modify Reorder to avoid the new block-off pattern.

Recent Work

- The bijections for types B_n and D_n given here are actually crystal isomorphisms. In progress with undergraduate student.
- Efficient combinatorial computation for energy function for types B and D.
- Explicit computation of so-called *non-Dual-demazure* arrows in types B and D.

What is next?

- Explicit computation of *non-Dual-demazure* arrows in types A, C.
- Explicit embedding of the more generation rectangular shape into tensor product of column shape KR crystals.

Thank you!