Computer Science Journal of Moldova, vol.4, no.3(12), 1996

Algorithms in Singular

Hans Schönemann

Abstract

Some algorithms for singularity theory and algebraic geometry

The use of Gröbner basis computations for treating systems of polynomial equations has become an important tool in many areas. This paper introduces of the concept of standard bases (a generalization of Gröbner bases) and the application to some problems from algebraic geometry. The examples are presented as SINGULAR commands. A general introduction to Gröbner bases can be found in the textbook [CLO], an introduction to syzygies in [E] and [St1].

SINGULAR is a computer algebra system for computing information about singularities, for use in algebraic geometry. The basic algorithms in SINGULAR are several variants of a general standard basis algorithm for general monomial orderings(see [GG]). This includes wellorderings (Buchberger algorithm ([B1],[B2]) and tangent cone orderings (Mora algorithm ([M1],[MPT])) as special cases: It is able to work with non-homogeneous and homogeneous input and also to compute in the localization of the polynomial ring in 0. Recent versions include algorithms to factorize polynomials and a factorizing Gröbner basis algorithm. For a complete description of SINGULAR see [Si].

1 Basic definitions

1.1 Monomial orderings

The basis ingredience for all standard basis algorithms is the ordering of the monomials (and the concept of the leading term: the term with the highest monomial).

A **monomial ordering** (term ordering) on $K[x_1,\ldots,x_n]$ is a total ordering $<$ on the set of monomials (power products) $\{x^\alpha | \alpha \in \mathbf{N}^n\}$ which is compatible with the natural semigroup structure, i.e. $x^\alpha < x^\beta$ implies $x^\gamma x^\alpha < x^\gamma x^\beta$ for any $\gamma \in \mathbf{N^n}$.

The ordering $<$ is called a **wellordering** iff 1 is the smallest monomial. Most of the algorithms work for general orderings.

Robbiano (cf.[R]) proved that any semigroup ordering can be defined by a matrix $A \in GL(n, \mathbf{R})$ as follows (**matrix ordering**):

Let $a_1,\ldots,a_k$ be the rows of A, then $x^\alpha < x^\beta$ if and only if there is an i with $a_j\alpha = a_j\beta$ for $j < i$ and $a_i\alpha < a_i\beta$. Thus, $x^\alpha < x^\beta$ if and only if $A\alpha$ is smaller than $A\beta$ with respect to the lexicographical ordering of vectors in $\mathbf{R}^n$.

We call an ordering a **degree ordering** if it is given by a matrix with coefficients of the first row either all positive or all negative.

Let K be a field; for $g \in K[\underline{x}]$, $g \neq 0$, let $\mathbf{L}(\mathbf{g})$ be the **leading monomial** with respect to the ordering $<$[1] and $\mathbf{c}(\mathbf{g})$ the coefficient of $L(g)$ in g, that is $g = c(g)L(g)+$ smaller terms with respect to $<$.

$<$ is an **elimination ordering** for $x_{r+1},\ldots,x_n$ iff $L(g) \in K[x_1,\ldots,x_r]$ implies $g \in K[x_1,\ldots,x_r]$).

1.2 Examples for monomial orderings

Important orderings for applications are:

- The **lexicographical ordering** *lp*, given by the matrix:

$$\begin{pmatrix} 1 & & & \\ & 1 & & 0 \\ & & \ddots & \\ & 0 & & 1 \end{pmatrix}$$

[1] we write the terms of a polynomial in decreasing order

resp. *ls*:

$$\begin{pmatrix} -1 & & & \\ & -1 & & 0 \\ & & \ddots & \\ & 0 & & -1 \end{pmatrix}$$

Remark 1.1 *The positive lexicraphic ordering lp on $K[x_1, \ldots, x_n]$ is an elimination ordering for $x_1, \ldots, x_i$ $\forall 1 \leq i \leq n$*

The definition of rings with these orderings in SINGULAR:
(Each line starting with // is a comment in SINGULAR.)

```
ring R1=0,(x(1..5)),lp;
ring R2=0,(x(1..5)),ls;
```

- The **weighted degree reverse lexicographical ordering**, given by the matrix

$$wp : \begin{pmatrix} w_1 & w_2 & \ldots & w_n \\ & & & -1 \\ & & \diagup & \\ 0 & -1 & & \end{pmatrix}$$

$w_i > 0 \forall i$, (resp. $ws : w_1 \neq 0, w_i \in \mathbf{Z}\ \forall i$).

If $w_i = 1$ (respectively $w_i = -1$) for all i we obtain the **degree reverse lexicographical ordering, dp** (respectively **ds**).

The definition of rings with these orderings in SINGULAR:

```
ring R3=0,(x(1..5)),wp(2,3,4,5,6);
// correspond to w_i:2,3,4,5,6
ring R4=0,(x(1..5)),ws(2,3,4,5,6);
// correspond to w_i:-2,-3,-4,-5,-6
ring R5=0,(x(1..4)),dp;
ring R6=0,(x(1..4)),ds;
```

- An example for an **elimination ordering** for $x_{r+1}, \dots, x_n$ in $K[\underline{x}] = \operatorname{Loc}_< K[\underline{x}]$ is given by the matrix

$$\left(\begin{array}{cccccccc}
0 & 0 & \dots & 0 & w_{r+1} & w_{r+2} & \dots & w_n \\
w_1 & w_2 & \dots & w_r & 0 & 0 & \dots & 0 \\
 & & & & & & & -1 \\
 & & & & & & \diagup & \\
 & & & & 0 & -1 & & \\
 & & & -1 & & & & \\
 & & \diagup & & & & & \\
0 & -1 & & & & & &
\end{array}\right)$$

with $w_1 > 0, \dots, w_n > 0$. In $K[x_1, \dots, x_r]_{(x_1,\dots,x_r)}[x_{r+1}, \dots, x_n] = \operatorname{Loc}_< K[\underline{x}]$ it is given by the same matrix with $w_1 < 0, \dots, w_r < 0$ and $w_{r+1} > 0, \dots, w_n > 0$.

The definition of a polynomial ring with an elimination ordering for x_3 and x_4 in Singular:

```
ring E=0,(x(1..4)),(a(0,0,1,1),a(1,1),dp);
// correspond to w_i=1 for all i, r=2
// or simpler:
ring EE=0,(x(1..4)),(a(0,0,1,1),dp);
```

- The **product ordering**, given by the matrix

$$\begin{pmatrix} A_1 & & & 0 \\ & A_2 & & \\ & & \ddots & \\ 0 & & & A_k \end{pmatrix}$$

if the A_i define orderings on monomials given by the corresponding subsets of $\{x_1, \dots, x_n\}$. Such an ordering can be used to compute in

  - $K(\underline{y})[\underline{x}]$ (A_1 : dp *on* $\underline{x}$, A_2 : dp *on* $\underline{y}$)
  - $(K[\underline{y}]_{(\underline{y})})[\underline{x}]$ (A_1 : dp *on* $\underline{x}$, A_2 : ds *on* $\underline{y}$)

– $(K[\underline{y}])[\underline{x}]_{([\underline{x})}$ (A_1 : ds *on* $\underline{x}$, A_2 : dp *on* $\underline{y}$)

(See [GTZ], [GG], definition 2.1).

The definition of a ring with this ordering in SINGULAR:

```
ring P=0,(x(1..6)),(dp(4),ds(2));
// correspond to
// a first block of 4 variables with ordering dp
// and a second block of 2 variables with ordering ds
```

1.3 Extension of monomial orderings to free modules

We consider also **module orderings** $<_m$ on the set of "monomials" $\{x^\alpha e_i\}$ of $K[\underline{x}]^r = \sum_{i=1,\dots,r} K[\underline{x}]e_i$ which are compatible with the ordering $<$ on $K[\underline{x}]$. That is for all monomials $f, f' \in K[\underline{x}]^r$ and $p, q \in K[\underline{x}]$ we have: $f <_m f'$ implies $pf <_m pf'$ and $p < q$ implies $pf <_m qf$.

We now fix an ordering $<_m$ on $K[\underline{x}]^r$ compatible with $<$ and denote it also with $<$. Again we have the notion of coefficient $c(f)$ and leading monomial $L(f)$. $<$ has the important property:

$$\begin{array}{ll} L(qf) = L(q)L(f) & \text{for } q \in K[\underline{x}] \text{ and } f \in K[\underline{x}]^r, \\ L(f+g) \le \max(L(f), L(g)) & \text{for } f, g \in K[\underline{x}]^r. \end{array}$$

2 Standard bases

Definition 2.1 *We define* $\mathbf{Loc_<K[\underline{x}]} := S_<^{-1}K[\underline{x}]$ *to be the localization of* $K[\underline{x}]$ *with respect to the multiplicative closed set* $S_< = \{1 + g \mid g = 0$ *or* $g \in K[\underline{x}]\backslash\{0\}$ *and* $1 > L(g)\}$.

Remark 2.2 1) $K[\underline{x}] \subseteq \mathrm{Loc}_<K[\underline{x}] \subseteq K[\underline{x}]_{(\underline{x})}$, where $K[\underline{x}]_{(\underline{x})}$ denotes the localization of $K[\underline{x}]$ with respect to the maximal ideal $(x_1, \dots, x_n)$. In particular, $\mathrm{Loc}_<K[\underline{x}]$ is noetherian, $\mathrm{Loc}_<K[\underline{x}]$ is $K[\underline{x}]$–flat and $K[\underline{x}]_{(\underline{x})}$ is $\mathrm{Loc}_<K[\underline{x}]$–flat.

2) If $<$ is a wellordering then $x^0 = 1$ is the smallest monomial and $\mathrm{Loc}_<K[\underline{x}] = K[\underline{x}]$. If $1 > x_i$ for all i, then $\mathrm{Loc}_<K[\underline{x}] = K[\underline{x}]_{(\underline{x})}$.

3) If, in general, $x_1, \ldots, x_r < 1$ and $x_{r+1}, \ldots, x_n > 1$ then

$$1+(x_1,\ldots,x_r)K[x_1,\ldots,x_r] \subseteq S_< \subseteq 1+(x_1,\ldots,x_r)K[\underline{x}] =: S,$$

hence

$$K[x_1,\ldots,x_r]_{(x_1,\ldots,x_r)}[x_{r+1},\ldots,x_n] \subseteq \mathrm{Loc}_< K[\underline{x}] \subseteq S^{-1}K[\underline{x}].$$

2.1 Definition

Definition 2.3 *Let $I \subseteq K[\underline{x}]$ be an ideal.*

1) $\mathbf{L}(I)$ *denotes the ideal of $K[\underline{x}]$ generated by $\{L(f) | f \in I\}$.*

2) $f_1, \ldots, f_s \in I$ *is called a* **standard basis** *of I if $\{L(f_1), \ldots, L(f_s)\}$ generates the ideal $L(I) \subset K[\underline{x}]$.*

Singular example: A standard basis computation:

```
// define a ring R= (Z/32003)[x,y,z]
ring R = 32003, (x,y,z), dp ;
// define 3 polynomials
poly s1  = x^3*y^2 + 151*x^5*y + 169*x^2*y4
          + 151*x^2*y*z3 + 186*x*y^6 + 169*y^9;
poly s2  = x^2*y^2*z^2 + 3*z^8;
poly s3  = 5*x^4*y^2 + 4*x*y^5 + 2x^2*y^2*z^3 + y^7 +
           11*x^10;
// define the ideal i generated by s1,s2,s3
ideal i = s1, s2, s3;
// compute standard basis j of i
ideal j = std(i);
// display j;
j;
```

2.2 Standard bases for submodules of free modules

Definition 2.4 *Let $M \subseteq K[\underline{x}]^r$ be an submodule of the free module $K[\underline{x}]^r$.*

1) $\mathbf{L}(M)$ *denotes the submodule of* $K[\underline{x}]^r$ *generated by* $\{L(f)|f \in M\}$.

2) $f_1, \ldots, f_s \in M$ *is called a* **standard basis** *of* M *if* $\{L(f_1), \ldots, L(f_s)\}$ *generates the submodule* $L(M) \subset K[\underline{x}]^r$.

In SINGULAR submodules of free modules are defined by a set of generators. These sets are of type `module`.

SINGULAR example (see [PS]):

```
// ===========Poincare complex ==========================
// counterexample to a possible generalization of a theorem
// of Kyoji Saito. A complete intersection with exact
// Poincare complex at 0 but which is in no coordinate
// system weighted homogeneous see [PS] for
// an exeact decription.
//
// define (Z/32003)[[x,y,z]]
ring Rp=32003,(x,y,z),(c,ds);
// load additional procedures (milnor, tjurina)
LIB "sing.lib";
// select an example, parametrized by n and m
int n=883; int m=937;
poly f1=xy+z^(n-1);
poly f2=xz+y^(m-1)+yz2;
ideal f=f1,f2;
// define the basering as Rp/f and fetch the data
qring R=std(f);
ideal f=fetch(Rp,f);
poly f1,f2=fetch(Rp,f1),fetch(Rp,f2);
// the module Omega2:
module omega2=
[diff(f1,y),diff(f1,z),0],
[diff(f1,x),0,-diff(f1,z)],
[0,diff(f1,x),diff(f1,y)],
[diff(f2,y),diff(f2,z),0],
```

```
[diff(f2,x),0,-diff(f2,z)],
[0,diff(f2,x),diff(f2,y)];
//it can be shown, that the Poincare complex is exact, if
//(in this case) Milnor number(f)+1 = multiplicity(omega2)
omega2=std(omega2);
multiplicity(omega2);
// The Milnor number of the complete intersection f;
milnor(f);
// The Tjurina number of the complete intersection f
tjurina(f);
//since the Milnor number and the Tjurina number do not
//coincide, the singularity is not weighted homogeneous
```

2.3 Basic properties

2.3.1 Normal form

Definition 2.5 *A function* $NF : K[\underline{x}]^r \times \{G|G \text{ standardbasis}\} \rightarrow K[\underline{x}]^r, (p, G) \mapsto NF(p|G)$*, is called a* **normal form** *if for any* $p \in K[\underline{x}]^r$ *and any* G *the following holds: if* $NF(p|G) \neq 0$ *then* $L(g) \not| L(NF(p|G))$ *for all* $g \in G$*.* $NF(g|G)$ *is called the* **normal form of p with respect to G***.*

2.3.2 Ideal/submodule membership

Lemma 2.6 $f \in I$ *iff* $NF(f, std(I)) = 0$ *for* $I \subseteq R$ *resp.* $I \subseteq R^r$*.*

Singular example:

```
//f defines a trimodal singularity for generic moduli
ring R = 0,(x,y),ds;
int a1,a2,a3=random(1,100),random(-100,1),random(1,100);
poly f = (x^2-y^3)*(y+a1*x)*(y+a2*x)*(y+a3*x);
ideal J = jacob(f);
ideal I = f;
// J:I, ideal of the closure of V(J) \ V(I)
ideal Q = quotient(J,I);
```

```
//the Hessian of f
poly Hess = det(jacob(jacob(f)));
//Hess is contained in Q iff NF is 0
reduce(Hess,std(Q));
```

2.3.3 Elimination

Lemma 2.7 *Let* $<$ *be an elimination order for* $y_1, \ldots, y_n$, $R = K[x_1, \ldots, x_r, y_1, \ldots, y_n]$.
Then $std(I) \cap K[x_1, \ldots, x_r] = std(I \cap K[x_1, \ldots, x_r])$.

Singular example:

```
// find the equations from a parametrization
// t->(t^3,t^4,t^5)
ring R=0,(x,y,z,t),dp;
ideal i=x-t^3,
        y-t^4,
        z-t^5;
ideal j=eliminate(i,t);
j;
```

Remark 2.8 *The elimination property applies also to modules.*

2.3.4 Hilbert series

Definition 2.9 *Let* M *be a graded module over* $K[\underline{x}]$. *The* **Hilbert series** *of* M *is the power series*

$$H(M)(t) = \sum_{t=-\infty}^{\infty} dim_K M_i t^i$$

.

Lemma 2.10 *Let* $<$ *be a (positive or negative) degree ordering and* $H(M)$ *the Hilbert function of (the homogenization of)* I. *Then* $H(M) = H(L(M))$.

Remark 2.11 *It turns out that $H(M)(t)$ can be written in two usefule ways:*

1. $H(M)(t) = Q(t)/(1-t)^n$, *where $Q(t)$ is a polynomial in t and n ist the number of variables in $K[\underline{x}]$.*
2. $H(M)(t) = P(t)/(1-t)^{dimM}$ *where $P(t)$ is a polynomial and $degM = P(1)$.*
3. *vector space dimension $dim_K(M) = dim_K(L(M))$.*

Remark 2.12 *Let $<$ be a degree ordering.*

- *Krull dimension: $dim(M) = dim(L(M))$.*
- *degree (for a positive degree ordering) resp. multiplicity (for a negative degree ordering) is equal for M and $L(M)$.*

SINGULAR example:

```
// the rational quartic curve J in P^3:
ring R=0,(a,b,c,d),dp;
ideal J=c3-bd2,bc-ad,b3-a2c,ac2-b2d;
// the output of hilb is Q, then P:
hilb(J);
```

2.4 Applications

2.4.1 Euclidian algorithm

Lemma 2.13 *If $<$ is a wellordering and $I = \{f, g\} \subseteq K[x]$ then the computation of the standard basis of I yields the greatest common divisor of f and g.*

SINGULAR example:

```
ring R=32003,x,dp;
poly f=(x^3+5)^2*(x-2)*(x^2+x+2)^4;
poly g=(x^3+5)*(x^2-3)*(x^2+x+2);
```

```
ideal I=f,g;
std(I);
// and the expected result:
(x^3+5)*(x^2+x+2);
```

2.4.2 Gaussian algorithm

Lemma 2.14 *If $<$ is a wellordering and the generators of* `I` *are linear then the computation of the standard basis of* `I` *is a Gaussian algorithm with the columns of* `matrix(I)`.

SINGULAR example:

```
ring R=32003,(x,y,z),dp;
ideal I=22*x+77*y+z-3,
         0*x+ 1*y+z-77,
         1*x+ 0*y+z+11;
std(I);
```

2.4.3 Kernel of a ring homomorphism

Lemma 2.15 *Let Φ be an affine ring homomorphism*

$$\Phi : R = K[x_1, \ldots, x_m]/I \longrightarrow K[y_1, \ldots, y_n]/(g_1, \ldots, g_s)$$

given by $f_i = \Phi(x_i) \in K[y_1, \ldots, y_n]/(g_1, \ldots, g_s)$, $i = 1, \ldots, m$. Then $Ker(\Phi)$ is generated by

$$(g_1(\underline{y}), \ldots, g_s(\underline{y}) \, , \, (x_1 - f_1(\underline{y})), \ldots, (x_m - f_m(\underline{y}))) \cap K[x_1, \ldots, x_m]$$

in $K[x_1, \ldots, x_m]/I$.

Remark 2.16 *For $std(H) \cap R$ use lemma 2.7.*

SINGULAR example:

```
ring r=...;
ideal F=...;
preimage(r,F,ideal(0));
```

2.4.4 Radical membership

Lemma 2.17 *Let* $I \subseteq R = Loc_{<}K[x_1, \ldots, x_n]$*,* I *generated by* F*.*
$f \in \sqrt{I}\ iff\ \ 1 \in std(F + (yf - 1) \subseteq R[y]$*.*

2.4.5 Principal ideal

Lemma 2.18 $I = (F)$ *is principal (i.e. has a one-element ideal basis) iff* $std(F)$ *has exactly one element.*

2.4.6 Trivial ideal

Lemma 2.19 (F) *is the whole ring* R *iff* $std(F) = \{1\}$*.*

2.4.7 Intersection 1

Lemma 2.20 *Let* $(F) \subseteq R^r$ *and* $(G) \subseteq R^r$*.*
Then $std((F) \cap (G)) = std(y(F) + (1 - y)(G)) \cap R$ *in* $R[y]$*.*

Remark 2.21 *For* $std(H) \cap R$ *use lemma 2.7.*

SINGULAR example:

```
ring r1 = 32003,(x,y,z),(c,ds);
poly s1=x2y3+45x6y3+68x4z5+80y6x8;
poly s2=6x5+3y6+8z6;
poly s3=12xyz3+2y3z6;
ideal i1=s1,s2,s3;
ideal i2=s1+s2,s2,s1;
intersect(i1,i2);
```

2.4.8 Ideal/module quotient 1

Lemma 2.22 *Let* $(F) \subseteq R^r, g \in R^r$*.*
Then $(F) : (g) = (h_1, \ldots, h_k)$ *if* $(F) \cap (g)) = (h_1 g, \ldots, h_k g)$*.*

Lemma 2.23 *Let* $(F) \subseteq R^r, (g_1, \ldots, g_s) \in R^r$*.*
Then $(F) : (g_1, \ldots, g_s) = (F) : (g_1) \cap \ldots \cap (F) : (g_s)$*.*

Remark 2.24 SINGULAR *uses another algorithm, see 4.8.*

2.4.9 Saturation 1

Lemma 2.25 *Let* $(F) \subseteq R^r, g \in R^r$.
Then $(F) : (g)^\infty = (h_1, \ldots, h_k)$ *if* $(F) \cap (g)) = (h_1 g, \ldots, h_k g)$.

Remark 2.26 SINGULAR *uses another algorithm, see 4.9*

3 Solving equations

The simplest way to "solve" systems of polynomial equations via standard basis computation is the computation of a lexicographical standard basis.

Other Possibilities include the command `eliminate` or preprocessing or a "factorizing" Groebner basis algorithm.

SINGULAR example:

```
// consider a line and a plane:
ring R=0,(x,y,z),lp;
number a,b,c,d,e=0,1,1,1,1;
ideal L=a*x+b*y,z;
poly P=c*x+d*y+e*z;
ideal I=L,P;
// force the complete reduction of the  standard basis:
option(redSB);
std(I);
eliminate(I,x);
```

3.1 Simplification

SINGULAR example:

```
ring R=...;
ideal I=....;
interred(I);
```

3.2 Solvability

Lemma 3.1 *The set of polynomials F is solvable iff $\{1\} \neq std(F)$. (See lemma 2.19.)*

SINGULAR example:

```
// consider two parallel lines:
ring R=0,(x,y),dp;
poly l1=x+y-3;
poly l2=x+y-500;
ideal F=l1,l2;
std(F);
// consider a circle and a line
poly c=x^2+y^2-4;
F=c,l1;
std(F);
F=c,l2;
std(F);
// solvable means: solvable in the algebaric closure !
```

3.3 Finite solvability

Lemma 3.2 *The set of polynomials $F \subseteq K[x_1, \ldots, x_n]$ has only finitly many solutions iff $\forall 1 \leq i \leq n : \exists f \in std(F) : L(f)$ is a power of x_i. (See remark 2.12.)*

SINGULAR example:

```
// consider a line and a plane:
ring R=0,(x,y,z),dp;
number a,b,c,d,e=1,1,1,1,1;
ideal L=a*x+b*y,z;
poly P=c*x+d*y+e*z;
ideal I=L,P;
std(I); // the zero set is a line
a=0;
```

```
L=a*x+b*y,z;
I=L,P;
std(I); // the zero set is finite (a point)
```

4 Syzygies

4.1 Definition

Definition 4.1 *Let* $I = \{g_1, \ldots, g_q\} \subseteq K[\underline{x}]^r$*.*
The **module of syzygies syz***(I) is ker* $(K[\underline{x}]^q \to K[\underline{x}]^r, \sum w_i e_i \mapsto \sum w_i g_i)$*.*

Lemma 4.2 *The module of syzygies of I is*

$$(g_1(\underline{x}) - e_{r+1}, \ldots, g_q(\underline{y}) - e_{r+q}) \cap \{0\}^r \times K[\underline{x}]^q$$

in $(K[x_1, \ldots, x_m]/J)^q$ *.*

Remark 4.3 *Use a module ordering with* $e_i > e_j \forall i \leq r < j$ *and the elimination property of lemma 2.7.*

Singular example:

```
ring R=0,(x,y,z),(c,dp);
ideal I=maxideal(1);
// the syzygies of the (x,y,z)
syz(I);
// syz yields a generating set for the module of syzygies
// but may not be a standard basis !
```

4.2 Resolutions

Iterating the `syz` command yields a free resolution of a module or ideal. Singular does this if the `res` or `mres` command is used.

Another algorithm due to Schreyer is presented in [S]. It will be used by the `sres` command.

For a comparision of these algorithms see [GG].

Singular example:

```
ring r=0,(x,y,z),dp;
ideal I=x,y,z;
list Ir=res(I,0);
// print the results:
Ir;
list Im=mres(I,0);
// print the results:
Im;
list Is=sres(std(I),0);
// print the results:
Is;
```

4.3 Kernel of a module homomorphism

Definition 4.4 *Let* $R = K[x_1, \ldots, x_n]/(h_1, \ldots, h_p)$, $A \in Mat(m \times r, R)$ *and* $B \in Mat(m \times s, R)$ *then define*

$$\mathbf{modulo}(\mathbf{A}, \mathbf{B}) := ker(R^r \stackrel{A}{\longrightarrow} R^m / Im(B))$$

($modulo(A, B)$ *is the preimage of* B *under the homomorphism given by* A*.)*

Lemma 4.5 *Let* $\{ (\underline{\alpha}_i, \underline{\beta}_i, \underline{\gamma}_i) \mid i = 1, \ldots, k \} \subset R^{r+s+p} =: R^N$ *be a* *generating set of syz(D) where*

$$C = \begin{pmatrix} h_1 & \cdots & h_p & 0 & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & h_1 & \cdots & h_p & 0 & \cdots & \cdots \\ \vdots & & & & & & & & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & h_1 & \cdots & h_p \end{pmatrix} \in$$

$$Mat(m \times pm, R)$$

and

$$D = \left(\begin{array}{ccc|ccc|ccc} a_{11} & \cdots & a_{1r} & b_{11} & \cdots & b_{1s} & c_{11} & \cdots & c_{1,pm} \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ a_{m1} & \cdots & a_{mr} & b_{m1} & \cdots & b_{ms} & c_{m1} & \cdots & c_{m,pm} \end{array} \right) \in$$

$$Mat(m \times r + s + pm, R)$$

Then

$$modulo\,(A, B) := (\,\alpha_1 \ldots \alpha_k\,) \in Mat(r \times k, R)$$

(see lemma 4.2.)

Remark 4.6 *In practice, one need not compute the entire syzygy module of D: it is better to find modulo(A,B) as:*

$$\left(\begin{array}{ccc|ccc|ccc} a_{11} & \cdots & a_{1r} & b_{11} & \cdots & b_{1s} & c_{11} & \cdots & c_{1,pm} \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ a_{m1} & \cdots & a_{mr} & b_{m1} & \cdots & b_{ms} & c_{m1} & \cdots & c_{m,pm} \\ 1 & & 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \ldots & 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & & 1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \end{array} \right) \cap \begin{pmatrix} 0 \\ \vdots \\ 0 \\ R \\ \vdots \\ R \end{pmatrix}$$

(see sections 4.2, 2.7.)

4.4 Module intersection 2

Let R be an affine ring, and let $I, J, K \subseteq R$ be ideals. One can compute generators for the intersection $L = I \cap J \cap K$ in the follwing way: L is the kernel of the R-module homomorphism $\phi : R \to R/I \oplus R/J \oplus R/K$ which sends 1 to (1,1,1).

Lemma 4.7

$$I \cap J \cap K = modulo(\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} I & 0 & 0 \\ 0 & J & 0 \\ 0 & 0 & K \end{pmatrix}).$$

4.5 Ideal quotient 2

Lemma 4.8 *The quotient $(I : J)$ of two ideals $I = (a_1, \ldots, a_r)$ and $J = (b_1, \ldots, b_s)$ in R is the kernel of the map*

$$\begin{array}{ccc} R & \longrightarrow & R/I \oplus \ldots \oplus R/I \\ 1 & \longmapsto & (b_1, \ldots, b_s) \end{array}$$

It can be computed as

$$(I : J) = modulo\Big((b_1|\ldots|b_s)^T,\ (a_1|\ldots|a_r) \oplus \ldots \oplus (a_1|\ldots|a_r)\Big)$$

(See also 2.23.)

Singular example (see example in section 2.3.2):

```
ring R=...;
ideal I=...;
ideal J=...;
quotient(I,J)
```

4.6 Saturation 2

Lemma 4.9 *The saturation $(I : J)^\infty$ of I with respect J can be computed by computing $(I : J), ((I : J) : J), \ldots$ until it stabilizes. (See also 2.25.)*

SINGULAR example:

```
ring R=...;
ideal I=...;
ideal J=...;
int ii;
I = std(I);
while ( ii<=size(II))
{
  II=quotient(I,J);
  for ( ii=1; ii <=size(II); ii=ii++)
  {
    if (reduce(II[ii],I)!=0) break;
  }
  I=std(II);
}

// II is now (I:J)∞.
```

4.7 Annihilator of a module

Lemma 4.10 *Let* $R = Loc_{<}K[x_1, \ldots, x_n]/(h_1, \ldots, h_p)$, $M \subseteq R^m$. $Ann_R(R^m/M) := \{ g \in R \,|\, gR^m \subset M \}$ *is generated by first entries of syzygies of the module*

$$\left(\begin{array}{c|c|c|c|c} e_1 & M & 0 & \cdots & 0 \\ e_2 & 0 & M & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ e_m & 0 & \cdots & 0 & M \end{array}\right)$$

where e_i *is the i-th unit vector in* R^m.
(We identify a matrix with the module generated by it columns.)

5 Examples

5.1 Ext modules Ext(M,R)

Let M be given as $R^m/Im(A_0)$, where $R = K[x_1, \ldots, x_n]/(h_1, \ldots, h_p)$. For each free resolution

$$\ldots \longrightarrow F_k \stackrel{A_{k-1}}{\longrightarrow} F_{k-1} \longrightarrow \ldots F_1 \stackrel{A_0}{\longrightarrow} F_0 = R^m \longrightarrow M \longrightarrow 0$$

is $Ext^j_R(M, R) = H^j(Hom(F_\bullet, R))$.

Algorithm 5.1 $Ext^j_R(M, R)$:

- *compute a free resolution of* M *by* $A_k = syz\,(A_{k-1})$ *for* $k = 1, \ldots, j$ *. (or use* `mres`*, see section 4.2).*
- $Ext^j_R(M, R) = Im\,(syz\,(A^T_j))\,/\,Im\,(A^T_{j-1})$ *is a free module modulo the image of the matrix*

$$modulo\,(syz\,(A^T_j)\,,\;A^T_{j-1})$$

Singular example: a complete version can be found in the Singular library `homog.lib`.

```
proc qmod (module M, module N)
//USAGE:   qmod(<module_M>,<module_N>);
//         N a submodule of M, a submodule of a free one
//COMPUTE: presentation S of M/N, i.e. M/N<<--F<--[S],
//         F free of rank = size(M),
//RETURNS: module(S)
{
  return(lift(M,N)+syz(M));
}
```

```
proc ext (int n, ideal i)
// COMPUTES:  Ext^n(R/i,R);   i ideal in the basering R
// USAGE:     ext(<int>,<ideal>);
// SHOWS:     degree of Ext^n
// RETURN:    Ext as quotient of a free module
{
//-----------------  compute resulution of R/i ----------
//          0<--R/i<--L(0)<--[i]--L(1)<--[RE[2]]--- ...
   list RE=mres(i,n+1);
//-----------------  apply Hom(_,R) at n-th place -------
   module g = module(transpose(matrix(RE[n+1])));
   module f = module(transpose(matrix(RE[n])));
//-----------------  ker(g)/im(f) ----------------------
   module ext = qmod(syz(g),f);
//---- return Ext as quotient of a free module (std) ----
   return(std(ext));
}
```

5.2 $Hom_R(M, N)$

Let M be given as $R^m/Im(A_0)$, N as $R^p/Im(B_0)$, where $R = K[x_1, \ldots, x_n]/(h_1, \ldots, h_p)$ together with free resolutions

$$\ldots \longrightarrow F_k \overset{A_{k-1}}{\longrightarrow} F_{k-1} \longrightarrow \ldots F_1 \overset{A_0}{\longrightarrow} F_0 = R^m \longrightarrow M \longrightarrow 0$$

and

$$\ldots \longrightarrow G_k \overset{B_{k-1}}{\longrightarrow} G_{k-1} \longrightarrow \ldots G_1 \overset{B_0}{\longrightarrow} G_0 = R^p \longrightarrow N \longrightarrow 0.$$

We get the following commutative diagram with exact columns and rows:

$$
\begin{array}{ccccccc}
 & & 0 & & & & \\
 & & \downarrow & & & & \\
 & & Hom_R(M,N) & & & & \\
 & & \downarrow & & & & \\
0 & \longleftarrow & F_0^* \otimes N & \longleftarrow & F_0^* \otimes G_0 & \longleftarrow & F_0^* \otimes G_1 \\
 & & \downarrow & & \downarrow & & \downarrow \\
0 & \longleftarrow & F_1^* \otimes N & \longleftarrow & F_1^* \otimes G_0 & \longleftarrow & F_1^* \otimes G_1
\end{array}
$$

.

Algorithm 5.2

$Hom_R(M,N) = Im\,(modulo\,(A_0^T \otimes id_{G_0},\, id_{F_1^*} \otimes B_0))\,/\,Im\,(id_{F_0^*} \otimes B_0)$

is a free module modulo the image of the matrix

$$modulo\,(modulo\,(A_0^T \otimes id_{G_0},\, id_{F_1^*} \otimes B_0)\,,\, id_{F_0^*} \otimes B_0)$$

5.3 T_X^1 and $T_X^1(M)$

This section and the corresponding SINGULAR procedure is a joint work of G.M. Greuel and B.Martin.

Let $X \in K^n$ be given by the ideal $I = (f_1, \ldots, f_k) \subseteq P = K[x_1, \ldots, x_n]$ and $R = P/I$.

Consider the canonical exact sequence

$$I/I^2 \xrightarrow{d} \Omega_P^1 \otimes_P R \longrightarrow \Omega_R^1 \to 0 \qquad (1)$$

where Ω^1_P denotes the module of Kähler differentials and d is induced by the exterior derivation. ($[d] \mapsto [df]$).

Ω^1_P is free with generators $dx_1, \ldots, dx_n$ (isomorph to P^n) and Ω^1_R (isomorph to R^n) is by definition the cokernel of the map

$$d : [f_i] \to [df_i] = [\sum_j (df_i/dx_j)]$$

Let M be any finitely generated R-module . The M-dual of (1) is

$$0 \to \Theta_X(M) \to Hom_R(\Omega_P \otimes R, M) \xrightarrow{d^*} Hom_R(I/I^2, M) \to T^1_X(M) \to 0$$

where $\Theta_X(M) = Hom_R(\Omega^1_R, M)$ is the module of the M-valued R-derivations and $T^1_X(M)$ is by definition the cokernel of d^* (the dual of d) and is called the **module of first order infinitesimal deformations** of X (resp. R) with values in M.

The module $T^1_{X/K^n}(M) := Hom_R(I/I^2, M)$ is called the **module of first order embedded deformations** of R with values in M. If M is ommitted, we define $T^1_X := T^1_X(R)$ as the **module of first order deformations of** X.

Remark: $Hom_R(\Omega^1_R, R) \cong Hom_P(R^n, R) \cong R^n$.

Algorithm 5.3 *Consider a presentation of I as a P-module:*

$$0 \leftarrow I \leftarrow P^k \xleftarrow{A} P^p \tag{2}$$

We note that for any R-module M

$$Hom_P(I/I^2, M) = Hom_R(I, M) \tag{3}$$

Hence, choosing $dx_1, \ldots, dx_n$ as a basis of Ω^1_P and the canonical basis of P^k, the right part of (3) is identified with the exact sequence

$$Hom_P(P^n, R) \xrightarrow{jac} Hom_P(I, M) \to T^1_X(M) \to 0 \tag{4}$$

where $jac : Hom_P(P^n, M) \to Hom_P(I, M) \subseteq Hom_P(P^k, M) = M^k$ is given by the Jacobian matrix $(df_i/dx_j)_{i,j}$ of I. In particuliar, for $M = R$, we get

$$Hom_P(P^n, R) \xrightarrow{jac} Hom_P(I, R) \to T^1_X \to 0 \tag{5}$$

as defining sequence of T^1_X.

Applying $Hom_P(-,R)$ *to (2), we get*

$$0 \to Hom_P(I,R) = ker(A^*) \to Hom_P(P^k,R) \xrightarrow{A^*} Hom_P(P^p,R) \quad (6)$$

where A^* *is the transposed matrix of* A. *Consider a 3-term partial resolution of* $im(A^*)$*:*

$$R^q \xrightarrow{B_3} R^r \xrightarrow{B_2} Hom_P(P^k,R) \overset{B_1:=A^*}{\longrightarrow} Hom_P(P^p,R) \quad (7)$$

together with the $J : Hom_P(P^n,R) \to Hom_P(P^k,R)$ *(induced by the Jacobian jac) and a lifting of* J *to a map* $L : Hom(P^n,R) \to R^r$. *This lifting exists since* $im(jac)$ *is contained in the normal bundle of* I*:*

$$im(J) \subseteq Hom_R(I/I^2,R) = ker(B_1) = im(B_2) \quad (8)$$

Finally we get (keeping notations for B_3 *and* L *when lifted to* P*)*

$$T^1_X = im(B_2)/im(J) = R^r/_{im(L)+im(B_3)} = P^r/_{im(L)+im(B_3)+I*P^r} \quad (9)$$

Singular example: a complete version can be found in the Singular library `sing.lib`.

```
ideal I=f1,...,fk;
list A=res(I,2); //compute the presentation (4) of I
module A'=transpose(A[2]); //A*=transposed 1st syzygy
                           //module of I
module jac=jacob(I); //jacobian matrix of I (as module)

// So far we are in the polynomial ring P,
// now we pass to the qring R=P/I:

qring R=std(I); //defines the quotient ring R=P/I
module A'=fetch(P,A'); //map A* to R
module J=fetch(P,jac); //map jac to R
list B=res(A',3); //compute the exact sequence (7)
module t1=lift(B[2],jac)+B[3]; //im(L)+im(B3)
```

```
int r=rank(t1); //compute the rank r
// Hence  T1_X = R^r/t1 as R-module. (see (9))

// Now we pass back to the original basering P:
setring P; //makes P the basering
module t1 = fetch(R,t1)+J*freemodule(r);
                          //im(L)+im(B3)+J*P^r=:T1
fetch(R,B(2)); // (generators of) normal bundle
fetch(R,B(3)); // presentation of normal bundle
```

References

[Ba] Bayer, D.: The division algorithm and the Hilbert scheme. Thesis, Harvard Univ. 1982.

[BS] Bayer, D.; Stillman, M.: Macaulay (Version 3.0). A computer algebra system for algebraic geometry.

[BW] Becker, T.; Weispfenning, V.: Gröbner Bases. A computational approach to commutative algebra. Springer–Verlag GTM 141 (1991).

[B1] Buchberger, B.: Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal. Thesis, Univ. Innsbruck, 1965.

[B2] Buchberger, B.: Gröbner bases: an algorithmic method in polynomial ideal theory, in N.K Bose (ed.) Recent trends in multidimensional system theory, Reidel (1985).

[CLO] Cox, D; Little, J.; O'Shea, D.: Ideals, Varieties and Algorithms. An Introduction to Computational Algebraic Geometry and Commutative Algebra. Springer 1992.

[E] Eisenbud, D.: Commutative Algebra with a view toward Algebraic Geometry. GTM 150 Springer, 1995.

[G] Gräbe, H.-G.: The tangent cone algorithm and homogenization. To appear in J. Pure Appl. Alg.

[GG] Grassmann, H; Greuel, G.-M.; Martin, B.; Neumann, W.; Pfister, G.; Pohl, W.; Schönemann, H.; Siebert, T.: Standard bases, syzygies and their implementation in SINGULAR. Preprint 251, Fachbereich Mathematik, Universität Kaiserslautern 1994.

[GM] Gebauer, R.; Möller, M.: On an installation of Buchberger's Algorithm. J. Symbolic Computation (1988) **6**, 275–286.

[GMNRT] Giovini, A.; Mora, T.; Niesi, G.; Robbiano, L.; Traverso, C.: "One sugar cube, please" or selection strategies in the Buchberger algorithm. Proceedings of the 1991 ISSAC, 55-63.

[GTZ] Gianni, P.; Trager, B.;Zacharias, G.: Gröbner bases and Primary Decomposition of Polynomial Ideals. Journal of Symbolic Computation. 1985.

[L] Lazard, D.: Gröbner bases, Gaussian elimination, and resolution of systems of algebraic equations. Proc. EUROCAL 83, LN Comp. Sci. **162**, 146–156.

[M1] Mora, T.: An algorithm to compute the equations of tangent cones. Proc. EUROCAM 82, Springer Lecture Notes in Computer Science (1982).

[M2] Mora, T.: Seven variations on standard bases. Preprint, Univ. Genova (1988).

[M3] Mora, T.: La Queste del Saint Graal: a computational approach to local algebra. Discrete Applied Math. **33**, 161-190 (1991).

[MMT] Möller, H.M.; Mora, T.; Traverso, C.: Gröbner bases computation using syzygies. Proc. of ISSAC 1992.

[MPT] Mora, T.; Pfister, G.; Traverso, C.: An introduction to the tangent cone algorithm . Advances in Computing research, Issues in Robotics and nonlinear geometry (**6**) 199–270 (1992).

[PS] Pfister, G.; Schönemann, H.: Singularities with exact Poincaré complex but not quasihomogeneous. Rev. Mat. de la Univ. Complutense de Madrid **2** (1989).

[R] Robbiano, L.: Termorderings on the polynomial ring. Proceedings of EUROCAL 85, Lecture Notes in Computer Science **204**, 513–517 (1985).

[S] Schreyer, F.-O.: A standard basis approach to syzygies of canonical curves. J. reine angew. Math. **421**, 83-123 (1991).

[St1] Stillman, M.: Methods for computing in algebraic geometry and commutative algebra. Acta Applicandae Mathematicae 21(77-103) 1990.

[St2] Stillman, M.: Macaulay. A tutorial. 1992.

[Si] Singular reference manual. Version 0.9.2. 1995. Available from ftp://helios.mathematik.uni-kl.de/pub/Math/Singular/bin

Hannes Schönemann,
Department of Mathematics,
University of Kaiserslautern
Germany
e-mail: *hannes@mathematik.uni − kl.de*

Received 27 November, 1996