SINGULAR 2-2 – A Computer Algebra System for Polynomial Computations

Viktor Levandovskyy and Hans Schönemann
Center for Computer Algebra
Fachbereich Mathematik
Universität Kaiserslautern
Postfach 3049 D-67653 Germany
E-mail: [levandov,hannes]@mathematik.uni-kl.de

June 4, 2004

Abstract

SINGULAR is a specialized computer algebra system for polynomial computations with emphasize on the needs of commutative algebra, algebraic geometry, and singularity theory. SINGULAR's main computational objects are polynomials, ideals and modules over a large variety of rings. SINGULAR features one of the fastest and most general implementations of various algorithms for computing standard resp. Gröbner bases. The new, upcoming version 2-2 includes also algorithms for a wide class of non-commutative algebras (PLURAL) and the possiblity for dynamic extension of the program at run-time (dynamic modules). Furthermore, it provides multivariate polynomial factorization, resultant, characteristic set and gcd computations, syzygy and free-resolution computations, numerical root–finding, visualisation, and many more related functionalities.

References

[GPBLS] G.-M. Greuel, G. Pfister: A SINGULAR Introduction to Commutative Algebra (with contributions by O. Bachmann, C. Lossen, H. Schönemann), Springer, 2002

[SINGULAR] `http://www.singular.uni-kl.de`

[PLURAL] V. Levandovskyy, H. Schönemann: Plural - a computer algebra system for non-commutative polynomial algebras, In Proceedings of ISSAC'03, ACM Press, 2003

Scenario for SINGULAR presentation

1 Overview on SINGULAR

We give a short overview of the abilities of SINGULAR (including some "running live examples"), with special emphasis on the new features of SINGULAR version 2-2.

2 Applications

The main part of the presentation will concentrate on following applications:

2.1 Dynamic modules

We demonstrate the intregration of a simple C routine into SINGULAR.

2.2 Gröbner walk

SINGULAR's primary computational objects are ideals resp. modules which are generated by polynomials resp. polynomial-vectors over polynomial rings or, more generally, over the localization of a polynomial ring with respect to any ordering on the set of monomials which is compatible with the semigroup structure.

The Gröbner walk implements another methods to compute Gröbner bases, we demonstrate its integration into the Gröbner base engine of SINGULAR.

2.3 Manipulation of basic data structures

The `ring` data structure describes mathematical and computer science properties for monomials and polynomials in SINGULAR: we demontrate SINGULAR's method: easy, convenient for the user and efficient for the data reprenstation.

Scenario for PLURAL presentation

3 Overview on PLURAL

We explain the main ideas and their realization in the non–commutative extension of SINGULAR, emphasizing the similarity and the difference between the commutative and non–commutative settings. Several nontrivial examples will be computed live.

3.1 Creating and manipulating non-commutative algebras

We illustrate all the methods for creating the algebras:

- *the general one*, by specifying the skew commutators explicitly;

- *the easiest one*, by just picking up one of predefined algebras from the libraries (one can choose from $U(\mathfrak{sl}_n)$, $U(\mathfrak{gl}_n)$, $U(\mathfrak{g}_2)$, $U'_q(\mathfrak{so}_3)$, $U_q(\mathfrak{sl}_{\{2,3\}})$ as well as quantum affine spaces $\mathcal{O}_q(\mathbb{A}^n)$, quantum matrices $\mathcal{O}_q(M_n(\mathbb{K}))$, Weyl, Heysenberg and various finite–dimensional algebras including exterior algebras);

- *the mixed one*, by defining a couple of algebras and building their tensor product over the ground field.

4 Applications

4.1 Maps and their kernels

We explain the concept of maps between algebras (data type `map`) and examine its usefullness by computing the kernel of the map from $U(\mathfrak{sl}_2)$ to the 1st Weyl algebra with two methods, available in PLURAL. No other computer algebra system can do this at the moment.

4.2 Homological algebra

We are going to show the abilities of PLURAL in computing free resolutions and Betti numbers over algebras and factor–algebras. As an example, we compute Betti numbers of a certain ideal in the exterior algebra, what is of actual interest in algebraic geometry.

4.3 Dimensions and annihilators

We present the library implementation of a computation of Gel'fand-Kirillov dimension (the most wanted dimension in the non-commutative world). We show how one can compute, using the GK-dimension, the annihilator of a Harish–Chandra module — the first known algorithm for solving this problem, which is very important, for instance, in the representation theory.