Hecke group algebras as degenerate affine Hecke algebras

Florent Hivert[1] Anne Schilling[2] Nicolas M. Thiéry[2,3]

[1]LITIS/LIFAR, Université Rouen, France

[3]University of California at Davis, USA

[3]Laboratoire de Mathématiques d'Orsay, Université Paris Sud, France

AMS Meeting, Claremont, May 4th, 2008

arXiv:0711.1561v1 [math.RT]
arXiv:0804.3781v1 [math.RT]

Goals

- Advertise Hecke group algebras:
  - Nice structure and representation theory
  - Connections with NCSF, parking functions, ...
  - Connections with 0-Hecke and affine Hecke algebras
- Where does this structure come from?
- Is it useful?

Coxeter groups

Definition (Coxeter group W)

Generators : $(s_i)_{i \in S}$ (simple reflections)
Relations: $s_i^2 = 1$ and $\underbrace{s_i s_j \cdots}_{m_{i,j}} = \underbrace{s_j s_i \cdots}_{m_{i,j}}$, for $i \neq j$

Group algebra: $\mathbb{C}[W]$

Example (Type A_n: symmetric group $\mathfrak{S}_{n+1}$)

Generators: $(s_i)_{i=1,\dots,n}$: elementary transpositions
Relations:

$$
\begin{aligned}
s_i^2 &= 1 && \text{for all } 1 \leq i \leq n,\\
s_i s_j &= s_j s_i && \text{for all } |i-j| > 1,\\
s_i s_{i+1} s_i &= s_{i+1} s_i s_{i+1} && \text{for all } 1 \leq i \leq n-1.
\end{aligned}
$$

Coxeter groups

Definition (Coxeter group W)

Generators : $(s_i)_{i\in S}$ (simple reflections)

Relations: $s_i^2 = 1$ and $\underbrace{s_i s_j \cdots}_{m_{i,j}} = \underbrace{s_j s_i \cdots}_{m_{i,j}}$, for $i \neq j$

Group algebra: $\mathbb{C}[W]$

Example (Type A_n: symmetric group $\mathfrak{S}_{n+1}$)

Generators: $(s_i)_{i=1,\dots,n}$: elementary transpositions

Relations:

$$\begin{aligned} s_i^2 &= 1 && \text{for all } 1 \leq i \leq n, \\ s_i s_j &= s_j s_i && \text{for all } |i-j| > 1, \\ s_i s_{i+1} s_i &= s_{i+1} s_i s_{i+1} && \text{for all } 1 \leq i \leq n-1. \end{aligned}$$

0-(Iwahori)-Hecke algebras

Definition (0-Hecke algebra H(W)(0))

Generators : $(\pi_i)_{i\in S}$

Relations: $\pi_i^2 = \pi_i$ and $\underbrace{\pi_i\pi_j\cdots}_{m_{i,j}} = \underbrace{\pi_j\pi_i\cdots}_{m_{i,j}}$ for $i \neq j$

Basis: $(\pi_w)_{w\in W}$

Example (Type A_n)

Generators: $(\pi_i)_{i=1,\dots,n}$:

Relations:

$$\pi_i^2 = \pi_i \qquad \text{for all } 1 \leq i \leq n,$$

$$\pi_i\pi_j = \pi_j\pi_i \qquad \text{for all } |i-j| > 1,$$

$$\pi_i\pi_{i+1}\pi_i = \pi_{i+1}\pi_i\pi_{i+1} \qquad \text{for all } 1 \leq i \leq n-1.$$

0-(Iwahori)-Hecke algebras

Definition (0-Hecke algebra $H(W)(0)$)

Generators : $(\pi_i)_{i\in S}$

Relations: $\pi_i^2 = \pi_i$ and $\underbrace{\pi_i\pi_j\cdots}_{m_{i,j}} = \underbrace{\pi_j\pi_i\cdots}_{m_{i,j}}$ for $i \neq j$

Basis: $(\pi_w)_{w\in W}$

Example (Type A_n)

Generators: $(\pi_i)_{i=1,\dots,n}$:

Relations:

$$\begin{aligned}
\pi_i^2 &= \pi_i && \text{for all } 1 \le i \le n,\\
\pi_i\pi_j &= \pi_j\pi_i && \text{for all } |i-j| > 1,\\
\pi_i\pi_{i+1}\pi_i &= \pi_{i+1}\pi_i\pi_{i+1} && \text{for all } 1 \le i \le n-1.
\end{aligned}$$

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$\begin{tikzcd} 13854627 \arrow[d, shift right=3, "s_2"'] \arrow[d, shift left=3, "s_6"] \\ 18354267 \arrow[d, shift right=3, "s_2"'] \arrow[d, shift left=3, "s_6"] \\ 13854627 \end{tikzcd} \qquad \begin{tikzcd} 13854627 \arrow[d, shift right=3, "\pi_2"'] \arrow[d, shift left=3, "\pi_6"] \\ 18354627 \arrow[d, shift right=3, "\pi_2"'] \arrow[d, shift left=3, "\pi_6"] \\ 18354627 \end{tikzcd} \qquad \begin{tikzcd} 13854627 \arrow[d, shift right=3, "\overline{\pi}_2"'] \arrow[d, shift left=3, "\overline{\pi}_6"] \\ 13854267 \arrow[d, shift right=3, "\overline{\pi}_2"'] \arrow[d, shift left=3, "\overline{\pi}_6"] \\ 13854267 \end{tikzcd}$$

Example (Bubble sort)

13854627

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$
\begin{tikzcd}
13854627 \arrow[d, "s_2"', "s_6"] \\
18354267 \arrow[d, "s_2"', "s_6"] \\
13854627
\end{tikzcd}
\qquad
\begin{tikzcd}
13854627 \arrow[d, "\pi_2"', "\pi_6"] \\
18354627 \arrow[d, "\pi_2"', "\pi_6"] \\
18354627
\end{tikzcd}
\qquad
\begin{tikzcd}
13854627 \arrow[d, "\overline{\pi}_2"', "\overline{\pi}_6"] \\
13854267 \arrow[d, "\overline{\pi}_2"', "\overline{\pi}_6"] \\
13854267
\end{tikzcd}
$$

Example (Bubble sort)

13854627

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$\begin{tikzcd} 13854627 \arrow[d, shift right=3, "s_2"'] \arrow[d, shift left=3, "s_6"] \\ 18354267 \arrow[d, shift right=3, "s_2"'] \arrow[d, shift left=3, "s_6"] \\ 13854627 \end{tikzcd}
\qquad
\begin{tikzcd} 13854627 \arrow[d, shift right=3, "\pi_2"'] \arrow[d, shift left=3, "\pi_6"] \\ 18354627 \arrow[d, shift right=3, "\pi_2"'] \arrow[d, shift left=3, "\pi_6"] \\ 18354627 \end{tikzcd}
\qquad
\begin{tikzcd} 13854627 \arrow[d, shift right=3, "\overline{\pi}_2"'] \arrow[d, shift left=3, "\overline{\pi}_6"] \\ 13854267 \arrow[d, shift right=3, "\overline{\pi}_2"'] \arrow[d, shift left=3, "\overline{\pi}_6"] \\ 13854267 \end{tikzcd}$$

Example (Bubble sort)

18354627

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$13854627 \xrightarrow{s_2,\ s_6} 18354267 \xrightarrow{s_2,\ s_6} 13854627$$

$$13854627 \xrightarrow{\pi_2,\ \pi_6} 18354627 \xrightarrow{\pi_2,\ \pi_6} 18354627$$

$$13854627 \xrightarrow{\overline{\pi}_2,\ \overline{\pi}_6} 13854267 \xrightarrow{\overline{\pi}_2,\ \overline{\pi}_6} 13854267$$

Example (Bubble sort)

81354627

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$\begin{array}{ccc}
13854627 & 13854627 & 13854627 \\
s_2 \big\downarrow \quad \big\downarrow s_6 & \pi_2 \big\downarrow \quad \big\downarrow \pi_6 & \overline{\pi}_2 \big\downarrow \quad \big\downarrow \overline{\pi}_6 \\
18354267 & 18354627 & 13854267 \\
s_2 \big\downarrow \quad \big\downarrow s_6 & \pi_2 \big\downarrow \quad \big\downarrow \pi_6 & \overline{\pi}_2 \big\downarrow \quad \big\downarrow \overline{\pi}_6 \\
13854627 & 18354627 & 13854267
\end{array}$$

Example (Bubble sort)

81354627

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$\begin{array}{ccc}
13854627 & 13854627 & 13854627 \\
s_2 \downarrow \quad \downarrow s_6 & \pi_2 \downarrow \quad \downarrow \pi_6 & \overline{\pi}_2 \downarrow \quad \downarrow \overline{\pi}_6 \\
18354267 & 18354627 & 13854267 \\
s_2 \downarrow \quad \downarrow s_6 & \pi_2 \downarrow \quad \downarrow \pi_6 & \overline{\pi}_2 \downarrow \quad \downarrow \overline{\pi}_6 \\
13854627 & 18354627 & 13854267
\end{array}$$

Example (Bubble sort)

81354672

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$13854627 \xrightarrow{s_2,\ s_6} 18354267 \xrightarrow{s_2,\ s_6} 13854627$$

$$13854627 \xrightarrow{\pi_2,\ \pi_6} 18354627 \xrightarrow{\pi_2,\ \pi_6} 18354627$$

$$13854627 \xrightarrow{\overline{\pi}_2,\ \overline{\pi}_6} 13854267 \xrightarrow{\overline{\pi}_2,\ \overline{\pi}_6} 13854267$$

Example (Bubble sort)

81354762

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$
\begin{array}{ccc}
13854627 & 13854627 & 13854627 \\
\downarrow s_2 \quad \downarrow s_6 & \downarrow \pi_2 \quad \downarrow \pi_6 & \downarrow \overline{\pi}_2 \quad \downarrow \overline{\pi}_6 \\
18354267 & 18354627 & 13854267 \\
\downarrow s_2 \quad \downarrow s_6 & \downarrow \pi_2 \quad \downarrow \pi_6 & \downarrow \overline{\pi}_2 \quad \downarrow \overline{\pi}_6 \\
13854627 & 18354627 & 13854267
\end{array}
$$

Example (Bubble sort)

81357462

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$\begin{array}{ccc}
13854627 & 13854627 & 13854627 \\
s_2 \downarrow \quad \downarrow s_6 & \pi_2 \downarrow \quad \downarrow \pi_6 & \overline{\pi}_2 \downarrow \quad \downarrow \overline{\pi}_6 \\
18354267 & 18354627 & 13854267 \\
s_2 \downarrow \quad \downarrow s_6 & \pi_2 \downarrow \quad \downarrow \pi_6 & \overline{\pi}_2 \downarrow \quad \downarrow \overline{\pi}_6 \\
13854627 & 18354627 & 13854267
\end{array}$$

Example (Bubble sort)

81375462

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$\begin{array}{ccc}
13854627 & 13854627 & 13854627 \\
s_2\downarrow \quad \downarrow s_6 & \pi_2\downarrow \quad \downarrow \pi_6 & \overline{\pi}_2\downarrow \quad \downarrow \overline{\pi}_6 \\
18354267 & 18354627 & 13854267 \\
s_2\downarrow \quad \downarrow s_6 & \pi_2\downarrow \quad \downarrow \pi_6 & \overline{\pi}_2\downarrow \quad \downarrow \overline{\pi}_6 \\
13854627 & 18354627 & 13854267
\end{array}$$

Example (Bubble sort)

81735462

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$\begin{tikzcd} 13854627 \arrow[d, "s_2"', "s_6"] \\ 18354267 \arrow[d, "s_2"', "s_6"] \\ 13854627 \end{tikzcd} \qquad \begin{tikzcd} 13854627 \arrow[d, "\pi_2"', "\pi_6"] \\ 18354627 \arrow[d, "\pi_2"', "\pi_6"] \\ 18354627 \end{tikzcd} \qquad \begin{tikzcd} 13854627 \arrow[d, "\overline{\pi}_2"', "\overline{\pi}_6"] \\ 13854267 \arrow[d, "\overline{\pi}_2"', "\overline{\pi}_6"] \\ 13854267 \end{tikzcd}$$

Example (Bubble sort)

87135462

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

13854627	13854627	13854627
$s_2 \downarrow \quad \downarrow s_6$	$\pi_2 \downarrow \quad \downarrow \pi_6$	$\overline{\pi}_2 \downarrow \quad \downarrow \overline{\pi}_6$
18354267	18354627	13854267
$s_2 \downarrow \quad \downarrow s_6$	$\pi_2 \downarrow \quad \downarrow \pi_6$	$\overline{\pi}_2 \downarrow \quad \downarrow \overline{\pi}_6$
13854627	18354627	13854267

Example (Bubble sort)

87135462

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$
\begin{tikzcd}
13854627 \arrow[d, "s_2\quad s_6"] & 13854627 \arrow[d, "\pi_2\quad \pi_6"] & 13854627 \arrow[d, "\overline{\pi}_2\quad \overline{\pi}_6"] \\
18354267 \arrow[d, "s_2\quad s_6"] & 18354627 \arrow[d, "\pi_2\quad \pi_6"] & 13854267 \arrow[d, "\overline{\pi}_2\quad \overline{\pi}_6"] \\
13854627 & 18354627 & 13854267
\end{tikzcd}
$$

Example (Bubble sort)

87135642

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$
\begin{array}{ccc}
13854627 & 13854627 & 13854627 \\
s_2 \downarrow \quad \downarrow s_6 & \pi_2 \downarrow \quad \downarrow \pi_6 & \overline{\pi}_2 \downarrow \quad \downarrow \overline{\pi}_6 \\
18354267 & 18354627 & 13854267 \\
s_2 \downarrow \quad \downarrow s_6 & \pi_2 \downarrow \quad \downarrow \pi_6 & \overline{\pi}_2 \downarrow \quad \downarrow \overline{\pi}_6 \\
13854627 & 18354627 & 13854267
\end{array}
$$

Example (Bubble sort)

87136542

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$\begin{tikzcd}
13854627 \arrow[d, "s_2"', "s_6"] & 13854627 \arrow[d, "\pi_2"', "\pi_6"] & 13854627 \arrow[d, "\overline{\pi}_2"', "\overline{\pi}_6"] \\
18354267 \arrow[d, "s_2"', "s_6"] & 18354627 \arrow[d, "\pi_2"', "\pi_6"] & 13854267 \arrow[d, "\overline{\pi}_2"', "\overline{\pi}_6"] \\
13854627 & 18354627 & 13854267
\end{tikzcd}$$

Example (Bubble sort)

87163542

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$\begin{array}{ccc}
13854627 & 13854627 & 13854627 \\
s_2 \downarrow \quad \downarrow s_6 & \pi_2 \downarrow \quad \downarrow \pi_6 & \overline{\pi}_2 \downarrow \quad \downarrow \overline{\pi}_6 \\
18354267 & 18354627 & 13854267 \\
s_2 \downarrow \quad \downarrow s_6 & \pi_2 \downarrow \quad \downarrow \pi_6 & \overline{\pi}_2 \downarrow \quad \downarrow \overline{\pi}_6 \\
13854627 & 18354627 & 13854267
\end{array}$$

Example (Bubble sort)

87613542

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$
\begin{tikzcd}
13854627 \arrow[d, "s_2"', "s_6"] \\
18354267 \arrow[d, "s_2"', "s_6"] \\
13854627
\end{tikzcd}
\qquad
\begin{tikzcd}
13854627 \arrow[d, "\pi_2"', "\pi_6"] \\
18354627 \arrow[d, "\pi_2"', "\pi_6"] \\
18354627
\end{tikzcd}
\qquad
\begin{tikzcd}
13854627 \arrow[d, "\overline{\pi}_2"', "\overline{\pi}_6"] \\
13854267 \arrow[d, "\overline{\pi}_2"', "\overline{\pi}_6"] \\
13854267
\end{tikzcd}
$$

Example (Bubble sort)

87613542

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$\begin{tikzcd} 13854627 \arrow[d, "s_2"', "s_6"] \\ 18354267 \arrow[d, "s_2"', "s_6"] \\ 13854627 \end{tikzcd} \qquad \begin{tikzcd} 13854627 \arrow[d, "\pi_2"', "\pi_6"] \\ 18354627 \arrow[d, "\pi_2"', "\pi_6"] \\ 18354627 \end{tikzcd} \qquad \begin{tikzcd} 13854627 \arrow[d, "\overline{\pi}_2"', "\overline{\pi}_6"] \\ 13854267 \arrow[d, "\overline{\pi}_2"', "\overline{\pi}_6"] \\ 13854267 \end{tikzcd}$$

Example (Bubble sort)

87615342

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$
\begin{tikzcd}
13854627 \arrow[d, "s_2\ s_6"] \\
18354267 \arrow[d, "s_2\ s_6"] \\
13854627
\end{tikzcd}
\qquad
\begin{tikzcd}
13854627 \arrow[d, "\pi_2\ \pi_6"] \\
18354627 \arrow[d, "\pi_2\ \pi_6"] \\
18354627
\end{tikzcd}
\qquad
\begin{tikzcd}
13854627 \arrow[d, "\overline{\pi}_2\ \overline{\pi}_6"] \\
13854267 \arrow[d, "\overline{\pi}_2\ \overline{\pi}_6"] \\
13854267
\end{tikzcd}
$$

Example (Bubble sort)

87651342

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$13854627 \xrightarrow{s_2,\ s_6} 18354267 \xrightarrow{s_2,\ s_6} 13854627$$

$$13854627 \xrightarrow{\pi_2,\ \pi_6} 18354627 \xrightarrow{\pi_2,\ \pi_6} 18354627$$

$$13854627 \xrightarrow{\overline{\pi}_2,\ \overline{\pi}_6} 13854267 \xrightarrow{\overline{\pi}_2,\ \overline{\pi}_6} 13854267$$

Example (Bubble sort)

87651342

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$13854627 \xrightarrow{s_2,\ s_6} 18354267 \xrightarrow{s_2,\ s_6} 13854627$$

$$13854627 \xrightarrow{\pi_2,\ \pi_6} 18354627 \xrightarrow{\pi_2,\ \pi_6} 18354627$$

$$13854627 \xrightarrow{\overline{\pi}_2,\ \overline{\pi}_6} 13854267 \xrightarrow{\overline{\pi}_2,\ \overline{\pi}_6} 13854267$$

Example (Bubble sort)

87651432

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$\begin{array}{ccc}
13854627 & 13854627 & 13854627 \\
s_2 \downarrow \quad \downarrow s_6 & \pi_2 \downarrow \quad \downarrow \pi_6 & \overline{\pi}_2 \downarrow \quad \downarrow \overline{\pi}_6 \\
18354267 & 18354627 & 13854267 \\
s_2 \downarrow \quad \downarrow s_6 & \pi_2 \downarrow \quad \downarrow \pi_6 & \overline{\pi}_2 \downarrow \quad \downarrow \overline{\pi}_6 \\
13854627 & 18354627 & 13854267
\end{array}$$

Example (Bubble sort)

87654132

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$13854627 \xrightarrow{s_2,\ s_6} 18354267 \xrightarrow{s_2,\ s_6} 13854627$$

$$13854627 \xrightarrow{\pi_2,\ \pi_6} 18354627 \xrightarrow{\pi_2,\ \pi_6} 18354627$$

$$13854627 \xrightarrow{\overline{\pi}_2,\ \overline{\pi}_6} 13854267 \xrightarrow{\overline{\pi}_2,\ \overline{\pi}_6} 13854267$$

Example (Bubble sort)

87654132

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$\begin{array}{ccc}
13854627 & 13854627 & 13854627 \\
s_2 \downarrow \quad \downarrow s_6 & \pi_2 \downarrow \quad \downarrow \pi_6 & \overline{\pi}_2 \downarrow \quad \downarrow \overline{\pi}_6 \\
18354267 & 18354627 & 13854267 \\
s_2 \downarrow \quad \downarrow s_6 & \pi_2 \downarrow \quad \downarrow \pi_6 & \overline{\pi}_2 \downarrow \quad \downarrow \overline{\pi}_6 \\
13854627 & 18354627 & 13854267
\end{array}$$

Example (Bubble sort)

87654312

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$\begin{tikzcd}
13854627 \arrow[d, "s_2\quad s_6"] & 13854627 \arrow[d, "\pi_2\quad \pi_6"] & 13854627 \arrow[d, "\overline{\pi}_2\quad \overline{\pi}_6"] \\
18354267 \arrow[d, "s_2\quad s_6"] & 18354627 \arrow[d, "\pi_2\quad \pi_6"] & 13854267 \arrow[d, "\overline{\pi}_2\quad \overline{\pi}_6"] \\
13854627 & 18354627 & 13854267
\end{tikzcd}$$

Example (Bubble sort)

87654312

0-Hecke algebras and bubble sort

Example (Right regular representation for type A)

$$
\begin{tikzcd}
13854627 \arrow[d, "s_2"', "s_6"] \\
18354267 \arrow[d, "s_2"', "s_6"] \\
13854627
\end{tikzcd}
\qquad
\begin{tikzcd}
13854627 \arrow[d, "\pi_2"', "\pi_6"] \\
18354627 \arrow[d, "\pi_2"', "\pi_6"] \\
18354627
\end{tikzcd}
\qquad
\begin{tikzcd}
13854627 \arrow[d, "\overline{\pi}_2"', "\overline{\pi}_6"] \\
13854267 \arrow[d, "\overline{\pi}_2"', "\overline{\pi}_6"] \\
13854267
\end{tikzcd}
$$

Example (Bubble sort)

87654321

Hecke algebras

Take q_1 and q_2 parameters, and set $q := -\frac{q_1}{q_2}$.

Definition (Hecke algebra $\mathrm{H}(W)(q_1, q_2)$)

Generators : $(T_i)_{i \in S}$ Relations: $(T_i - q_1)(T_i - q_2) = 0$ and

$\underbrace{T_i T_j \cdots}_{m_{i,j}} = \underbrace{T_j T_i \cdots}_{m_{i,j}}$, for $i \neq j$

Basis: $(T_w)_{w \in W}$

- At $q = 1$: group algebra $\mathbb{C}[W]$
- At $q = 0$: 0-Hecke algebra $\mathrm{H}(W)(0)$
- At q not 0 nor a root of unity: isomorphic to $\mathbb{C}[W]$

Realization of T_i as operator in $\mathrm{End}(\mathbb{C}W)$:

$$T_i := (q_1 + q_2)\pi_i - q_1 s_i$$

Hecke algebras

Take q_1 and q_2 parameters, and set $q := -\frac{q_1}{q_2}$.

Definition (Hecke algebra $\mathrm{H}(W)(q_1, q_2)$)

Generators : $(T_i)_{i\in S}$ Relations: $(T_i - q_1)(T_i - q_2) = 0$ and $\underbrace{T_i T_j \cdots}_{m_{i,j}} = \underbrace{T_j T_i \cdots}_{m_{i,j}}$, for $i \neq j$

Basis: $(T_w)_{w\in W}$

- At $q = 1$: group algebra $\mathbb{C}[W]$
- At $q = 0$: 0-Hecke algebra $\mathrm{H}(W)(0)$
- At q not 0 nor a root of unity: isomorphic to $\mathbb{C}[W]$

Realization of T_i as operator in $\mathrm{End}(\mathbb{C}W)$:

$$T_i := (q_1 + q_2)\pi_i - q_1 s_i$$

Hecke algebras

Take q_1 and q_2 parameters, and set $q := -\frac{q_1}{q_2}$.

Definition (Hecke algebra $\mathrm{H}(W)(q_1, q_2)$)

Generators : $(T_i)_{i\in S}$ Relations: $(T_i - q_1)(T_i - q_2) = 0$ and $\underbrace{T_i T_j \cdots}_{m_{i,j}} = \underbrace{T_j T_i \cdots}_{m_{i,j}}$, for $i \neq j$

Basis: $(T_w)_{w\in W}$

- At $q = 1$: group algebra $\mathbb{C}[W]$
- At $q = 0$: 0-Hecke algebra $\mathrm{H}(W)(0)$
- At q not 0 nor a root of unity: isomorphic to $\mathbb{C}[W]$

Realization of T_i as operator in $\mathrm{End}(\mathbb{C}W)$:

$$T_i := (q_1 + q_2)\pi_i - q_1 s_i$$

Hecke group algebras

A silly idea during a brainstorm (Thibon, Novelli, H., T., 2003)

Definition (Hecke group algebra HW of a Coxeter group W)

Glue $\mathbb{C}[W]$ and $\mathrm{H}(W)(0)$ on their right regular representations:

$$HW := \langle (\pi_i, s_i)_{i\in S}\rangle \quad \subset \quad \mathrm{End}(\mathbb{C}W)$$

- Any interesting structure?
- Contains all Hecke algebras by construction
- Type A: dimension and dimension of the radical in the Sloane!

Hecke group algebras

A silly idea during a brainstorm (Thibon, Novelli, H., T., 2003)

Definition (Hecke group algebra HW of a Coxeter group W)

Glue $\mathbb{C}[W]$ and H$(W)(0)$ on their right regular representations:

$$HW := \langle (\pi_i, s_i)_{i \in S} \rangle \quad \subset \quad \mathrm{End}(\mathbb{C}W)$$

- Any interesting structure?
- Contains all Hecke algebras by construction
- Type A: dimension and dimension of the radical in the Sloane!

Hecke group algebras

A silly idea during a brainstorm (Thibon, Novelli, H., T., 2003)

Definition (Hecke group algebra HW of a Coxeter group W)

Glue $\mathbb{C}[W]$ and H(W)(0) on their right regular representations:

$$HW := \langle (\pi_i, s_i)_{i \in S} \rangle \quad \subset \quad \mathrm{End}(\mathbb{C}W)$$

- Any interesting structure?
- Contains all Hecke algebras by construction
- Type A: dimension and dimension of the radical in the Sloane!

Hecke group algebras

A silly idea during a brainstorm (Thibon, Novelli, H., T., 2003)

Definition (Hecke group algebra HW of a Coxeter group W)

Glue $\mathbb{C}[W]$ and H(W)(0) on their right regular representations:

$$HW := \langle(\pi_i, s_i)_{i \in S}\rangle \quad \subset \quad \mathrm{End}(\mathbb{C}W)$$

- Any interesting structure?
- Contains all Hecke algebras by construction
- Type A: dimension and dimension of the radical in the Sloane!

The Hecke group algebra of rank 1

$$W := \{1, s\} \qquad \mathbb{C}W := \mathbb{C}.1 \oplus \mathbb{C}.s$$

Basis

$$\left\{ \quad \mathrm{id} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad s = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \pi = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} \quad \right\}$$

Relations

$$s\pi = \pi, \qquad \pi s = \overline{\pi}, \qquad \overline{\pi} + \pi = 1 + s$$

Dimension 1 simple and projective module

$$(1-s).\mathrm{id} = (1-s), \qquad (1-s).s = -(1-s), \qquad (1-s).\pi = 0$$

The Hecke group algebra of rank 1

$$W := \{1, s\} \qquad \mathbb{C}W := \mathbb{C}.1 \oplus \mathbb{C}.s$$

Basis

$$\left\{ \quad \mathrm{id} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad s = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \pi = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} \quad \right\}$$

Relations

$$s\pi = \pi, \qquad \pi s = \overline{\pi}, \qquad \overline{\pi} + \pi = 1 + s$$

Dimension 1 simple and projective module

$$(1-s).\mathrm{id} = (1-s), \qquad (1-s).s = -(1-s), \qquad (1-s).\pi = 0$$

The Hecke group algebra of rank 1

$$W := \{1, s\} \qquad \mathbb{C}W := \mathbb{C}.1 \oplus \mathbb{C}.s$$

Basis

$$\left\{ \quad \mathrm{id} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad s = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \pi = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} \quad \right\}$$

Relations

$$s\pi = \pi, \qquad \pi s = \overline{\pi}, \qquad \overline{\pi} + \pi = 1 + s$$

Dimension 1 simple and projective module

$$(1-s).\mathrm{id} = (1-s), \qquad (1-s).s = -(1-s), \qquad (1-s).\pi = 0$$

Structure theory of Hecke group algebras (I)

Definition

Left / right descent sets:

$$\mathrm{D}_R(w) := \{i \in S \mid w = w's_i\} \qquad \mathrm{D}_L(w) := \{i \in S \mid w = s_i w'\}$$

$v \in \mathbb{C}W$ *i-left antisymmetric* if $s_i v = -v$

Theorem (H.,T., 2005)

Basis of $\mathrm{H}W$*:* $\{w\pi_{w'} \mid \mathrm{D}_R(w) \cap \mathrm{D}_L(w') = \emptyset\}$

$\mathrm{H}W$ *algebra of left antisymmetry preserving operators*

$\mathrm{H}W^*$ *algebra of left symmetry preserving operators*

- Proof: simultaneous triangularity of basis and linear relations
- Straightening relations? Presentation?

Structure theory of Hecke group algebras (I)

Definition

Left / right descent sets:

$$\mathrm{D}_R(w) := \{i \in S \mid w = w' s_i\} \qquad \mathrm{D}_L(w) := \{i \in S \mid w = s_i w'\}$$

$v \in \mathbb{C}W$ *i-left antisymmetric* if $s_i v = -v$

Theorem (H.,T., 2005)

Basis of $\mathrm{H}W$*:* $\{w\pi_{w'} \mid \mathrm{D}_R(w) \cap \mathrm{D}_L(w') = \emptyset\}$

$\mathrm{H}W$ *algebra of left antisymmetry preserving operators*

$\mathrm{H}W^*$ *algebra of left symmetry preserving operators*

- Proof: simultaneous triangularity of basis and linear relations
- Straightening relations? Presentation?

Structure theory of Hecke group algebras (I)

Definition

Left / right descent sets:

$$\mathrm{D}_R(w) := \{i \in S \mid w = w' s_i\} \qquad \mathrm{D}_L(w) := \{i \in S \mid w = s_i w'\}$$

$v \in \mathbb{C}W$ *i-left antisymmetric* if $s_i v = -v$

Theorem (H.,T., 2005)

Basis of $\mathrm{H}W$*:* $\{w\pi_{w'} \mid \mathrm{D}_R(w) \cap \mathrm{D}_L(w') = \emptyset\}$
$\mathrm{H}W$ *algebra of left antisymmetry preserving operators*
$\mathrm{H}W^*$ *algebra of left symmetry preserving operators*

- Proof: simultaneous triangularity of basis and linear relations
- Straightening relations? Presentation?

Structure theory of Hecke group algebras (I)

Definition

Left / right descent sets:

$$\mathrm{D}_R(w) := \{i \in S \mid w = w's_i\} \qquad \mathrm{D}_L(w) := \{i \in S \mid w = s_iw'\}$$

$v \in \mathbb{C}W$ *i-left antisymmetric* if $s_iv = -v$

Theorem (H.,T., 2005)

Basis of $\mathrm{H}W$*:* $\{w\pi_{w'} \mid \mathrm{D}_R(w) \cap \mathrm{D}_L(w') = \emptyset\}$

$\mathrm{H}W$ *algebra of left antisymmetry preserving operators*

$\mathrm{H}W^*$ *algebra of left symmetry preserving operators*

- Proof: simultaneous triangularity of basis and linear relations
- Straightening relations? Presentation?

Structure theory of Hecke group algebras (II)

- Modules P_I for $I \subset S$:

$$P_I := \{v \in \mathbb{C}W \mid s_i v = -s_i, \forall i \in I\}$$

- Boolean lattice: $I \subset J \Longrightarrow P_J \subset P_I$
- Combinatorics of descent classes: $\dim P_I = |{}^I_S W|$
- *Good basis* of $\mathbb{C}W$: compatible with restriction on each P_I:
  - Par ex: $\left\{ v_w := \sum_{w' \in W_{S \setminus D_L(w)}} (-1)^{l(w')} w' w \quad \middle| \quad w \in W \right\}$
  - Or the Kazhdan-Lusztig basis to be fancy

Proposition

Realization of the Hecke group algebra as digraph algebra:
Basis: $\{e_{w,w'} \mid D_L(w) \subset D_L(w')\}$, *where* $e_{w,w'}(v_{w''})\delta_{w'',w} v_{w'}$

Structure theory of Hecke group algebras (II)

- Modules P_I for $I \subset S$:

$$P_I := \{v \in \mathbb{C}W \mid s_i v = -s_i, \forall i \in I\}$$

- Boolean lattice: $I \subset J \Longrightarrow P_J \subset P_I$
- Combinatorics of descent classes: $\dim P_I = |{}^I_S W|$
- *Good basis* of $\mathbb{C}W$: compatible with restriction on each P_I:
  - Par ex: $\left\{ v_w := \sum_{w' \in W_{S \setminus D_L(w)}} (-1)^{l(w')} w'w \mid w \in W \right\}$
  - Or the Kazhdan-Lusztig basis to be fancy

Proposition

Realization of the Hecke group algebra as digraph algebra:
Basis: $\{e_{w,w'} \mid D_L(w) \subset D_L(w')\}$, *where* $e_{w,w'}(v_{w''})\delta_{w'',w}v_{w'}$

Structure theory of Hecke group algebras (II)

- Modules P_I for $I \subset S$:

$$P_I := \{v \in \mathbb{C}W \mid s_i v = -s_i, \forall i \in I\}$$

- Boolean lattice: $I \subset J \Longrightarrow P_J \subset P_I$
- Combinatorics of descent classes: $\dim P_I = |{}^I_S W|$
- *Good basis* of $\mathbb{C}W$: compatible with restriction on each P_I:
  - Par ex: $\left\{ v_w := \sum_{w' \in W_{S \setminus D_L(w)}} (-1)^{l(w')} w' w \quad \middle| \quad w \in W \right\}$
  - Or the Kazhdan-Lusztig basis to be fancy

Proposition

Realization of the Hecke group algebra as digraph algebra:
Basis: $\{e_{w,w'} \mid D_L(w) \subset D_L(w')\}$, *where* $e_{w,w'}(v_{w''})\delta_{w'',w} v_{w'}$

Structure theory of Hecke group algebras (II)

- Modules P_I for $I \subset S$:

$$P_I := \{v \in \mathbb{C}W \mid s_i v = -s_i, \forall i \in I\}$$

- Boolean lattice: $I \subset J \Longrightarrow P_J \subset P_I$
- Combinatorics of descent classes: $\dim P_I = |{}^I_S W|$
- *Good basis* of $\mathbb{C}W$: compatible with restriction on each P_I:
  - Par ex: $\left\{ v_w := \sum_{w' \in W_{S \setminus \mathrm{D}_L(w)}} (-1)^{l(w')} w'w \quad \middle| \quad w \in W \right\}$
  - Or the Kazhdan-Lusztig basis to be fancy

Proposition

Realization of the Hecke group algebra as digraph algebra:
Basis: $\{e_{w,w'} \mid \mathrm{D}_L(w) \subset \mathrm{D}_L(w')\}$, *where* $e_{w,w'}(v_{w''})\delta_{w'',w}v_{w'}$

Structure theory of Hecke group algebras (II)

- Modules P_I for $I \subset S$:

$$P_I := \{v \in \mathbb{C}W \mid s_i v = -s_i, \forall i \in I\}$$

- Boolean lattice: $I \subset J \Longrightarrow P_J \subset P_I$
- Combinatorics of descent classes: $\dim P_I = |{}^I_S W|$
- *Good basis* of $\mathbb{C}W$: compatible with restriction on each P_I:
    - Par ex: $\left\{ v_w := \sum_{w' \in W_{S \setminus \mathrm{D}_L(w)}} (-1)^{l(w')} w' w \quad \mid \quad w \in W \right\}$
    - Or the Kazhdan-Lusztig basis to be fancy

Proposition

Realization of the Hecke group algebra as digraph algebra:
Basis: $\{e_{w,w'} \mid \mathrm{D}_L(w) \subset \mathrm{D}_L(w')\}$, *where* $e_{w,w'}(v_{w''}) \delta_{w'',w} v_{w'}$

Representation theory of Hecke group algebras

Theorem (H.,T., 2005)

- $e_{w,w}$: *max. decomposition of* id *into orthogonal idempotents*
- HW *Morita equivalent to the poset algebra of boolean lattice*
- *Projective modules:* P_I
- *Simple modules:* $S_I := P_I \,/\, \sum_{J \supset I} P_J$

  Left-antisymmetries on I*, left-symmetries on the complement*
  By restriction:
  - *Exactly the Young's ribbon representation of* W
  - *Exactly the projective modules of* $\mathrm{H}(W)(0)$

Question

Is there a link with the affine Hecke algebra?

Representation theory of Hecke group algebras

Theorem (H.,T., 2005)

- $e_{w,w}$: *max. decomposition of* id *into orthogonal idempotents*
- HW *Morita equivalent to the poset algebra of boolean lattice*
- *Projective modules:* P_I
- *Simple modules:* $S_I := P_I \;/\; \sum_{J \supset I} P_J$

  Left-antisymmetries on I, *left-symmetries on the complement*
  By restriction:
  - *Exactly the Young's ribbon representation of* W
  - *Exactly the projective modules of* $\mathrm{H}(W)(0)$

Question

Is there a link with the affine Hecke algebra?

Representation theory of Hecke group algebras

Theorem (H.,T., 2005)

- $e_{w,w}$: *max. decomposition of* id *into orthogonal idempotents*
- $\mathrm{H}W$ *Morita equivalent to the poset algebra of boolean lattice*
- *Projective modules:* P_I
- *Simple modules:* $S_I := P_I \;/\; \sum_{J \supset I} P_J$

  Left-antisymmetries on I, *left-symmetries on the complement*
  By restriction:
  - *Exactly the Young's ribbon representation of* W
  - *Exactly the projective modules of* $\mathrm{H}(W)(0)$

Question

Is there a link with the affine Hecke algebra?

Representation theory of Hecke group algebras

Theorem (H.,T., 2005)

- $e_{w,w}$: *max. decomposition of* id *into orthogonal idempotents*
- $\mathrm{H}W$ *Morita equivalent to the poset algebra of boolean lattice*
- *Projective modules:* P_I
- *Simple modules:* $S_I := P_I \ / \ \sum_{J \supset I} P_J$

  Left-antisymmetries on I*, left-symmetries on the complement*
  By restriction:
  - *Exactly the Young's ribbon representation of* W
  - *Exactly the projective modules of* $\mathrm{H}(W)(0)$

Question

Is there a link with the affine Hecke algebra?

Representation theory of Hecke group algebras

Theorem (H.,T., 2005)

- $e_{w,w}$: *max. decomposition of* id *into orthogonal idempotents*
- $\mathrm{H}W$ *Morita equivalent to the poset algebra of boolean lattice*
- *Projective modules:* P_I
- *Simple modules:* $S_I := P_I \ / \ \sum_{J \supset I} P_J$

  Left-antisymmetries on I*, left-symmetries on the complement*
  By restriction:
  - *Exactly the Young's ribbon representation of* W
  - *Exactly the projective modules of* $\mathrm{H}(W)(0)$

Question

Is there a link with the affine Hecke algebra?

Representation theory of Hecke group algebras

Theorem (H.,T., 2005)

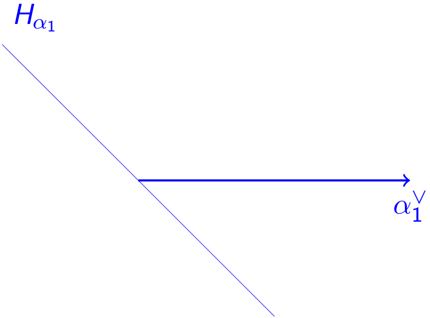
- $e_{w,w}$: *max. decomposition of* id *into orthogonal idempotents*
- HW *Morita equivalent to the poset algebra of boolean lattice*
- *Projective modules:* P_I
- *Simple modules:* $S_I := P_I \ / \ \sum_{J \supset I} P_J$

  Left-antisymmetries on I*, left-symmetries on the complement*
  By restriction:
  - *Exactly the Young's ribbon representation of* W
  - *Exactly the projective modules of* H$(W)(0)$

Question

Is there a link with the affine Hecke algebra?

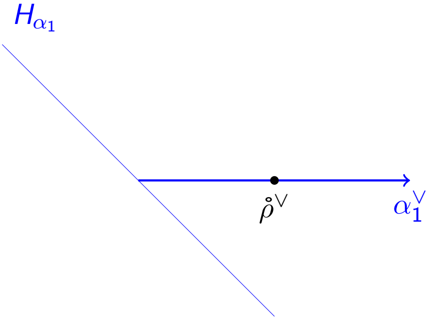
Level 0 action of the type A_1^1 affine Hecke algebra



Remark (At level 0)

- *W degenerates trivially to $\mathring{W}$ of type A_1; $W = \mathring{W} \ltimes T$*
- $\mathrm{H}(W)(0)$ *acts transitively on* $\mathring{W}$
- $\mathrm{H}(W)(0)$ *degenerates to* $\mathrm{H}\mathring{W}$, *not* $\mathrm{H}(\mathring{W})(0)$*!*

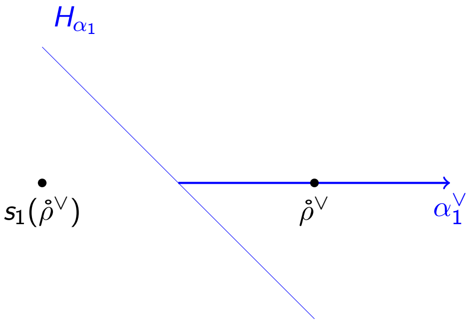
Level 0 action of the type A_1^1 affine Hecke algebra



Remark (At level 0)

- *W degenerates trivially to $\mathring{W}$ of type A_1; $W = \mathring{W} \ltimes T$*
- $\mathrm{H}(W)(0)$ *acts transitively on* $\mathring{W}$
- $\mathrm{H}(W)(0)$ *degenerates to* $\mathrm{H}\mathring{W}$*, not* $\mathrm{H}(\mathring{W})(0)$*!*

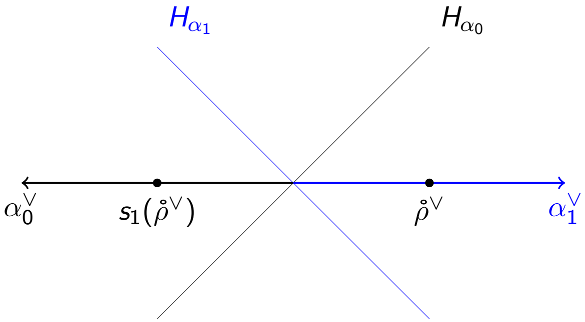
Level 0 action of the type A_1^1 affine Hecke algebra



Remark (At level 0)

- *W degenerates trivially to $\mathring{W}$ of type A_1; $W = \mathring{W} \ltimes T$*
- $\mathrm{H}(W)(0)$ *acts transitively on* $\mathring{W}$
- $\mathrm{H}(W)(0)$ *degenerates to* $\mathrm{H}\mathring{W}$, *not* $\mathrm{H}(\mathring{W})(0)$*!*

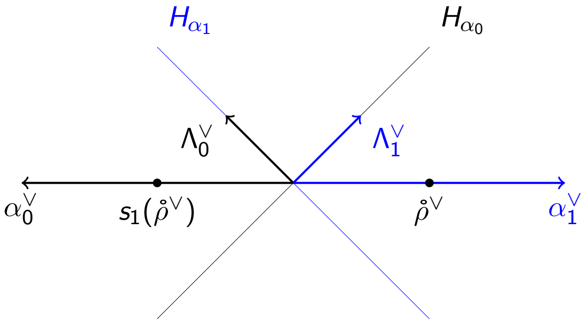
Level 0 action of the type A_1^1 affine Hecke algebra



Remark (At level 0)

- *W degenerates trivially to $\mathring{W}$ of type A_1; $W = \mathring{W} \ltimes T$*
- $\mathrm{H}(W)(0)$ *acts transitively on $\mathring{W}$*
- $\mathrm{H}(W)(0)$ *degenerates to* $\mathrm{H}\mathring{W}$, *not* $\mathrm{H}(\mathring{W})(0)$*!*

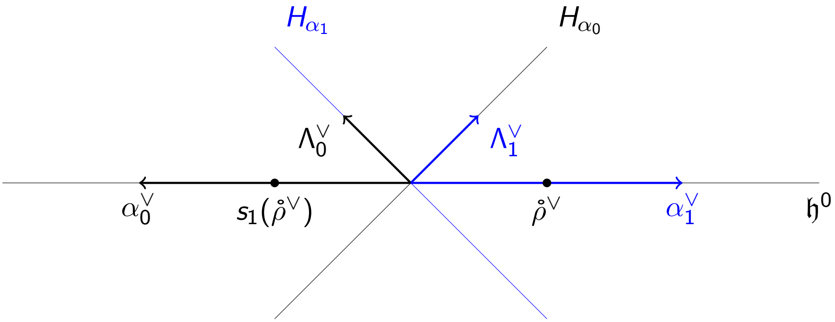
Level 0 action of the type A_1^1 affine Hecke algebra



Remark (At level 0)

- *W degenerates trivially to $\mathring{W}$ of type A_1; $W = \mathring{W} \ltimes T$*
- $\mathrm{H}(W)(0)$ *acts transitively on* $\mathring{W}$
- $\mathrm{H}(W)(0)$ *degenerates to* $\mathrm{H}\mathring{W}$*, not* $\mathrm{H}(\mathring{W})(0)$*!*

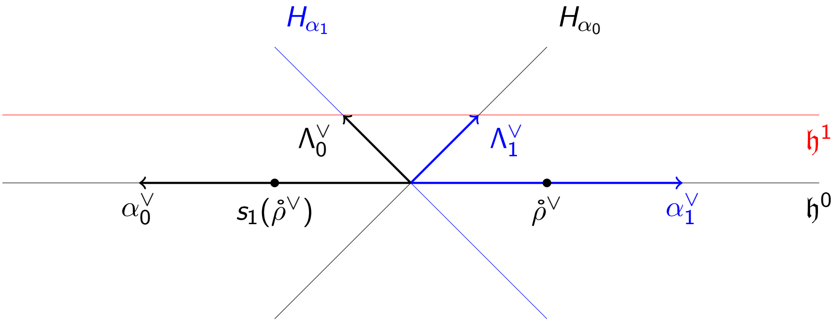
Level 0 action of the type A_1^1 affine Hecke algebra



Remark (At level 0)

- *W degenerates trivially to $\mathring{W}$ of type A_1; $W = \mathring{W} \ltimes T$*
- $\mathrm{H}(W)(0)$ *acts transitively on $\mathring{W}$*
- $\mathrm{H}(W)(0)$ *degenerates to* $\mathrm{H}\mathring{W}$*, not* $\mathrm{H}(\mathring{W})(0)$*!*

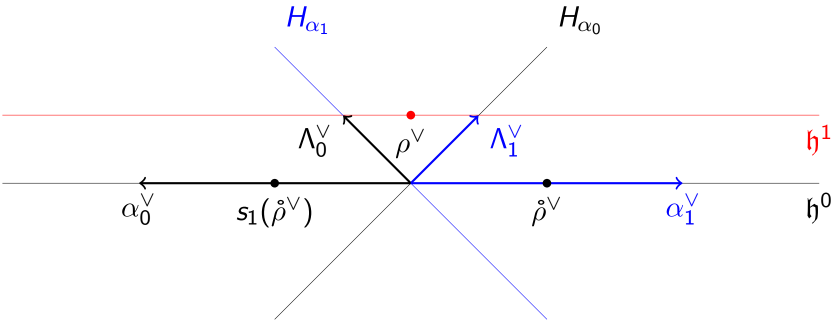
Level 0 action of the type A_1^1 affine Hecke algebra



Remark (At level 0)

- W *degenerates trivially to* $\mathring{W}$ *of type* A_1; $W = \mathring{W} \ltimes T$
- $\mathrm{H}(W)(0)$ *acts transitively on* $\mathring{W}$
- $\mathrm{H}(W)(0)$ *degenerates to* $\mathrm{H}\mathring{W}$, *not* $\mathrm{H}(\mathring{W})(0)$*!*

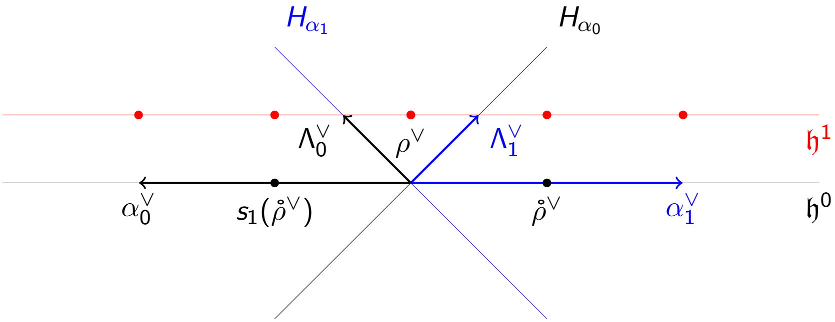
Level 0 action of the type A_1^1 affine Hecke algebra



Remark (At level 0)

- *W degenerates trivially to $\mathring{W}$ of type A_1; $W = \mathring{W} \ltimes T$*
- $\mathrm{H}(W)(0)$ *acts transitively on* $\mathring{W}$
- $\mathrm{H}(W)(0)$ *degenerates to* $\mathrm{H}\mathring{W}$*, not* $\mathrm{H}(\mathring{W})(0)$*!*

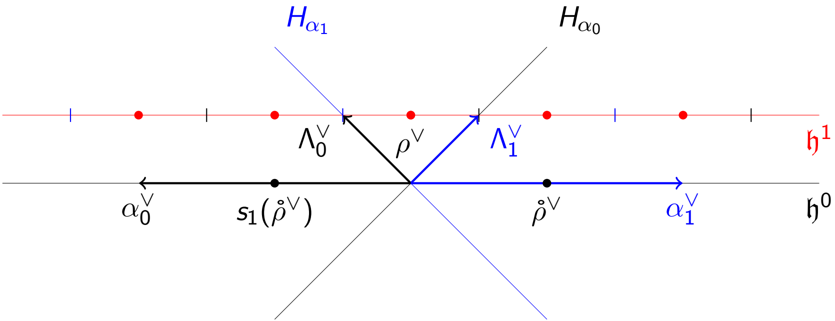
Level 0 action of the type A_1^1 affine Hecke algebra



Remark (At level 0)

- *W degenerates trivially to $\mathring{W}$ of type A_1; $W = \mathring{W} \ltimes T$*
- $\mathrm{H}(W)(0)$ *acts transitively on* $\mathring{W}$
- $\mathrm{H}(W)(0)$ *degenerates to* $\mathrm{H}\mathring{W}$, *not* $\mathrm{H}(\mathring{W})(0)$*!*

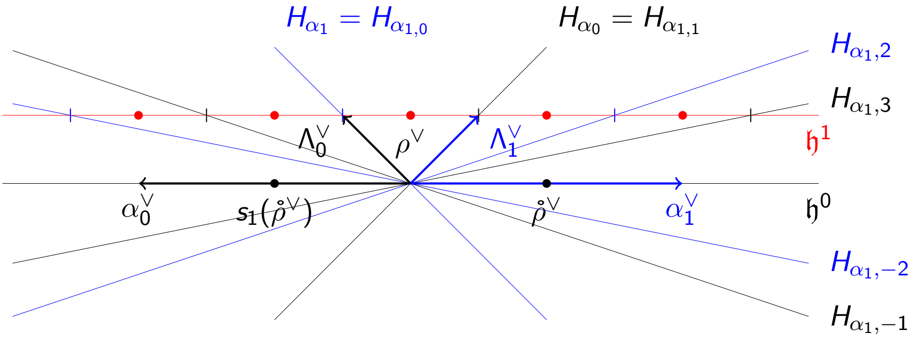
Level 0 action of the type A_1^1 affine Hecke algebra



Remark (At level 0)

- *W degenerates trivially to $\mathring{W}$ of type A_1; $W = \mathring{W} \ltimes T$*
- $\mathrm{H}(W)(0)$ *acts transitively on* $\mathring{W}$
- $\mathrm{H}(W)(0)$ *degenerates to* $\mathrm{H}\mathring{W}$, *not* $\mathrm{H}(\mathring{W})(0)$*!*

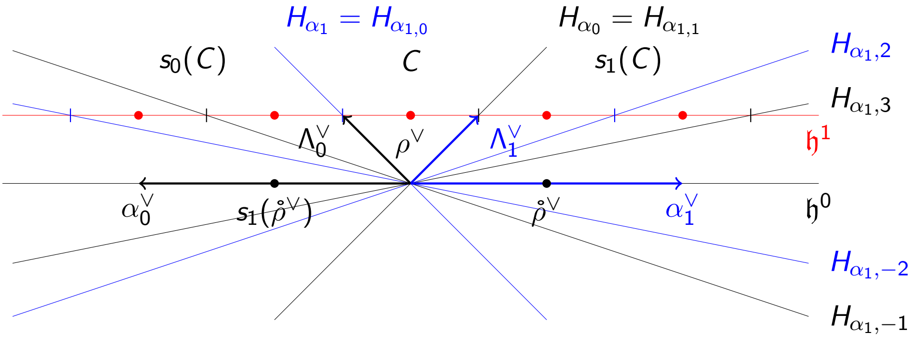
Level 0 action of the type A_1^1 affine Hecke algebra



Remark (At level 0)

- *W degenerates trivially to $\mathring{W}$ of type A_1; $W = \mathring{W} \ltimes T$*
- $\mathrm{H}(W)(0)$ *acts transitively on $\mathring{W}$*
- $\mathrm{H}(W)(0)$ *degenerates to* $\mathrm{H}\mathring{W}$*, not* $\mathrm{H}(\mathring{W})(0)$*!*

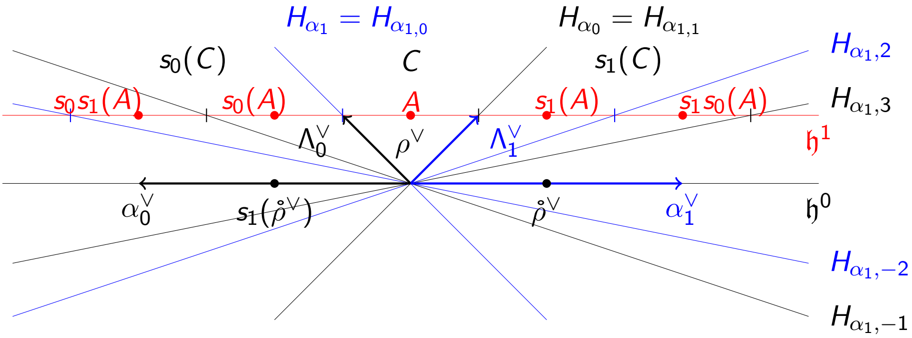
Level 0 action of the type A_1^1 affine Hecke algebra



Remark (At level 0)

- *W degenerates trivially to $\mathring{W}$ of type A_1; $W = \mathring{W} \ltimes T$*
- $\mathrm{H}(W)(0)$ *acts transitively on $\mathring{W}$*
- $\mathrm{H}(W)(0)$ *degenerates to* $\mathrm{H}\mathring{W}$*, not* $\mathrm{H}(\mathring{W})(0)$*!*

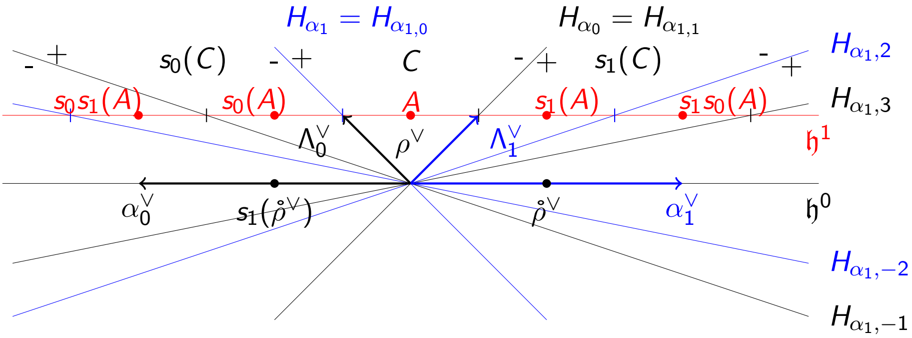
Level 0 action of the type A_1^1 affine Hecke algebra



Remark (At level 0)

- *W degenerates trivially to $\mathring{W}$ of type A_1; $W = \mathring{W} \ltimes T$*
- $\mathrm{H}(W)(0)$ *acts transitively on* $\mathring{W}$
- $\mathrm{H}(W)(0)$ *degenerates to* $\mathrm{H}\mathring{W}$, *not* $\mathrm{H}(\mathring{W})(0)$*!*

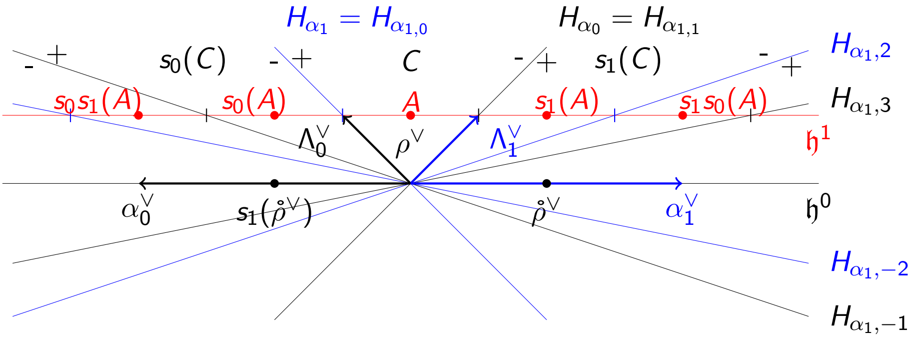
Level 0 action of the type A_1^1 affine Hecke algebra



Remark (At level 0)

- W *degenerates trivially to* $\mathring{W}$ *of type* A_1; $W = \mathring{W} \ltimes T$
- $\mathrm{H}(W)(0)$ *acts transitively on* $\mathring{W}$
- $\mathrm{H}(W)(0)$ *degenerates to* $\mathrm{H}\mathring{W}$, *not* $\mathrm{H}(\mathring{W})(0)$*!*

Level 0 action of the type A_1^1 affine Hecke algebra



Remark (At level 0)

- *W degenerates trivially to $\mathring{W}$ of type A_1; $W = \mathring{W} \ltimes T$*
- $\mathrm{H}(W)(0)$ *acts transitively on* $\mathring{W}$
- $\mathrm{H}(W)(0)$ *degenerates to* $\mathrm{H}\mathring{W}$, *not* $\mathrm{H}(\mathring{W})(0)$*!*

Level 0 action of affine Hecke algebras

Theorem (H.,S.,T. 2008)

W: affine Weyl group
$\mathring{W}$: finite Weyl group induced by the level 0 action
At level 0:

- *W degenerates trivially to $\mathring{W}$*
- *$\pi_0, \pi_1, \ldots, \pi_n$ act transitively on $\mathring{W}$*
- *$\mathrm{H}(W)(0)$ degenerates to $\mathrm{H}\mathring{W}$*
- *$\mathrm{H}(W)(q)$ degenerates to $\mathrm{H}\mathring{W}$, for q generic*

Level 0 action of affine Hecke algebras

Theorem (H.,S.,T. 2008)

W: affine Weyl group
$\mathring{W}$: finite Weyl group induced by the level 0 action
At level 0:

- *W degenerates trivially to $\mathring{W}$*
- *$\pi_0, \pi_1, \dots, \pi_n$ act transitively on $\mathring{W}$*
- *$\mathrm{H}(W)(0)$ degenerates to $\mathrm{H}\mathring{W}$*
- *$\mathrm{H}(W)(q)$ degenerates to $\mathrm{H}\mathring{W}$, for q generic*

Level 0 action of affine Hecke algebras

Theorem (H.,S.,T. 2008)

W: affine Weyl group
$\mathring{W}$: finite Weyl group induced by the level 0 action
At level 0:

- *W degenerates trivially to $\mathring{W}$*
- *$\pi_0, \pi_1, \dots, \pi_n$ act transitively on $\mathring{W}$*
- $\mathrm{H}(W)(0)$ *degenerates to* $\mathrm{H}\mathring{W}$
- $\mathrm{H}(W)(q)$ *degenerates to* $\mathrm{H}\mathring{W}$*, for q generic*

Level 0 action of affine Hecke algebras

Theorem (H.,S.,T. 2008)

W: affine Weyl group
$\mathring{W}$: finite Weyl group induced by the level 0 action
At level 0:

- *W degenerates trivially to $\mathring{W}$*
- *$\pi_0, \pi_1, \dots, \pi_n$ act transitively on $\mathring{W}$*
- *$\mathrm{H}(W)(0)$ degenerates to $\mathrm{H}\mathring{W}$*
- *$\mathrm{H}(W)(q)$ degenerates to $\mathrm{H}\mathring{W}$, for q generic*

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ *act transitively on permutations*

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

5

1 4

2 3

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ act transitively on permutations

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

1

5 4

2 3

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ *act transitively on permutations*

Recursive sorting algorithm in type A

$\pi_1, \dots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

1

2 4

5 3

Proposition

$\pi_0, \pi_1, \dots, \pi_n$ act transitively on permutations

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

1
2 4
3 5

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ act transitively on permutations

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

1

2 5

3 4

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ *act transitively on permutations*

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

5

2 1

3 4

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ act transitively on permutations

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

2

5 1

3 4

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ act transitively on permutations

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

2

3 1

5 4

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ act transitively on permutations

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

2

3 4

5 1

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ act transitively on permutations

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

2

3 4

1 5

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ act transitively on permutations

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

4

3 2

1 5

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ *act transitively on permutations*

Recursive sorting algorithm in type A

$\pi_1, \dots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

4

3 5

1 2

Proposition

$\pi_0, \pi_1, \dots, \pi_n$ act transitively on permutations

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

4

1 5

3 2

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ *act transitively on permutations*

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

1

4 5

3 2

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ *act transitively on permutations*

Recursive sorting algorithm in type A

$\pi_1, \dots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

5

4 1

3 2

Proposition

$\pi_0, \pi_1, \dots, \pi_n$ act transitively on permutations

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

4

5 1

3 2

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ *act transitively on permutations*

Recursive sorting algorithm in type A

$\pi_1, \dots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

4

3 1

5 2

Proposition

$\pi_0, \pi_1, \dots, \pi_n$ *act transitively on permutations*

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

4

3 1

2 5

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ act transitively on permutations

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

4

3 5

2 1

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ *act transitively on permutations*

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

5

3 4

2 1

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ act transitively on permutations

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

3

5 4

2 1

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ *act transitively on permutations*

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

3

2 4

5 1

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ act transitively on permutations

Recursive sorting algorithm in type A

$\pi_1, \dots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

3

2 4

1 5

Proposition

$\pi_0, \pi_1, \dots, \pi_n$ act transitively on permutations

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

3

2 5

1 4

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ act transitively on permutations

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

5

2 3

1 4

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ act transitively on permutations

Recursive sorting algorithm in type A

$\pi_1, \dots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

2

5 3

1 4

Proposition

$\pi_0, \pi_1, \dots, \pi_n$ act transitively on permutations

Recursive sorting algorithm in type A

$\pi_1, \dots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

2

1 3

5 4

Proposition

$\pi_0, \pi_1, \dots, \pi_n$ act transitively on permutations

Recursive sorting algorithm in type A

$\pi_1, \dots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

2

1 3

4 5

Proposition

$\pi_0, \pi_1, \dots, \pi_n$ act transitively on permutations

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

2

1 5

4 3

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ act transitively on permutations

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

5

1 2

4 3

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ act transitively on permutations

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

1

5 2

4 3

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ *act transitively on permutations*

Recursive sorting algorithm in type A

$\pi_1, \ldots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

1

5 2

4 3

Proposition

$\pi_0, \pi_1, \ldots, \pi_n$ act transitively on permutations

Recursive sorting algorithm in type A

$\pi_1, \dots, \pi_n$ can antisort 12345 to 54321 (bubble sort)
But not back! (going down the permutohedron)

Definition (Affine action)

Put the permutation on a circle
π_0 acts between last and first letter

1

5   2

4   3

Proposition

$\pi_0, \pi_1, \dots, \pi_n$ *act transitively on permutations*

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$\underline{1234}$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

  $\underline{1234}$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$21\underline{34}$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$21\underline{34}$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$2\underline{3}1\underline{4}$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$2\underline{34}1$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$2\underline{34}1$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$\underline{3}\,{\color{red}2}\,\underline{4}\,1$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$\underline{34}21$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$\underline{34}21$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

  4321

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

4321

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$432\underline{1}$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$43\underline{1}2$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$4\underline{1}32$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$\underline{1}432$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$\underline{1}432$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$\underline{1}43\underline{2}$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$\underline{1}4\underline{2}3$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$\underline{1}\,\underline{2}43$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$\underline{12}43$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$\underline{12}4\color{red}{\underline{3}}$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$\underline{123}4$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$\underline{12}\underline{3}4$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$21\underline{3}4$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$2\textcolor{red}{1}\underline{3}4$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$2\underline{3}14$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$2\underline{3}14$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$\underline{3}{\color{red}2}14$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$\underline{3}214$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$\underline{3}241$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$\underline{3}24\underline{1}$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$\underline{3}2\underline{1}4$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$\underline{3}\,\underline{1}24$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$\underline{31}24$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

1324

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

13**2**4

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

  1342

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$134\underline{2}$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$13\underline{2}4$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$1\underline{2}34$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$\underline{2}134$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$\underline{2}134$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$\underline{2}3\textcolor{red}{1}4$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$\underline{2}341$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$\underline{2}34\underline{1}$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$\underline{2}3\underline{1}4$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$\underline{2}\underline{1}34$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$\underline{21}34$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

$$1234$$

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

1234

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

  1234

- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Recursive sorting algorithms in other types

Proposition (S.,T.,2007)

- *Similar algorithms for types B, C, D*
- *Existence for all types (including twisted):*

Proof

- Type B: $1 < 2 < 3 < 4 < \underline{4} < \underline{3} < \underline{2} < \underline{1}$

  1234

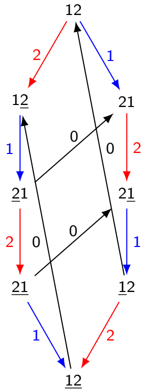
- Type-free induction strategy
- Case by case induction step
  Brute force on computer for the exceptional types (E_8!)

Type free geometric argument (I)

$$\begin{tikzcd}
& 12 \arrow[dl, "2"'] \arrow[dr, "1"] & \\
1\underline{2} \arrow[d, "1"'] & & 21 \arrow[d, "2"] \\
\underline{2}1 \arrow[d, "2"'] & & 2\underline{1} \arrow[d, "1"] \\
\underline{21} \arrow[dr, "1"'] & & \underline{1}2 \arrow[dl, "2"] \\
& \underline{12} &
\end{tikzcd}$$

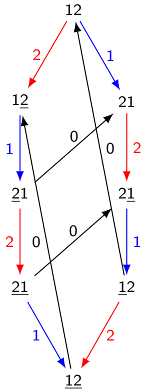
π_0, π_1, π_2 on C_2

Type free geometric argument (I)

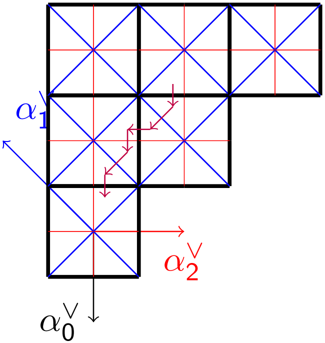


π_0, π_1, π_2 on C_2

Type free geometric argument (I)

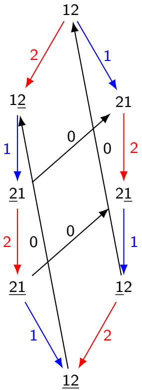


π_0, π_1, π_2 on C_2

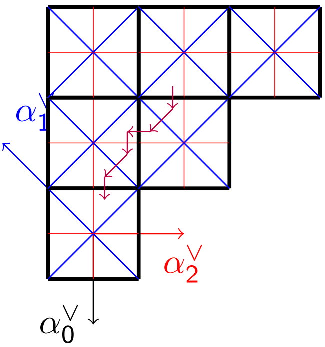


Alcove picture at level 1

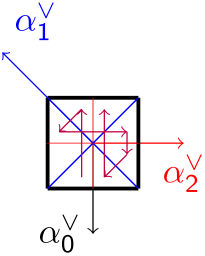
Type free geometric argument (I)



π_0, π_1, π_2 on C_2

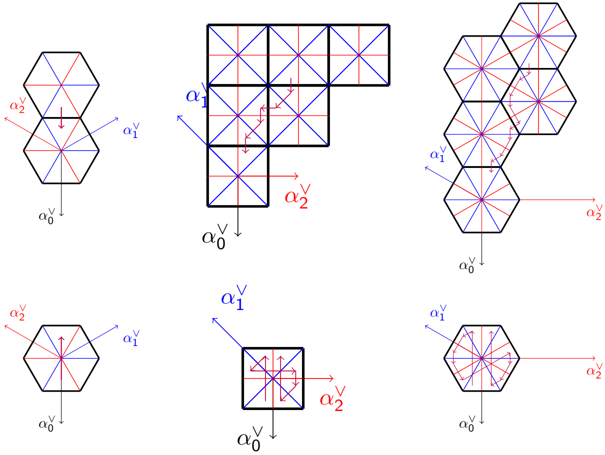


Alcove picture at level 1



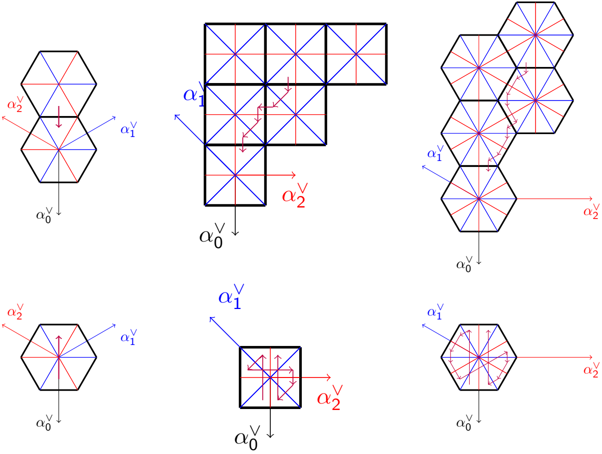
Quotient at level 0

Type free geometric argument (II)



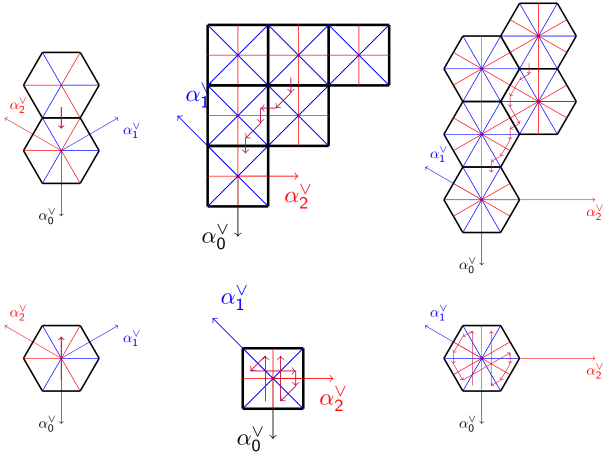
- Covers all rank 2 affine Weyl groups
- Generalization to I_n? Meaning of π_0?

Type free geometric argument (II)



- Covers all rank 2 affine Weyl groups
- Generalization to I_n? Meaning of π_0?

Type free geometric argument (II)



- Covers all rank 2 affine Weyl groups
- Generalization to I_n? Meaning of π_0?

Hecke group algebras and principal series representations

- $Y^{\lambda^\vee} \in \mathrm{H}(W)(q)$ (analog of translations in W)
- $\mathbb{C}[Y]$: commutative algebra, $\mathbb{C}[Y]^{\mathring{W}}$: center of $\mathrm{H}(W)(q)$
- t: character on $\mathbb{C}[Y]$
  - Central specialization of $\mathbb{C}[Y]^W$ on t:
    Quotient $\mathcal{H}(q,t)$ of $\mathrm{H}(W)(q)$ of dimension $|W|^2$
  - Representation $\rho_t := t \uparrow_{\mathbb{C}[Y]}^{\mathrm{H}(W)(q)}$ of $\mathrm{H}(W)(q)$ on $\mathbb{C}\mathring{W}$:
    Quotient of $\mathcal{H}(q,t)$, generically trivial

Theorem (S.,T., 2008)

Take q non zero and non root of unity, and $t : Y^{\lambda^\vee} \mapsto q^{-\operatorname{ht}(\lambda^\vee)}$
Then, $\rho_t(\mathrm{H}(W)(q)) = \mathrm{H}\mathring{W}$
I.e. $\mathrm{H}\mathring{W}$ *(non trivial!) quotient of* $\mathcal{H}(q,t)$ *and of* $\mathrm{H}(W)(q)$

- Proof: diagonalization of the action of Y on $\mathbb{C}\mathring{W}$, using alcove walks and the intertwining operators τ_i
- What happens at roots of unity?

Hecke group algebras and principal series representations

- $Y^{\lambda^\vee} \in \mathrm{H}(W)(q)$ (analog of translations in W)
- $\mathbb{C}[Y]$: commutative algebra, $\mathbb{C}[Y]^{\mathring{W}}$: center of $\mathrm{H}(W)(q)$
- t: character on $\mathbb{C}[Y]$
  - Central specialization of $\mathbb{C}[Y]^W$ on t:
    Quotient $\mathcal{H}(q,t)$ of $\mathrm{H}(W)(q)$ of dimension $|W|^2$
  - Representation $\rho_t := t \uparrow_{\mathbb{C}[Y]}^{\mathrm{H}(W)(q)}$ of $\mathrm{H}(W)(q)$ on $\mathbb{C}\mathring{W}$:
    Quotient of $\mathcal{H}(q,t)$, generically trivial

Theorem (S.,T., 2008)

Take q non zero and non root of unity, and $t : Y^{\lambda^\vee} \mapsto q^{-\operatorname{ht}(\lambda^\vee)}$
Then, $\rho_t(\mathrm{H}(W)(q)) = \mathrm{H}\mathring{W}$
I.e. $\mathrm{H}\mathring{W}$ (non trivial!) quotient of $\mathcal{H}(q,t)$ and of $\mathrm{H}(W)(q)$

- Proof: diagonalization of the action of Y on $\mathbb{C}\mathring{W}$, using alcove walks and the intertwining operators τ_i
- What happens at roots of unity?

Hecke group algebras and principal series representations

- $Y^{\lambda^\vee} \in \mathrm{H}(W)(q)$ (analog of translations in W)
- $\mathbb{C}[Y]$: commutative algebra, $\mathbb{C}[Y]^{\mathring{W}}$: center of $\mathrm{H}(W)(q)$
- t: character on $\mathbb{C}[Y]$
  - Central specialization of $\mathbb{C}[Y]^W$ on t:
    Quotient $\mathcal{H}(q,t)$ of $\mathrm{H}(W)(q)$ of dimension $|W|^2$
  - Representation $\rho_t := t \uparrow_{\mathbb{C}[Y]}^{\mathrm{H}(W)(q)}$ of $\mathrm{H}(W)(q)$ on $\mathbb{C}\mathring{W}$:
    Quotient of $\mathcal{H}(q,t)$, generically trivial

Theorem (S.,T., 2008)

Take q non zero and non root of unity, and $t : Y^{\lambda^\vee} \mapsto q^{-\operatorname{ht}(\lambda^\vee)}$
Then, $\rho_t(\mathrm{H}(W)(q)) = \mathrm{H}\mathring{W}$
I.e. $\mathrm{H}\mathring{W}$ (non trivial!) quotient of $\mathcal{H}(q,t)$ and of $\mathrm{H}(W)(q)$

- Proof: diagonalization of the action of Y on $\mathbb{C}\mathring{W}$, using alcove walks and the intertwining operators τ_i
- What happens at roots of unity?

Hecke group algebras and principal series representations

- $Y^{\lambda^\vee} \in \mathrm{H}(W)(q)$ (analog of translations in W)
- $\mathbb{C}[Y]$: commutative algebra, $\mathbb{C}[Y]^{\mathring{W}}$: center of $\mathrm{H}(W)(q)$
- t: character on $\mathbb{C}[Y]$
  - Central specialization of $\mathbb{C}[Y]^W$ on t:
    Quotient $\mathcal{H}(q,t)$ of $\mathrm{H}(W)(q)$ of dimension $|W|^2$
  - Representation $\rho_t := t \uparrow_{\mathbb{C}[Y]}^{\mathrm{H}(W)(q)}$ of $\mathrm{H}(W)(q)$ on $\mathbb{C}\mathring{W}$:
    Quotient of $\mathcal{H}(q,t)$, generically trivial

Theorem (S.,T., 2008)

Take q non zero and non root of unity, and $t : Y^{\lambda^\vee} \mapsto q^{-\operatorname{ht}(\lambda^\vee)}$
Then, $\rho_t(\mathrm{H}(W)(q)) = \mathrm{H}\mathring{W}$
I.e. $\mathrm{H}\mathring{W}$ (non trivial!) quotient of $\mathcal{H}(q,t)$ and of $\mathrm{H}(W)(q)$

- Proof: diagonalization of the action of Y on $\mathbb{C}\mathring{W}$, using alcove walks and the intertwining operators τ_i
- What happens at roots of unity?

Hecke group algebras and principal series representations

- $Y^{\lambda^\vee} \in \mathrm{H}(W)(q)$ (analog of translations in W)
- $\mathbb{C}[Y]$: commutative algebra, $\mathbb{C}[Y]^{\mathring{W}}$: center of $\mathrm{H}(W)(q)$
- t: character on $\mathbb{C}[Y]$
  - Central specialization of $\mathbb{C}[Y]^{W}$ on t:
    Quotient $\mathcal{H}(q,t)$ of $\mathrm{H}(W)(q)$ of dimension $|W|^2$
  - Representation $\rho_t := t \uparrow_{\mathbb{C}[Y]}^{\mathrm{H}(W)(q)}$ of $\mathrm{H}(W)(q)$ on $\mathbb{C}\mathring{W}$:
    Quotient of $\mathcal{H}(q,t)$, generically trivial

Theorem (S.,T., 2008)

Take q non zero and non root of unity, and $t : Y^{\lambda^\vee} \mapsto q^{-\operatorname{ht}(\lambda^\vee)}$
Then, $\rho_t(\mathrm{H}(W)(q)) = \mathrm{H}\mathring{W}$
I.e. $\mathrm{H}\mathring{W}$ *(non trivial!) quotient of* $\mathcal{H}(q,t)$ *and of* $\mathrm{H}(W)(q)$

- Proof: diagonalization of the action of Y on $\mathbb{C}\mathring{W}$, using alcove walks and the intertwining operators τ_i
- What happens at roots of unity?

Hecke group algebras and principal series representations

- $Y^{\lambda^\vee} \in \mathrm{H}(W)(q)$ (analog of translations in W)
- $\mathbb{C}[Y]$: commutative algebra, $\mathbb{C}[Y]^{\mathring{W}}$: center of $\mathrm{H}(W)(q)$
- t: character on $\mathbb{C}[Y]$
  - Central specialization of $\mathbb{C}[Y]^W$ on t:
    Quotient $\mathcal{H}(q,t)$ of $\mathrm{H}(W)(q)$ of dimension $|W|^2$
  - Representation $\rho_t := t \uparrow_{\mathbb{C}[Y]}^{\mathrm{H}(W)(q)}$ of $\mathrm{H}(W)(q)$ on $\mathbb{C}\mathring{W}$:
    Quotient of $\mathcal{H}(q,t)$, generically trivial

Theorem (S.,T., 2008)

Take q non zero and non root of unity, and $t : Y^{\lambda^\vee} \mapsto q^{-\operatorname{ht}(\lambda^\vee)}$
Then, $\rho_t(\mathrm{H}(W)(q)) = \mathrm{H}\mathring{W}$
I.e. $\mathrm{H}\mathring{W}$ (non trivial!) quotient of $\mathcal{H}(q,t)$ and of $\mathrm{H}(W)(q)$

- Proof: diagonalization of the action of Y on $\mathbb{C}\mathring{W}$, using alcove walks and the intertwining operators τ_i
- What happens at roots of unity?