Combinatoire et représentations des monoïdes de tris

Nicolas M. Thiéry[1,2]

[1]University of California at Davis, USA

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

LRI, 31 mars 2011

arXiv:0711.1561 [math.RT]
arXiv:0804.3781 [math.RT]
arXiv:0912.2212 [math.CO]
arXiv:1010.3455 [math.RT]
arXiv:1012.1361 [math.CO]
+ research in progress

Bubble (anti) sort algorithm

1234

Bubble (anti) sort algorithm

1234

Bubble (anti) sort algorithm

1243

Bubble (anti) sort algorithm

1423

Bubble (anti) sort algorithm

4123

Bubble (anti) sort algorithm

4123

Bubble (anti) sort algorithm

4132

Bubble (anti) sort algorithm

4312

Bubble (anti) sort algorithm

4312

Bubble (anti) sort algorithm

4321

Bubble (anti) sort algorithm

4321

Bubble (anti) sort algorithm

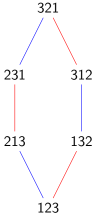
4321

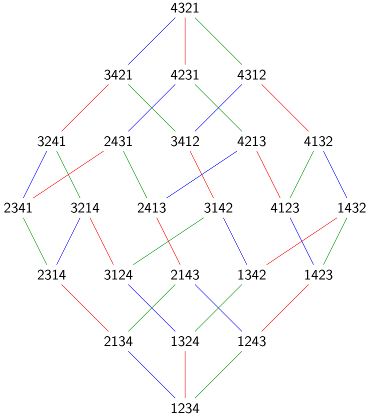
Underlying combinatorics: right permutahedron

Bubble (anti) sort algorithm

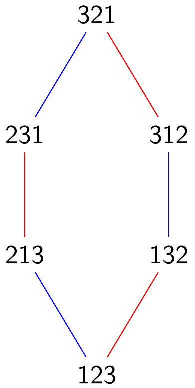
4321

Underlying combinatorics: right permutahedron

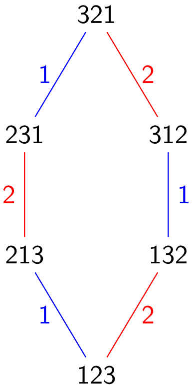




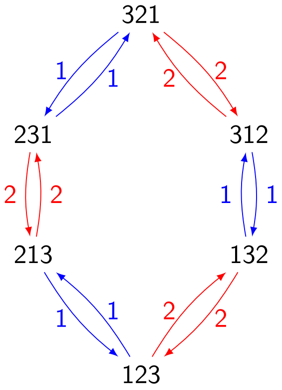
The permutohedron, as an automaton



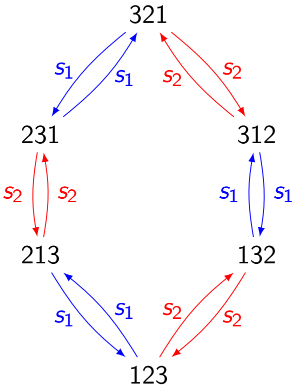
The permutohedron, as an automaton



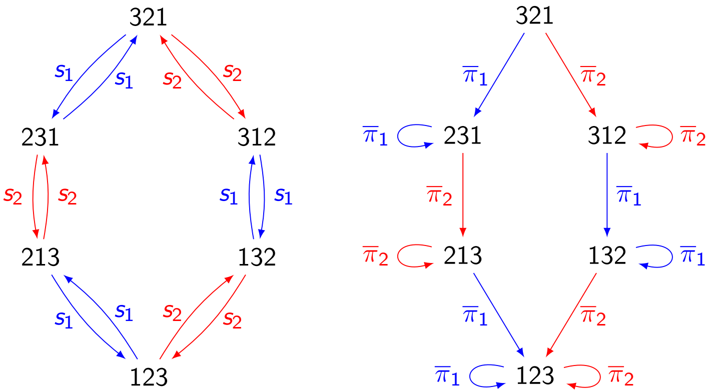
The permutohedron, as an automaton



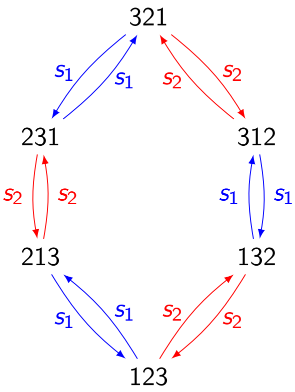
The permutohedron, as an automaton

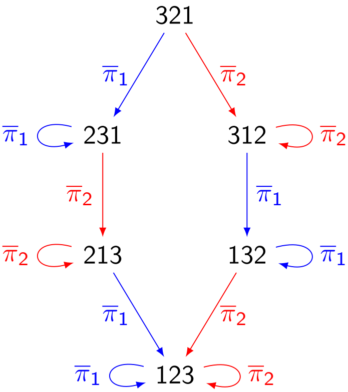


The permutohedron, as an automaton

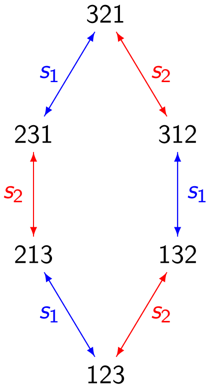


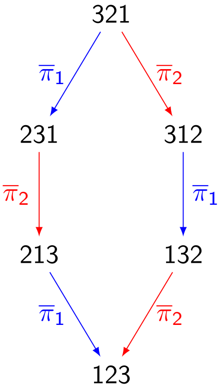
The permutohedron, as an automaton



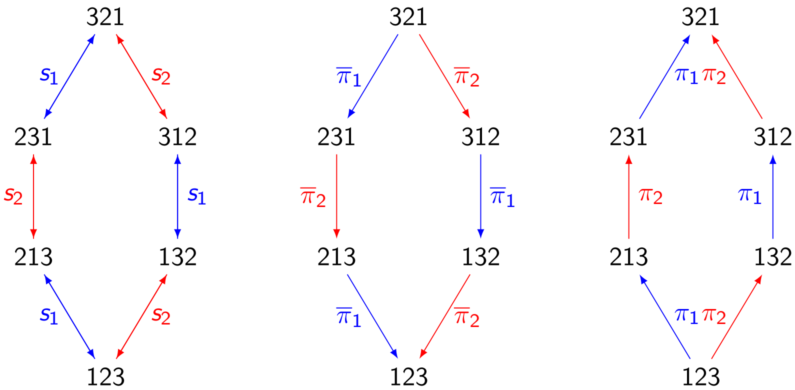


The permutohedron, as an automaton





The permutohedron, as an automaton



The transition monoid of an automaton

An automaton

A set of states: Q
An alphabet: $A = \{a_1, \ldots, a_n\}$
A set of transitions: $q \xrightarrow{a_i} q'$

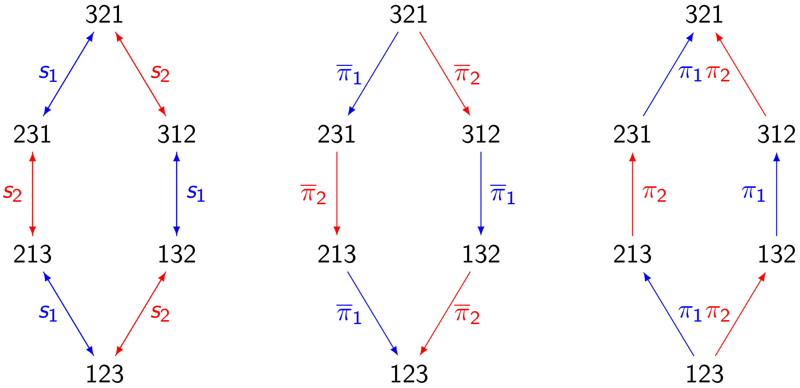
The free monoid and its action on the automaton

The free monoid A^*: words w over A, concatenation product
ρ maps a word w to a function $f_w : Q \mapsto Q$

The transition monoid

The monoid $\rho(A^*)$ generated by $f_{a_1}, \ldots, f_{a_n}$

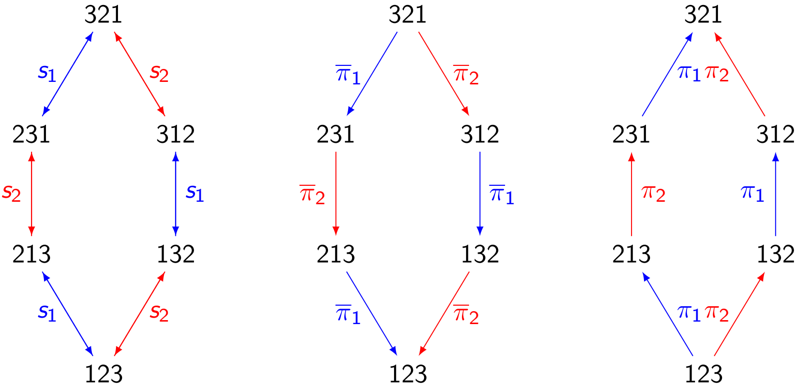
Monoids associated to the permutohedron



$$s_i^2 = 1$$

$$s_1 s_2 s_1 = s_2 s_1 s_2$$

Monoids associated to the permutohedron

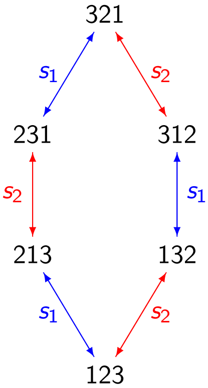


$$s_i^2 = 1$$

$$s_1 s_2 s_1 = s_2 s_1 s_2$$

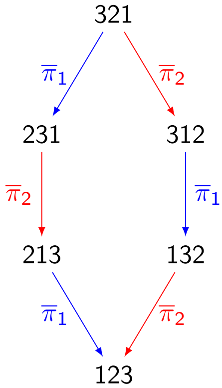
Symmetric group $\mathfrak{S}_3$

Monoids associated to the permutohedron



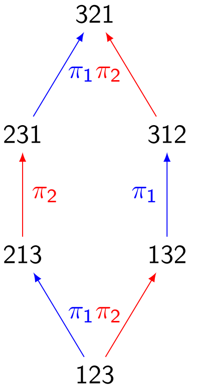
$s_i^2 = 1$

$s_1 s_2 s_1 = s_2 s_1 s_2$



$\overline{\pi}_i^2 = \overline{\pi}_i$

$\overline{\pi}_1 \overline{\pi}_2 \overline{\pi}_1 = \overline{\pi}_2 \overline{\pi}_1 \overline{\pi}_2$

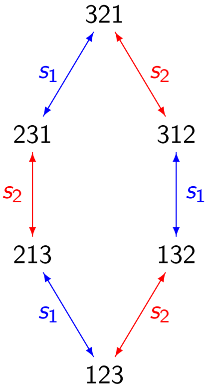


$\overline{\pi}_i^2 = \overline{\pi}_i$

$\pi_1 \pi_2 \pi_1 = \pi_2 \pi_1 \pi_2$

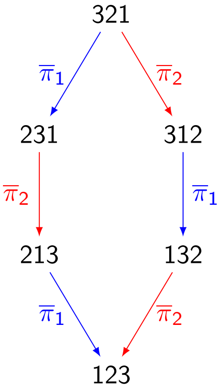
Symmetric group $\mathfrak{S}_3$

Monoids associated to the permutohedron



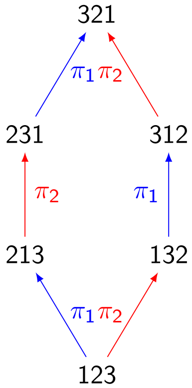
$$s_i^2 = 1$$
$$s_1 s_2 s_1 = s_2 s_1 s_2$$

Symmetric group $\mathfrak{S}_3$



$$\overline{\pi}_i^2 = \overline{\pi}_i$$
$$\overline{\pi}_1 \overline{\pi}_2 \overline{\pi}_1 = \overline{\pi}_2 \overline{\pi}_1 \overline{\pi}_2$$

0-Hecke monoide $\overline{H}_0(\mathfrak{S}_3)$



$$\overline{\pi}_i^2 = \overline{\pi}_i$$
$$\pi_1 \pi_2 \pi_1 = \pi_2 \pi_1 \pi_2$$

$H_0(\mathfrak{S}_3)$

Coxeter groups

Definition (Coxeter group W)

Generators : $s_1, s_2, \dots$ (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 $i \neq j$

Reduced words

0-Hecke monoid

Definition (0-Hecke monoid $H_0(W)$ of a Coxeter group W)

Generators : $\langle \pi_1, \pi_2, \dots \rangle$ (simple reflections)
Relations: $\pi_i^2 = \pi_i$ and braid relations

Theorem

$|H_0(W)| = |W|$
+ lots of nice properties

Motivation: simple combinatorial model (bubble sort)
appears in Iwahori-Hecke algebras, Schur symmetric functions, Schubert, Macdonald, Kazhdan-Lusztig polynomials, (affine) Stanley symmetric functions, mathematical physics, Schur-Weyl duality for quantum groups, representations of $GL(\mathbb{F}_q)$, probabilities, ...

0-Hecke monoid

Definition (0-Hecke monoid $H_0(W)$ of a Coxeter group W)

Generators : $\langle \pi_1, \pi_2, \ldots \rangle$ (simple reflections)
Relations: $\pi_i^2 = \pi_i$ and braid relations

Theorem

$|H_0(W)| = |W|$
+ *lots of nice properties*

Motivation: simple combinatorial model (bubble sort)
appears in Iwahori-Hecke algebras, Schur symmetric functions, Schubert, Macdonald, Kazhdan-Lusztig polynomials, (affine) Stanley symmetric functions, mathematical physics, Schur-Weyl duality for quantum groups, representations of $GL(\mathbb{F}_q)$, probabilities, ...

0-Hecke monoid

Definition (0-Hecke monoid $H_0(W)$ of a Coxeter group W)

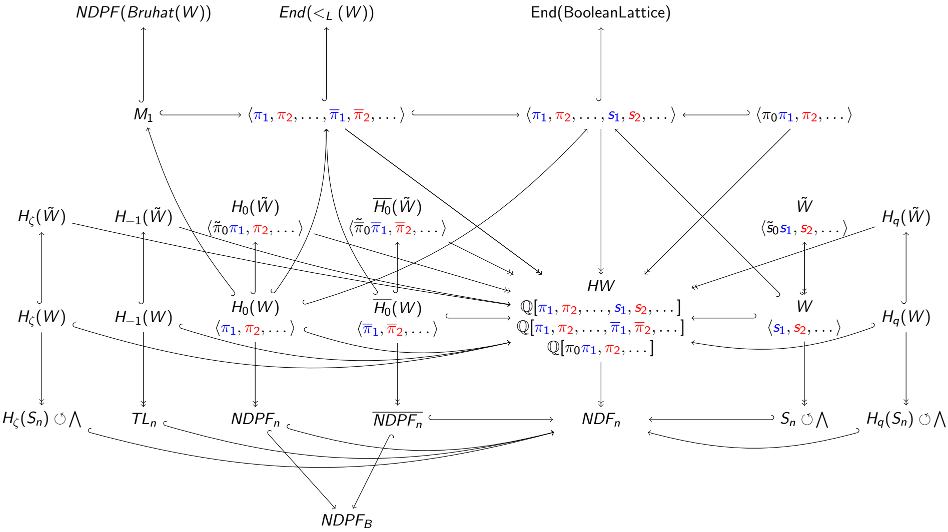
Generators : $\langle \pi_1, \pi_2, \ldots \rangle$ (simple reflections)
Relations: $\pi_i^2 = \pi_i$ and braid relations

Theorem

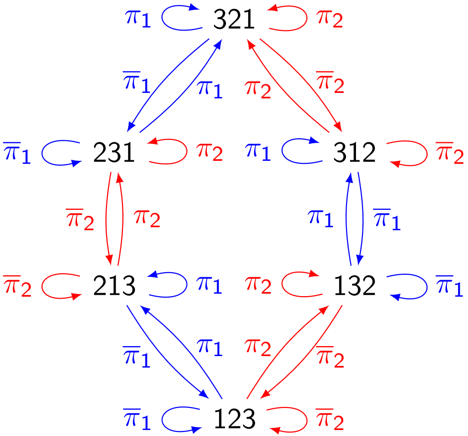
$|H_0(W)| = |W|$
$+$ *lots of nice properties*

Motivation: simple combinatorial model (bubble sort)
appears in Iwahori-Hecke algebras, Schur symmetric functions, Schubert, Macdonald, Kazhdan-Lusztig polynomials, (affine) Stanley symmetric functions, mathematical physics, Schur-Weyl duality for quantum groups, representations of $GL(\mathbb{F}_q)$, probabilities, ...

The Big Picture

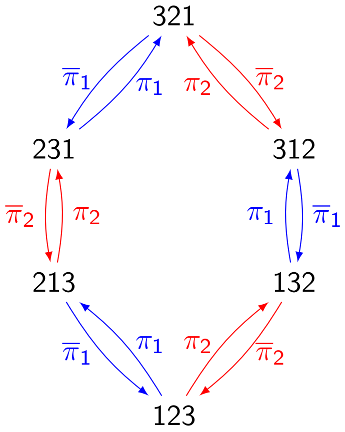


A strange cocktail: the biHecke monoid



What's the transition monoid?

A strange cocktail: the biHecke monoid



What's the transition monoid?

The biHecke monoid

Question

Size of $M(W) = \langle \pi_1, \pi_2, \ldots, \overline{\pi}_1, \overline{\pi}_2, \ldots \rangle$

$|M(S_n)| = 1, 3, 23, 477, 31103, ?$

- How to attack such a problem?
- Generators and relations?
- Representation theory?

Theorem (Hivert, Schilling, T. '08)

$M(W)$ admits $|W|$ simple / indecomposable projective modules

- Why do we care?

$$|M(W)| = \sum_{w \in W} \dim S_w . \dim P_w$$

The biHecke monoid

Question

Size of $M(W) = \langle \pi_1, \pi_2, \ldots, \overline{\pi}_1, \overline{\pi}_2, \ldots \rangle$

$|M(S_n)| = 1, 3, 23, 477, 31103, ?$

- How to attack such a problem?
- Generators and relations?
- Representation theory?

Theorem (Hivert, Schilling, T. '08)

$M(W)$ admits $|W|$ simple / indecomposable projective modules

- Why do we care?

$$|M(W)| = \sum_{w \in W} \dim S_w . \dim P_w$$

The biHecke monoid

Question

Size of $M(W) = \langle \pi_1, \pi_2, \ldots, \overline{\pi}_1, \overline{\pi}_2, \ldots \rangle$
$|M(S_n)| = 1, 3, 23, 477, 31103, ?$

- How to attack such a problem?
- Generators and relations?
- Representation theory?

Theorem (Hivert, Schilling, T. '08)

$M(W)$ admits $|W|$ simple / indecomposable projective modules

- Why do we care?

$$|M(W)| = \sum_{w \in W} \dim S_w . \dim P_w$$

The biHecke monoid

Question

Size of $M(W) = \langle \pi_1, \pi_2, \dots, \overline{\pi}_1, \overline{\pi}_2, \dots \rangle$

$|M(S_n)| = 1, 3, 23, 477, 31103, ?$

- How to attack such a problem?
- Generators and relations?
- Representation theory?

Theorem (Hivert, Schilling, T. '08)

$M(W)$ admits $|W|$ simple / indecomposable projective modules

- Why do we care?

$$|M(W)| = \sum_{w \in W} \dim S_w . \dim P_w$$

The biHecke monoid

Question

Size of $M(W) = \langle \pi_1, \pi_2, \ldots, \overline{\pi}_1, \overline{\pi}_2, \ldots \rangle$
$|M(S_n)| = 1, 3, 23, 477, 31103, ?$

- How to attack such a problem?
- Generators and relations?
- Representation theory?

Theorem (Hivert, Schilling, T. '08)

$M(W)$ admits $|W|$ simple / indecomposable projective modules

- Why do we care?

$$|M(W)| = \sum_{w \in W} \dim S_w . \dim P_w$$

The biHecke monoid

Question

Size of $M(W) = \langle \pi_1, \pi_2, \ldots, \overline{\pi}_1, \overline{\pi}_2, \ldots \rangle$

$|M(S_n)| = 1, 3, 23, 477, 31103, ?$

- How to attack such a problem?
- Generators and relations?
- Representation theory?

Theorem (Hivert, Schilling, T. '08)

$M(W)$ *admits* $|W|$ *simple / indecomposable projective modules*

- Why do we care?

$$|M(W)| = \sum_{w \in W} \dim S_w . \dim P_w$$

The biHecke monoid

Question

Size of $M(W) = \langle \pi_1, \pi_2, \ldots, \overline{\pi}_1, \overline{\pi}_2, \ldots \rangle$
$|M(S_n)| = 1, 3, 23, 477, 31103, ?$

- How to attack such a problem?
- Generators and relations?
- Representation theory?

Theorem (Hivert, Schilling, T. '08)

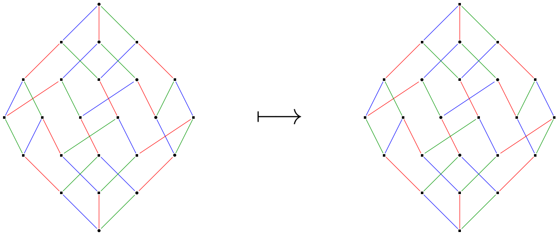
$M(W)$ admits $|W|$ simple / indecomposable projective modules

- Why do we care?

$$|M(W)| = \sum_{w \in W} \dim S_w . \dim P_w$$

Key combinatorial lemma

$13\overline{2}$



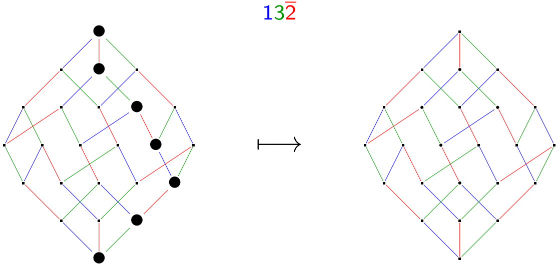
Lemma

For $f \in M(W)$ and $w \in W$: $\quad (s_i w).f = w.f \quad$ or $\quad s_i\,(w.f)$

Proof.

Exchange property / associativity □

Key combinatorial lemma



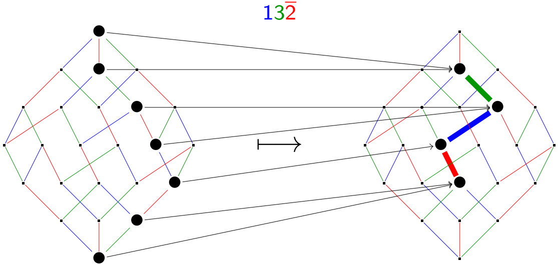
Lemma

For $f \in M(W)$ and $w \in W$: $\quad (s_i w).f = w.f \quad$ *or* $\quad s_i\,(w.f)$

Proof.

Exchange property / associativity □

Key combinatorial lemma



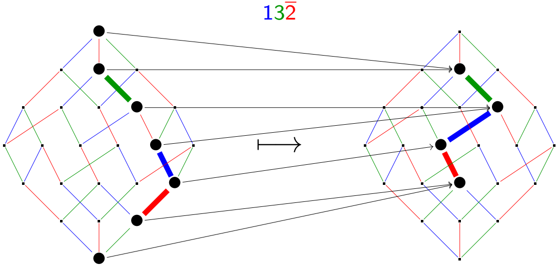
Lemma

For $f \in M(W)$ and $w \in W$: $\quad (s_i w).f = w.f \quad$ *or* $\quad s_i\,(w.f)$

Proof.

Exchange property / associativity □

Key combinatorial lemma



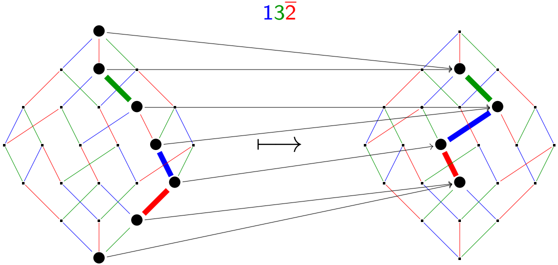
Lemma

For $f \in M(W)$ and $w \in W$: $\quad (s_i w).f = w.f \quad$ *or* $\quad s_i\,(w.f)$

Proof.

Exchange property / associativity □

Key combinatorial lemma



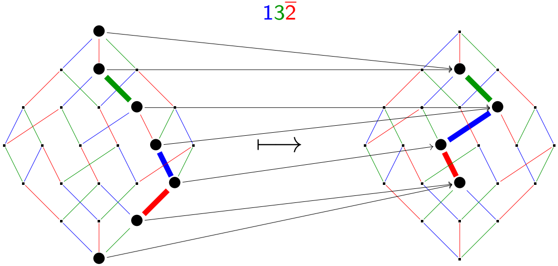
Lemma

For $f \in M(W)$ and $w \in W$: $(s_i w).f = w.f$ *or* $s_i\,(w.f)$

Proof.

Exchange property / associativity □

Key combinatorial lemma



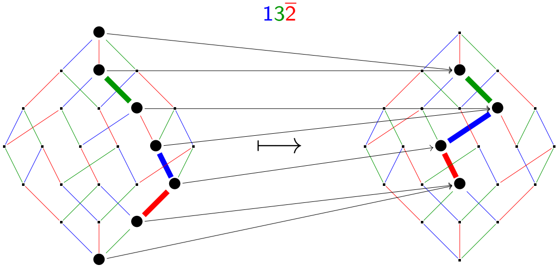
Lemma

For $f \in M(W)$ and $w \in W$: $\quad (s_i w).f = w.f \quad$ *or* $\quad s_i\,(w.f)$

Proof.

Exchange property / associativity □

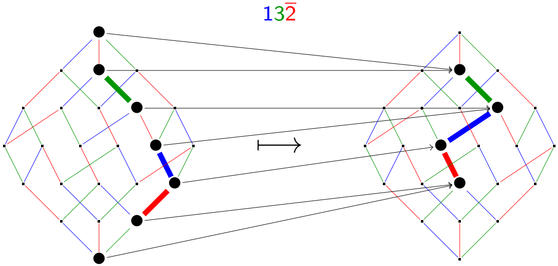
Key combinatorial lemma



Corollary

- *Preservation of left order:* $u \leq_L v \Longrightarrow u.f \leq_L v.f$
- *Preservation of Bruhat order:* $u \leq_B v \Longrightarrow u.f \leq_B v.f$
- $M(W)$ *is aperiodic*
- f *in* $M(W)$ *is determined by its fibers and* $f(1)$

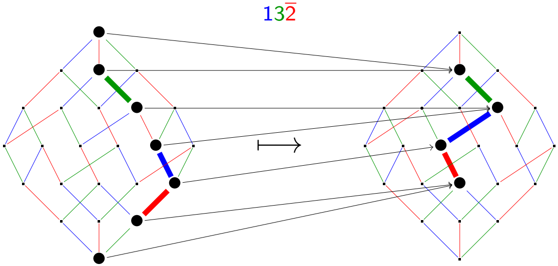
Key combinatorial lemma



Corollary

- *Preservation of left order:* $u \leq_L v \Longrightarrow u.f \leq_L v.f$
- *Preservation of Bruhat order:* $u \leq_B v \Longrightarrow u.f \leq_B v.f$
- $M(W)$ *is aperiodic*
- f *in* $M(W)$ *is determined by its fibers and* $f(1)$

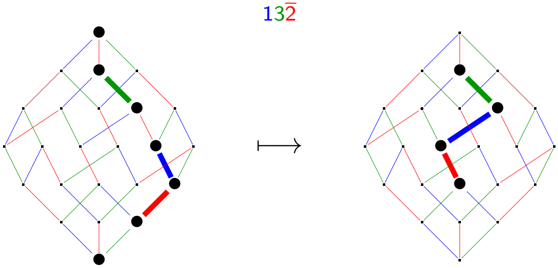
Key combinatorial lemma



Corollary

- *Preservation of left order:* $u \leq_L v \Longrightarrow u.f \leq_L v.f$
- *Preservation of Bruhat order:* $u \leq_B v \Longrightarrow u.f \leq_B v.f$
- $M(W)$ *is aperiodic*
- f *in* $M(W)$ *is determined by its fibers and* $f(1)$

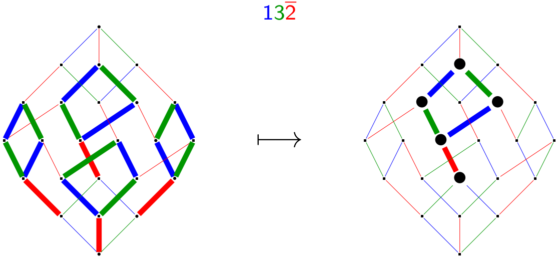
Key combinatorial lemma



Corollary

- *Preservation of left order:* $u \leq_L v \Longrightarrow u.f \leq_L v.f$
- *Preservation of Bruhat order:* $u \leq_B v \Longrightarrow u.f \leq_B v.f$
- $M(W)$ *is aperiodic*
- f *in* $M(W)$ *is determined by its fibers and* $f(1)$

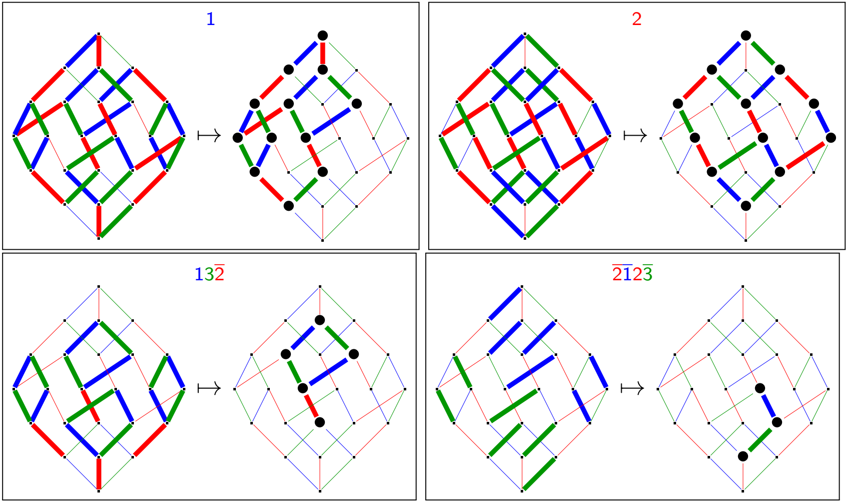
Key combinatorial lemma



Corollary

- *Preservation of left order:* $u \leq_L v \Longrightarrow u.f \leq_L v.f$
- *Preservation of Bruhat order:* $u \leq_B v \Longrightarrow u.f \leq_B v.f$
- $M(W)$ *is aperiodic*
- f *in* $M(W)$ *is determined by its fibers and* $f(1)$

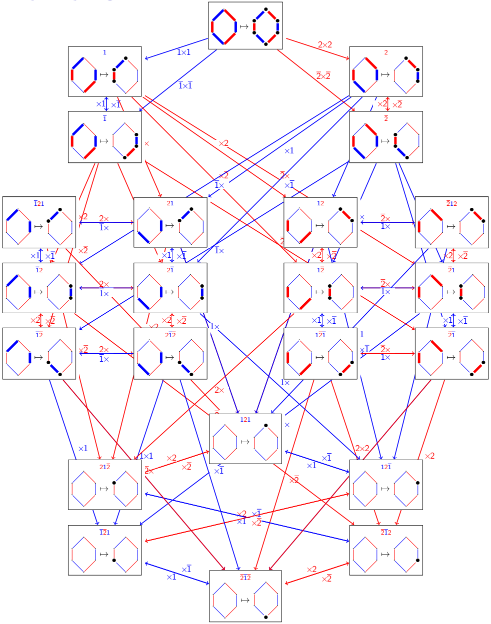
Some elements of the monoid



Lemma

The image set of an idempotent is an interval in left order

Cayley graph and Green relations



Green relations

Theorem (HST'08-10)

- *Regular $\mathcal{J}$-classes are indexed by W*
- *$\mathcal{J}$-order on regular classes: left-right order on W*
- *$\mathcal{R}$-classes: intervals in right order on W*
- *$\mathcal{R}$-order on regular $\mathcal{R}$-classes: $\approx$ right order on W*
- *$\mathcal{L}$-order on regular $\mathcal{L}$-classes: $\approx$ left order on W*

Problems

- *$\mathcal{L}$,$\mathcal{R}$,$\mathcal{J}$-order between non regular classes?*
- *$\mathcal{L}$-classes?*

Representation theory

Definition

A: algebra / group / monoid

Linear representation: vector space V with a morphism

$$\rho \ : \ A \longmapsto \mathrm{End}(V)$$

(Left) Module: Bilinear operation $a.v$ (for $a \in A$, $v \in V$) such that

$$a.(b.v) = (ab).v$$

Idea:

- Better understanding of A via multiple views on it
- Reduce the study of A to linear algebra

Representation theory (building blocks)

Definition

Submodule $W \subset V$ is a stable subspace (if $x \in W$ then $a.x \in W$).

Simple (irreducible) module: no nontrivial submodule.

The smallest possible modules.
Indecomposable module: V cannot be written as $V = V_1 \oplus V_2$
Projective module: $V \oplus \cdots = \mathbb{C}[M] \oplus \cdots \oplus \mathbb{C}[M]$

Representation theory

Theorem (See e.g. Curtis-Reiner)

Simple modules $\leftrightarrow$ indecomposable projective modules
Dimension formula, ...

It's computational!

$\Rightarrow$ computer exploration

- Algebras: linear algebra $\gg O(n^3)$
- Groups: characters $o(n)$
- What about monoids?

Representation theory

Theorem (See e.g. Curtis-Reiner)

Simple modules $\leftrightarrow$ *indecomposable projective modules*
Dimension formula, . . .

It's computational!

$\Rightarrow$ computer exploration

- Algebras: linear algebra $\gg O(n^3)$
- Groups: characters $o(n)$
- What about monoids?

Representation theory

Theorem (See e.g. Curtis-Reiner)

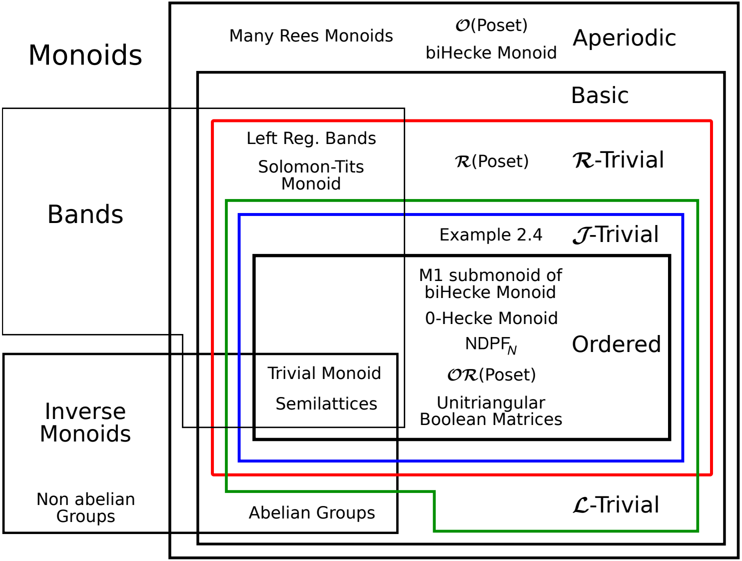
Simple modules $\leftrightarrow$ *indecomposable projective modules*
Dimension formula, ...

It's computational!

$\Rightarrow$ computer exploration

- Algebras: linear algebra $\gg O(n^3)$
- Groups: characters $o(n)$
- What about monoids?

Zoology of monoids



Representation theory of monoids

Theorem (See e.g. Ganyushkin, Mazorchuk, Steinberg 07)

The regular J-classes determine the simple modules.

Definition (Schüztenberger)

Aperiodic monoid: no trivial subgroup

$\Longrightarrow$ Combinatorial Representation Theory

Representation theory of monoids

Theorem (See e.g. Ganyushkin, Mazorchuk, Steinberg 07)

The regular J-classes (essentially) determine the simple modules.

Definition (Schüztenberger)

Aperiodic monoid: no trivial subgroup

$\Longrightarrow$ Combinatorial Representation Theory

Representation theory of monoids

Theorem (See e.g. Ganyushkin, Mazorchuk, Steinberg 07)

The regular J-classes (essentially) determine the simple modules.

Definition (Schüztenberger)

Aperiodic monoid: no trivial subgroup

$\Longrightarrow$ Combinatorial Representation Theory

Representation theory of monoids

Theorem (See e.g. Ganyushkin, Mazorchuk, Steinberg 07)

The regular J-classes (essentially) determine the simple modules.

Definition (Schüztenberger)

Aperiodic monoid: no trivial subgroup

$\Longrightarrow$ Combinatorial Representation Theory

Representation theory of $M(W)$

Theorem (HST'08)

$M(W)$ admits $|W|$ simple modules / indecomposable projective modules

Problem

Dimension of simple and indecomposable projective modules?

Representation theory of $M(W)$

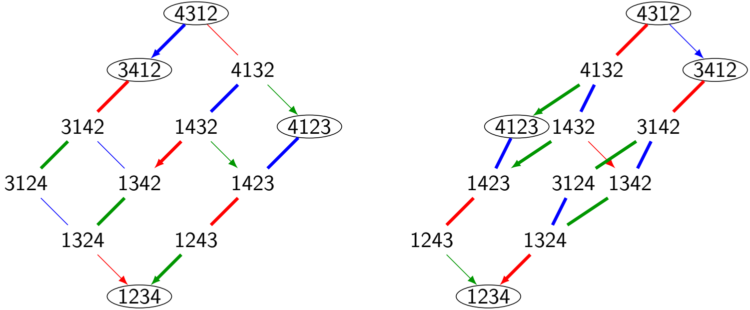
Theorem (HST'08)

$M(W)$ admits $|W|$ simple modules / indecomposable projective modules

Problem

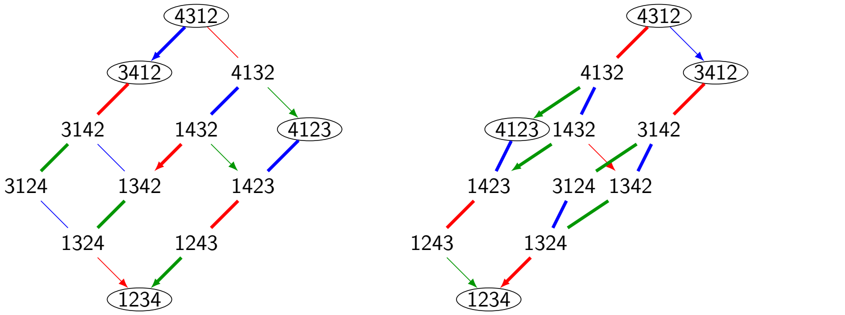
Dimension of simple and indecomposable projective modules?

$\mathcal{R}$-class modules



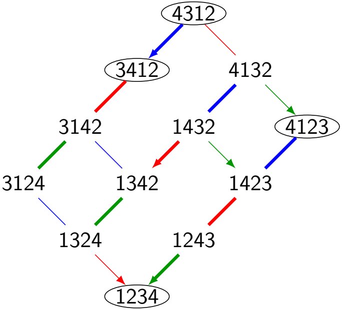
- Blocks: $J = \{\}, \{1, 2\}, \{3\}, \{1, 2, 3\} \implies$ Submodules P_J
- Description of the simple module S_w
- Dimension: inclusion-exclusion along the **cutting poset**
- Generating series calculation?

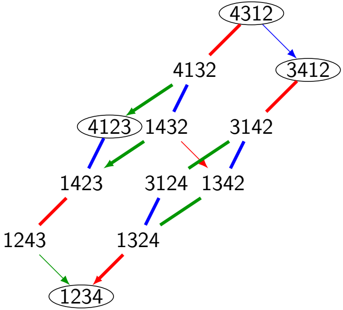
$\mathcal{R}$-class modules



- Blocks: $J = \{\}, \{1, 2\}, \{3\}, \{1, 2, 3\} \implies$ Submodules P_J
- Description of the simple module S_w
- Dimension: inclusion-exclusion along the **cutting poset**
- Generating series calculation?

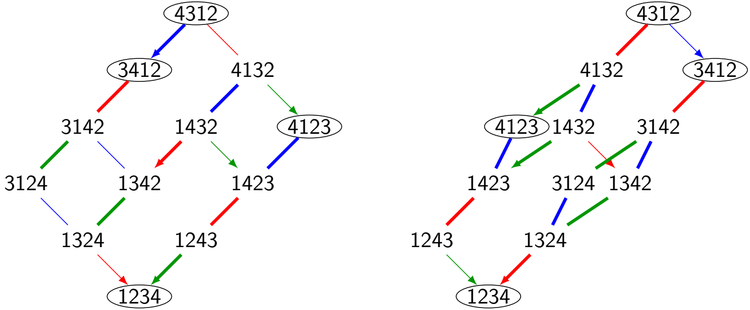
$\mathcal{R}$-class modules





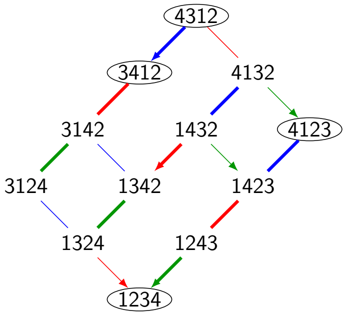
- Blocks: $J = \{\}, \{1, 2\}, \{3\}, \{1, 2, 3\} \quad \Longrightarrow \quad$ Submodules P_J
- Description of the simple module S_w
- Dimension: inclusion-exclusion along the **cutting poset**
- Generating series calculation?

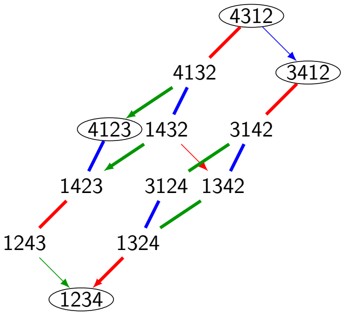
$\mathcal{R}$-class modules



- Blocks: $J = \{\}, \{1, 2\}, \{3\}, \{1, 2, 3\} \implies$ Submodules P_J
- Description of the simple module S_w
- Dimension: inclusion-exclusion along the **cutting poset**
- Generating series calculation?

$\mathcal{R}$-class modules





- Blocks: $J = \{\}, \{1, 2\}, \{3\}, \{1, 2, 3\} \quad \Longrightarrow \quad$ Submodules P_J
- Description of the simple module S_w
- Dimension: inclusion-exclusion along the **cutting poset**
- Generating series calculation?

The cutting poset

Definition (Cutting poset $(W, \sqsubseteq)$ ST'09)

$u \sqsubseteq w$ if $u = w^J$ with J block

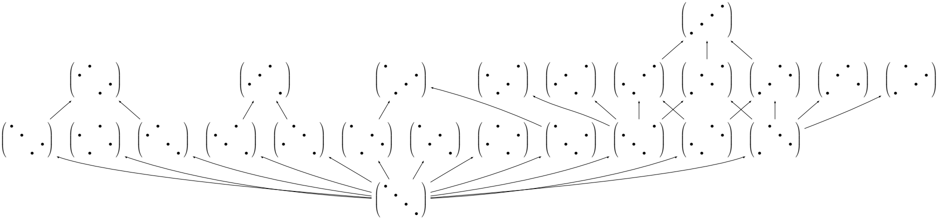
Theorem

- *Intervals are distributive lattices*
- *Distributive meet-semi lattice*
- *Möbius function: inclusion-exclusion along minimal blocks*

The cutting poset

Definition (Cutting poset $(W, \sqsubseteq)$ ST'09)

$u \sqsubseteq w$ if $u = w^J$ with J block



Theorem

- *Intervals are distributive lattices*
- *Distributive meet-semi lattice*
- *Möbius function: inclusion-exclusion along minimal blocks*

Work in progress

Problem

Projective modules? Cartan Matrix?

Results on aperiodic monoids (T'11)

- It is sufficient to look at left and right class modules
- Fast (mostly) combinatorial algorithms based on embeddings of automata

Work in progress

Problem

Projective modules? Cartan Matrix?

Results on aperiodic monoids (T'11)

- It is sufficient to look at left and right class modules
- Fast (mostly) combinatorial algorithms based on embeddings of automata

Overall strategy

Combinatorics Algebra

Computer exploration

- Use algebra to raises questions
- Use combinatorics to make them concrete
- Use computer exploration to suggest answers and produce combinatorial data
- Analyse the data
- Find combinatorial regularities and proofs
- Derive conceptual interpretations
- Derive better algorithms
- Reiterate!

Overall strategy

Combinatorics Algebra

Computer exploration

- Use algebra to raises questions
- Use combinatorics to make them concrete
- Use computer exploration to suggest answers and produce combinatorial data
- Analyse the data
- Find combinatorial regularities and proofs
- Derive conceptual interpretations
- Derive better algorithms
- Reiterate!

Software needs for computer exploration

The previous calculations required:

A wide set of features

- Groups, root systems, ...
- Monoids of transformations, automatic monoids
- Automatons
- Graphs: standard algorithmic, isomorphism, visualization
- Posets, lattices
- Linear algebra (vector spaces, morphisms, quotients, ...)
- Serialization, Parallelism, ...
- **Representations of monoids**

A tight modelling of mathematics

Software needs for computer exploration

The previous calculations required:

A wide set of features

- Groups, root systems, ...
- Monoids of transformations, automatic monoids
- Automatons
- Graphs: standard algorithmic, isomorphism, visualization
- Posets, lattices
- Linear algebra (vector spaces, morphisms, quotients, ...)
- Serialization, Parallelism, ...
- **Representations of monoids**

A tight modelling of mathematics

The ∗-Combinat projet: combinat.sagemath.org

Mission statement

"To improve MuPAD/Sage as an extensible toolbox for computer exploration in combinatorics, and foster code sharing among researchers in this area"

Stratégie

- Licence libre pour partager avec le plus grand nombre
  En restant pragmatique dans les collaborations
- Développement décentralisé et international
  Garantie d'indépendance vis-à-vis des tutelles
- Développé par des chercheurs pour des chercheurs
  Avec un usage plus large en vue
- Coeur du développement par des permanents
  Les doctorants se concentrent sur leurs propres besoins
- Chaque ligne de code justifiée par un projet de recherche
  Avec une vision à long terme (développement agile)
- Inspiration des informaticiens:
  Concepts et méthodologies de programmation
  Outils de développement coopératif

The $*$-Combinat projet: combinat.sagemath.org

Mission statement

"To improve MuPAD/Sage as an extensible toolbox for computer exploration in combinatorics, and foster code sharing among researchers in this area"

Stratégie

- Licence libre pour partager avec le plus grand nombre
  En restant pragmatique dans les collaborations
- Développement décentralisé et international
  Garantie d'indépendance vis-à-vis des tutelles
- Développé par des chercheurs pour des chercheurs
  Avec un usage plus large en vue
- Coeur du développement par des permanents
  Les doctorants se concentrent sur leurs propres besoins
- Chaque ligne de code justifiée par un projet de recherche
  Avec une vision à long terme (développement agile)
- Inspiration des informaticiens:
  Concepts et méthodologies de programmation
  Outils de développement coopératif

The *-Combinat project: 10 years after

In a nutshell

- `MuPAD-Combinat`: 115k lines of `MuPAD`, 15k lines of C++, 32k lines of tests, 600 pages of doc
- `Sage-Combinat`: 300 tickets / 100k lines integrated in `Sage`
- Sponsors: ANR, PEPS, NSF, Google Summer of Code
- **70+ research articles**

A community:

Nicolas Borie, Daniel Bump, Jason Bandlow, Adrien Boussicault, Frédéric Chapoton, Vincent Delecroix, Paul-Olivier Dehaye, Tom Denton, François Descouens, Dan Drake, Teresa Gomez Diaz, Valentin Feray, Mike Hansen, Ralf Hemmecke, Florent Hivert, Brant Jones, Sébastien Labbé, Yann Laigle-Chapuy, Éric Laugerotte, Patrick Lemeur, Andrew Mathas, Xavier Molinero, Thierry Monteil, Olivier Mallet, Gregg Musiker, Jean-Christophe Novelli, Janvier Nzeutchap, Steven Pon, Viviane Pons, Franco Saliola, Anne Schilling, Mark Shimozono, Christian Stump, Lenny Tevlin, Nicolas M. Thiéry, Justin Walker, Qiang Wang, Mike Zabrocki, ... **And you?**

The ∗-Combinat project: 10 years after

In a nutshell

- MuPAD-Combinat: 115k lines of MuPAD, 15k lines of C++, 32k lines of tests, 600 pages of doc
- Sage-Combinat: 300 tickets / 100k lines integrated in Sage
- Sponsors: ANR, PEPS, NSF, Google Summer of Code
- **70+ research articles**

A community:

Nicolas Borie, Daniel Bump, Jason Bandlow, Adrien Boussicault, Frédéric Chapoton, Vincent Delecroix, Paul-Olivier Dehaye, Tom Denton, François Descouens, Dan Drake, Teresa Gomez Diaz, Valentin Feray, Mike Hansen, Ralf Hemmecke, Florent Hivert, Brant Jones, Sébastien Labbé, Yann Laigle-Chapuy, Éric Laugerotte, Patrick Lemeur, Andrew Mathas, Xavier Molinero, Thierry Monteil, Olivier Mallet, Gregg Musiker, Jean-Christophe Novelli, Janvier Nzeutchap, Steven Pon, Viviane Pons, Franco Saliola, Anne Schilling, Mark Shimozono, Christian Stump, Lenny Tevlin, Nicolas M. Thiéry, Justin Walker, Qiang Wang, Mike Zabrocki, ... **And you?**

The $*$-Combinat project: 10 years after

In a nutshell

- MuPAD-Combinat: 115k lines of MuPAD, 15k lines of C++, 32k lines of tests, 600 pages of doc
- Sage-Combinat: 300 tickets / 100k lines integrated in Sage
- Sponsors: ANR, PEPS, NSF, Google Summer of Code
- **70+ research articles**

A community:

Nicolas Borie, Daniel Bump, Jason Bandlow, Adrien Boussicault, Frédéric Chapoton, Vincent Delecroix, Paul-Olivier Dehaye, Tom Denton, François Descouens, Dan Drake, Teresa Gomez Diaz, Valentin Feray, Mike Hansen, Ralf Hemmecke, Florent Hivert, Brant Jones, Sébastien Labbé, Yann Laigle-Chapuy, Éric Laugerotte, Patrick Lemeur, Andrew Mathas, Xavier Molinero, Thierry Monteil, Olivier Mallet, Gregg Musiker, Jean-Christophe Novelli, Janvier Nzeutchap, Steven Pon, Viviane Pons, Franco Saliola, Anne Schilling, Mark Shimozono, Christian Stump, Lenny Tevlin, Nicolas M. Thiéry, Justin Walker, Qiang Wang, Mike Zabrocki, ...

And you?

The $*$-Combinat project: 10 years after

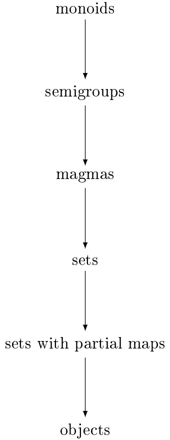
In a nutshell

- MuPAD-Combinat: 115k lines of MuPAD, 15k lines of C++, 32k lines of tests, 600 pages of doc
- Sage-Combinat: 300 tickets / 100k lines integrated in Sage
- Sponsors: ANR, PEPS, NSF, Google Summer of Code
- **70+ research articles**

A community:

Nicolas Borie, Daniel Bump, Jason Bandlow, Adrien Boussicault, Frédéric Chapoton, Vincent Delecroix, Paul-Olivier Dehaye, Tom Denton, François Descouens, Dan Drake, Teresa Gomez Diaz, Valentin Feray, Mike Hansen, Ralf Hemmecke, Florent Hivert, Brant Jones, Sébastien Labbé, Yann Laigle-Chapuy, Éric Laugerotte, Patrick Lemeur, Andrew Mathas, Xavier Molinero, Thierry Monteil, Olivier Mallet, Gregg Musiker, Jean-Christophe Novelli, Janvier Nzeutchap, Steven Pon, Viviane Pons, Franco Saliola, Anne Schilling, Mark Shimozono, Christian Stump, Lenny Tevlin, Nicolas M. Thiéry, Justin Walker, Qiang Wang, Mike Zabrocki, … **And you?**

Managing large hierarchies of classes I

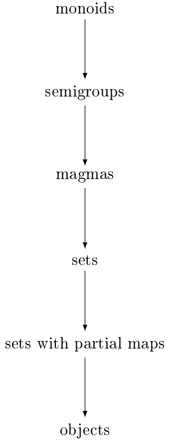


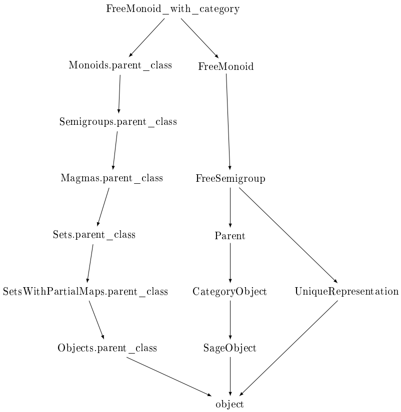
Categories

(and also: classes for tests, for morphisms)

Manual class hierarchies violate the “Single Point Of Truth” pattern

Managing large hierarchies of classes I



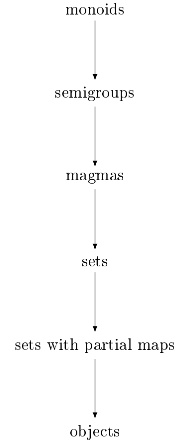


Categories	Classes for parents	

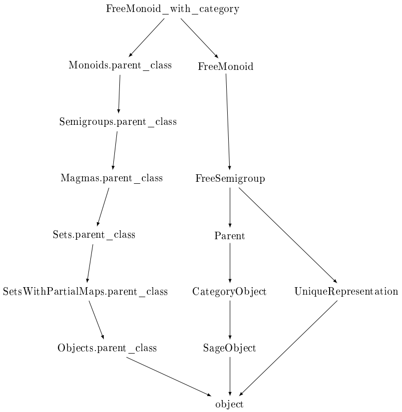
(and also: classes for tests, for morphisms)

Manual class hierarchies violate the "Single Point Of Truth" pattern

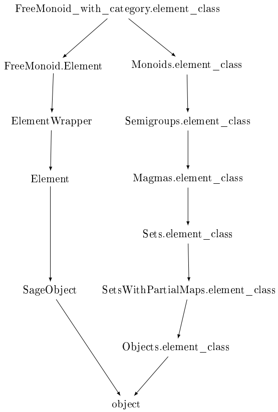
Managing large hierarchies of classes I



Categories



Classes for parents

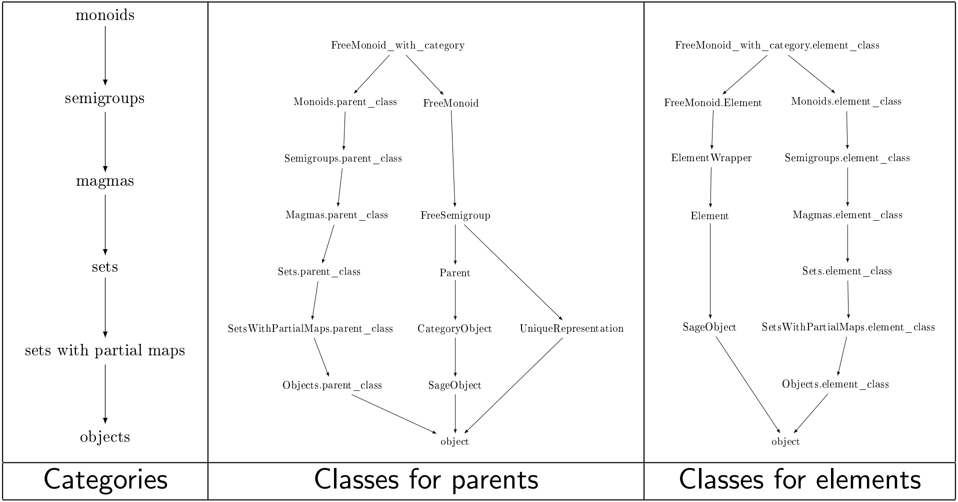


Classes for elements

(and also: classes for tests, for morphisms)

Manual class hierarchies violate the “Single Point Of Truth” pattern

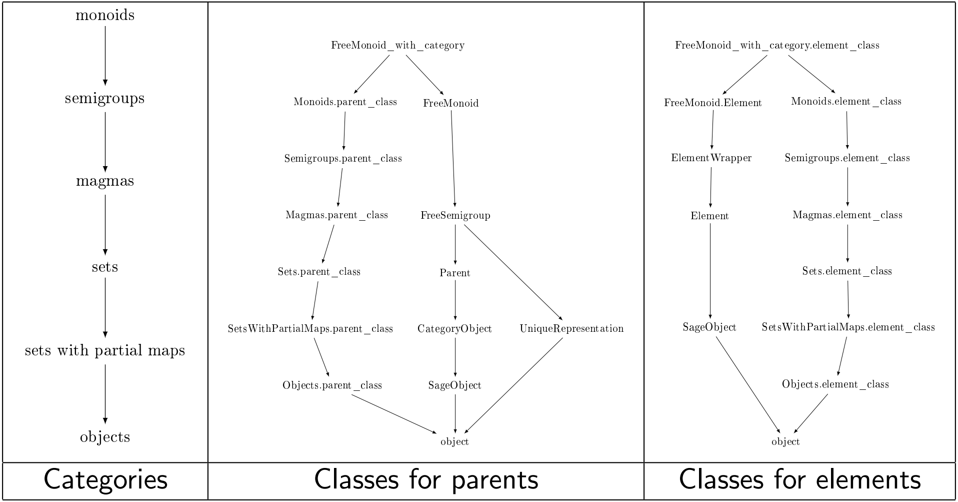
Managing large hierarchies of classes I



(and also: classes for tests, for morphisms)

Manual class hierarchies violate the “Single Point Of Truth” pattern

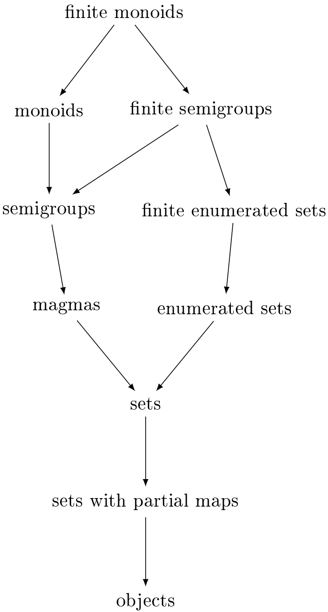
Managing large hierarchies of classes I



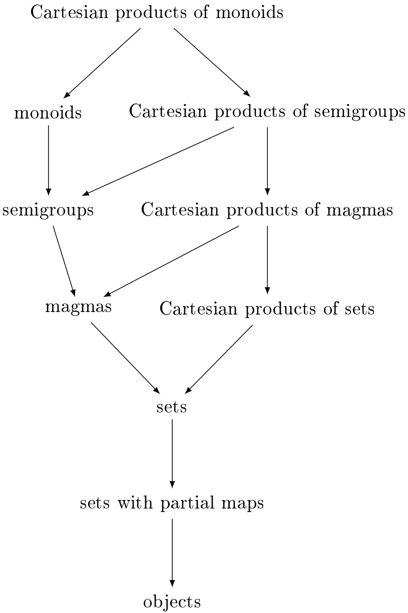
(and also: classes for tests, for morphisms)

Manual class hierarchies violate the “Single Point Of Truth” pattern

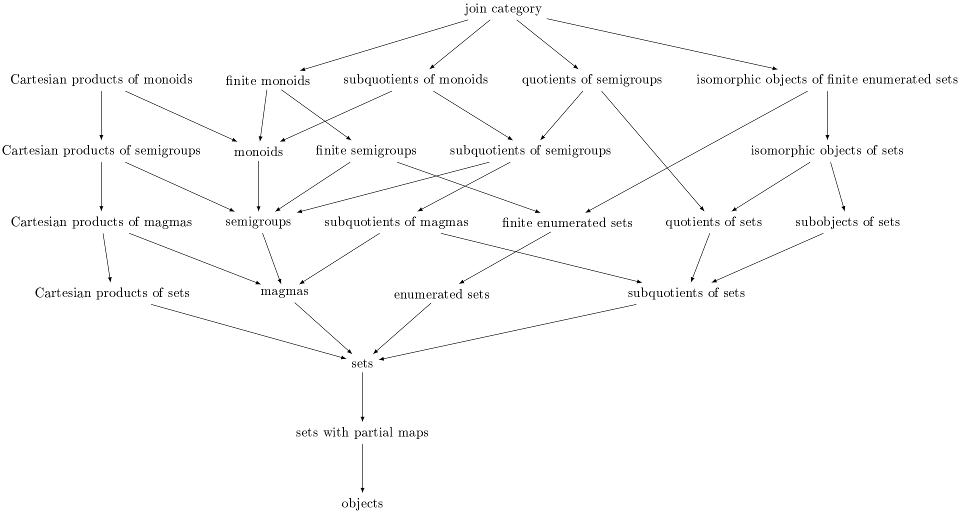
Managing large hierarchies of classes II



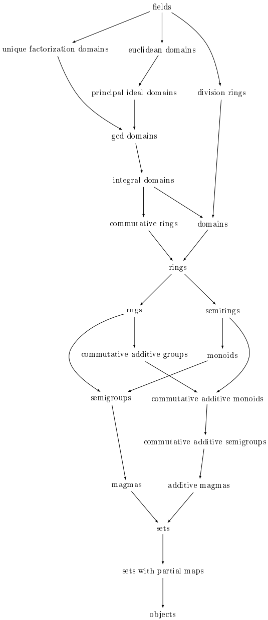
Managing large hierarchies of classes II



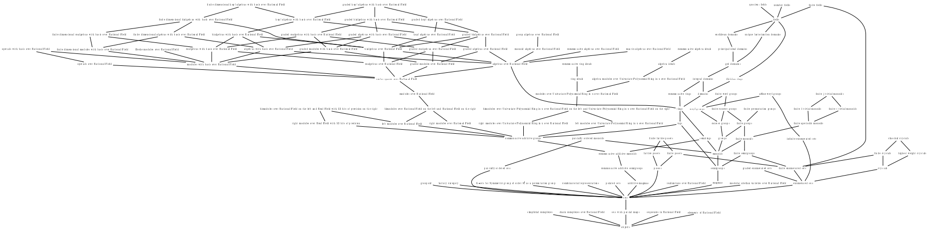
Managing large hierarchies of classes II



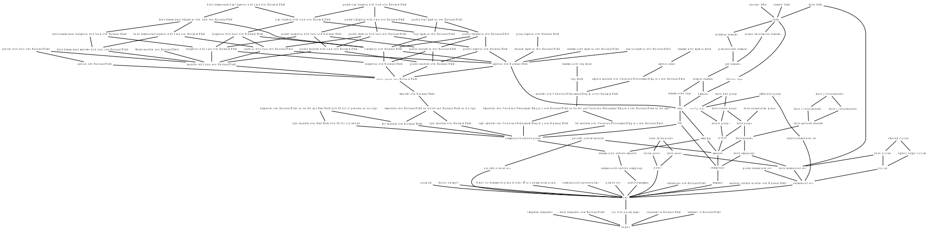
Managing large hierarchies of classes II



Managing large hierarchies of classes II



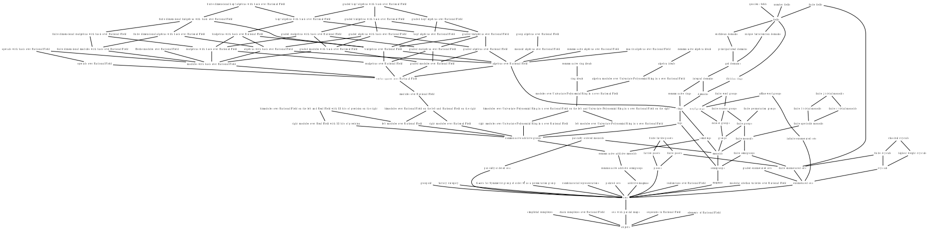
Managing large hierarchies of classes II



Problem

- *Control the* **combinatorial explosion**
- *Keep the size of the code and of the class hierarchy linear with the number of non trivial classes*

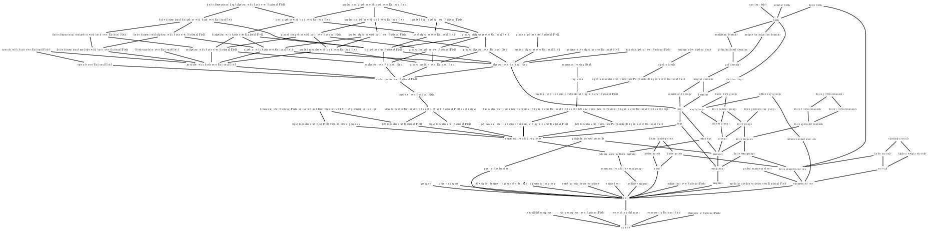
Managing large hierarchies of classes II



Solution

- Hierarchy of classes generated **dynamically** and **lazily**
- Lattice algorithmic to minimize the hierarchy

Managing large hierarchies of classes II



Challenges

- Refactoring of 400k lines / 10 referees / 1 year
- Compilation / partial compilation
- Serialization
- Documentation tools
- Dynamic class updates