An Introduction to Computer Algebra System SINGULAR. Part I

Viktor Levandovskyy, RWTH Aachen, Germany

2.09.2013, Rolduc

Where to find the information about SINGULAR?

On the SINGULAR homepage

- `http://www.singular.uni-kl.de/` (Download, Online documentation, SINGULAR Discussion Forum etc.)

In the book

- "A Singular Introduction to Commutative Algebra"

  by G.-M. Greuel and G. Pfister, Springer 2002 and 2008 (2nd ed.)

SINGULAR is a free service to the mathematical community, it is distributed under GPL license.

Background

The development of SINGULAR started in early 80's in order to support the research in

- commutative algebra
- algebraic geometry
- singularity theory

as well as aiming at real life applications of these disciplines.

- SINGULAR is one of the fastest computer algebra systems in the area of polynomial computations and Gröbner bases.

Meanwhile, the area of applications of SINGULAR grew significantly. Now it includes

- symbolic-numerical solving
- invariant theory
- integer programming
- coding theory
- cryptoanalysis
- systems and control theory
- development of electric circuits
- tropical geometry
- noncommutative computer algebra, *D*-modules etc.
- and many more...

• In 2004, SINGULAR was awarded with the *Richard D. Jenks Memorial Prize for Excellence in Software Engineering Applied to Computer Algebra*.

Usage

SINGULAR is not a general but a *specialized* computer algebra system. There is no fancy interface like in big "M" systems (Mathematica, Maple, MuPad), but a simple-to-use terminal interface (very good for using on remote machines).
As a background Gröbner basis engine, SINGULAR is used by e. g.

- SAGE, an open alternative to big "M" commercial systems
- HOMALG, open package for homological algebra

You can use SINGULAR from your favourite system like

- MATHEMATICA
- MAPLE
- GAP etc.

Useful links

You can find test files I present for SINGULAR at

```
http://www.math.rwth-aachen.de/~Viktor.Levandovskyy/
filez/rolduc/
```

Singular Online Manual: keep it open

```
http://www.singular.uni-kl.de/Manual/latest/index.htm
```

Introduction to Data Types

”Lord Of The Rings” Principle

Almost all computations in SINGULAR are done inside of some **ring**, which must be defined explicitly.

Example (There are data types, not depending on a ring)

- `int, intvec, intmat`: integer number, vector and matrix
- `bigint` : play with factorial
- `string` : play with ”Hello World!”
- `list` : a collection of any data
- `def` : special universal data type (chameleon)

Rings::Generalities

”Lord Of The Rings” Principle

Almost all computations in SINGULAR are done inside of some ring, which has to be defined explicitly.

Assumption

A ring R contains the identity 1 and is finitely generated.

For constructing a ring, we need

- a field $\mathbb{K}$ (together with *parameters* $p_1, \ldots, p_m$)
- a set of variables, e.g. `x,Y1,Psmall,Dt,u',XA_3`
- a monomial (module) ordering $\prec$ on the variables

Rings::Possibilities

In SINGULAR, one can set up the following commutative rings

(P) a polynomial ring $\mathbb{K}[x_1, \ldots, x_n]$ over a field $\mathbb{K}$

(S) a localization of a polynomial ring, e.g. $\mathbb{K}[x_1, \ldots, x_n]_{\langle x_1,\ldots,x_n \rangle}$

- a factor ring (also called quotient ring) by an ideal P/I or S/J
- a tensor product over a field $P/I \otimes_{\mathbb{K}} S/J$

The noncommutative subsystem PLURAL provides a possibility to set up and to work with non-commutative polynomial algebras (*GR*–algebras a.k.a. PBW algebras), where the variables $x_1, \ldots, x_n$ obey the relations

$$x_j x_i = c_{ij} x_i x_j + d_{ij} \quad \forall 1 \leq i < j \leq n, \qquad c_{ij} \in \mathbb{K}^*$$

with some more technical conditions.

Rings::Fields

Finite Fields

- $\mathbb{Z}/\mathbb{Z}p,\ p \leq 2147483629$, p a prime
- Galois fields $GF(p^n)$ with $p^n \leq 2^{15}$ elements

Extensions

- transcendental field extension by *parameters* $\mathbb{K}(p_1, \ldots, p_m)$
- simple algebraic extension with a parameter and it minimal polynomial $\mu(a)$ produces $\mathbb{K}[a]/\mu(a)$
- multiparametric algebraic extensions may be converted to a simple algebraic extension by using a library PRIMITIV.LIB

Numerical fields

- `(real,10,20)` for $\mathbb{R}$: 10 valid digits, 20 digits for the rest
- `(complex,30,50)` for $\mathbb{C}$, where $\sqrt{-1} =: i$

Rings::Fields Examples

Finite Fields

- `ring r1 = 11111,(x),dp;` gives $(\mathbb{Z}/11093\mathbb{Z})[x]$
- `ring G = (1024,g),(x,y),dp;` gives $GF(2^{10})[x,y]$, where g is a generator of the cyclic group of units of $GF(2^{10})$

Extensions

- $\mathbb{Z}/7\mathbb{Z}(a,b,c)[X_1,X_2]$: `ring t = (7,a,b,c),(X1,X2),dp;`
- $(\mathbb{Q}[i]/(i^2+1))[z]$:

```
ring A = (0,i),(z),dp;
minpoly = i^2+1;
```

Remark

Arbitrarily long integers are handled with the data type `bigint`.

Rings::Orderings

There is the following classification:

Definition (Monomial Ordering)

Let $\mathrm{Mon}(R) = \{x^\alpha \mid \alpha \in \mathbb{N}^n\}$.

- $\prec$ is a global ordering, if $1 \prec x^\alpha \ \forall\ \alpha \neq 0$ (polynomials)
- $\prec$ is a local ordering, if $x^\alpha \prec 1 \ \forall\ \alpha \neq 0$ (series)
- otherwise, $\prec$ is a mixed (product) ordering

Robbiano's Construction

Indeed, any monomial ordering can be represented by a matrix $M \in \mathbb{GL}(n, \mathbb{Z})$ by

$$\alpha \prec_M \beta \ \Leftrightarrow\ M\alpha \prec_{lex} M\beta$$

Rings::Orderings Examples

Global and Product Monomial Orderings

- `lp` lexicographical ordering
- `dp` degree reverse lexicographical ordering
- `wp(`$w_1, \ldots, w_n$`)` w–weighted degrevlex ordering
- `Dp` degree lexicographical ordering
- `Wp(`$w_1, \ldots, w_n$`)` w–weighted deglex ordering
- `(ord1,...,ordN)` a product ordering (e.g. `(dp(2),lp(3))`
- `M(`$m_{11}, \ldots, m_{nn}$`)` matrix–defined ordering
- `(a(`$w_1, \ldots, w_n$`),ord)` extra weight ordering

Module Orderings

- Position–over–Term `(c,dp)` resp. Term–over–Position `(dp,C)`
- descending (”C”) resp. ascending (”c”) order of components

Rings::Orderings

"Lord Of The Rings" Principle Implication

If you wish to change the ordering (or the ground field), you have to create a new ring with that ordering (or that field).

There is an object called `basering`, where you are currently in.

Tools to transfer objects between rings and/or qrings

- `imap` works between rings with compatible ground fields
  - `imap` is the identity on variables and parameters of the same name and 0 otherwise
  - `imap` can map parameters to variables
- `fetch` works between rings with compatible ground fields
  - the *i*-th variable of the source ring is mapped to the *i*-th variable of the basering
- `map` works between quite different rings
  - the target of a map is always the actual basering
  - maps between rings with different coefficient fields are possible

Data Types `poly` and `ideal`

Polynomials

`poly` corresponds to a finite sum of monomials in variables of the ring with coefficients from the ground field of the ring, where the monomials are ordered, accoring to the monomial ordering of the ring.

```
ring r = (0,a),(x,y,z),Dp;
poly p = a^7*x^2*y - 343*xz*(y - (az +x)^3);
p; // prints p in the expanded form
factorize(p); // factorization
```

Data Type `ideal`

Constructively it is a list of generators of type `poly`. `ncols` gives the total number of elements, `size` gives the number of nonzero elements. For numerous reasons, we want to compute Gröbner bases of ideals with respect to a fixed monomial ordering.

Engine: Gröbner basis

There are many possibilities to compute GB

- GB of a submodule of a free module of finite rank
- w.r.t. any monomial module ordering
- reduced resp. completely reduced Gröbner basis
- quite fast in general
- `groebner` computes a GB with heuristically chosen method
- classical all-purpose standard basis `std`
- GB and minimal basis together `mstd`
- FGLM method for 0-dimensional ideals `stdfglm`
- Hilbert-driven method `stdhilb`
- factorizing Groebner basis algorithm `facstd`
- a recent addition: slim Groebner basis `slimgb`

Symbolic–Numerical Solving

A system of equations S over the field $\mathbb{K}$ corresponds to the ideal $I = I(S)$. There is a finite number of solutions over $\bar{\mathbb{K}}$ if and only if the *dimension* (Krull dimension) of I is 0.

What does solving mean?

There might be different wishes, like

- compute one, some or all the roots with or without multiplicities numerically with a given precision
- compute a field extension $\mathbb{K} \subseteq L$, such that there are exact symbolic expressions for the roots of S in L

SINGULAR has procedures for both ways of solving. The second way can be done, using the primary decomposition.