SINGULAR – A Computer Algebra System for Polynomial Computations

G.M. Greuel, G.Pfister, H. Schönemann
Fachbereich Mathematik; Universität Kaiserslautern
67653 Kaiserslautern; Germany
email: [greuel,pfister,hannes]@mathematik.uni-kl.de

What is SINGULAR?

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, including important non-commutative rings. SINGULAR features one of the fastest and most general implementations of various algorithms for computing standard resp. Gröbner bases. Furthermore, it provides multivariate polynomial factorization, resultant, characteristic set and gcd computations, syzygy and free-resolution computations, numerical root–finding, visualization, and many more related functionalities.

What is new?

Gröbner bases over rings

The Gröbner basis routines were modified to work for polynomial rings with coefficients from a ring like Z or Z/m instead of coming from a field. Coefficient rings of the form $Z/(2^n)$ are important for the applications, for example in proving the arithmetic correctness of data paths in System-on-Chip modules:

We start with a set of equations $G_j, j = 1,\ldots,m$ given by polynomials $f_j \in \mathbb{Z}[X]$, X a set of variables, which are of the form

$$G_j : \sum_{i=0}^{n_j-1} 2^i r_i^{(j)} = f_j\left(a_1^{(j)}, a_2^{(j)}, \ldots, a_{m_j}^{(j)}\right) \mod 2^{n_j}.$$

For the variables $r_i^{(j)}, a_k^{(l)} \in X$ in this equation we assume $r_i^{(j)} \neq a_k^{(l)}$ for $1 \leq l \leq j$ and all i,k. We call the variables $a_i^{(j)}$ *inputs* and $r_i^{(j)}$ *outputs* of G_j.

For every proof goal, we obtain an additional polynomial g depending on a subset of variables $\{a_1,\ldots,a_t\} \subset X$ and need to check whether

$$g(a_1,\ldots,a_t) = 0 \mod 2^n$$

for all solutions of the set of equations $\{G_j\}$.

Example 1 *A k-bit comparator of operands a and b is modeled by the polynomial*

$$g = \sum_{i=0}^{k-1} 2^i (a_i - b_i)$$

Denote the set of all solutions to $\{G_j\}$ as $V(\{G_j\})$. Analogously let $V(g)$ be the set of all roots of g. We create an equivalent variety subset problem $V(\{h_i\}) \subset V(g)$ where h_i and g are polynomials over a single ring $\mathbb{Z}/2^N$ with appropriate N. This problem can be solved effectively using Gröbner basis techniques, such as normal form computations.

This methods was used to verify a multiply-accumulate-unit (which shall compute $a + b * cmod2^64$) and was much faster than other methods.

Additional to the definition of Gröbner basis over rings (a generating set of polynomials for the ideal, whose leading terms generate the leading term ideal) we need the notion of a strong Gröbner basis: a Gröbner basis, where the leading term of each element of the ideal is divisible by an element of the strong Gröbner basis.

SINGULAR computes a strong Gröbner basis with a variant of the algorithm of Buchberger.

SINGULAR **as a library**

Although SINGULAR is developed as a complete system, it can now also be compiled as a library and provide its routines to other systems via direct linkage as a C++ library. This allows access to all commands provides by the kernel of SINGULAR, and, if the high level libraries can be found and loaded, also to the library routines.

One example is the SAGE framework ([SAGE]) which uses libSingular for its polynomial arithmetic, optional for its Gröbner base computations.

Another example is `gfan` ([GFAN]), a software package for computing Gröbner fans and tropical varieties. These are polyhedral fans associated to polynomial ideals. The maximal cones of a Gröbner fan are in bijection with the marked reduced Gröbner bases of its defining ideal. For the computation of these many Gröbner basis an experimental version of `gfan` uses now libSingular.

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.

2 New stuff

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

2.1 SINGULAR as a library modules

We demonstrate a simple C++ routine which calls the routine to compute a Gröbner base from the SINGULAR kernel.

2.2 Gröbner bases and normal forms for polynomial rings over Z/m

An example for normal form computation coming from proving arithmetic correctness of data paths in System-on-Chip modules will be given. Also we describe what is working for such polynomial rings (arithmetic, Gröbner bases, normal forms) and what not (everything which assumes the coefficients to come from a field).

References

[GFAN]	`http://www.math.tu-berlin.de/~jensen/software/gfan/gfan.html`
[GPBLS]	G.-M. Greuel, G. Pfister: A SINGULAR Introduction to Commutative Algebra (with contributions by O. Bachmann, C. Lossen, H. Schönemann), Springer, 2002. Second edition appeared in 2007.
[SAGE]	`http://www.sagemath.org`
[SINGULAR]	`http://www.singular.uni-kl.de`