Algebraic combinatorics, representation theory, and Sage

Anne Schilling, UC Davis

ICERM, Providence
January 28, 2013
Topics

- **Representation theory:** crystal bases, Lie theory, Hecke algebras, root systems, Coxeter groups, ...
- **Algebraic combinatorics:** symmetric functions, (nonsymmetric) Macdonald polynomials, Demazure characters, tableaux, ...
- **Schubert calculus:** Schubert polynomials, Kazhdan–Lusztig polynomials, Demazure operators, ...
Topics

- **Representation theory**: crystal bases, Lie theory, Hecke algebras, root systems, Coxeter groups, ...

- **Algebraic combinatorics**: symmetric functions, (nonsymmetric) Macdonald polynomials, Demazure characters, tableaux, ...

- **Schubert calculus**: Schubert polynomials, Kazhdan–Lusztig polynomials, Demazure operators, ...
Topics

- **Representation theory**: crystal bases, Lie theory, Hecke algebras, root systems, Coxeter groups, ...

- **Algebraic combinatorics**: symmetric functions, (nonsymmetric) Macdonald polynomials, Demazure characters, tableaux, ...

- **Schubert calculus**: Schubert polynomials, Kazhdan–Lusztig polynomials, Demazure operators, ...
Crystals

\[ B(\Lambda_1) \quad B(\Lambda_1 + \Lambda_2) \]
A \( U_q(g) \)-crystal is a nonempty set \( B \) with maps

\[
\begin{align*}
\text{wt} & : B \to P \\
e_i, f_i & : B \to B \cup \{\emptyset\} \quad \text{for all } i \in I
\end{align*}
\]

satisfying

\[
\begin{align*}
f_i(b) = b' & \iff e_i(b') = b & \text{if } b, b' \in B \\
\text{wt}(f_i(b)) & = \text{wt}(b) - \alpha_i & \text{if } f_i(b) \in B \\
\langle h_i, \text{wt}(b) \rangle & = \varphi_i(b) - \varepsilon_i(b)
\end{align*}
\]

Write \( \bullet \overline{i} \bullet \) for \( b' = f_i(b) \)
Tensor products of crystals

$B(\Lambda_1) \otimes B(\Lambda_1 + \Lambda_2)$
**Tensor products**

**Definition**

$B, B'$ crystals

$B \otimes B'$ is $B \times B'$ as sets with

\[
\text{wt}(b \otimes b') = \text{wt}(b) + \text{wt}(b')
\]

\[
f_i(b \otimes b') = \begin{cases} 
  f_i(b) \otimes b' & \text{if } \varepsilon_i(b) \geq \varphi_i(b') \\
  b \otimes f_i(b') & \text{otherwise}
\end{cases}
\]
Tensor products

Definition

\( B, B' \) crystals

\( B \otimes B' \) is \( B \times B' \) as sets with

\[
\text{wt}(b \otimes b') = \text{wt}(b) + \text{wt}(b')
\]

\[
f_i(b \otimes b') = \begin{cases} 
  f_i(b) \otimes b' & \text{if } \varepsilon_i(b) \geq \varphi_i(b') \\
  b \otimes f_i(b') & \text{otherwise}
\end{cases}
\]
Tensor products

Definition

$B, B'$ crystals

$B \otimes B'$ is $B \times B'$ as sets with

$$\text{wt}(b \otimes b') = \text{wt}(b) + \text{wt}(b')$$

$$f_i(b \otimes b') = \begin{cases} f_i(b) \otimes b' & \text{if } \varepsilon_i(b) \geq \varphi_i(b') \\ b \otimes f_i(b') & \text{otherwise} \end{cases}$$
Tensor products

**Definition**

$B, B'$ crystals

$B \otimes B'$ is $B \times B'$ as sets with

\[ \text{wt}(b \otimes b') = \text{wt}(b) + \text{wt}(b') \]

\[ f_i(b \otimes b') = \begin{cases} f_i(b) \otimes b' & \text{if } \varepsilon_i(b) \geq \varphi_i(b') \\ b \otimes f_i(b') & \text{otherwise} \end{cases} \]

\[
\begin{array}{ccc}
\varphi_i(b) & \varepsilon_i(b) & \varphi_i(b') & \varepsilon_i(b') \\
\hline
- - - & +++ & - - & +++
\end{array}
\]
Littlewood-Richardson rule in terms of crystals

\[ V(\lambda) \otimes V(\mu) \cong \bigoplus_{\nu} c_{\lambda\mu}^\nu V(\nu) \]

\( c_{\lambda\mu}^\nu = \text{LR coefficient} \)

**Theorem (Kashiwara-Nakashima)**

\( c_{\lambda\mu}^\nu \) is the number of highest weight vectors in \( B(\lambda) \otimes B(\mu) \) of weight \( \nu \).

Crystals are now on lmfdb.org. Try to explore them there!

Thematic Tutorial
Littlewood-Richardson rule in terms of crystals

\[ V(\lambda) \otimes V(\mu) \cong \bigoplus_{\nu} c^{\nu}_{\lambda \mu} V(\nu) \]

\( c^{\nu}_{\lambda \mu} = \text{LR coefficient} \)

**Theorem (Kashiwara-Nakashima)**

\( c^{\nu}_{\lambda \mu} \) is the number of highest weight vectors in \( B(\lambda) \otimes B(\mu) \) of weight \( \nu \).

Crystals are now on lmfdb.org. Try to explore them there!

**Thematic Tutorial**
Littlewood-Richardson rule in terms of crystals

\[ V(\lambda) \otimes V(\mu) \cong \bigoplus_{\nu} c_{\lambda\mu}^{\nu} V(\nu) \]

\( c_{\lambda\mu}^{\nu} \) = LR coefficient

**Theorem (Kashiwara-Nakashima)**

\( c_{\lambda\mu}^{\nu} \) is the number of highest weight vectors in \( B(\lambda) \otimes B(\mu) \) of weight \( \nu \).

Crystals are now on lmfdb.org. Try to explore them there!

Thematic Tutorial
Littlewood-Richardson rule in terms of crystals

\[ V(\lambda) \otimes V(\mu) \cong \bigoplus_{\nu} c_{\lambda\mu}^{\nu} V(\nu) \]

\[ c_{\lambda\mu}^{\nu} = \text{LR coefficient} \]

**Theorem (Kashiwara-Nakashima)**

\[ c_{\lambda\mu}^{\nu} \text{ is the number of highest weight vectors in } B(\lambda) \otimes B(\mu) \text{ of weight } \nu. \]

Crystals are now on lmfdb.org. Try to explore them there!

Thematic Tutorial
Macdonald polynomials

$s_\lambda[X]$ Schur functions (basis for ring of symmetric functions)
$p_\lambda[X]$ power sum symmetric functions

Macdonald symmetric functions $H_\lambda[X; q, t]$ unique basis for ring of symmetric functions over $\mathbb{Q}((q, t))$ s.t.:

1. $H_\lambda[X; q, t] = \sum_{\mu \geq \lambda} r_{\lambda \mu}(q, t) s_\mu[X/(1 - q)];$

2. $\langle H_\lambda[X; q, t], H_\mu[X; q, t] \rangle/qt = 0$ if $\lambda \neq \mu$

where scalar product $\langle \cdot , \cdot \rangle/qt$ is defined as

$$\langle p_\lambda, p_\mu \rangle/qt = z_\lambda \delta_{\lambda \mu} \prod_i (1 - q^{\lambda_i})(1 - t^{\lambda_i});$$

3. $\langle H_\lambda[X; q, t], s_{(n)}[X] \rangle = t^n(\lambda).$

Disadvantage: This is a very implicit definition!
Macdonald polynomials

\[ s_\lambda [X] \] Schur functions (basis for ring of symmetric functions)
\[ p_\lambda [X] \] power sum symmetric functions

Macdonald symmetric functions \( H_\lambda [X; q, t] \) unique basis for ring of symmetric functions over \( \mathbb{Q}((q, t)) \) s.t.:

1. \( H_\lambda [X; q, t] = \sum_{\mu \geq \lambda} r_{\lambda \mu} (q, t) s_\mu [X/(1 - q)] \);

2. \( \langle H_\lambda [X; q, t], H_\mu [X; q, t] \rangle_{qt} = 0 \) if \( \lambda \neq \mu \)

where scalar product \( \langle \cdot , \cdot \rangle_{qt} \) is defined as

\[
\langle p_\lambda, p_\mu \rangle_{qt} = z_\lambda \delta_{\lambda \mu} \prod_i (1 - q^{\lambda_i})(1 - t^{\lambda_i});
\]

3. \( \langle H_\lambda [X; q, t], s_{(n)} [X] \rangle = t^{n(\lambda)} \).

Disadvantage: This is a very implicit definition!
Macdonald polynomials

\( s_\lambda[X] \) Schur functions (basis for ring of symmetric functions)
\( p_\lambda[X] \) power sum symmetric functions

Macdonald symmetric functions \( H_\lambda[X; q, t] \) unique basis for ring of symmetric functions over \( \mathbb{Q}((q, t)) \) s.t.:

1. \( H_\lambda[X; q, t] = \sum_{\mu \geq \lambda} r_{\lambda \mu}(q, t) s_\mu[X/(1-q)] \);

2. \( \langle H_\lambda[X; q, t], H_\mu[X; q, t] \rangle_{qt} = 0 \) if \( \lambda \neq \mu \)

where scalar product \( \langle \cdot, \cdot \rangle_{qt} \) is defined as

\[
\langle p_\lambda, p_\mu \rangle_{qt} = z_\lambda \delta_{\lambda \mu} \prod_i (1 - q^{\lambda_i})(1 - t^{\lambda_i});
\]

3. \( \langle H_\lambda[X; q, t], s^{(n)}[X] \rangle = t^n(\lambda) \).

Disadvantage: This is a very implicit definition!
Macdonald polynomials

\(s_\lambda[X]\) Schur functions (basis for ring of symmetric functions)
\(p_\lambda[X]\) power sum symmetric functions

Macdonald symmetric functions \(H_\lambda[X; q, t]\) unique basis for ring of symmetric functions over \(\mathbb{Q}((q, t))\) s.t.:

1. \(H_\lambda[X; q, t] = \sum_{\mu \geq \lambda} r_{\lambda \mu}(q, t) s_\mu[X/(1 - q)]\);

2. \(\langle H_\lambda[X; q, t], H_\mu[X; q, t] \rangle_{qt} = 0 \text{ if } \lambda \neq \mu\)

where scalar product \(\langle \cdot, \cdot \rangle_{qt}\) is defined as

\(\langle p_\lambda, p_\mu \rangle_{qt} = z_\lambda \delta_{\lambda \mu} \prod_i (1 - q^{\lambda_i})(1 - t^{\lambda_i})\);

3. \(\langle H_\lambda[X; q, t], s_{(n)}[X] \rangle = t^n(\lambda)\).

Disadvantage: This is a very implicit definition!
Macdonald polynomials

$s_\lambda[X]$ Schur functions (basis for ring of symmetric functions)

$p_\lambda[X]$ power sum symmetric functions

Macdonald symmetric functions $H_\lambda[X; q, t]$ unique basis for ring of symmetric functions over $\mathbb{Q}((q, t))$ s.t.:

1. $H_\lambda[X; q, t] = \sum_{\mu \geq \lambda} r_{\lambda \mu}(q, t) s_\mu[X/(1-q)];$

2. $\langle H_\lambda[X; q, t], H_\mu[X; q, t] \rangle_{qt} = 0$ if $\lambda \neq \mu$

where scalar product $\langle \cdot, \cdot \rangle_{qt}$ is defined as

$$\langle p_\lambda, p_\mu \rangle_{qt} = z_\lambda \delta_{\lambda \mu} \prod_i (1 - q^{\lambda_i})(1 - t^{\lambda_i});$$

3. $\langle H_\lambda[X; q, t], s_{(n)}[X] \rangle = t^n(\lambda).$

Disadvantage: This is a very implicit definition!
Macdonald polynomials

\( s_{\lambda}[X] \) Schur functions (basis for ring of symmetric functions)

\( p_{\lambda}[X] \) power sum symmetric functions

Macdonald symmetric functions \( H_{\lambda}[X; q, t] \) unique basis for ring of symmetric functions over \( \mathbb{Q}((q, t)) \) s.t.:

1. \( H_{\lambda}[X; q, t] = \sum_{\mu \geq \lambda} r_{\lambda\mu}(q, t) s_{\mu}[X/(1 - q)] \);

2. \( \langle H_{\lambda}[X; q, t], H_{\mu}[X; q, t] \rangle_{qt} = 0 \) if \( \lambda \neq \mu \)

where scalar product \( \langle \cdot, \cdot \rangle_{qt} \) is defined as

\[
\langle p_{\lambda}, p_{\mu} \rangle_{qt} = z_{\lambda} \delta_{\lambda\mu} \prod_{i} (1 - q^{\lambda_i})(1 - t^{\lambda_i});
\]

3. \( \langle H_{\lambda}[X; q, t], s_{(n)}[X] \rangle = t^{n(\lambda)}. \)

Disadvantage: This is a very implicit definition!
Macdonald polynomials and crystals

From the Ram-Yip formula in terms of crystals

Theorem (Lenart)

In types $A$ and $C$, we have

$$H_\mu(x; q, 0) = \sum_{b \in B_1 \otimes B_2 \otimes \ldots} q^{\text{charge}(b)} x^{\text{weight}(b)}$$

Using quantum alcove paths

Theorem (Lenart, Naito, Sagaki, S, Shimozono)

For general untwisted types

$$H_\mu(x; q, 0) = \sum_{\Pi \in A(\mu)} q^{\text{level}(\Pi)} x^{\text{weight}(\Pi)}$$
Macdonald polynomials and crystals

From the Ram-Yip formula in terms of crystals

**Theorem (Lenart)**

*In types $A$ and $C$, we have*

$$H_\mu(x; q, 0) = \sum_{b \in B^\mu_{1,1} \otimes B^\mu_{2,1} \otimes ...} q^{charge(b)} x^{weight(b)}$$

**Using quantum alcove paths**

**Theorem (Lenart, Naito, Sagaki, S, Shimozono)**

*For general untwisted types*

$$H_\mu(x; q, 0) = \sum_{\Pi \in A(\mu)} q^{level(\Pi)} x^{weight(\Pi)}$$
Demazure crystals

Littelmann conjectured/ Kashiwara proved that there is a subset $B_w(\Lambda)$ (Demazure crystal) of $B(\Lambda)$ s.t.

$$\sum_{b \in B_w(\Lambda)} b = D_{i_1} \cdots D_{i_\ell} u_\Lambda$$

where

1. $i_1 \ldots i_\ell$ is a reduced word of $w$

2. 

$$D_i b = \begin{cases} \sum_{0 \leq k \leq \langle h_i, \text{wt}(b) \rangle} f_i^k b & \text{if } \langle h_i, \text{wt}(b) \rangle \geq 0 \\ -\sum_{1 \leq k < -\langle h_i, \text{wt}(b) \rangle} e_i^k b & \text{if } \langle h_i, \text{wt}(b) \rangle < 0. \end{cases}$$

Corollary

$\chi(B_w(\Lambda))$ is the Demazure character.
Demazure crystals

Littelmann conjectured/ Kashiwara proved that there is a subset $B_w(\Lambda)$ (Demazure crystal) of $B(\Lambda)$ s.t.

$$\sum_{b\in B_w(\Lambda)} b = D_{i_1} \cdots D_{i_\ell} u_\Lambda$$

where

1. $i_1 \ldots i_\ell$ is a reduced word of $w$
2. $D_i b = \begin{cases} \sum_{0 \leq k \leq \langle h_i, \text{wt}(b) \rangle} f_i^k b & \text{if } \langle h_i, \text{wt}(b) \rangle \geq 0 \\ -\sum_{1 \leq k < \langle h_i, \text{wt}(b) \rangle} e_i^k b & \text{if } \langle h_i, \text{wt}(b) \rangle < 0. \end{cases}$

Corollary

$\chi(B_w(\Lambda))$ is the Demazure character.
Demazure crystals

$B_{s_2s_1}(\omega_1 + \omega_2)$ Demazure crystal

vertices given by $\{f_2^a f_1^b (u \omega_1 + \omega_2) \mid a, b \geq 0\}$
Kyoto path model, affine Demazure and KR crystals

Theorem (Kyoto path model)

*Highest weight infinite-dimensional crystal in terms of semi-infinite tensor product*

\[ B(s\Lambda_0) \cong \cdots \otimes B^{r,s} \otimes B^{r,s} \otimes B(s\Lambda_i) \]

Theorem (Fourier, S, Shimozono; S, Tingley)

*Affine Demazure crystal*

\[ D(\omega_r, s) \cong B^{r,s} \otimes u_s\Lambda_0 \]
Kyoto path model, affine Demazure and KR crystals

**Theorem (Kyoto path model)**

*Highest weight infinite-dimensional crystal in terms of semi-infinite tensor product*

\[ B(\Lambda_0) \cong \cdots \otimes B^{r,s} \otimes B^{r',s} \otimes B(\Lambda_i) \]

**Theorem (Fourier, S, Shimozono; S, Tingley)**

*Affine Demazure crystal*

\[ D(\omega_r, s) \cong B^{r,s} \otimes u_{s\Lambda_0} \]
Affine Demazure crystals

→ Demazure 0-arrows

→ non-Demazure 0-arrows (not all given)
Macdonald polynomials and Demazure characters

Theorem (Ion)

For simply-laced or twisted root system the (nonsymmetric) Macdonald polynomials at $t = 0$ equal Demazure characters:

$$H_\lambda[X; q, 0] = \chi(B_{w_\lambda}(\Lambda_0))$$

To do! Generalize to untwisted types.
Theorem (Ion)

For simply-laced or twisted root system the (nonsymmetric) Macdonald polynomials at $t = 0$ equal Demazure characters:

$$H_\lambda[X; q, 0] = \chi(B_{w_\lambda}(\Lambda_0))$$

To do! Generalize to untwisted types.
Sage’s mission: Creating a viable free open source alternative to Magma™, Maple™, Mathematica™, and Matlab™.

- free
- open source
- anyone can contribute!

Design: Sage is build around Python (general-purpose programming language)
Sage’s mission: Creating a viable free open source alternative to Magma™, Maple™, Mathematica™, and Matlab™.

- free
- open source
- anyone can contribute!

Design: Sage is build around Python (general-purpose programming language)
Sage’s mission: Creating a viable free open source alternative to Magma™, Maple™, Mathematica™, and Matlab™.

- free
- open source
- anyone can contribute!

Design: Sage is build around Python (general-purpose programming language)
Sage’s mission: Creating a viable free open source alternative to Magma™, Maple™, Mathematica™, and Matlab™.

- free
- open source
- anyone can contribute!

Design: Sage is build around Python (general-purpose programming language)
Sage

Sage’s mission: Creating a viable free open source alternative to Magma™, Maple™, Mathematica™, and Matlab™.

- free
- open source
- anyone can contribute!

Design: Sage is build around Python (general-purpose programming language)
Sage

History:
- Sage project began by William Stein in 2005
  SAGE=“Software for Arithmetic Geometry Experimentation”
- Quickly expanded beyond number theory; attracted more users, developers, funding
- sagenb.org now has over 90,000 accounts

Sage-combinat: “To improve the open source mathematical system Sage as an extensible toolbox for computer exploration in (algebraic) combinatorics, and foster code sharing between researchers in this area.”

Try it!
- Use Sage online http://sagenb.org
- Install Sage http://sagemath.org
Sage

History:

- Sage project began by William Stein in 2005
  SAGE=“Software for Arithmetic Geometry Experimentation”
- Quickly expanded beyond number theory; attracted more users, developers, funding
  - sagenb.org now has over 90,000 accounts

Sage-combinat: “To improve the open source mathematical system Sage as an extensible toolbox for computer exploration in (algebraic) combinatorics, and foster code sharing between researchers in this area.”

Try it!

- Use Sage online http://sagenb.org
- Install Sage http://sagemath.org
Sage

History:

- Sage project began by William Stein in 2005
  SAGE=“Software for Arithmetic Geometry Experimentation”
- Quickly expanded beyond number theory; attracted more users, developers, funding
- sagenb.org now has over 90,000 accounts

Sage-combinat: “To improve the open source mathematical system Sage as an extensible toolbox for computer exploration in (algebraic) combinatorics, and foster code sharing between researchers in this area.”

Try it!

- Use Sage online http://sagenb.org
- Install Sage http://sagemath.org
Sage

History:

- Sage project began by William Stein in 2005
  SAGE=“Software for Arithmetic Geometry Experimentation”
- Quickly expanded beyond number theory; attracted more users, developers, funding
- sagenb.org now has over 90,000 accounts

Sage-combinat: “To improve the open source mathematical system Sage as an extensible toolbox for computer exploration in (algebraic) combinatorics, and foster code sharing between researchers in this area.”

Try it!

- Use Sage online http://sagenb.org
- Install Sage http://sagemath.org
Sage

History:

- Sage project began by William Stein in 2005
  SAGE=“Software for Arithmetic Geometry Experimentation”
- Quickly expanded beyond number theory; attracted more users, developers, funding
- sagenb.org now has over 90,000 accounts

Sage-combinat: “To improve the open source mathematical system Sage as an extensible toolbox for computer exploration in (algebraic) combinatorics, and foster code sharing between researchers in this area.”

Try it!

- Use Sage online http://sagenb.org
- Install Sage http://sagemath.org