# Infrastructure pour du code générique dans SageMath : catégories, axiomes, constructions, ...

Nicolas M. Thiéry

LRI, Université Paris Sud 11

30 mai 2016, Séminaire «Computations and Proofs», SpecFun

# Résumé

Le logiciel libre SageMath permet de manipuler des milliers d'objets mathématiques différents avec des dizaines de milliers de fonctions. Un système de cette taille requiert une infrastructure permettant l'écriture et l'organisation de code, de documentation et de tests génériques s'appliquant uniformément à tous les objets satisfaisant certaines propriétés.
Dans cet exposé, nous décrirons l'infrastructure implantée dans SageMath – qui s'appuie sur la programmation objet classique en Python, avec des mécanismes pour passer à l'échelle (classes dynamiques, mixins, ...) s'appuyant sur la forte sémantique disponible (catégories, axiomes, constructions) – et la comparerons avec ce qui est fait dans d'autres systèmes.

# SageMath : sagemath.org

## Mission

« *Créer une alternative libre et viable à Maple$^{TM}$, Mathematica$^{TM}$, Magma$^{TM}$ et MATLAB$^{TM}$* »

## En quelques chiffres

- Initié en 2004 par William Stein
- Basé sur NumPy, SciPy, matplotlib, Sympy, Maxima, GAP, FLINT, Singular, R, ...
- 40000 utilisateurs ?
- 200 contributeurs

# SageMath : sagemath.org

## Mission

« *Créer une alternative libre et viable à Maple$^{TM}$, Mathematica$^{TM}$, Magma$^{TM}$ et MATLAB$^{TM}$* »

## En quelques chiffres

- Initié en 2004 par William Stein
- Basé sur NumPy, SciPy, matplotlib, Sympy, Maxima, GAP, FLINT, Singular, R, ...
- 40000 utilisateurs ?
- 200 contributeurs

# Sage : un logiciel de calcul mathématique

Nombres : 42, $\frac{7}{9}$, $\frac{I+sqrt(3)}{2}$, $\pi$, 2.718281828459045235360287471?

Matrices : $\begin{pmatrix} 4 & -1 & 1 & -1 \\ -1 & 2 & -1 & -1 \\ 0 & 5 & 1 & 3 \end{pmatrix}$

Polynômes : $-9x^8 + x^7 + x^6 - 13x^5 - x^3 - 3x^2 - 8x + 4$

Séries : $1 + 1x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4 + \frac{1}{120}x^5 + \cdots$

Corps finis, extensions algébriques, courbes elliptiques, ...

Expressions symboliques, équations, ...

# Sage : un logiciel de calcul mathématique

Nombres : 42, $\frac{7}{9}$, $\frac{I+sqrt(3)}{2}$, $\pi$, 2.71828182845904523536028747?

Matrices : $\begin{pmatrix} 4 & -1 & 1 & -1 \\ -1 & 2 & -1 & -1 \\ 0 & 5 & 1 & 3 \end{pmatrix}$

Polynômes : $-9x^8 + x^7 + x^6 - 13x^5 - x^3 - 3x^2 - 8x + 4$

Séries : $1 + 1x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4 + \frac{1}{120}x^5 + \cdots$

Corps finis, extensions algébriques, courbes elliptiques, ...

Expressions symboliques, équations, ...

# Sage : un logiciel de calcul mathématique

Nombres : $42$, $\frac{7}{9}$, $\frac{I+sqrt(3)}{2}$, $\pi$, $2.718281828459045235360287471?$

Matrices : $\begin{pmatrix} 4 & -1 & 1 & -1 \\ -1 & 2 & -1 & -1 \\ 0 & 5 & 1 & 3 \end{pmatrix}$

Polynômes : $-9x^8 + x^7 + x^6 - 13x^5 - x^3 - 3x^2 - 8x + 4$

Séries : $1 + 1x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4 + \frac{1}{120}x^5 + \cdots$

Corps finis, extensions algébriques, courbes elliptiques, ...

Expressions symboliques, équations, ...

# Sage : un logiciel de calcul mathématique

Nombres : 42, $\frac{7}{9}$, $\frac{I+sqrt(3)}{2}$, $\pi$, 2.71828182845904523536028747?

Matrices : $\begin{pmatrix} 4 & -1 & 1 & -1 \\ -1 & 2 & -1 & -1 \\ 0 & 5 & 1 & 3 \end{pmatrix}$

Polynômes : $-9x^8 + x^7 + x^6 - 13x^5 - x^3 - 3x^2 - 8x + 4$

Séries : $1 + 1x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4 + \frac{1}{120}x^5 + \cdots$

Corps finis, extensions algébriques, courbes elliptiques, ...

Expressions symboliques, équations, ...

# Sage : un logiciel de calcul mathématique

Nombres : 42, $\frac{7}{9}$, $\frac{I+sqrt(3)}{2}$, $\pi$, 2.718281828459045235360287471?

Matrices : $\begin{pmatrix} 4 & -1 & 1 & -1 \\ -1 & 2 & -1 & -1 \\ 0 & 5 & 1 & 3 \end{pmatrix}$

Polynômes : $-9x^8 + x^7 + x^6 - 13x^5 - x^3 - 3x^2 - 8x + 4$

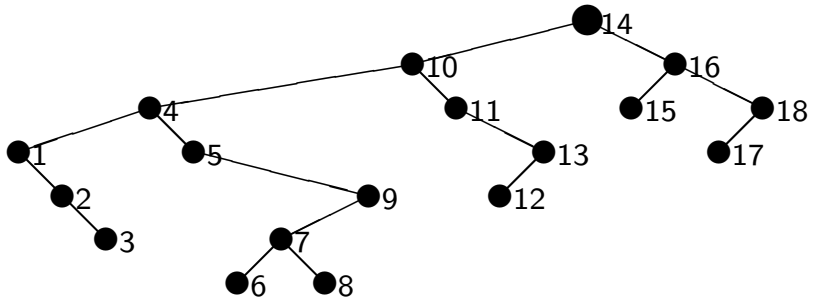
Séries : $1 + 1x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4 + \frac{1}{120}x^5 + \cdots$

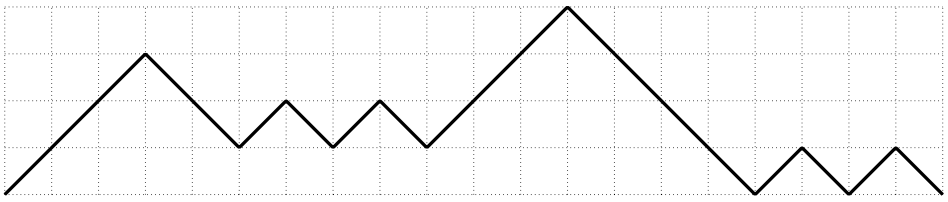
Corps finis, extensions algébriques, courbes elliptiques, ...

Expressions symboliques, équations, ...

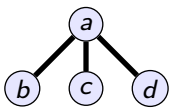
# Objets combinatoires

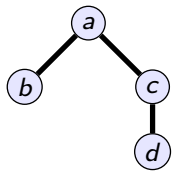
| | | | |
|---|---|---|---|
| 1 | 3 | 4 | 7 |
| 2 | 5 | 6 | |
| 8 | | | |

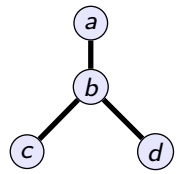


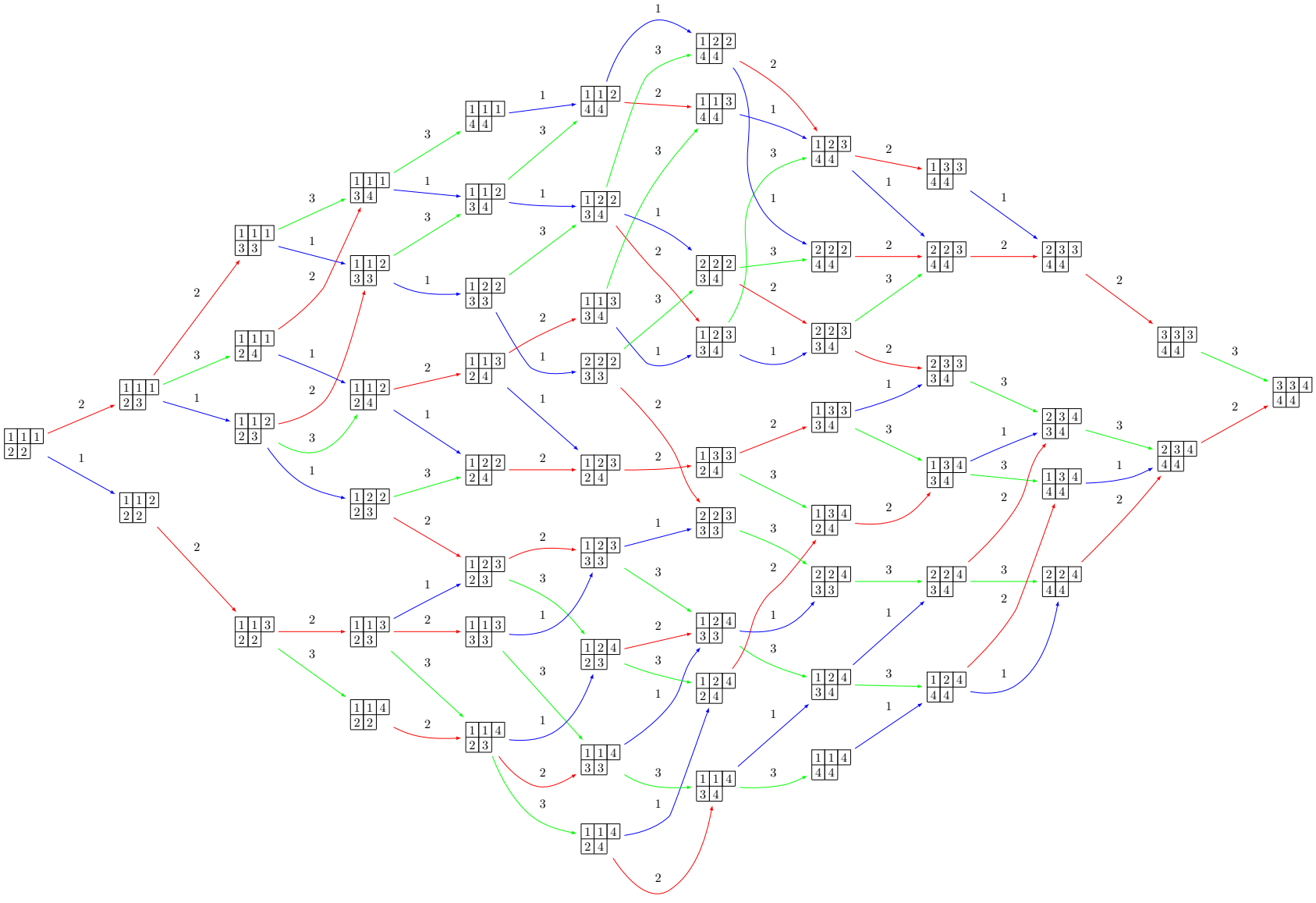
$$0100101001001010010100100101001001010010\cdots$$

$$\frac{\frac{1}{6}q^2-\frac{1}{6}q}{q^5+2q^4+3q^3+3q^2+2q+1}$$ 

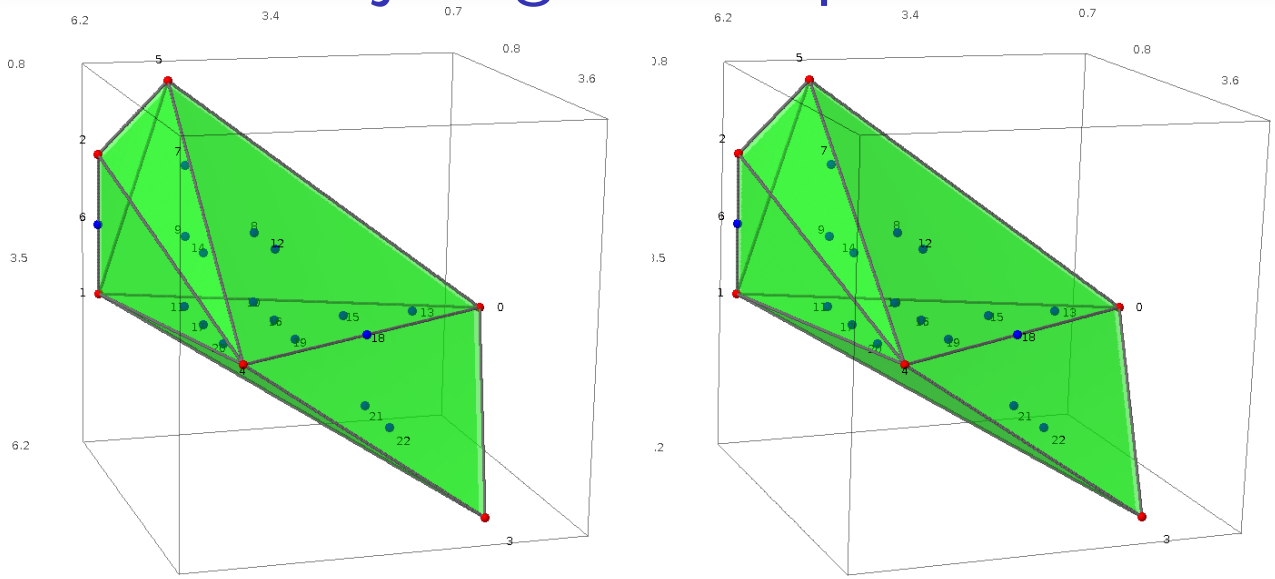
$$+\frac{q^2}{q^5+2q^4+3q^3+3q^2+2q+1}$$ 

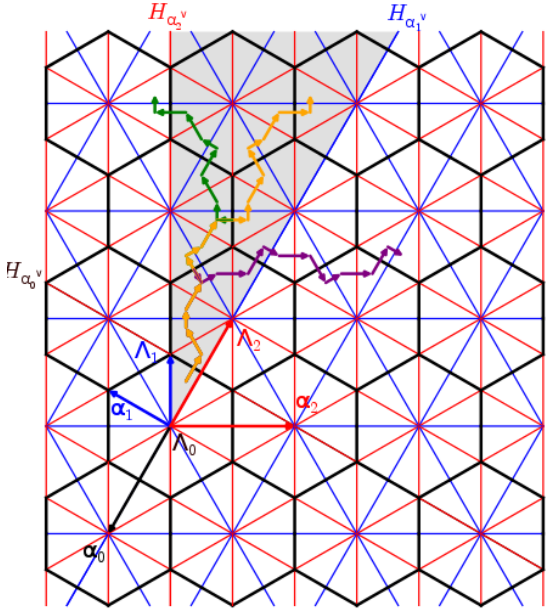
$$+\frac{\frac{1}{2}q}{q^4+q^3+2q^2+q+1}$$ 

# Graphes



# Objets géométriques





# Sage : une bibliothèque large d'objets mathématiques et d'algorithmes

- 1.5G lignes de code/doc/tests (Python/Cython) plus dépendances
- 1000+ types d'objets
- 20000+ méthodes and fonctions
- 300 contributeurs

## Problèmes

- Comment structurer cette bibliothèque ?
- Comment guider l'utilisateur
- Comment garantir la cohérence, la robustesse ?
- Comment éviter les duplications ?

# Exemple : Exponentiation rapide I

```
sage : m = 3
sage : m^8 == m*m*m*m*m*m*m*m == ((m^2)^2)^2
True
```

```
sage : m = random_matrix(QQ, 4)
sage : m^8 == m*m*m*m*m*m*m*m == ((m^2)^2)^2
True
```

- Complexité $O(\log(n))$ au lieu de $O(n)$ pour calculer $x^n$
- On voudrait une implantation *générique* unique !

# Exemple : Exponentiation rapide II

## Contexte algébrique

- *Semigroupe* : ensemble $S$ muni d'une loi $*$ associative.
- Les entiers forment un semigroupe
- Les matrices carrées forment un semigroupe

## On veut :

- Implanter pow_exp(x,k)
- Spécifier au système que
  - si $x$ est un *élément* d'un semigroupe
  - alors $x^k$ peut être calculé avec pow_exp(x,k)

- Si $x$ est un élément d'un groupe ?
- Si $x$ est un élément d'un anneau en petite caractéristique ?

# Exemple : Exponentiation rapide II

## Contexte algébrique

- *Semigroupe* : ensemble $S$ muni d'une loi $*$ associative.
- Les entiers forment un semigroupe
- Les matrices carrées forment un semigroupe

## On veut :

- Implanter pow_exp(x,k)
- Spécifier au système que
  - si $x$ est un *élément* d'un semigroupe
  - alors $x^k$ peut être calculé avec pow_exp(x,k)

- Si $x$ est un élément d'un groupe ?
- Si $x$ est un élément d'un anneau en petite caractéristique ?

# Exemple : Exponentiation rapide II

## Contexte algébrique

- *Semigroupe* : ensemble $S$ muni d'une loi $*$ associative.
- Les entiers forment un semigroupe
- Les matrices carrées forment un semigroupe

## On veut :

- Implanter pow_exp(x,k)
- Spécifier au système que
  - si $x$ est un *élément* d'un semigroupe
  - alors $x^k$ peut être calculé avec pow_exp(x,k)

- Si $x$ est un élément d'un groupe ?
- Si $x$ est un élément d'un anneau en petite caractéristique ?

# Mécanismes de sélection

## On veut

- Concevoir une hiérarchie de contextes
- Implanter des fonctions génériques
- Spécifier dans quel contexte elles sont valables
- Spécifier dans quel(s) contexte(s) est chaque objet

## Besoin d'un *mécanisme de sélection*

Pour résoudre l'appel `f(x)` en sélectionnant l'implantation de l'opération `f` la plus spécifique

# Mécanismes de sélection

## On veut

- Concevoir une hiérarchie de contextes
- Implanter des fonctions génériques
- Spécifier dans quel contexte elles sont valables
- Spécifier dans quel(s) contexte(s) est chaque objet

## Besoin d'un *mécanisme de sélection*

Pour résoudre l'appel `f(x)` en sélectionnant l'implantation de l'opération `f` la plus spécifique

# Hiérarchie de contextes pour les maths

## En général

Problème difficile : isoler les bons concepts métiers ?

## En maths

- Un petit nombre de concepts fondamentaux : opérations, axiomes
- Concepts connus de la plupart des utilisateurs
- La richesse est dans la combinaison de ceux-ci : Corps : $+$, $*$, associatif, commutatif, distributif, ...

# Hiérarchie de contextes pour les maths

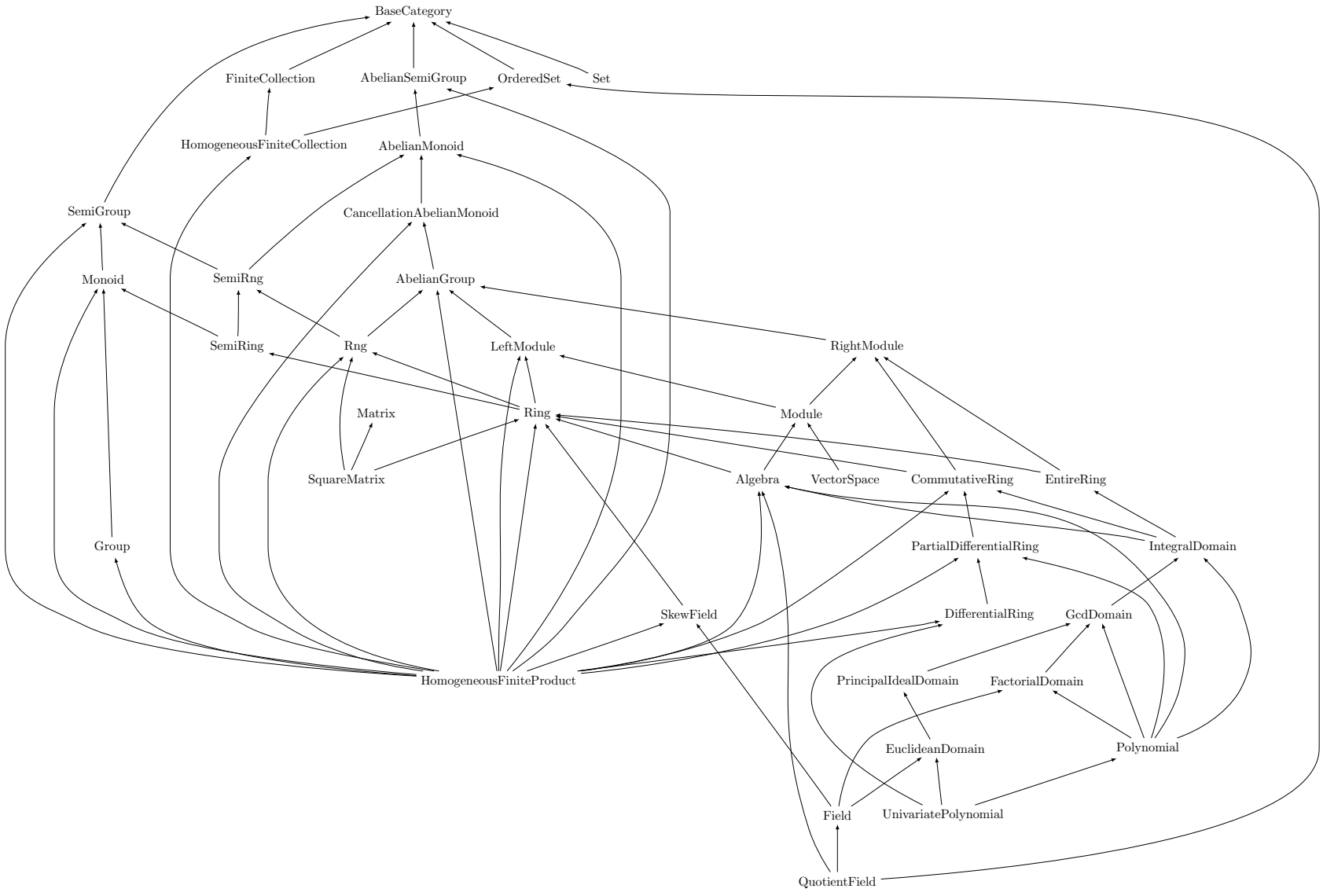
## En général

Problème difficile : isoler les bons concepts métiers ?

## En maths

- Un petit nombre de concepts fondamentaux : opérations, axiomes
- Concepts connus de la plupart des utilisateurs
- La richesse est dans la combinaison de ceux-ci : Corps : $+$, *, associatif, commutatif, distributif, ...

# Une hiérarchie de contextes basée sur les catégories




Hiérarchie robuste car reposant sur un siècle d'algèbre abstraite.

# Quelques pioniers 1980- I

## Axiom, Aldor, MuPAD

- Langage spécifique
- Mécanisme de sélection : programmation objet
- Un contexte est modélisé par une classe abstraite C
- C : étagère contenant toutes les implantations valides dans ce contexte
- Hiérarchie de classes abstraites modélisant les catégories

### Exemple

```
category Semigroups :
    category Magmas;

    intpow := proc(x, k) ...
    // other methods
```

# Quelques pioniers 1980- I

## Axiom, Aldor, MuPAD

- Langage spécifique
- Mécanisme de sélection : programmation objet
- Un contexte est modélisé par une classe abstraite C
- C : étagère contenant toutes les implantations valides dans ce contexte
- Hiérarchie de classes abstraites modélisant les catégories

### Exemple

```
category Semigroups :
    category Magmas;

    intpow := proc(x, k) ...
    // other methods
```

# Quelques pioniers 1980- I

## Axiom, Aldor, MuPAD

- Langage spécifique
- Mécanisme de sélection : programmation objet
- Un contexte est modélisé par une classe abstraite C
- C : étagère contenant toutes les implantations valides dans ce contexte
- Hiérarchie de classes abstraites modélisant les catégories

## Exemple

```
category Semigroups :
    category Magmas;

    intpow := proc(x, k) ...
    // other methods
```

# Quelques pioniers 1980- II

## GAP

- Langage spécifique
- Filtres : `IsMagma(G)`, `IsAssociative(G)`, ...
- `InstallMethod(Operation, filters, method)`
- Sélection selon les filtres connus pour être vrais
- *Modélisation implicite de la hiérarchie*

### Exemple

```
powExp := function(n, k) ...

InstallMethod(pow, [IsMagma, IsAssociative],
              powExp)
```

# Quelques pioniers 1980- II

## GAP

- Langage spécifique
- Filtres : `IsMagma(G)`, `IsAssociative(G)`, ...
- `InstallMethod(Operation, filters, method)`
- Sélection selon les filtres connus pour être vrais
- *Modélisation implicite de la hiérarchie*

### Exemple

```
powExp := function(n, k) ...

InstallMethod(pow, [IsMagma, IsAssociative],
              powExp)
```

# Quelques pioniers 1980- II

## GAP

- Langage spécifique
- Filtres : `IsMagma(G)`, `IsAssociative(G)`, ...
- `InstallMethod(Operation, filters, method)`
- Sélection selon les filtres connus pour être vrais
- *Modélisation implicite de la hiérarchie*

## Exemple

```
powExp := function(n, k) ...

InstallMethod(pow, [IsMagma, IsAssociative],
                    powExp)
```

# Implantation dans Sage (2008-)

## Choix stratégiques

- Langage standard (Python)
- Mécanisme de sélection : programmation objet

## Contraintes

- Compilation partielle (Cython), sérialization
- Héritage multiple avec Python / Cython
- Passage à l'échelle

## Spécificités

- Distinction Élément/Parent (à la Magma)
- Morphismes
- Constructions fonctorielles
- Axiomes

# Implantation dans Sage (2008-)

## Choix stratégiques

- Langage standard (Python)
- Mécanisme de sélection : programmation objet

## Contraintes

- Compilation partielle (Cython), sérialization
- Héritage multiple avec Python / Cython
- Passage à l'échelle

## Spécificités

- Distinction Élément/Parent (à la Magma)
- Morphismes
- Constructions fonctorielles
- Axiomes

# Implantation dans Sage (2008-)

## Choix stratégiques

- Langage standard (Python)
- Mécanisme de sélection : programmation objet

## Contraintes

- Compilation partielle (Cython), sérialization
- Héritage multiple avec Python / Cython
- Passage à l'échelle

## Spécificités

- Distinction Élément/Parent (à la Magma)
- Morphismes
- Constructions fonctorielles
- Axiomes

# Implantation dans Sage (2008-)

## Choix stratégiques

- Langage standard (Python)
- Mécanisme de sélection : programmation objet

## Contraintes

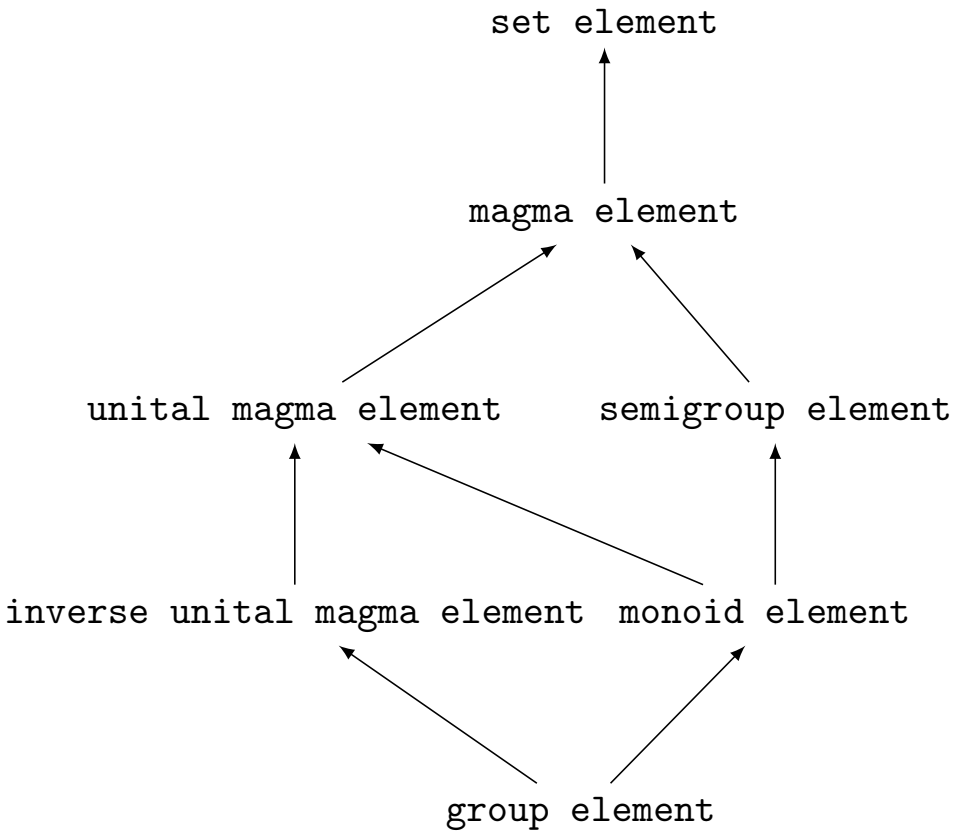
- Compilation partielle (Cython), sérialization
- Héritage multiple avec Python / Cython
- Passage à l'échelle

## Spécificités

- Distinction Élément/Parent (à la Magma)
- Morphismes
- Constructions fonctorielles
- Axiomes

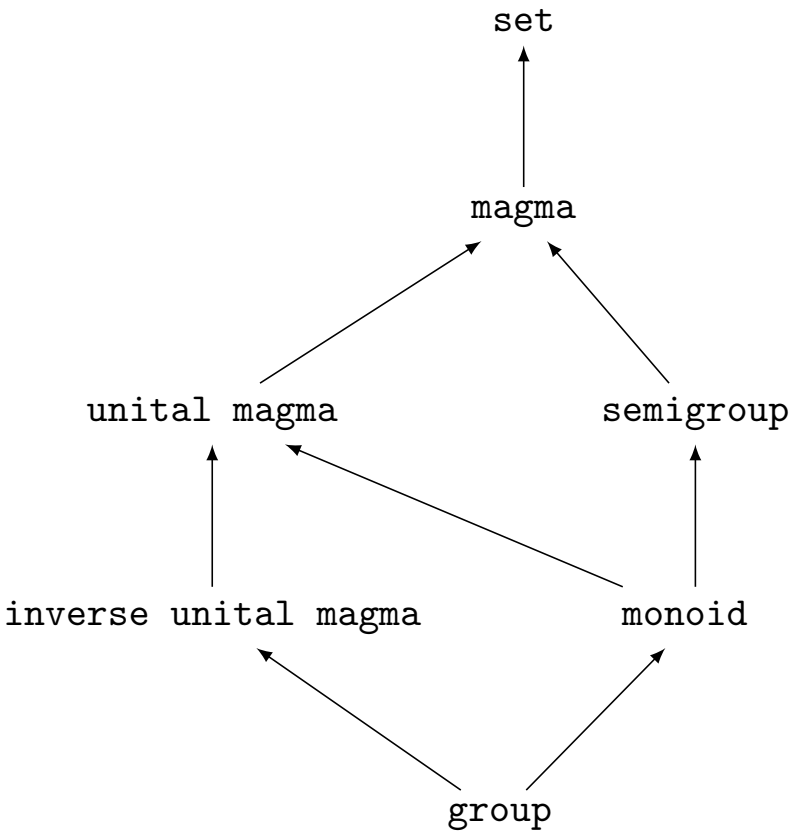
# Parent, éléments, morphismes, catégories

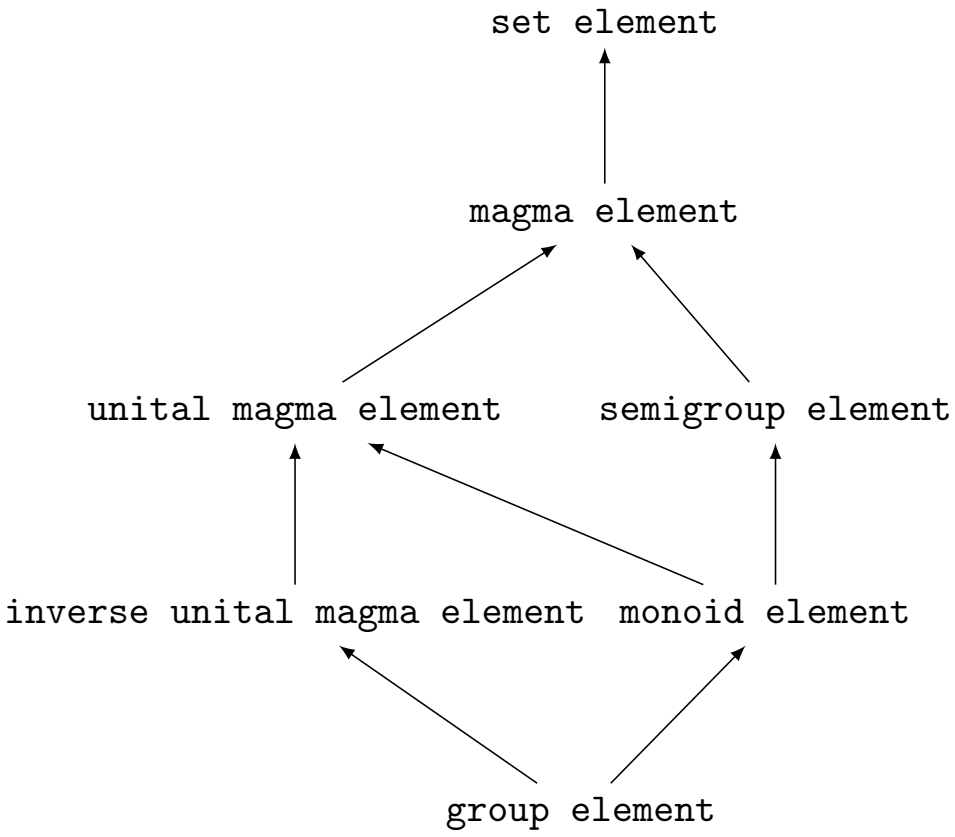
Demo



# Parent, éléments, morphismes, catégories

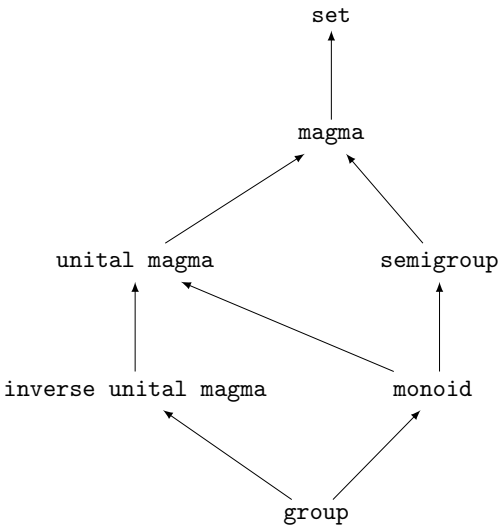
Demo

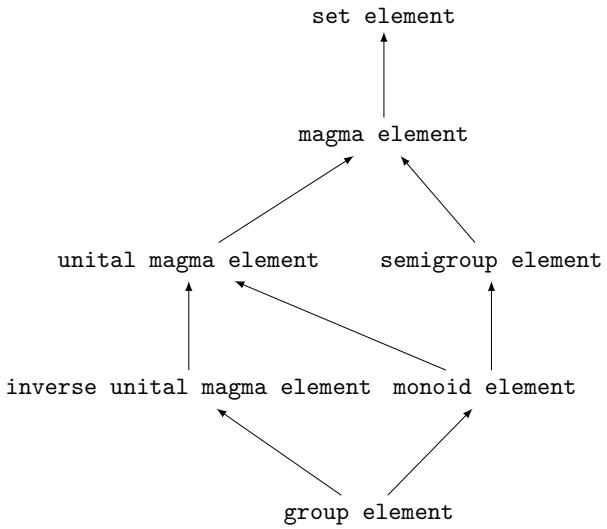


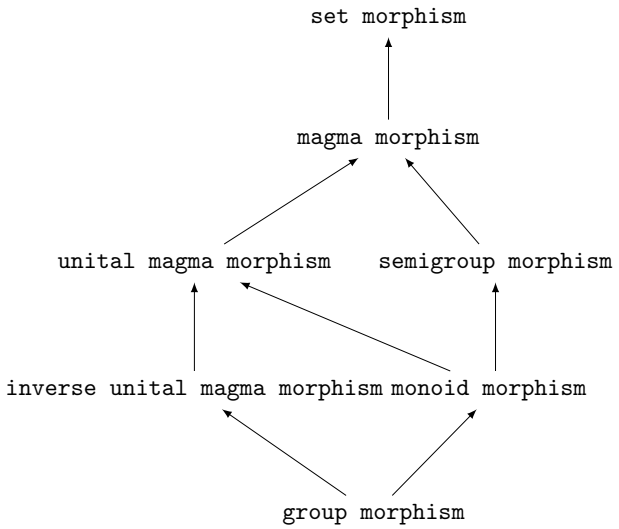



# Parent, éléments, morphismes, catégories

## Demo



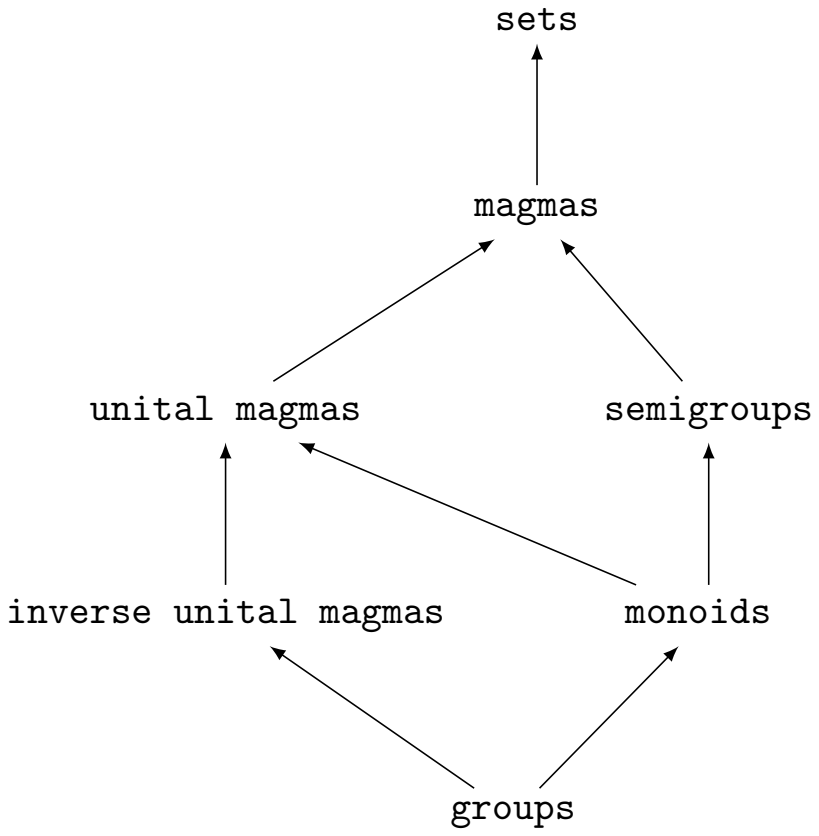



# Passage à l'échelle ?

Démo

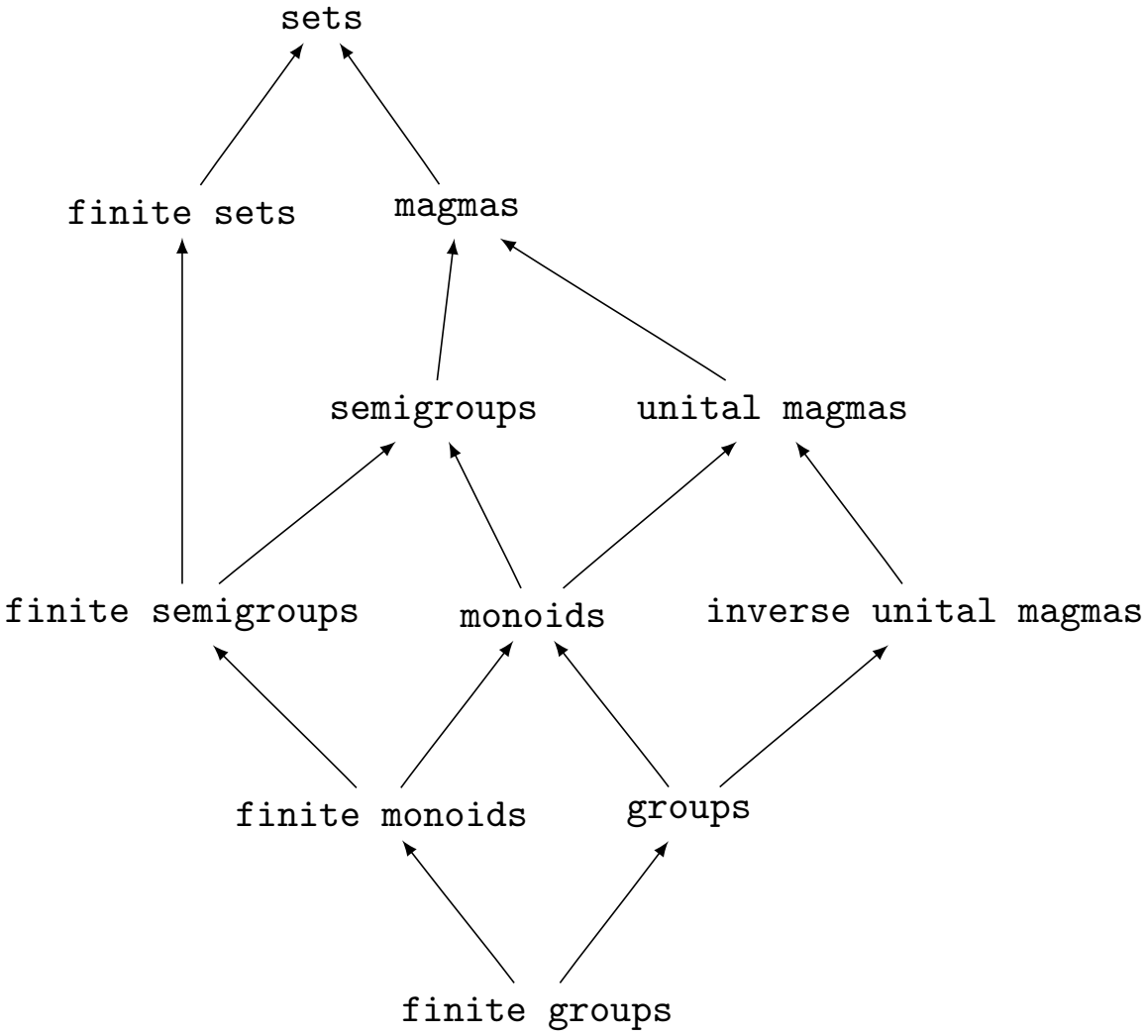
# Passage à l'échelle ?

Catégories pour les groupes :



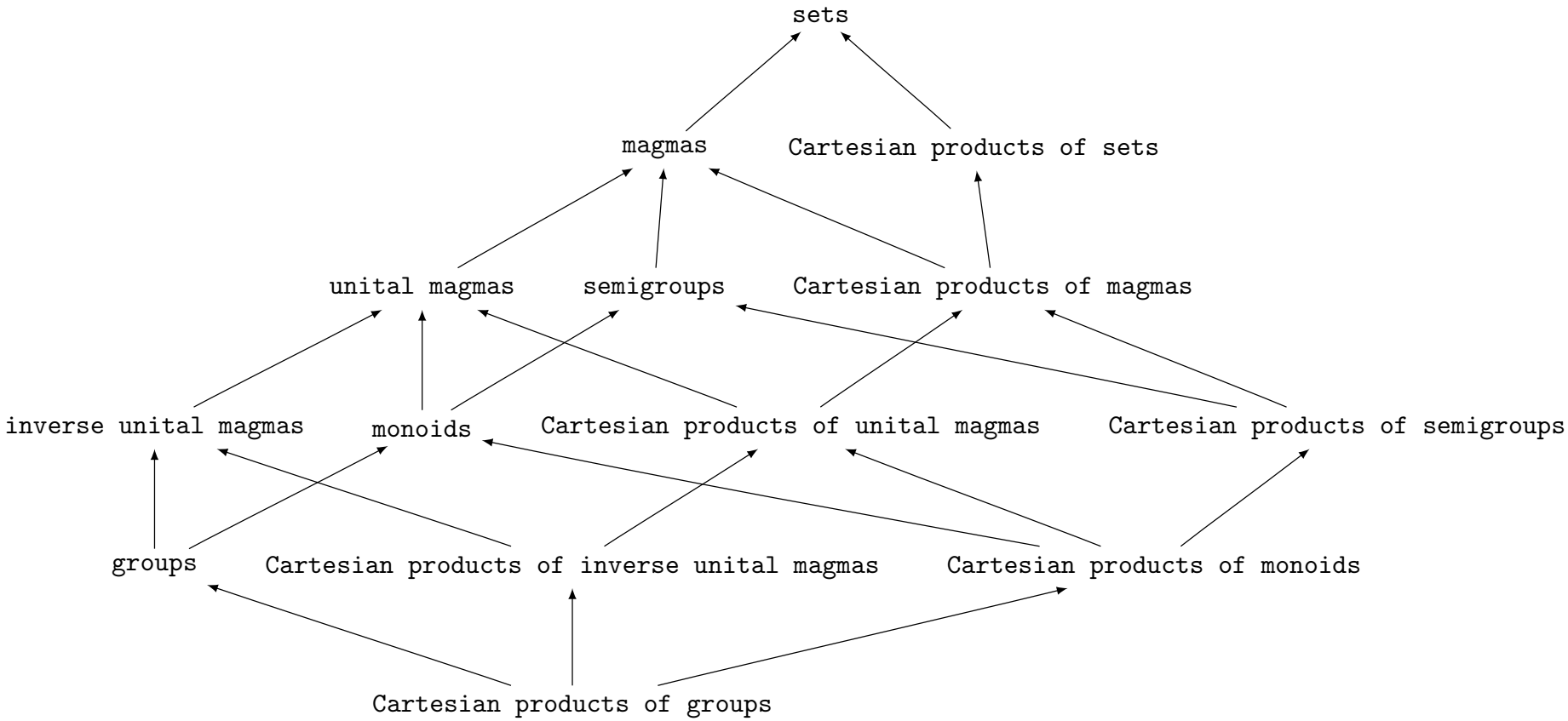
# Passage à l'échelle ?

Catégories pour les groupes finis :



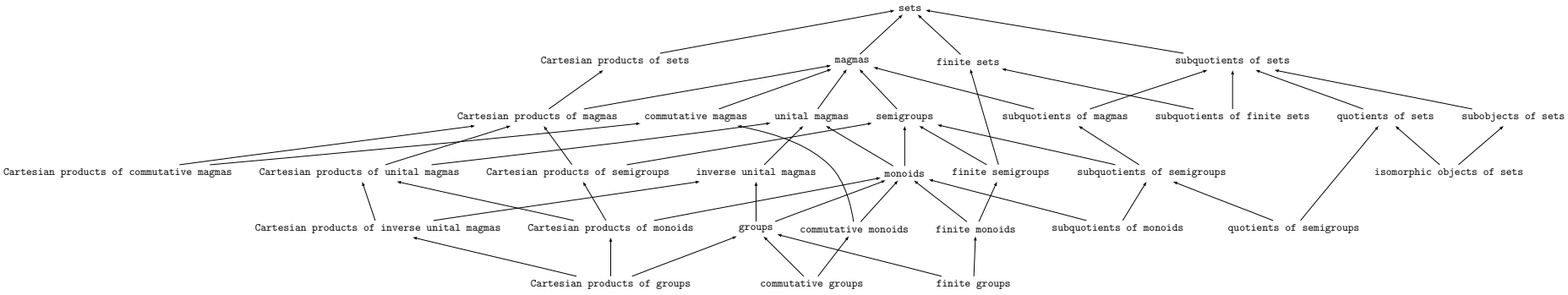
# Passage à l'échelle ?

Catégories pour les produits cartésiens de groupes :



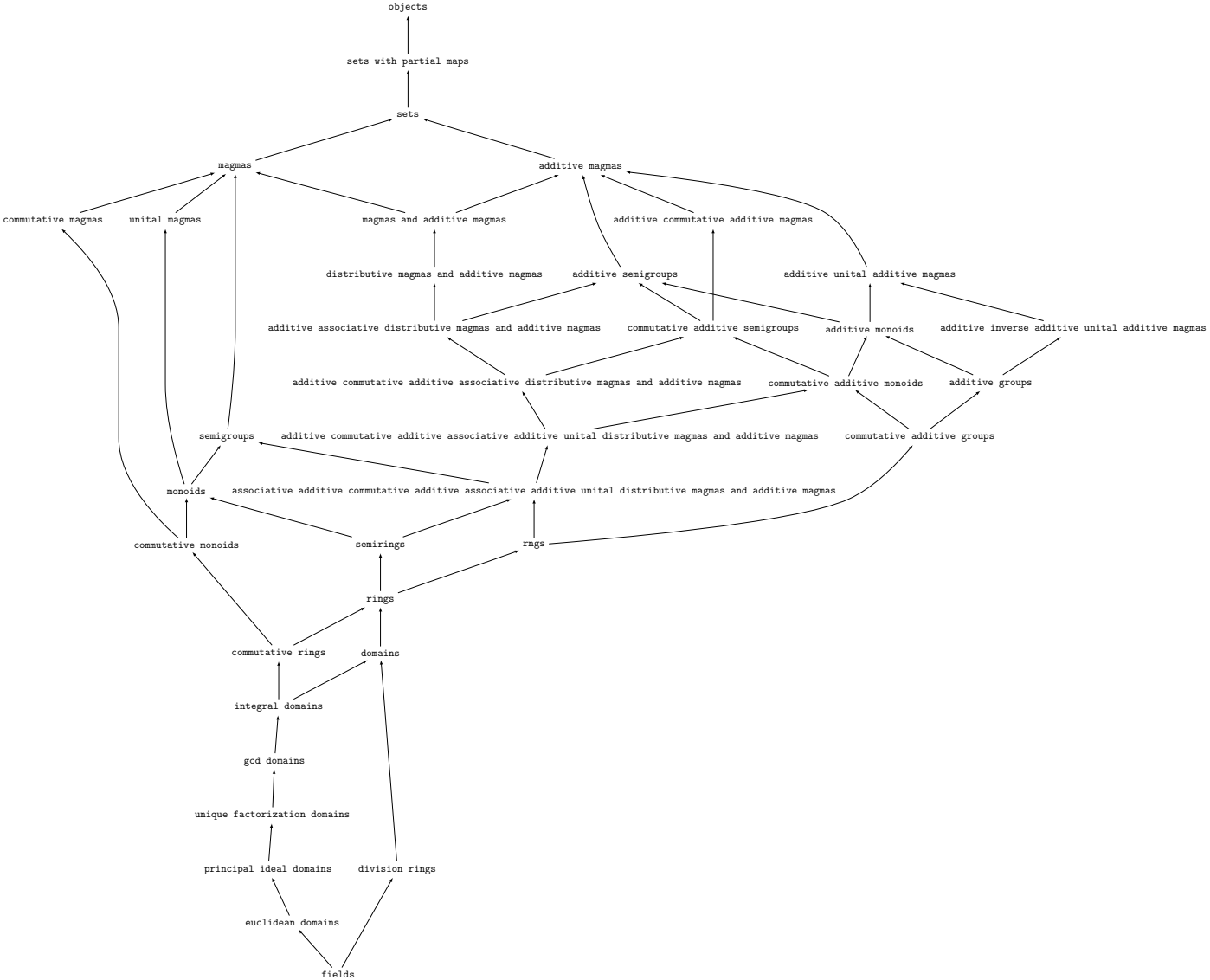
# Passage à l'échelle ?

Toutes les catégories pour les groupes :



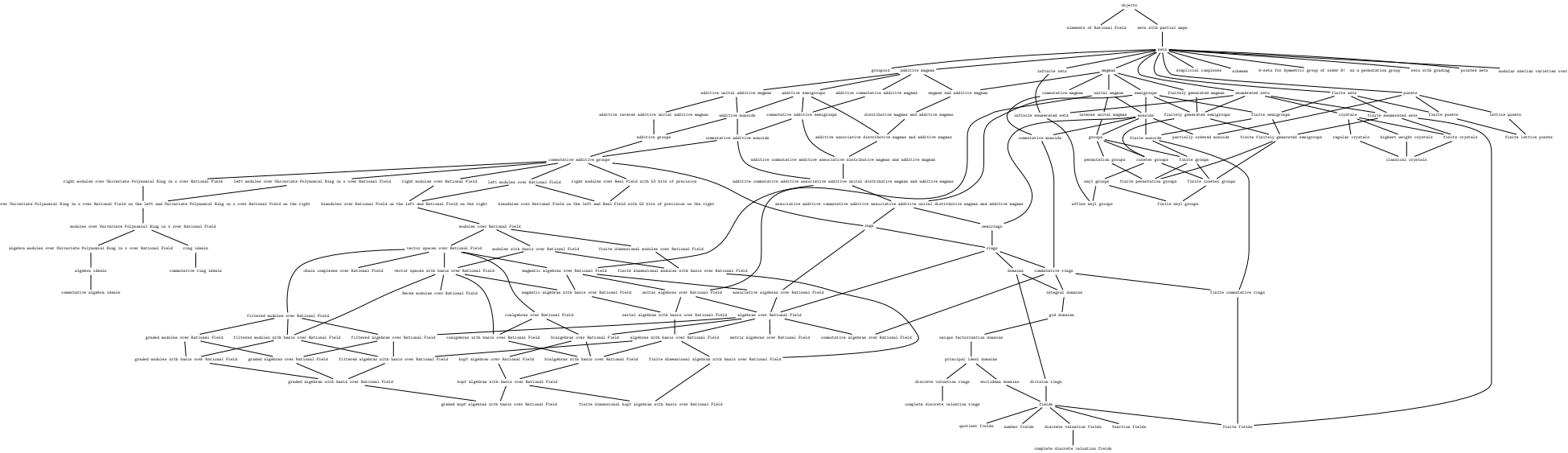
# Passage à l'échelle ?

## Catégories pour les corps :



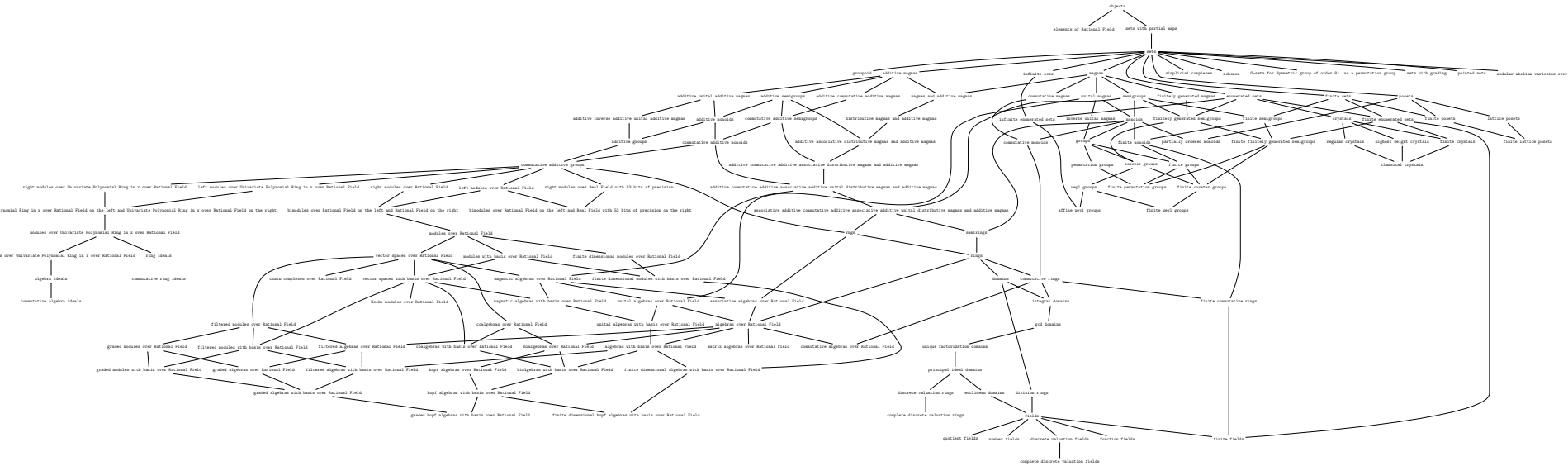
# Passage à l'échelle ?

Toutes les catégories :



# Passage à l'échelle ?

Toutes les catégories :




## Problème

Maîtriser l'*explosion combinatoire* du nombre de classes

# Implantation dans Sage

## Une catégorie pour chaque contexte *intéressant*

*Étagère* contenant :

- Des informations sémantiques (théorèmes, ...)
- Des «Mixins» pour les parents, éléments, morphismes :
  - Documentation
  - Code générique
  - Tests
- D'autres catégories

## Hiérarchie de classes

- Génération dynamique et paresseuse à partir des «Mixins»
- Algorithmique des treillis pour minimizer la hiérarchie

# Implantation dans Sage

## Une catégorie pour chaque contexte *intéressant*

*Étagère* contenant :

- Des informations sémantiques (théorèmes, ...)
- Des «Mixins» pour les parents, éléments, morphismes :
  - Documentation
  - Code générique
  - Tests
- D'autres catégories

## Hiérarchie de classes

- Génération dynamique et paresseuse à partir des «Mixins»
- Algorithmique des treillis pour minimizer la hiérarchie

# Résumé

- Sage modélise de très nombreux objets mathématiques
- Grande hiérarchie de catégories :
  - Sémantique
  - Mixins pour parents, éléments, morphismes
  - Documentation, Tests, Code générique
  - Constructions
- Hiérarchie robuste car elle modélise une hiérarchie existante (catégories en algèbre)
- Passage à l'échelle :
  - Construction dynamique à partir d'*information sémantique* et de *mixins* fournis par les catégories
  - Contrôle de la linéarisation

# Résumé

- Sage modélise de très nombreux objets mathématiques
- Grande hiérarchie de catégories :
  - Sémantique
  - Mixins pour parents, éléments, morphismes
  - Documentation, Tests, Code générique
  - Constructions
- Hiérarchie robuste car elle modélise une hiérarchie existante (catégories en algèbre)
- Passage à l'échelle :
  - Construction dynamique à partir d'*information sémantique* et de *mixins* fournis par les catégories
  - Contrôle de la linéarisation

# Le paradigme est bon ; est-ce une bonne implantation ?

## Naturelle dans son contexte

- Langage dynamique (Python)
- Programmation orientée objet

## En dehors de ce contexte ?

- Performances ? Compilation ?
  - Le code générique peut appeler du code compilé
  - Le code générique pourrait être compilé
  - Mais via appels de méthodes virtuelles
- Vérification statique ?

# Le paradigme est bon ; est-ce une bonne implantation ?

## Naturelle dans son contexte

- Langage dynamique (Python)
- Programmation orientée objet

## En dehors de ce contexte ?

- Performances ? Compilation ?
  - Le code générique peut appeler du code compilé
  - Le code générique pourrait être compilé
  - Mais via appels de méthodes virtuelles
- Vérification statique ?

# Pistes à explorer

## Implantations alternatives du paradigme ?

- Dans un langage à typage statique ou graduel ?
- À coup de templates et de traits ?
  Par exemple en C++ / Scala
- Dans des assistants de preuve ?
  Par exemple Coq
- Gérant le multi-paramètres
  Par exemple Julia, GAP

# Pistes à explorer

## Modéliser plus de connaissances mathématiques ?
## Formaliser des systèmes de calculs ?

- Collaboration avec des systèmes de représentation de connaissances et documents mathématiques OMDoc/MMT
- Génération automatique d'interfaces entre systèmes de calculs ?

## Systèmes de documentation et de navigation ?

**Bourse de thèse ou postdoc 2016-2019 au LRI !**

(financée par OpenDreamKit)

# Pistes à explorer

## Modéliser plus de connaissances mathématiques ?
## Formaliser des systèmes de calculs ?

- Collaboration avec des systèmes de représentation de connaissances et documents mathématiques OMDoc/MMT
- Génération automatique d'interfaces entre systèmes de calculs ?

## Systèmes de documentation et de navigation ?

**Bourse de thèse ou postdoc 2016-2019 au LRI !**

(financée par OpenDreamKit)

# Pistes à explorer

## Modéliser plus de connaissances mathématiques ?
## Formaliser des systèmes de calculs ?

- Collaboration avec des systèmes de représentation de connaissances et documents mathématiques OMDoc/MMT
- Génération automatique d'interfaces entre systèmes de calculs ?

## Systèmes de documentation et de navigation ?

**Bourse de thèse ou postdoc 2016-2019 au LRI !**

(financée par OpenDreamKit)