MuPAD-Combinat

Copyleft 2000-2005 by the MuPAD-Combinat team

`http://mupad-combinat.sf.net/`

MuPAD-Combinat is an open-source algebraic combinatorics package for the computer algebra system MuPAD. Its main purpose is to provide an extensible toolbox for computer exploration, and to foster code sharing between researchers in this area.

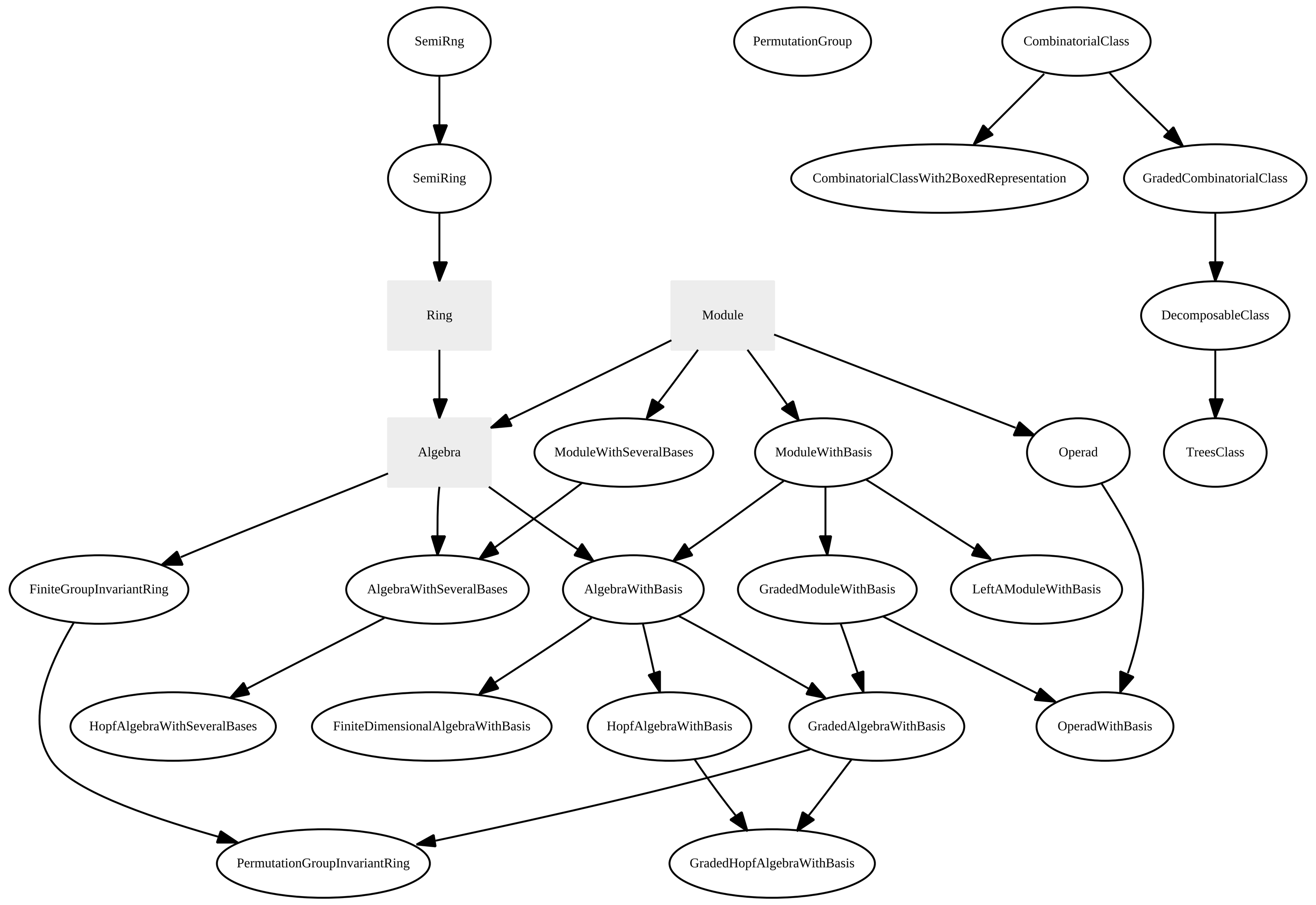
Why contribute to a Computer Algebra System?

- Research in algebraic combinatorics benefits greatly from computer aided exploration, which requires flexible and powerful tools.
- Once developed, these tools are best shared with the community where they find further applications, both in research and teaching.
- This increases our visibility and helps spreading our research results to other scientific areas where similar computations are needed.

Why MuPAD?

- The kernel and language of MuPAD (`http://www.mupad.de/`) are built on sound bases, which eases development.
- The collaboration with the MuPAD team is based on mutual commitments which are satisfactory for both parts:
  - backward compatibility and stability of MuPAD,
  - technical support, addition of new functionalities in the kernel,
  - reasonably open (albeit unfortunately not free software).

MuPAD-Combinat categories



State of the art (version 1.3.0, 28th of May 2005)

- Stable version, general project architecture in place.
- Usual combinatorial classes: words, permutations, tableaux, trees, set partitions, graphs, decomposable objects, ...
- Tools to define new combinatorial classes.
- Examples of combinatorial Hopf algebras: symmetric functions and their generalisations, Hecke, Loday-Ronco, Steenrod, Weyl algebras, Schubert poynomials, invariants of permutation groups, ...
- Tools to define new combinatorial algebras.
- Representation theory of finite dimensional algebras.
- Weighted automata, combinatorial operads, ...

MuPAD-Combinat facts

- Born in december 2000 by merging the projects ACE, μ-EC (IGM Marne-la-vallée), CS (Loria Nancy, LaBRI Bordeaux, LRI Orsay), PerMuVAR (LaPCS Lyon), and the combinat library from MuPAD.
- Official MuPAD combinat library starting from January 2002 on.
- Played a central role in 15+ publications.
- Nine official releases among which four stable ones.
- Code: 70000 lines of MuPAD, 15000 lines of C++, 80000 lines of C (Symmetrica); documentation: 575 pages; tests: 17000 lines.
- OS: GNU/Linux, MacOS X, Windows.

Implementing Free Symmetric Functions

```
/*************************************************************
   Free Littlewood-Richardson rule:
   The product of two tableaux t1 and t2 is defined as the sum
   of the tableaux appearing in the shuffle of t1 with the
   plactic class of t2 shifted by the size of t1.
************************************************************/

export(combinat, Alias, tableaux, words): // Shortcuts

// Plactic class of a standard tableau
placticClasstableaux :=
proc(t : tableaux::typeStandard)
begin
    map(tableaux::list(tableaux::shape(t)),
        t2 -> tableaux::invSchensted(t, t2));
end_proc:

// Is a word the natural row-reading of some standard tableau?
isWordStandardTableaux :=
proc(word : words)
begin
    testtype(tableaux::fromWord(word), tableaux::typeStandard);
end_proc:
```

```
// Free Littlewood-Richardson rule
freeLR :=
proc (t1 : tableaux::typeStandard, t2 : tableaux::typeStandard)
    local m1, t2shifted, k, res;
begin
    m1  := tableaux::toWord(t1);
    k   := nops(m1);
    t2  := map(t2, map, _plus, k);
    res := _concat(select(words::shuffle::list(m1, i),
                          isWordStandardTableaux)
                   $ i in placticClassTab(t2));
    map(res, z -> tableaux::insertWord([], z));
end_proc:

// The algebra of Free Symmetric Functions
domain FreeSymmetricFunctions(R)
    inherits Dom::FreeModule(R, tableaux);
    category Cat::AlgebraWithBasis(R);

    oneBasis   := tableaux([]);
    mult2Basis :=
    (t1,t2) -> dom::plus(dom::term(t) $ t in freeLR(t1,t2));
end_domain:
```

Using Free Symmetric Functions

```
// The algebra of Free Symmetric Functions over the field of expressions
>> FS := FreeSymmetricFunctions(Dom::ExpressionField()):

// Some sample computations
>> x := FS([[1]])^2;
                                +---+
                                | 2 |   +---+---+
                                +---+ + | 1 | 2 |
                                | 1 |   +---+---+
                                +---+
                                                            Time: 20 msec

>> y := a*FS([[2],[1]]) - b*FS([[1,2]]);
                              +---+
                              | 2 |        +---+---+
                            a +---+ + (-b) | 1 | 2 |
                              | 1 |        +---+---+
                              +---+
                                                            Time: 30 msec

>> x * y;
                                                                    +---+                              +---+
     +---+---+        +---+                                         | 3 |            +---+             | 4 |
     | 3 | 4 |        | 3 |                +---+---+---+---+        +---+            | 2 |             +---+
(-b) +---+---+ + (-b) +---+---+---+ + (-b) | 1 | 2 | 3 | 4 | + (-b) | 2 |     + (-b) +---+---+---+ + a | 3 |     +
     | 1 | 2 |        | 1 | 2 | 4 |        +---+---+---+---+        +---+---+        | 1 | 3 | 4 |     +---+---+
     +---+---+        +---+---+---+                                 | 1 | 4 |        +---+---+---+     | 1 | 2 |
                                                                    +---+---+                          +---+---+

                    +---+
                    | 4 |     +---+
  +---+             +---+     | 4 |         +---+---+
  | 4 |             | 3 |     +---+         | 2 | 4 |
a +---+---+---+ + a +---+ + a | 2 |     + a +---+---+
  | 1 | 2 | 3 |     | 2 |     +---+---+     | 1 | 3 |
  +---+---+---+     +---+     | 1 | 3 |     +---+---+
                    | 1 |     +---+---+
                    +---+
                                                            Time: 120 msec
```

The morphism from Free Symmetric Functions to Symmetric Functions

```
// The algebra of symmetric functions
>> Sym := examples::SymmetricFunctions():

// Implementing the morphism:
//  -A tableau is sent to the Schur function indexed by the corresponding partition
//  - tableaux::shape computes the shape of a tableau
//  - Sym::s::term creates a Schur function from a partition
>> FS2Sym := FS::moduleMorphism(Sym::s::term @ tableaux::shape, ImageSet=Sym::s):

// Using the morphism
>> FS([[3],[1,2]])^2;

+---+       +---+
| 6 |       | 6 |           +---+           +---+                       +---+---+   +---+
+---+       +---+           | 6 |           | 6 |                       | 4 | 6 |   | 4 |           +---+---+---+
| 4 |       | 4 |           +---+---+       +---+           +---+---+   +---+---+   +---+---+       | 3 | 4 | 6 |
+---+---+ + +---+         + | 3 | 4 |     + | 3 |             + +---+---+---+---+ + | 3 | 5 | + | 3 | 6 |     + +---+---+---+
| 3 | 5 |   | 3 |           +---+---+---+   +---+---+---+---+   | 1 | 2 | 4 | 5 |   +---+---+   +---+---+---+   | 1 | 2 | 5 |
+---+---+   +---+---+---+   | 1 | 2 | 5 |   | 1 | 2 | 4 | 5 |   +---+---+---+---+   | 1 | 2 |   | 1 | 2 | 5 |   +---+---+---+
| 1 | 2 |   | 1 | 2 | 5 |   +---+---+---+   +---+---+---+---+                       +---+---+   +---+---+---+
+---+---+   +---+---+---+
                                                            Time: 510 msec

>> FS2Sym(%);

        s[3, 3] + s[2, 2, 2] + s[4, 2] + s[4, 1, 1] + 2 s[3, 2, 1] + s[3, 1, 1, 1] + s[2, 2, 1, 1]
                                                            Time: 20 msec

// Checking that this is indeed a morphism
>> Sym::s([2,1])^2;

        s[3, 3] + s[2, 2, 2] + s[4, 2] + s[4, 1, 1] + 2 s[3, 2, 1] + s[3, 1, 1, 1] + s[2, 2, 1, 1]
                                                            Time: 80 msec
```

Contact: `mupad-combinat-users@users.sf.net`

Team: Nicolas M. Thiéry, Florent Hivert, François Descouens, Teresa Gomez-Diaz, Jean-Christophe Novelli, Eric Laugerotte, Houda Abbad, Patrick Lemeur, Christophe Carré, Frédéric Chapoton, Janvier Nzeutchap, Xavier Molinero, … and you?

FPSAC, June 2005