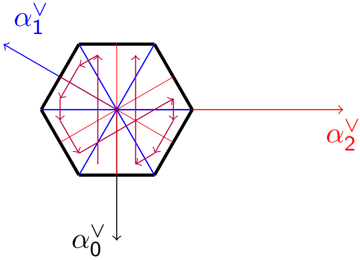
*-Combinat
Sharing algebraic combinatorics since 2000

Florent Hivert, Nicolas M. Thiéry, et al.

LIFAR, University of Rouen, France
Laboratoire de Mathématiques d'Orsay, Université Paris Sud, France
et al.

Sage Days 10, October 10-th of 2008



*-Combinat
An open-source framework
For computer exploration
In algebraic combinatorics

*-Combinat in a nutshell

Demo: combinatorics for all

Computer exploration in algebraic combinatorics: Cayley trees

Short history, switch to Sage

Status of the switch, advanced demo, future directions

It all started there

*-Combinat figures

- Community: Abbad, Bandlow, *Borie*, Bump, Carré, Chapotton, Denton, Descouens, Gomez-Diaz, *Hansen*, Hemmecke, *Hivert*, Jones, Laugerotte, Lecouvey, Lemeur, *Miller*, Molinero, Musiker, Novelli, Nzeutchap, Qiang, Rubey, Schilling, Shimozono, *Thiéry*, *Tevlin*, Walker, Zabrocki, *Zimmermann*
- End result: 50 publications
- 600 classes, 5000 methods, 130k lines of MuPAD
  60k lines of Sage
- 32k lines of tests, 600 pages of doc

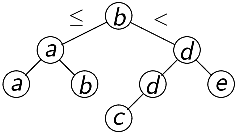
Demo: combinatorics for all

Algebraic combinatorics

Motivation: relations algorithms ↔ algebraic structures

Combinatorial objects appear in algebraic computations

(()((()())(()))(()()))



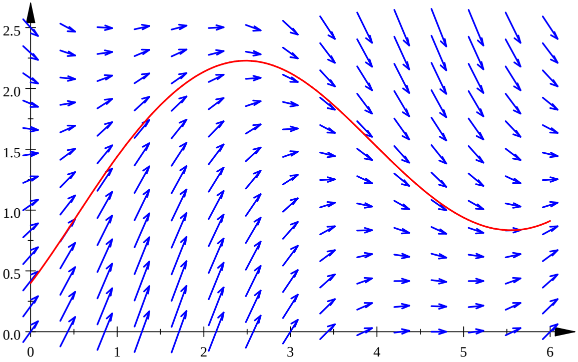
5			
2	4	8	
1	3	6	7

The (recursive) structure of the combinatorial objects encodes the algebraic structure.

Cayley: *On the theory of the analytical forms called trees*

Given a *vector field* $\vec{V}_x$, $x \in \mathbb{R}^d$
Find the *flow* integrating the vector field, *i.e.*, $x(t)$ such that:

$$x(0) = x_0 \qquad \text{and} \qquad x'(t) = \vec{V}_{x(t)}$$



The differential of a vector field

Let $\vec{V}$ and $\vec{U}_1, \ldots, \vec{U}_k$ be vector fields
The k-th differential $D^k\vec{V}$ of $\vec{V}$ is defined by

$$[D^k\vec{V}(\vec{U}_1, \ldots, \vec{U}_k)]_i := \sum_{j_1 \ldots j_k = 1}^{d} \frac{\partial^k [\vec{V}]_i}{\partial x_{j_1} \ldots \partial x_{j_k}} [\vec{U}_1]_{j_1} \ldots [\vec{U}_k]_{j_k} \,,$$

where $[\vec{W}]_i$ denotes the i-th coordinate of the vector field $\vec{W}$

$D^k\vec{V}$ maps k vector fields to one vector field

The differential of a vector field

Let $\vec{V}$ and $\vec{U}_1, \ldots, \vec{U}_k$ be vector fields
The k-th differential $D^k\vec{V}$ of $\vec{V}$ is defined by

$$[D^k\vec{V}(\vec{U}_1, \ldots, \vec{U}_k)]_i := \sum_{j_1 \ldots j_k = 1}^{d} \frac{\partial^k [\vec{V}]_i}{\partial x_{j_1} \ldots \partial x_{j_k}} [\vec{U}_1]_{j_1} \ldots [\vec{U}_k]_{j_k} ,$$

where $[\vec{W}]_i$ denotes the i-th coordinate of the vector field $\vec{W}$

$D^k\vec{V}$ maps k vector fields to one vector field

The derivatives of $x(t)$

$$\frac{dx}{dt}(t) = \vec{V}_{x(t)} = (\vec{V} \circ x)(t)$$

Using the chain rule (derivative of composed functions):

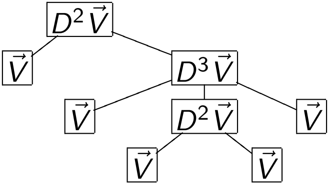
$$\frac{d^2x}{dt^2} = D\vec{V}_x\left(\frac{dx}{dt}\right) = D\vec{V}_x(\vec{V}_x)$$

Third and fourth derivative with implicit $x(t)$:

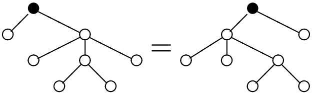
$$\begin{aligned}\frac{d^3x}{dt^3} =& D^2\vec{V}(\vec{V},\vec{V}) + D\vec{V}(D\vec{V}(\vec{V}))\\ \frac{d^4x}{dt^4} =& D^3\vec{V}(\vec{V},\vec{V},\vec{V}) + 3\,D^2\vec{V}(\vec{V},D\vec{V}(\vec{V})) +\\ & D\vec{V}(D^2\vec{V}(\vec{V},\vec{V})) + D\vec{V}(D\vec{V}(D\vec{V}(\vec{V})))\end{aligned}$$

A better notation: expression trees

$$D^2\vec{V}(\vec{V}, D^3\vec{V}(\vec{V}, D^2\vec{V}(\vec{V}, \vec{V}), \vec{V})) =$$



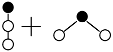
Clairaut's theorem $\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x}$: rooted topological trees

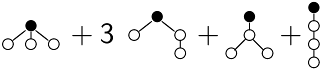


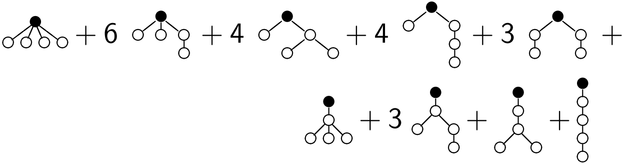
The derivatives of $x(t)$ (continued)

$$\frac{dx}{dt} =$$ 

$$\frac{d^2x}{dt^2} =$$ 

$$\frac{d^3x}{dt^3} =$$ 

$$\frac{d^4x}{dt^4} =$$ 

$$\frac{d^5x}{dt^5} =$$ 

$\dots$

Closed formula

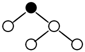
Theorem

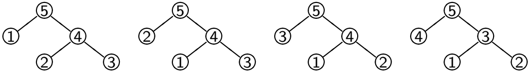
The n-th derivative of $x(t)$ is given by

$$\frac{d^n x}{dt^n} = \sum_{T:\ \text{tree of size } n} c_T\, T$$

where c_T is the number of standard decreasing labelings of T.

Example:

c  $= 4$:

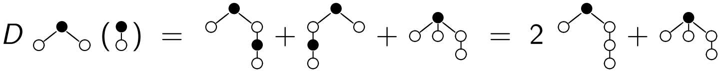


Some algebraic structure

Pre-Lie operation: $(\vec{U}, \vec{V}) \mapsto \vec{U} * \vec{V} := D\vec{U}(\vec{V})$

$$DT_1(T_2) = \sum_{n:\text{node of } T_1} \text{grafting of the root of } T_2 \text{ on } n$$

For example:



Pre-Lie Identity:

$$(x * y) * z - x * (y * z) = (x * z) * y - x * (z * y)$$

Requirements for a typical computation

- A bit of standard combinatorics
- A bit of standard linear algebra
- A bit of standard group theory
- A bit of standard computer algebra
- A bit of standard ...
- And your own little touch

There is nothing like a complete algebraic combinatorics package

Requirements for a typical computation

- A bit of standard combinatorics
- A bit of standard linear algebra
- A bit of standard group theory
- A bit of standard computer algebra
- A bit of standard ...
- And your own little touch

There is nothing like a complete algebraic combinatorics package

Requirements for a typical computation

- A bit of standard combinatorics
- A bit of standard linear algebra
- A bit of standard group theory
- A bit of standard computer algebra
- A bit of standard ...
- And your own little touch

There is nothing like a complete algebraic combinatorics package

Requirements for a typical computation

- A bit of standard combinatorics
- A bit of standard linear algebra
- A bit of standard group theory
- A bit of standard computer algebra
- A bit of standard ...
- And your own little touch

There is nothing like a complete algebraic combinatorics package

Importance of good and consistent large-scale design

*-Combinat is all about

- Sharing

  Remember: this goes both ways!

- Building a community

- Integration: a well organized repository body of code (?)

- Using a bit of computer science
  To organize the code and the community
  To put a bit more math in the machine

- Doing research!

*-Combinat is all about

- Sharing

  Remember: this goes both ways!

- Building a community

- Integration: a well organized repository body of code (?)

- Using a bit of computer science
  To organize the code and the community
  To put a bit more math in the machine

- Doing research!

*-Combinat is all about

- Sharing

  Remember: this goes both ways!

- Building a community

- Integration: a well organized repository body of code (?)

- Using a bit of computer science
  To organize the code and the community
  To put a bit more math in the machine

- Doing research!

*-Combinat is all about

- Sharing

  Remember: this goes both ways!

- Building a community

- Integration: a well organized repository body of code (?)

- Using a bit of computer science
  To organize the code and the community
  To put a bit more math in the machine

- Doing research!

*-Combinat is all about

- Sharing

  Remember: this goes both ways!

- Building a community

- Integration: a well organized repository body of code (?)

- Using a bit of computer science
  To organize the code and the community
  To put a bit more math in the machine

- Doing research!

*-Combinat is all about

- Sharing

  Remember: this goes both ways!

- Building a community
- Integration: a well organized repository body of code (?)
- Using a bit of computer science
  To organize the code and the community
  To put a bit more math in the machine
- Doing research!

2000: state of the art

Algebraic combinatorics packages

- guess, combstruct, gfun, CS (Projet Algo, INRIA)
- SF (Stembridge)
- ACE, μ-EC (Marne-la-Vallée)
- Symmetrica (Bayreuth)
- Rate, ...

Platforms

- Maple / Maxima
- GAP
- Magma
- Axiom / Aldor
- MuPAD

2000: state of the art

Algebraic combinatorics packages

- guess, combstruct, gfun, CS (Projet Algo, INRIA)
- SF (Stembridge)
- ACE, μ-EC (Marne-la-Vallée)
- Symmetrica (Bayreuth)
- Rate, ...

Platforms

- Maple / Maxima
- GAP
- Magma
- Axiom / Aldor
- MuPAD

2007: The MuPAD-Combinat team in action

2008: How to scale further?

- Design thingies, speed, ...
- Architecture, modeling, ...
- Identifying highest return value algorithms

Our community was not broad enough

- Sharing did not yet pay off (for me)!
- Too much management work
- Too much engineering underground work
- Too much off topic work

2008: How to scale further?

- Design thingies, speed, ...
- Architecture, modeling, ...
- Identifying highest return value algorithms

Our community was not broad enough

- Sharing did not yet pay off (for me)!
- Too much management work
- Too much engineering underground work
- Too much off topic work

The platform was not broad enough

- MuPAD is not a widely used language and system
- MuPAD is not open-source

Dealing with it:

- 1999: Negotiation with the MuPAD team
- 2002: Open source computer algebra workshop
- 2005: Sage on radar
- 2006-2007: Axiom meetings → Aldor-Combinat
- 2007: Mike on radar
- February: Sage Days 7
- June 19th: Let's switch!
- September: MuPAD bought out by Mathworks
- Today: Sage Days 10!

The platform was not broad enough

- MuPAD is not a widely used language and system
- MuPAD is not open-source

Dealing with it:

- 1999: Negotiation with the MuPAD team
- 2002: Open source computer algebra workshop
- 2005: Sage on radar
- 2006-2007: Axiom meetings → Aldor-Combinat
- 2007: Mike on radar
- February: Sage Days 7
- June 19th: Let's switch!
- September: MuPAD bought out by Mathworks
- Today: Sage Days 10!

The platform was not broad enough

- MuPAD is not a widely used language and system
- MuPAD is not open-source

Dealing with it:

- 1999: Negotiation with the MuPAD team
- 2002: Open source computer algebra workshop
- 2005: Sage on radar
- 2006-2007: Axiom meetings → Aldor-Combinat
- 2007: Mike on radar
- February: Sage Days 7
- June 19th: Let's switch!
- September: MuPAD bought out by Mathworks
- Today: Sage Days 10!

The platform was not broad enough

- MuPAD is not a widely used language and system
- MuPAD is not open-source

Dealing with it:

- 1999: Negotiation with the MuPAD team
- 2002: Open source computer algebra workshop
- 2005: Sage on radar
- 2006-2007: Axiom meetings → Aldor-Combinat
- 2007: Mike on radar
- February: Sage Days 7
- June 19th: Let's switch!
- September: MuPAD bought out by Mathworks
- Today: Sage Days 10!

The platform was not broad enough

- MuPAD is not a widely used language and system
- MuPAD is not open-source

Dealing with it:

- 1999: Negotiation with the MuPAD team
- 2002: Open source computer algebra workshop
- 2005: Sage on radar
- 2006-2007: Axiom meetings → Aldor-Combinat
- 2007: Mike on radar
- February: Sage Days 7
- June 19th: Let's switch!
- September: MuPAD bought out by Mathworks
- Today: Sage Days 10!

The platform was not broad enough

- MuPAD is not a widely used language and system
- MuPAD is not open-source

Dealing with it:

- 1999: Negotiation with the MuPAD team
- 2002: Open source computer algebra workshop
- 2005: Sage on radar
- 2006-2007: Axiom meetings → Aldor-Combinat
- 2007: Mike on radar
- February: Sage Days 7
- June 19th: Let's switch!
- September: MuPAD bought out by Mathworks
- Today: Sage Days 10!

The platform was not broad enough

- MuPAD is not a widely used language and system
- MuPAD is not open-source

Dealing with it:

- 1999: Negotiation with the MuPAD team
- 2002: Open source computer algebra workshop
- 2005: Sage on radar
- 2006-2007: Axiom meetings → Aldor-Combinat
- 2007: Mike on radar
- February: Sage Days 7
- June 19th: Let's switch!
- September: MuPAD bought out by Mathworks
- Today: Sage Days 10!

The platform was not broad enough

- MuPAD is not a widely used language and system
- MuPAD is not open-source

Dealing with it:

- 1999: Negotiation with the MuPAD team
- 2002: Open source computer algebra workshop
- 2005: Sage on radar
- 2006-2007: Axiom meetings → Aldor-Combinat
- 2007: Mike on radar
- February: Sage Days 7
- June 19th: Let's switch!
- September: MuPAD bought out by Mathworks
- Today: Sage Days 10!

The platform was not broad enough

- MuPAD is not a widely used language and system
- MuPAD is not open-source

Dealing with it:

- 1999: Negotiation with the MuPAD team
- 2002: Open source computer algebra workshop
- 2005: Sage on radar
- 2006-2007: Axiom meetings → Aldor-Combinat
- 2007: Mike on radar
- February: Sage Days 7
- June 19th: Let's switch!
- September: MuPAD bought out by Mathworks
- Today: Sage Days 10!

The platform was not broad enough

- MuPAD is not a widely used language and system
- MuPAD is not open-source

Dealing with it:

- 1999: Negotiation with the MuPAD team
- 2002: Open source computer algebra workshop
- 2005: Sage on radar
- 2006-2007: Axiom meetings → Aldor-Combinat
- 2007: Mike on radar
- February: Sage Days 7
- June 19th: Let's switch!
- September: MuPAD bought out by Mathworks
- Today: Sage Days 10!

The platform was not broad enough

- MuPAD is not a widely used language and system
- MuPAD is not open-source

Dealing with it:

- 1999: Negotiation with the MuPAD team
- 2002: Open source computer algebra workshop
- 2005: Sage on radar
- 2006-2007: Axiom meetings → Aldor-Combinat
- 2007: Mike on radar
- February: Sage Days 7
- June 19th: Let's switch!
- September: MuPAD bought out by Mathworks
- Today: Sage Days 10!

*-Combinat building steps (combinatorics I)

Concentrate on the foundations

- Counting functions / generating series $\surd$?
  lazy Karatsuba product, plethysm, implicit equation, ...
- Generators / iterators, continuations (yield) $\surd$
- Data structures for combinatorial objects
  partitions$\surd$, graphs$\surd$, trees, tableaux$\surd$, permutations$\surd$, ...)
- Lexicographic enumeration of list of integers $\surd$?
- Integral points of a polyhedron $\surd$?
- Decomposable classes / species (Hansen) $\surd$
- Objects mod a group action (Miller) $\surd$?

*-Combinat building steps (combinatorics I)

Concentrate on the foundations

- Counting functions / generating series √?
  lazy Karatsuba product, plethysm, implicit equation, ...
- Generators / iterators, continuations (yield) √
- Data structures for combinatorial objects
  partitions√, graphs√, trees, tableaux√, permutations√, ...)
- Lexicographic enumeration of list of integers √?
- Integral points of a polyhedron √?
- Decomposable classes / species (Hansen) √
- Objects mod a group action (Miller) √?

∗-Combinat building steps (combinatorics I)

Concentrate on the foundations

- Counting functions / generating series √?
  lazy Karatsuba product, plethysm, implicit equation, ...
- Generators / iterators, continuations (yield) √
- Data structures for combinatorial objects
  partitions√, graphs√, trees, tableaux√, permutations√, ...)
- Lexicographic enumeration of list of integers √?
- Integral points of a polyhedron √?
- Decomposable classes / species (Hansen) √
- Objects mod a group action (Miller) √?

*-Combinat building steps (combinatorics I)

Concentrate on the foundations

- Counting functions / generating series $\surd$?
  lazy Karatsuba product, plethysm, implicit equation, ...
- Generators / iterators, continuations (yield) $\surd$
- Data structures for combinatorial objects
  partitions$\surd$, graphs$\surd$, trees, tableaux$\surd$, permutations$\surd$, ...)
- Lexicographic enumeration of list of integers $\surd$?
- Integral points of a polyhedron $\surd$?
- Decomposable classes / species (Hansen) $\surd$
- Objects mod a group action (Miller) $\surd$?

*-Combinat building steps (combinatorics I)

Concentrate on the foundations

- Counting functions / generating series $\surd$?
  lazy Karatsuba product, plethysm, implicit equation, ...
- Generators / iterators, continuations (yield) $\surd$
- Data structures for combinatorial objects
  partitions$\surd$, graphs$\surd$, trees, tableaux$\surd$, permutations$\surd$, ...)
- Lexicographic enumeration of list of integers $\surd$?
- Integral points of a polyhedron $\surd$?
- Decomposable classes / species (Hansen) $\surd$
- Objects mod a group action (Miller) $\surd$?

$*$-Combinat building steps (combinatorics I)

Concentrate on the foundations

- Counting functions / generating series $\surd$?
  lazy Karatsuba product, plethysm, implicit equation, ...
- Generators / iterators, continuations (yield) $\surd$
- Data structures for combinatorial objects
  partitions$\surd$, graphs$\surd$, trees, tableaux$\surd$, permutations$\surd$, ...)
- Lexicographic enumeration of list of integers $\surd$?
- Integral points of a polyhedron $\surd$?
- Decomposable classes / species (Hansen) $\surd$
- Objects mod a group action (Miller) $\surd$?

∗-Combinat building steps (combinatorics I)

Concentrate on the foundations

- Counting functions / generating series $\surd$?
  lazy Karatsuba product, plethysm, implicit equation, ...
- Generators / iterators, continuations (yield) $\surd$
- Data structures for combinatorial objects
  partitions$\surd$, graphs$\surd$, trees, tableaux$\surd$, permutations$\surd$, ...)
- Lexicographic enumeration of list of integers $\surd$?
- Integral points of a polyhedron $\surd$?
- Decomposable classes / species (Hansen) $\surd$
- Objects mod a group action (Miller) $\surd$?

*-Combinat building steps (combinatorics II)

- Trees (Saliola)
- Posets and linear extensions (Saliola, Hivert, Nzeutchap) √?
- Words (Saliola)
- Root systems, crystals, Weyl groups√
  (Schilling, Bandlow, Bump, Walker, Hansen, T)
- (Weighted) Automatons (Saliola, Laugerotte)
- Guess (Rubey)
- Packages from the Algo project!

*-Combinat building steps (combinatorics II)

- Trees (Saliola)
- Posets and linear extensions (Saliola, Hivert, Nzeutchap) √?
- Words (Saliola)
- Root systems, crystals, Weyl groups√
  (Schilling, Bandlow, Bump, Walker, Hansen, T)
- (Weighted) Automatons (Saliola, Laugerotte)
- Guess (Rubey)
- Packages from the Algo project!

*-Combinat building steps (combinatorics II)

- Trees (Saliola)
- Posets and linear extensions (Saliola, Hivert, Nzeutchap) $\surd$?
- Words (Saliola)
- Root systems, crystals, Weyl groups$\surd$ (Schilling, Bandlow, Bump, Walker, Hansen, T)
- (Weighted) Automatons (Saliola, Laugerotte)
- Guess (Rubey)
- Packages from the Algo project!

*-Combinat building steps (combinatorics II)

- Trees (Saliola)
- Posets and linear extensions (Saliola, Hivert, Nzeutchap) $\surd$?
- Words (Saliola)
- Root systems, crystals, Weyl groups$\surd$
  (Schilling, Bandlow, Bump, Walker, Hansen, T)
- (Weighted) Automatons (Saliola, Laugerotte)
- Guess (Rubey)
- Packages from the Algo project!

*-Combinat building steps (combinatorics II)

- Trees (Saliola)
- Posets and linear extensions (Saliola, Hivert, Nzeutchap) √?
- Words (Saliola)
- Root systems, crystals, Weyl groups√
  (Schilling, Bandlow, Bump, Walker, Hansen, T)
- (Weighted) Automatons (Saliola, Laugerotte)
- Guess (Rubey)
- Packages from the Algo project!

∗-Combinat building steps (combinatorics II)

- Trees (Saliola)
- Posets and linear extensions (Saliola, Hivert, Nzeutchap) $\surd$?
- Words (Saliola)
- Root systems, crystals, Weyl groups$\surd$ (Schilling, Bandlow, Bump, Walker, Hansen, T)
- (Weighted) Automatons (Saliola, Laugerotte)
- Guess (Rubey)
- Packages from the Algo project!

*-Combinat building steps (combinatorics II)

- Trees (Saliola)
- Posets and linear extensions (Saliola, Hivert, Nzeutchap) √?
- Words (Saliola)
- Root systems, crystals, Weyl groups√
  (Schilling, Bandlow, Bump, Walker, Hansen, T)
- (Weighted) Automatons (Saliola, Laugerotte)
- Guess (Rubey)
- Packages from the Algo project!

*-Combinat building steps (algebra)

*-Combinat building steps (algebra I)

Trivial code that solves trivial problems is precious

- `FreeModule(Combinatorial Class, Coefficient Ring)`$\surd$
- Categories: (Hopf)AlgebraWithBasis and friends
  Unification with polynomials, Weyl & Ore algebras, ...
- Seamless linear algebra (Linbox)
  Hom(..., ...)
- Spaces with several representations, ...
- Functors: tensor products, tensor, exterior, and symmetric algebra, submodules, quotients, ...

*-Combinat building steps (algebra I)

Trivial code that solves trivial problems is precious

- `FreeModule(Combinatorial Class, Coefficient Ring)` $\surd$
- Categories: (Hopf)AlgebraWithBasis and friends
  Unification with polynomials, Weyl & Ore algebras, ...
- Seamless linear algebra (Linbox)
  Hom(..., ...)
- Spaces with several representations, ...
- Functors: tensor products, tensor, exterior, and symmetric algebra, submodules, quotients, ...

∗-Combinat building steps (algebra I)

Trivial code that solves trivial problems is precious

- `FreeModule(Combinatorial Class, Coefficient Ring)` $\surd$
- Categories: (Hopf)AlgebraWithBasis and friends
  Unification with polynomials, Weyl & Ore algebras, ...
- Seamless linear algebra (Linbox)
  Hom(..., ...)
- Spaces with several representations, ...
- Functors: tensor products, tensor, exterior, and symmetric algebra, submodules, quotients, ...

*-Combinat building steps (algebra I)

Trivial code that solves trivial problems is precious

- `FreeModule(Combinatorial Class, Coefficient Ring)` $\surd$
- Categories: (Hopf)AlgebraWithBasis and friends
  Unification with polynomials, Weyl & Ore algebras, ...
- Seamless linear algebra (Linbox)
  Hom(..., ...)
- Spaces with several representations, ...
- Functors: tensor products, tensor, exterior, and symmetric algebra, submodules, quotients, ...

*-Combinat building steps (algebra I)

Trivial code that solves trivial problems is precious

- `FreeModule(Combinatorial Class, Coefficient Ring)` $\surd$
- Categories: (Hopf)AlgebraWithBasis and friends
  Unification with polynomials, Weyl & Ore algebras, ...
- Seamless linear algebra (Linbox)
  Hom(..., ...)
- Spaces with several representations, ...
- Functors: tensor products, tensor, exterior, and symmetric algebra, submodules, quotients, ...

∗-Combinat building steps (algebra I)

Trivial code that solves trivial problems is precious

- `FreeModule(Combinatorial Class, Coefficient Ring)` $\surd$
- Categories: (Hopf)AlgebraWithBasis and friends
  Unification with polynomials, Weyl & Ore algebras, ...
- Seamless linear algebra (Linbox)
  Hom(..., ...)
- Spaces with several representations, ...
- Functors: tensor products, tensor, exterior, and symmetric algebra, submodules, quotients, ...

*-Combinat building steps (algebra II)

- Symmetric functions (Hansen) √?
- Non commutative and quasi-symmetric functions (Tevlin, Luoto)
- Database of Hopf algebras
- Operads (Chapoton, Saliola)
- Permutation groups (GAP) √
- Quivers (Lemeur?)
- Generic Gröbner/Involutive elimination tools
- Representation Theory (Hivert)
- Crystals, Weyl groups, ... √?

*-Combinat building steps (algebra II)

- Symmetric functions (Hansen) √?
- Non commutative and quasi-symmetric functions (Tevlin, Luoto)
- Database of Hopf algebras
- Operads (Chapoton, Saliola)
- Permutation groups (GAP) √
- Quivers (Lemeur?)
- Generic Gröbner/Involutive elimination tools
- Representation Theory (Hivert)
- Crystals, Weyl groups, ... √?

∗-Combinat building steps (algebra II)

- Symmetric functions (Hansen) √?
- Non commutative and quasi-symmetric functions (Tevlin, Luoto)
- Database of Hopf algebras
- Operads (Chapoton, Saliola)
- Permutation groups (GAP) √
- Quivers (Lemeur?)
- Generic Gröbner/Involutive elimination tools
- Representation Theory (Hivert)
- Crystals, Weyl groups, ... √?

∗-Combinat building steps (algebra II)

- Symmetric functions (Hansen) √?
- Non commutative and quasi-symmetric functions (Tevlin, Luoto)
- Database of Hopf algebras
- Operads (Chapoton, Saliola)
- Permutation groups (GAP) √
- Quivers (Lemeur?)
- Generic Gröbner/Involutive elimination tools
- Representation Theory (Hivert)
- Crystals, Weyl groups, ... √?

*-Combinat building steps (algebra II)

- Symmetric functions (Hansen) $\surd$?
- Non commutative and quasi-symmetric functions (Tevlin, Luoto)
- Database of Hopf algebras
- Operads (Chapoton, Saliola)
- Permutation groups (GAP) $\surd$
- Quivers (Lemeur?)
- Generic Gröbner/Involutive elimination tools
- Representation Theory (Hivert)
- Crystals, Weyl groups, ... $\surd$?

∗-Combinat building steps (algebra II)

- Symmetric functions (Hansen) √?
- Non commutative and quasi-symmetric functions (Tevlin, Luoto)
- Database of Hopf algebras
- Operads (Chapoton, Saliola)
- Permutation groups (GAP) √
- Quivers (Lemeur?)
- Generic Gröbner/Involutive elimination tools
- Representation Theory (Hivert)
- Crystals, Weyl groups, ... √?

*-Combinat building steps (algebra II)

- Symmetric functions (Hansen) $\surd$?
- Non commutative and quasi-symmetric functions (Tevlin, Luoto)
- Database of Hopf algebras
- Operads (Chapoton, Saliola)
- Permutation groups (GAP) $\surd$
- Quivers (Lemeur?)
- Generic Gröbner/Involutive elimination tools
- Representation Theory (Hivert)
- Crystals, Weyl groups, ... $\surd$?

∗-Combinat building steps (algebra II)

- Symmetric functions (Hansen) $\surd$?
- Non commutative and quasi-symmetric functions (Tevlin, Luoto)
- Database of Hopf algebras
- Operads (Chapoton, Saliola)
- Permutation groups (GAP) $\surd$
- Quivers (Lemeur?)
- Generic Gröbner/Involutive elimination tools
- Representation Theory (Hivert)
- Crystals, Weyl groups, ... $\surd$?

∗-Combinat building steps (algebra II)

- Symmetric functions (Hansen) $\surd$?
- Non commutative and quasi-symmetric functions (Tevlin, Luoto)
- Database of Hopf algebras
- Operads (Chapoton, Saliola)
- Permutation groups (GAP) $\surd$
- Quivers (Lemeur?)
- Generic Gröbner/Involutive elimination tools
- Representation Theory (Hivert)
- Crystals, Weyl groups, ... $\surd$?

∗-Combinat building steps (algebra II)

- Symmetric functions (Hansen) $\surd$?
- Non commutative and quasi-symmetric functions (Tevlin, Luoto)
- Database of Hopf algebras
- Operads (Chapoton, Saliola)
- Permutation groups (GAP) $\surd$
- Quivers (Lemeur?)
- Generic Gröbner/Involutive elimination tools
- Representation Theory (Hivert)
- Crystals, Weyl groups, ... $\surd$?

*-Combinat building steps (coercion)

- We need automatic coercion (implicit conversions)
- At interactive-level *and* inside code
- Possibly with > 1000 domains simultaneously

A good test case for the new Sage coercion model!

Want to join?

mupad-combinat.sf.net

wiki.sagemath.org/combinat