UNIVERSITÄT KAISERSLAUTERN
Zentrum für Computeralgebra

REPORTS ON COMPUTER ALGEBRA
NO. 01

Standard bases, syzygies and their implementation in singular

by

H. Grassmann, G.-M. Greuel, B. Martin, W. Neumann, G. Pfister, W. Pohl, H. Schönemann, and T. Siebert

July 1996

The Zentrum für Computeralgebra (Centre for Computer Algebra) at the University of Kaiserslautern was founded in June 1993 by the Ministerium für Wissenschaft und Weiterbildung in Rheinland-Pfalz (Ministry of Science and Education of the state of Rheinland-Pfalz). The centre is a scientific institution of the departments of **Mathematics, Computer Science, and Electrical Engineering** at the University of Kaiserslautern.
The goals of the centre are to advance and to support the use of Computer Algebra in industry, research, and teaching. More concrete goals of the centre include

- the development, integration, and use of software for Computer Algebra
- the development of curricula in Computer Algebra under special consideration of interdisciplinary aspects
- the realisation of seminars about Computer Algebra
- the cooperation with other centres and institutions which have similar goals

The present coordinator of the Reports on Computer Algebra is:
Olaf Bachmann (email: `obachman@mathematik.uni-kl.de`)

Zentrum für Computeralgebra
c/o Prof. Dr. G.-M. Greuel, FB Mathematik
Erwin-Schrödinger-Strasse
D-67663 Kaiserslautern; Germany
Phone: 49 - 631/205-2850 Fax: 49 - 631/205-5052
email: `greuel@mathematik.uni-kl.de`
URL: `http://www.mathematik.uni-kl.de/~zca/`

Standard bases, syzygies and their implementation in SINGULAR

*Grassmann, H.; Greuel, G.-M.;
Martin, B.; Neumann, W.;
Pfister, G.; Pohl, W.;
Schönemann, H.; Siebert, T.

Herrn Prof. Dr. H. Brakhage
aus Anlaß seiner Emeritierung gewidmet

Contents

* **Acknowledgement**: The authors were partially supported by the VW-Stiftung, the Stiftung für Innovation Rheinland–Pfalz, the DFG and the ESPRIT BRA contract 6846 POSSO.

Introduction

The aim of this article is to describe recent advances and improvements on the tangent cone algorithm of T. Mora. This tangent cone algorithm is itself a variant of B. Buchberger's celebrated algorithm for constructing a Gröbner basis of an ideal in a polynomial ring over a field. In the same manner as the knowledge of a Gröbner basis allows the computation of numerous invariants of the coordinate ring of a projective algebraic variety, a standard basis (computed by the tangent cone algorithm) does so for invariants of the local ring of an algebraic variety at a given point. In this paper we describe a generalization which includes Buchberger's and Mora's algorithm as special cases. That is, we prove — with an appropriate definition of ecart — that Mora's algorithm terminates for any ordering on the monomials of $K[x_1, \ldots, x_n]$, which is compatible with the natural semigroup structure (a fact which was found independently by Gräbe [G]), in particular, the variables may have as well negative, positive or zero weights (cf. §1). This generalization has already had mathematical applications for examples to deformations of projective varieties or to the theory of moduli spaces for singularities, which will be published independently. Moreover, more or less all algorithms using Gröbner bases (such as computation of syzygies, ideal theoretic operations, etc.) are now available in this general context. Our generalization provides also an easy manner to implement standard bases for modules over the Weyl algebra (cf. §6).

This general standard basis algorithm is described in §1, its implementation in the computer algebra system SINGULAR in §2, which has been developed by the authors over the last few years. The need for an efficient implementation of the tangent cone algorithm arose from applications in the theory of singularities. First of all there was the success in disproving a conjectured generalization of a theorem of K. Saito (cf. [PS]). Later, the attempt to attack Zariski's multiplicity question was one of the most stimulating points. Indeed, none of the existing computer algebra systems were able to compute a series of possible counter examples. Therefore, we had to invent special strategies in particular for zero–dimensional ideals in the local case.

These strategies and their implementation in SINGULAR are described in §2. The most relevant new strategies are the HCtest and the ecartMethod. The HCtest has dramatic impact on the efficiency of the algorithm for zero–dimensional ideals in the local case. It was first found by Pfister and Schönemann and has been implemented in a forerunner of SINGULAR since 1985. It consists of the computation of the minimal monomial not contained in the initial ideal and discarding all bigger monomials in further computations. The Pfister–Schönemann HCtest has also proved to be fairly successful in an early implementation of the tangent cone algorithm by Zimnol [Zi] (which was based on ideas of H. Brakhage).

The ecartMethod consists of the choice of a weight vector of positive integers such that the weighted ecart of the input polynomials, with respect to this vector, become as small as possible. Of course, this method is most useful if the system offers an automatic choice of the ecart vector. In SINGULAR the automatic choice is realized by optimizing a certain functional but the user also has the option to choose his own ecart vectors. The ecartMethod and the functional will be described in more detail in [Po], together with the weighted sugarMethod and another method to compute weights for the variables, which applies in the classifical Buchberger case.

Whilst the HCtest applies only to 0–dimensional ideals, the ecartMethod works in any case and is usually superior to other strategies. But we noticed that computing a standard basis with respect to the lexicographical ordering is extremely fast in the local case. However, the meaning of this ordering is less clear (it is not an elimination ordering in this case). Nevertheless, it can be used to compute a lexicographical standard basis and then to recompute a standard basis from this one with respect to the desired ordering. In many cases this is surprisingly fast. §3 contains a comparison of these strategies by giving the timing for many examples.

In §4 we prove that Schreyer's method to compute syzygies generalizes to arbitrary semigroup orderings. It seems to be the first algorithmic proof of the fact that the length of a free resolution is equal to the number of variables which actually occur in the equations (and not on all variables of the ring) in the local and mixed local–global case. It follows basically Schreyer's original proof [S1] but contains some new ideas, since Macaulay's lemma, which is usually applied, does not hold for orderings which are not well–orderings. We describe the implementation in SINGULAR, which is the first implementation even in the classical Buchberger case. We believe that this algorithm, and further improvements, will have theoretical as well as practical advantages in many cases of interest, in particular in cases which are far from a complete intersection. On the other hand, it does not give a minimal resolution and computing a minimal resolution step by step is sometimes more efficient. We describe some essentials for an efficient implementation and include a table of experiments with comparisons of different strategies.

Chapter §5 contains a partial positive answer to Zariski's multiplicity conjecture or question. This question is concerned with the topological nature of the algebraically defined multiplicity of a hypersurface singularity and belongs certainly to the most outstanding open problems in singularity theory. Although there are partial positive answers, e.g. by Zariski, Lê, Lipmann, Laufer, O'Shea, Yau and the second named author, it has basically resisted all attacks. Our partial result, which supports the conjecture, was prompted by computer experiments with SINGULAR in order to find a counter example. The proof (given in §5) does not use any computer computation but the computer experiments were essential in guessing the result. In the last chapter we include for completeness a proof that the module of leading terms (with respect to any semigroup ordering) is a flat specialization of the original module. This is the basis of most applications, e.g. for computing Milnor numbers or multiplicities and Hilbert functions of singularities.

SINGULAR is implemented in C and C++. A test version is available unter ftp from saturn.mathematik.uni-kl.de. The ground fields for standard basis computations are $\mathbb{Z}/p\mathbb{Z}$ (p a small prime number $< 2^{16}$), $\mathbb{Q}$, $\mathbb{Q}[Z]/(f)$, $(\mathbb{Z}/p\mathbb{Z})[Z]/(f)$ (f irreducible), $\mathbb{Q}(Z, Y, X, \ldots)$ and $(\mathbb{Z}/p\mathbb{Z})\ (Z, Y, X, \ldots)$ (finitely many parameters). Of course, the computations in algebraic or transcendental extensions are very time consuming. SINGULAR contains, besides the usual functions such as Krull dimension, Hilbert function and ideal theoretic operations, also several libraries with useful procedures (such as degree for projective varieties, Milnor numbers and Tjurinia numbers for isolated complete intersection singularities, T^1 and T^2 for affine varieties and isolated singularities, etc.). Finally, there is a quite capable interface with online help and comfortable programming facilities.

1 A standard basis algorithm for any semigroup ordering

This algorithm is a generalization of Buchberger's algorithm (which works for wellorderings cf. [B1], [B2]) and Mora's tangent cone algorithm (which works for tangent cone orderings, cf. [M1], [MPT]) and which includes a mixture of both (which is useful for certain applications, in particular to algorithms described by Mora in his Graal paper [M3]). In fact, it is an easy extension of Mora's idea by introducing the "correct" definition of ecart. But we present it in a new way which, as we hope, makes the relation to the existing standard basis algorithms transparent.

Let K be a field, $x = (x_1, \ldots, x_n)$ and α, β, γ column vectors in $\mathbb{N}^n$, $\mathbb{N} = \{0, 1, 2, \ldots\}$. Let $<$ be a **semigroup ordering** on the set of monomials $\{x^\alpha | \alpha \in \mathbb{N}^n\}$ of $K[x]$, that is, $<$ is a total ordering and $x^\alpha < x^\beta$ implies $x^\gamma x^\alpha < x^\gamma x^\beta$ for any $\gamma \in \mathbb{N}^n$. Robbiano (cf. [R]) proved that any semigroup ordering can be defined by a matrix $A \in GL(n, \mathbb{R})$ as follows:

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 $\mathbb{R}^n$.

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

Definition 1.1 *We define* $\mathbf{Loc_< K[x]} := S_<^{-1}K[x]$ *to be the localization of* $K[x]$ *with respect to the multiplicative closed set* $S_< := \{1 + g \mid g = 0$ *or* $g \in K[x]\backslash\{0\}$ *and* $1 > L(g)\}$.

Remark 1.2 1) $K[x] \subseteq \mathrm{Loc}_< K[x] \subseteq K[x]_{(x)}$, where $K[x]_{(x)}$ denotes the localization of $K[x]$ with respect to the maximal ideal $(x_1, \ldots, x_n)$. In particular, $\mathrm{Loc}_< K[x]$ is noetherian, $\mathrm{Loc}_< K[x]$ is $K[x]$–flat and $K[x]_{(x)}$ is $\mathrm{Loc}_< K[x]$–flat.

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

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

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

hence

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

4) Let $<$ be an **elimination ordering** for $x_{r+1}, \ldots, x_n$ (that is $L(g) \in K[x_1, \ldots, x_r]$ implies $g \in K[x_1, \ldots, x_r]$), then $x^\alpha < 1$ implies $x^\alpha \in K[x_1, \ldots, x_r]$. In particular, $<$ is necessarily a wellordering on the set of monomials in $K[x_{r+1}, \ldots, x_n]$. Note that lex^+ below eliminates but lex^- does not.

Important orderings for applications are:

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

$$\mathrm{lex} \text{ or } \mathrm{lex}^+ : \begin{pmatrix} 1 & & & \\ & 1 & & 0 \\ & & \ddots & \\ & 0 & & 1 \end{pmatrix} \quad \text{resp. } \mathrm{lex}^- : \begin{pmatrix} -1 & & & \\ & -1 & & 0 \\ & & \ddots & \\ & 0 & & -1 \end{pmatrix}$$

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

$$W^+ : \begin{pmatrix} w_1 & w_2 & \dots & w_n \\ & & & -1 \\ & & \iddots & \\ 0 & -1 & & \end{pmatrix}, w_i > 0 \text{ for all } i, \text{ (resp. } W^- : w_i < 0 \text{ for all } i).$$

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

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

$$\begin{pmatrix} 0 & 0 & \dots & 0 & w_{r+1} & w_{r+2} & \dots & w_n \\ w_1 & w_2 & \dots & w_r & 0 & 0 & \dots & 0 \\ & & & & & & & -1 \\ & & & & & & \iddots & \\ & & & & 0 & -1 & & \\ & & & -1 & & & & \\ & & \iddots & & & & & \\ 0 & -1 & & & & & & \end{pmatrix}$$

  with $w_1 > 0, \dots, w_n > 0$. In $K[x_1, \dots, x_r]_{(x_1,\dots,x_r)}[x_{r+1}, \dots, x_n] = \mathrm{Loc}_< K[x]$ 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 **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 $\{x_1, \dots, x_n\}$.

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. In the positive (respectively negative) case $\mathrm{Loc}_< K[x] = K[x]$ (respectively $\mathrm{Loc}_< K[x] = K[x]_{(x)}$).

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

We now fix an ordering $<_m$ on $K[x]^r$ 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:

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

Definition 1.3 *Let* $I \subseteq K[x]^r$ *be a submodule.*

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

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

3) *A standard basis* $f_1, \dots, f_s$ *is called* **reduced** *if, for any* i, $L(f_i)$ *does not divide any of the monomials of* $f_1, \dots, f_s$ *(except itself).*

Proposition 1.4 *If* $\{f_1, \dots, f_s\}$ *is a standard basis of* I *then* $I\mathrm{Loc}_< K[x] = (f_1, \dots, f_s)\mathit{Loc}_< K[x]$.

Note that a reduced standard basis of polynomials does not necessarily exist (cf. Remark 1.12).

The proof will be deduced from the normal form used in the standard basis algorithm (cf. Corollary 1.11). Note that, in general, it is not true that $f_1, \dots, f_s$ generate I as $K[x]$–module (take $I = (x)K[x]$, $n = 1$, $f = x + x^2$ with lex^-).

This is also not true if $I \subset K[x]$ is $(x_1, \dots, x_n)$–primary and if $\{f_1, \dots, f_s\}$ is a reduced standard basis: Consider the ideal $I \subset K[x, y]$ generated by $x^{10} - y^2x^9$, $y^8 - x^2y^7$, $x^{10}y^7$ which is (x, y)–primary. The first two elements are a reduced standard basis of $I\,\mathrm{Loc}_< K[x, y] = I\,K[x, y]_{(x,y)}$ where $<$ is $\mathrm{degrevlex}^-$ und hence generate $I\,K[x, y]_{(x,y)}$ but they do not generate $I\,K[x, y]$. (This answers a question of T. Mora.)

Notations:

Let $f, g \in K[x]^r$, $L(f) = x^\alpha e_i$ and $L(g) = x^\beta e_j$. If $i = j$ and $x^\alpha | x^\beta$ then we write $L(f)|L(g)$.

If $i = j$ and $x^\gamma = \mathrm{lcm}(x^\alpha, x^\beta)$, $\gamma = (\max(\alpha_1, \beta_1), \dots, \max(\alpha_n, \beta_n))$ then

$$\begin{aligned} \mathbf{lcm}(\mathbf{L}(\mathbf{f}), \mathbf{L}(\mathbf{g})) &:= x^\gamma \text{ and} \\ \mathbf{spoly}(\mathbf{f}, \mathbf{g}) &:= x^{\gamma-\alpha} f - \tfrac{c(f)}{c(g)} x^{\gamma-\beta} g. \end{aligned}$$

If $i \neq j$ then, by definition, $L(f) \not| L(g)$, $\mathrm{spoly}(f, g) := 0$ and $\mathrm{lcm}(L(f), L(g)) := 0$.

Let $\mathcal{F} = \{G \subseteq K[x]^r | G \text{ finite and ordered }\}$.

Definition 1.5 *A function* $\mathsf{NF} : K[x]^r \times \mathcal{F} \to K[x]^r, (p, G) \mapsto \mathsf{NF}(p|G)$, *is called a* **normal form** *if for any* $p \in K[x]^r$ *and any* $G \in \mathcal{F}$ *the following holds: if* $\mathsf{NF}(p|G) \neq 0$ *then* $L(g) \not| L(\mathsf{NF}(p|G))$ *for all* $g \in G$. $\mathsf{NF}(g|G)$ *is called the* **normal form of p with respect to G**.

Example 1.6 Let $<$ be a wellordering then the following procedure NFBuchberger is a normal form:

$h :=$ **NFBuchberger** $(\mathbf{p}|\mathbf{G})$

```
h := p
WHILE exist f ∈ G such that L(f)|L(h) DO
    choose the first f ∈ G with this property
    h := spoly(h, f)
END.
```

The principle for many standard basis algorithms depending on a chosen normal form is the following:

$S :=$ **Standard** $(\mathbf{G}, \mathbf{NF})$

```
S := G
P := {(f, g)|f, g ∈ S}
WHILE P ≠ ∅ DO
    choose (f, g) ∈ P; P := P\{(f, g)}
    h := NF(spoly (f, g) | S)
    IF h ≠ 0 THEN
        P := P ∪ {(h, f) | f ∈ S}
        S := S ∪ {h}
    END
END
```

In this language **Buchberger's algorithm** is just

$$\mathbf{Buchberger}(\mathbf{G}) = \text{ Standard } (\mathtt{G},\ \text{NFBuchberger}).$$

If $<$ is any ordering (not necessarily a wellordering) and A the corresponding matrix, then the matrix

$$\begin{pmatrix} 1 & 1\ldots1 \\ 0 & \\ \vdots & A \\ 0 & \end{pmatrix}$$

defines a wellordering on the monomials of $K[t,x]$ which we denote also by $<$. For $f \in K[x]$ let f^h be the **homogenization** of f with respect to t and for $G \subseteq K[x]$ let $G^h = \{f^h \mid f \in G\}$. If $f \in K[x]^r$, $f = \sum f_i e_i$ then $f^h = \sum t^{\alpha_i} f_i^h e_i$, deg $f_i^h + \alpha_i =$ deg $f_j^h + \alpha_j$ for all i, j and the α_i minimal with this property. Similarly, we define $G^h = \{f^h \mid f \in G\}$ for $G \subseteq K[x]^r$.

This ordering has the following property:

Lemma 1.7 *If $t^\alpha > x^\gamma$, for some $\gamma = (\gamma_1, \ldots, \gamma_n)$, and $\alpha = \gamma_1 + \cdots + \gamma_n$ then $x^\gamma < 1$. Especially, $<$ is not a wellordering in this case on $K[x]$.*

The **Lazard method** (cf. [L]) to compute a standard basis is the following:

S := **Lazard** (**G**)

```
S := G^h
S := Buchberger (S)
S := S(t = 1)
```

Remark 1.8 The result S is a standard basis of the submodule $\langle G\rangle$ generated by G in $K[x]^r$ with the additional property that $\langle G\rangle$ is generated by S as $K[x]$–module (we need not pass to $\mathrm{Loc}_< K[x]$!). If we are only interested in a standard basis of $\langle G\rangle$ this algorithm computes usually too much and this might be the reason why it is often too slow (although there are cases where it is surprisingly fast).

Mora found for tangent cone orderings (cf. [M1], [MPT]) an algorithm which computes a standard basis over $\mathrm{Loc}_< K[x]$. This algorithm can be generalized to any ordering and we can describe it as follows:

S := **Standard basis** (**G**)

```
S := G^h
S := Standard (S, NFMora)
S := S(t = 1)
```

Let $G \subseteq K[t,x]^r$ be a finite set of homogeneous elements and $p \in K[t,x]^r$ homogeneous. Note that an element of $K[t,x]^r$ is **homogeneous** if its components are homogeneous polynomials of the same degree. The generalization of Mora's normal form to any semigroup ordering is as follows:

```
h := NFMora (p|G)

    h := p
    T := G
    WHILE exist f ∈ T, such that L(f) | t^α L(h) for some α DO
        choose f ∈ T with L(f) | t^α L(h) and α minimal
        IF α > 0 THEN
              T := T ∪ {h}
        END
        h := spoly (t^α h, f)
        IF t | h THEN
           choose α maximal such that t^α divides h
           h := h / t^α
        END
    END
```

Proposition 1.9 *1) NFMora terminates.*

2) If h is a normal form of p with respect to $G = \{f_1, \ldots, f_s\}$ computed by NFMora then there are homogeneous polynomials $g,\ \xi_1, \ldots, \xi_s \in K[t,x]$ such that

- $gp = \sum \xi_i f_i + h$
- $L(g) = t^\alpha$
- *deg* $p + \alpha = \deg\ \xi_i + \deg\ f_i = \deg(h)$ *(if $\xi_i \neq 0,\ h \neq 0$)*
- $L(f_i) \not| t^\alpha L(h)$ *for all* i, α

If $<$ is a wellordering on $K[x]$ then $g = t^\alpha$.

Proof: 2) By induction suppose that after the ν–th step in NFMora we have

$$g_\nu p = \sum \xi_{i\nu} f_i + h_\nu,$$

and

- $L(g_\nu) = t^{\alpha_\nu}$,
- $\deg\ p + \alpha_\nu = \deg\ \xi_{i,\nu} + \deg\ f_i = \deg\ h_\nu$ (if $\xi_{i,\nu} \neq 0,\ h_\nu \neq 0$)
- $t^{-\alpha_\mu} L(h_\mu) > t^{-\alpha_\nu} L(h_\nu)$ for $\mu < \nu$.

If $L(f_i) \not| t^\alpha L(h_\nu)$ for all i, α then we have finished.

Since T consists of elements $f_k \in G$ and of h_μ constructed in previous steps we have to consider two cases:

- If $L(f_k) \mid t^\alpha L(h_\nu)$ and α is minimal for all possible choices for $f_k \in G$ then

$$t^\alpha g_\nu p = \sum t^\alpha \xi_{i\nu} f_i + t^\alpha h_\nu - \eta f_k + \eta f_k$$

with $L(f_k)\eta = t^\alpha L(h_\nu)$.

We obtain

$$\begin{aligned} h_{\nu+1} &= t^{\alpha}h_{\nu} - \eta f_k \\ g_{\nu+1} &= t^{\alpha}g_{\nu} \\ \xi_{i\nu+1} &= t^{\alpha}\xi_{i\nu} \quad \text{if } \nu \neq k \\ \xi_{k\nu+1} &= t^{\alpha}\xi_{k\nu} + \eta \end{aligned}$$

and the induction step follows with $\alpha_{\nu+1} = \alpha + \alpha_{\nu}$.

- If $L(h_{\mu}) \mid t^{\alpha}L(h_{\nu})$ for some $\mu < \nu$ and α is minimal for all possible choices from T then

$$t^{\alpha}g_{\nu}p = \sum t^{\alpha}\xi_{i\nu}f_i + t^{\alpha}h_{\nu} - \eta h_{\mu} + \eta h_{\mu}$$

with $L(h_{\mu})\eta = t^{\alpha}L(h_{\nu})$.

$$\begin{aligned} h_{\nu+1} &= t^{\alpha}h_{\nu} - \eta h_{\mu} \\ g_{\nu+1} &= t^{\alpha}g_{\nu} - \eta g_{\mu} \\ \xi_{i\nu+1} &= t^{\alpha}\xi_{i\nu} - \eta\xi_{i\mu}. \end{aligned}$$

Now $t^{\alpha_{\nu}-\alpha_{\mu}}L(h_{\mu}) > L(h_{\nu})$ implies $t^{\alpha+\alpha_{\nu}} > L(\eta)t^{\alpha_{\mu}}$, that is $t^{\alpha+\alpha_{\nu}} = L(g_{\nu+1})$.

This proves 2).

To prove 1) let $I_{\nu} = \langle L(f) \mid f \in T_{\nu}\rangle$, T_{ν} be the set T after the ν–th reduction. Let N be an integer such that $I_N = I_{N+1} = \ldots$ (such N exists because $K[t,x]^r$ is noetherian). This implies $T_N = T_{N+1} = \ldots$. The algorithm continues with fixed T and terminates because $<$ is a wellordering on $K[t,x]^r$.

Remark 1.10 1) If the ordering $<$ on $K[x]$ is a wellordering, then the standard basis algorithm is essentially Buchberger's algorithm because then $x^{\alpha} \mid x^{\beta}$ implies $x^{\alpha} < x^{\beta}$. This shows that only elements from G are used for the reduction in NFMora. Moreover, if G is homogeneous but $<$ arbitrary, the standard basis algorithm coincides with Buchberger's algorithm.

2) If $<$ is a tangent cone ordering then the algorithm is Mora's tangent cone algorithm. In his algorithm Mora uses the same normal form, just in another language. Instead of passing from $K[x]$ to $K[t,x]$ by homogenizing and extending the ordering, he uses the notion of **ecart**, where $\text{ecart}(p) = \deg_t(p^h)$. During the implementation of SINGULAR we discovered that the normal form with $\text{ecart}(p) := \deg_t(L(p^h))$ terminates for any ordering, not only for tangent cone orderings. This was found also by Gräbe (cf. [G]).

Corollary 1.11 *Let $S = \{f_1, \ldots, f_s\}$ be a finite subset of the submodule $I \subseteq K[x]^r$.*

1) If S is a standard basis of I then:

(i) For any $f \in K[x]^r$ there are $g, \xi_i \in K[x]$, $h \in K[x]^r$, such that

$$(1+g)f = \sum \xi_i f_i + h,$$

$L(g) < 1$ *if* $g \neq 0$, $L(\xi_i f_i) \leq L(f)$ *if* $\xi_i \neq 0$ *and, for all* i, $L(f_i) \not| L(h)$.

(ii) $f \in I$ *if and only if NFMora* $(f^h \mid S^h) = 0$.

(ii') $f \in I$ *if and only if* $(1+g)f = \sum \xi_i f_i$ *for suitable* $g, \xi_i \in K[x]$, $L(g) < 1$ *if* $g \neq 0$ *and* $L(\xi_i f_i) \leq L(f)$ *if* $\xi_i \neq 0$.

(iii) $I\, Loc_< K[x] = \langle S\rangle\, Loc_< K[x]$.

2) The following are equivalent:

(i) S *is a standard basis of* I.

(ii) $S^h =$ *Standard* (S^h, *NFMora*).

(iii) *NFMora* (*spoly*(f,g), $S^h) = 0$ *for all* $f, g \in S^h$.

The corollary is an easy consequence of 1.9.

Remark 1.12 1) If one extends the ordering $<$ given by the matrix A on $K[x]$ to $K[t,x]$ by

$$\begin{pmatrix} 1 & w_1,\dots,w_n \\ 0 & \\ \vdots & A \\ 0 & \end{pmatrix}, \; w_i > 0$$

and use homogenization with respect to the weights $w_1,\dots,w_n$ then the standard basis algorithm works as well.

Gräbe discovered (cf. [G]) that for a suitable choice of the weights adapted to the input (the polynomials should become as homogeneous as possible with respect to these weights) the algorithm can become faster. We call this the **(weighted) ecartMethod**.

2) If $<$ is a wellordering, we can apply the normal form algorithm to each monomial of h and we can achieve that for any $f \in K[x]^r$,

$$(*) \qquad f = \sum \xi_i f_i + h,$$

for suitable $\xi_i \in K[x]$, $h \in K[x]^r$ such that $L(\xi_i f_i) \le L(f)$ if $\xi_i \ne 0$ and, for all i, no monomial of h is divisible by $L(f_i)$; h is then unique.

If we try the same for an arbitrary semigroup ordering, this procedure will, in general, not terminate. We can only derive a presentation $(*)$ with $\xi_i \in K[[x]]$ and $h \in K[[x]]^r$ (formal power series) having the above properties.

3) A reduced standard basis is uniquely determined by I and $<$. If $<$ is a wellordering or $\dim_K Loc_< K[x]^r/I < \infty$ then there exists always a reduced standard basis in $K[x]^r$. In general, it exists only in $K[[x]]^r$.

2 The standard basis algorithm in SINGULAR

In SINGULAR the standard basis algorithm is implemented as follows:

$S :=$ **StandardBasis** $(\mathbf{G})$
INPUT: G, a set of polynomial vectors $\subset K[x]^r$
OUTPUT: S, a standard base of the submodule generated by G with respect to the given monomial ordering

```
S := G^h
update (S)
HCtest
T := S
L := initPairs (S)
clear (S)
WHILE L ≠ ∅ DO
    p := (a, b, s) the last element from L
    L := L\{(a, b, s)}
    IF the spoly of a and b is not computed yet THEN
        s := spoly (a, b)
    END
    h := LazyNF (p|T)
    IF h ≠ 0 THEN
        HCtest
        updatePairs (h)
        S := S ∪ {h}
        clear (S)
        T := T ∪ {h}
    END
END
S := completeReduce (S).
```

Remark 2.1 In cases when either $<$ is a wellordering, or $G \subseteq K[x]$ and $\dim_K K[x]/\langle G\rangle < \infty$ the standard basis is uniquely determined. Moreover, in this case we can achieve a presentation $(*)$ as in 1.12 (2) with $\xi_i \in K[x]$, $h \in K[x]^r$.

During the algorithm the set S is sorted by increasing monomial ordering of the leading terms of the elements with respect to

$$\begin{pmatrix} 0 & 0,\dots,0 \\ 0 & \\ \vdots & A \\ 0 & \end{pmatrix}.$$

The set T (respectively L) is sorted by increasing (respectively decreasing) monomial ordering of the leading terms of the elements (respectively the S-polynomial of the corresponding pair) with respect to

$$\begin{pmatrix} 1 & 1,\dots,1 \\ 0 & \\ \vdots & A \\ 0 & \end{pmatrix}$$

or

$$\begin{pmatrix} 1 & w_1, \dots, w_n \\ 0 & \\ \vdots & A \\ 0 & \end{pmatrix} \text{ if we use the weighted ecart method.}$$

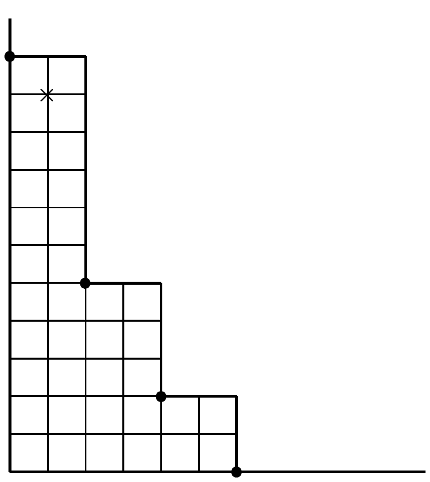
Now we explain the procedures used in the standard basis algorithm:

Update $(\mathbf{S})$
INPUT: S, a set of polynomial vectors
OUTPUT: S, the set of polynomial vectors after the interreduction

`for all` $s \in S$ `DO` $s :=$ `NFBuchberger` $(s \mid S \backslash \{s\})$.

HCtest
INPUT: S, a set of polynomials
OUTPUT: a boolean value (does a "highest corner" exist?) and, if so, the monomial "highest corner"
If $G \subseteq K[x]$ (that is $r = 1$) then it tests if there are $a_i, \alpha_i \geq 0$ such that $t^{a_i} x_i^{\alpha_i}$ occur as leading terms in S for all i. If this is true it computes the minimal monomial x^α (with respect to the ordering $<$ in $K[x]$) which is not in the ideal generated by the leading terms of the elements of $S(t = 1)$. This monomial is called ''highest corner''.
It changes the polynomial arithmetic to cancel all monomials $t^a x^\beta$ with $x^\beta < x^\alpha$ in further computations.

Remark 2.2 Notice that the highest corner x^α is equal to 1 in the case of a wellordering. Therefore, the procedure is called only if $x_i < 1$ for some i. The highest corner is computed in a combinatorial manner.



• correspond to leading terms of $S(t = 1)$, $\times$ corresponds to the "highest corner" x^α.

$\mathbf{L} :=$ **initPairs** $(\mathbf{S})$
INPUT: S, a set of (interreduced) polynomial vectors
OUTPUT: L, the set of critical pairs
It creates the pairset $L = \{(f, g, s) \mid f, g \in S$, s the leading term

of spoly$(f, g)\}$. Using the criteria similar to Gebauer-Möller (cf. [GM]) useless pairs are cancelled.
We have two options: usually the criteria are applied for the pairs $\{(f(t = 1),\ g(t = 1)) \mid f, g \in S\}$, that is in $K[x]$. In the option sugar crit we apply it to L using the idea of [GMNRT].

Clear $(\mathbf{S})$
Input and Output: S, a set of polynomial vectors
Deletes f from S if $L(g) \mid t^{\alpha}L(f)$ for some α and $g \in S$.

Update $(\mathbf{h})$
Input and Output: h, a polynomial vector

```
IF  sugar   THEN RETURN END
IF  t|h   THEN
    choose α maximal such that t^α|h
    h := h / t^α
END
```

UpdatePairs $(\mathbf{h})$
Input: h, a polynomial vector
S, a set of polynomial vectors
Output: L, the set of critical pairs from S and h

It updates the pairset L.
$L := L \cup \{(f, h, s) \mid f \in S,\ s$ the leading term of spoly$(f, h)\}$.
The criteria to cancel useless pairs are used as in initPairs.

$h :=$ **LazyNF** $((\mathbf{a}, \mathbf{b}, \mathbf{s}) \mid \mathbf{T})$
Input: s, a polynomial vector to reduce
(a, b), the critical pair from which s is the spoly
T, a set of polynomials with which to reduce
Output: h, the reduced polynomial vector

```
h := s
WHILE exist f ∈ T such that L(f) | t^α L(h) for some α  DO
    choose the first possible f with respect to the ordering
    in T such that α is minimal.
    IF  α > 0  THEN
        IF the position of (a,b,h) in L is not the last one
        THEN
            L := L ∪ {(a,b,h)}
            RETURN 0
        END
        T := T ∪ {h}
        NFupdate pairs (h)
    END
    h := spoly(t^α h, f)
    update (h)
    IF degree (h) > degree(s)   and
    the position of (a,b,h) in L is not the last one THEN
```

```
        L := L ∪ {(a, b, h)}
        RETURN 0
    END
END
```

NFupdatePairs $(\mathbf{h})$
INPUT: h, a polynomial vector
S, a set of polynomial vectors
OUTPUT: L the set of some critical pairs from S and h

```
IF NOT more pairs THEN RETURN END
L := L ∪ {(f, h, s) | f ∈ S, deg_t L(f) ≤ deg_t L(h), s the leading term of
spoly(f, h)}
The criteria to cancel useless pairs are used as in
initPairs.
```

Remark 2.3 The option **morePairs** (= `more pairs` in NFupdatePairs) has the effect of keeping some useless pairs in order to have better candidates for reduction. The leading idea is to keep (some of) those pairs which could not be discarded if we had used Lazard's method. In the case of a non–wellordering this has turned out to be useful, since it can help in not creating polynomials with too long tails during the normal form computation.

$\mathbf{S} :=$ **completeReduce** $(\mathbf{S})$
INPUT and OUTPUT: S, a set of polynomial vectors

```
IF < is a wellordering or (S ⊆ K[x] and dim_K K[x]/⟨S⟩ < ∞)
THEN
    S := S(t = 1)
    FOR all s ∈ S DO
        s := redtail (s|S)
    END
ELSE
    FOR all s ∈ S DO
        s := redtail (s|S)
    END
    S := S(t = 1)
END
```

$\mathbf{s} :=$ **redtail** $(\mathbf{s}|\mathbf{S})$
INPUT and OUTPUT: s, a polynomial vector

```
reduces the monomial below L(s) with elements from S as long as
possible.
```

3 Examples and comparisons

Let

$$\begin{array}{lll} f_t^{a,b,c} & := & x^a + y^b + z^{3c} + x^{c+2}y^{c-1} + x^{c-1}y^{c-1}z^3 + x^{c-2}y^c(y^2+tx)^2 \\ h^a & := & x^a + y^a + z^a + xyz(x+y+z)^2 + (x+y+z)^3 \\ g_t^{a,b,c,d,e} & := & x^a + y^b + z^c + x^d y^{e-5} + x^{d-2}y^{e-3} + x^{d-3}y^{e-4}z^2 + x^{d-4}y^{e-4}(y^2+tx)^2 \end{array}$$

These three series stem from our attempts to disprove Zariski's conjecture.

Examples:

1. Alex 1
   $I = (5t^3x^2z + 2t^2y^3x^5,\ 7y + 4x^2y + y^2x + 2zt,\ 3tz + 3yz^2 + 2yz^4)$
2. $I = (x^3y^2 - \frac{22}{3}x^5y - \frac{2}{3}x^2y^4 - \frac{22}{3}x^2yz^3 - \frac{5}{3}xy^6 - \frac{2}{3}y^9,$
   $1x^2y^2z^2 + 3z^8, 5x^4y^2 + 4xy^5 + 2x^2y^2z^3 + 1y^7 + 11x^{10})$
3. $f = f_1^{11,10,3},\ I = \left(\frac{\partial f}{\partial x},\ \frac{\partial f}{\partial y},\ \frac{\partial f}{\partial z}\right)$
4. $f = h^6,\quad I = \left(\frac{\partial f}{\partial x},\ \frac{\partial f}{\partial y},\ \frac{\partial f}{\partial z}\right)$
5. $f = h^7,\ I = \left(\frac{\partial f}{\partial x},\ \frac{\partial f}{\partial y},\ \frac{\partial f}{\partial z}\right)$
6. $f = g_0^{6,8,10,5,5},\ I = \left(f, \frac{\partial f}{\partial x},\ \frac{\partial f}{\partial y},\ \frac{\partial f}{\partial z}\right)$
7. $f = g_0^{6,8,10,5,5},\ I = \left(\frac{\partial f}{\partial x},\ \frac{\partial f}{\partial y},\ \frac{\partial f}{\partial z}\right)$
8. $f = f_1^{15,10,3},\ g = f_1^{10,11,3},\ I = \left(\frac{\partial f}{\partial x},\ \frac{\partial f}{\partial y},\ \frac{\partial f}{\partial z},\ \frac{\partial g}{\partial x},\ \frac{\partial g}{\partial y},\ \frac{\partial g}{\partial z}\right)$
9. $f = f_1^{11,10,3},\ I = \left(f, \frac{\partial f}{\partial x},\ \frac{\partial f}{\partial y},\ \frac{\partial f}{\partial z}\right)$
10. $f = h^6, I = \left(f, \frac{\partial f}{\partial x},\ \frac{\partial f}{\partial y},\ \frac{\partial f}{\partial z}\right)$
11. $f = h^7,\ I = \left(f,\ \frac{\partial f}{\partial x},\ \frac{\partial f}{\partial y},\ \frac{\partial f}{\partial z}\right)$
12. Gräbe
    $I = (x^2 - z^{10} - z^{20},\ xy^3 - z^{10} - z^{30},\ y^6 - xy^3w^{40})$
13. $f = (x+y)^{20} + y^{21} + (x+y)^{10} + x^9y^9 + x^3y^{15} + x^2y^{11} + x^4y^4 + (x+y)^{17},$
    $I = \left(\frac{\partial f}{\partial x},\ \frac{\partial f}{\partial y}\right)$
14. $f = (x+y)^{20} + y^{21} + (x+y)^{10} + x^9y^9 + x^3y^{15} + x^2y^{11} + x^4y^4 + (x+y)^{17},$
    $I = \left(f, \frac{\partial f}{\partial x},\ \frac{\partial f}{\partial y}\right)$
15. Random 1
    $I = (47x^7y^8z^3 + 91x^7y^4z^7 + 28x^3y^6z^8 + 63x^2y,$
    $21x^3y^2z^{10} + 57xy^7z + 15x^3yz^5 + 51xy^3z^3,$
    $32x^7y^4z^8 + 53x^6y^6z^2 + 17x^3y^7z^2 + 74xy^5z,$
    $32x^{10}y^9z^6 + 23x^5y^8z^8 + 21x^2y^3z^7 + 27y^5z,$
    $81x^{10}y^{10}z + 19x^3y^5z^5 + 79x^5z^7 + 36xy^2z^3)$
16. Random 2
    $I = (57xy^2z^3 + 28xy^7z + 63xy^5z^4 + 91x^2y^3z^7 + 47x^7y^8z^3,$
    $51x^3y^2z^{10} + 21x^3y^6z^8 + 15x^7y^4z^8 + 32x^{10}y^{10}z + 74x^{10}y^9z^6,$
    $53x^2y + 17xy^3z^3 + 23x^3y^5z^5 + 21x^6y^6z^2 + 32x^5y^8z^8,$
    $19y^5z + 36x^3yz^5 + 81x^3y^7z^2 + 79x^5z^7 + 27x^7y^4z^7)$

17. Random 3

$I = (x^3 + y^4 + 2xz^3 + z^5 - 3x^4y^2 + 2z^6 + 3z^7,$
$xz^3 - 2x^4y^2 + z^6 + 2z^7,$
$9x^3z^2 + 18x^2z^5 - 5z^7 + 12x^4y^2z^2 + 42x^2z^6 + 40x^3y^2z^4 + 7z^9 + 24x^3y^2z^5,$
$-4y^3z^3 - 12x^6y + 32x^3y^5 - 2x^4yz^3,$
$12xy^3z^2 + 6x^5yz^2 + 24y^3z^5 + 20x^4yz^4 + 56y^3z^6 + 12x^4yz^5)$

18. Alex 2

$I = (4t^2z + 6z^3t + 3z^3 + tz, 5t^2z^7y^3x + 5x^2z^4t^3y + 3t^7, 6zt^2y + 2x^8 + 6z^2y^2t + 2y^5)$

19. $f = f_1^{19,19,4}, I = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right)$

20. $f = f_1^{24,23,6}, I = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right)$

Options:
The ordering of the variables is t, x, y, z, w, the computations were done in characteristic 32003.
Lazard stands for the algorithm "Lazard" and Mora for the algorithm "Standard basis" with NFMora, cf. Chapter 1.
The columns have the following meaning:

1. Lazard with MACAULAY, elimination order for homogenizing variable (MAC)
2. Lazard with SINGULAR, elimination order for homogenizing variable (SING)
3. Mora with degrevlex$^-$ (–)
4. Mora with morePairs, degrevlex$^-$ (mP)
5. Mora with sugar and morePairs, degrevlex$^-$ (su/mP)
6. Mora with sugar, sugarCrit and morePairs, degrevlex$^-$ (suC/mP)
7. Mora with sugar, degrevlex$^-$ (su)
8. Mora with fast HCtest, degrevlex$^-$ (fHC)
9. Mora with weighted ecartMethod and sugar, but without HCtest, degrevlex$^-$ (ecartM).
   w denotes the ecart vector, if $w = (1, \ldots, 1)$ this is the same as 3. An asterisk in front of the ecart vector indicates that it was set by hand, otherwise it was set automtically by the method described in [Po].
10. Mora with lex$^-$ (lex)
11. Mora with degrevlex$^-$ applied to the standardbasis computed in 10 (d(lex)).

We used

- the test version 8.6 of SINGULAR (January 1994) and
- the version 4.0 alpha of MACAULAY (15th October, 1993)

for the tests on HP–UX, 9000/735 with 80 megabyte of main memory.
The symbol "#" indicates usage of more than 120 megabyte and "-" indicates that this option makes no sense.
The time is given in seconds (up to one decimal place). The time 0 indicates less than 0.05 seconds.

Compare the table on the next page.

	MAC	SING	—	mP	su/mP	suC/mP	su	fHC	ecartM		lex	d(lex)
	1	2	3	4	5	6	7	8	9	w	10	11
1	53	35.7	27.9	28.2	27.6	#	27.9	-	0	(5, 2, 2, 1)	0	0
2	21	12.9	419.8	9.9	9.3	9.5	289.6	0	1.0	(95, 94, 63)	0.5	1.1
3	20	13.6	4.3	8.6	8.4	6.2	4.4	0.2	0.9	(86, 67, 51)	0.4	1.18
4	2	1.6	6.9	3.5	3.4	1.4	6.6	0	1.4	*(4, 3, 3)	0.9	0.4
5	2	0.8	2.8	2.8	2.7	2.5	2.7	0.5	1.4	*(5, 3, 3)	2.1	1.3
6	12	7.3	26.5	4.5	4.5	4.7	26.9	0	1.0	(16, 8, 5)	0	0.1
7	6	3.5	21.1	2.2	2.2	2.5	20.6	0.4	0	(16, 8, 5)	0.1	1.7
8	3	1.9	0	0	0	0	0	0	0.1	(60, 55, 64)	0.1	0.4
9	7	4.1	6.8	3.0	3.1	2.2	7.4	0	6.2	(10, 11, 7)	0.2	0.3
10	2	1.3	4.3	4.2	4.2	2.0	6.3	0	3.0	*(5, 4, 4)	0.3	0.2
11	1	0.7	4.9	4.6	4.4	2.2	4.9	0.3	3.5	*(5, 4, 4)	0.3	0.6
12	4	1.2	#	#	559.5	0.3	#	-	0	(20, 20, 2, 1)	0	0
13	237	84.2	#	43.4	44.1	87.4		0	1.0	*(4, 3)	0	0
14	33	14.2	2520.8	56.7	53.4	44.3	2438.7	0	7.7	*(4, 3)	0.2	0.1
15	2405	1480.4	0	0	0	0	0	-	0	(47, 64, 62)	0	0
16	6156	4113.7	#	#	#	#	#	-	#	(35, 38, 61)	0	0
17	5	3.3	14.7	1.6	1.6	1.3	15.0	0	1.2	(2, 3, 2)	0.1	0
18	4383	3838.4	0.6	0.1	0.1	#	0.5	-	0.3	(32, 12, 19, 13)	0	#
19	17641	9190.2	#	2205.2	2033.6	2253.6	#	2.2	92.9	*(4, 3, 2)	6.3	8.7
20	83965	50265.5	#	32421.8	33447.3	8855.7	#	315.0	#	(76, 65, 47)	135.9	816.6

Conclusions: (for the case $\mathrm{Loc}_< \mathsf{K}[\mathsf{x}] = \mathsf{K}[\mathsf{x}]_{(\mathsf{x})}$)
The above examples and many more tests which have not been documented show the following pattern:

1. The HCtest is essential for 0–dimensional ideals. In SINGULAR it is checked automatically whether all axes contain a leading monomial, regardless of whether the ideal is 0–dimensional or not (which is not very expansive). If this is so, the highest corner is computed and used as described in Chapter 2. In almost all cases this is the fastest option. Of course, it may be combined with the ecartMethod (and is so in SINGULAR if one chooses the ecartMethod). Moreover, if it is known in advance that the ideal is 0–dimensional we have the option fast HCtest (fHC) which does the following: in case all but one axes contain a leading monomial the search for a leading monomial on the last axes has priority. This (as for other options) is achieved by sorting the sets appropriately.

2. The ecartMethod applies to any ideal and can be extremely fast if one has a good feeling for the weights. In practice this option is most useful if the system automatically offers an ecart vector w. In SINGULAR this is implemented by choosing w such that the sum of the weighted ecart over all generators of the ideal (normalized in a certain way) is minimal. Other variants are under experimentation.

3. If the ecartMethod is not successful, it is usually recommended to use the option morePairs.

4. If the above options are not successful one should use morePairs and sugarCrit.

5. The sugar option is generally good and is a default option in SINGULAR.

6. Lazard's method is in general slower than the other options, although it is sometimes surprisingly fast. In general it seems to be least a safe option (there are no #).

7. As the above timings show, the lex ordering is usually the fastest. Since it is not an elimination ordering in the local case, its use is very limited (e.g. the dimension is computed correctly but not the multiplicity). But it can be used as a preprocessing in the sense that we first compute a standard bases with lex$^-$ and then transform this standard basis into one with the desired ordering, either by linear algebra methods or just by using it as an input to another standard basis computation. We see that the sum of timings in columns 10 and 11 is, in general, a very good strategy. There are examples where this method, combined with the ecartMethod, is the only way to compute a standard basis for degrevlex$^-$.

These are general principles which have proved useful. Moreover, in SINGULAR the options can be combined by the user. Of course, there are always special examples with different behaviour. For example, in example 12 sugarCrit + morePairs is very good, but in example 1 it is the worst option. There is no option which is universally the best. The standard basis algorithm is extremely sensitive to the choice of the strategy, in the local case ($\mathrm{Loc}_< \mathsf{K}[\mathsf{x}] = \mathsf{K}[\mathsf{x}]_{(\mathsf{x})}$) even more than in the inhomogeneous Buchberger case. The homogeneous Buchberger case, on the contrary, is much less sensitive to the choice of the strategy and is very stable. This explains why Lazard's method did always succeed, but on average is much slower. The ecartMethod (in connection with HC) seems to be by far the best. But we did not succeed in finding a good ecart vector for example 16. Here lex together with d(lex) is the best (less than 0.1 seconds). The conclusion is that at the moment a system has to offer several strategies, a default one which is good in most cases, but also gives the user the possibility of another choice.

Moreover, for computations in characterstic zero or over algebraic extensions the growth of the coefficients has to be taken into account in the strategies.

We tried to compare the timings also with Faugère's GB, but GB neither offers the ordering used in Columns 1 and 2 nor the tangent cone orderings used in the remaining columns. Testing several homogeneous examples in positive characteristic with degrevlex ordering showed, however, that SINGULAR is about 1.4 times faster than GB (on a Sparc2 and on an IBM RS/6000).

4 On Schreyer's method to compute syzygies

In this chapter we shall prove that Schreyer's method to compute syzygies (cf. [S1], [S2], [E]) extends to any semigroup ordering $<$ on $K[x]^r = \sum_{i=1}^r K[x]e_i$. For the treatment of syzygies in a different context, or for different algorithms see [M2], [Ba], [MM], [MMT] and [LS].
Let $S = \{g_1, \ldots, g_q\}$ be a standard basis of $I \subseteq K[x]^r$.

For $K[x]^q = \sum_{i=r+1}^{q+r} K[x]e_i$ we choose the following **Schreyer ordering** $<_1$ (depending on S): $x^\alpha e_{i+r} <_1 x^\beta e_{j+r}$ if and only if either $L(x^\alpha g_i) < L(x^\beta g_j)$ or $L(x^\alpha g_i) = L(x^\beta g_j)$ and $i > j$.

For g_i, g_j having the leading term in the same component, that is $L(g_i) = x^{\alpha_i} e_k, L(g_j) = x^{\alpha_j} e_k$ we consider $\mathrm{spoly}(g_i, g_j) := m_{ji} g_i - m_{ij} g_j$ with $m_{ji} = c(g_j)\frac{\mathrm{lcm}(L(g_i), L(g_j))}{x^{\alpha_i}}$.

Because S is a standard basis we obtain (Corollary 1.11)

$$(1 + h_{ij})(m_{ji} g_i - m_{ij} g_j) = \sum \xi_\nu^{ij} g_\nu$$

with $L(h_{ij}) < 1$ if $h_{ij} \neq 0$ and $L(\xi_\nu^{ij} g_\nu) < L(m_{ji} g_i)$.

For $j > i$ such that g_i, g_j have leading term in the same component, let

$$\tau_{ij} := (1 + h_{ij})(m_{ji} e_{i+r} - m_{ij} e_{j+r}) - \sum \xi_\nu^{ij} e_{\nu+r}.$$

Let $\ker(K[x]^q \to K[x]^r, \sum w_i e_{i+r} \mapsto \sum w_i g_i)$ denote the **module of syzygies, syz**(I), of $\{g_1, \ldots, g_q\}$. The following proposition is essentially due to Schreyer.

Proposition 4.1 *With respect to the ordering $<_1$ the following holds:*

1) $L(\tau_{ij}) = m_{ji} e_{i+r}$.

2) $\{\tau_{ij} \mid i < j$ *s.t.* $L(g_i), L(g_j)$ *are in the same component* $\}$ *is a standard basis for syz*(I)

Proof: 1) $L(\tau_{ij}) = L(m_{ji} e_{i+r} - m_{ij} e_{j+r}) = m_{ji} e_{i+r}$ holds by definition of $<_1$.

To prove 2) it has to be shown that $L(\mathrm{syz}(I)) = \langle\{m_{ji} e_{i+r}\}\rangle$ (Definition 1.3).

Let $\sum w_i g_i = 0$, that is $\tau := \sum w_i e_{i+r} \in \mathrm{syz}(I)$, and let $m e_{k+r} = L(\tau)$ with respect to $<_1$. Let

$$T := \{n e_{r+l} \mid n e_{r+l} \text{ be a monomial of } \tau, L(n g_l) = L(m g_k)\}.$$

Then, obviously, $\tau|_T := \sum_{n e_{r+l} \in T} n e_{r+l}$ is a syzygy of $L(g_1), \ldots, L(g_q)$. Especially, $\#T \geq 2$. Choose l such that $n e_{r+l} \in T$ for some n and $n e_{r+l} \neq m e_{k+r}$.
Because $L(\tau) = m e_{k+r}$ and the definition of $<_1$ we have $k < l$.
Since $m L(g_k) = n L(g_l)$ we have $m_{lk} \mid m$.
But $L(\tau_{kl}) = m_{lk} e_{k+r}$ implies $L(\tau_{kl}) \mid \tau$, that is $L(\tau) \in L(\langle\{m_{ji} e_{i+r}\}\rangle)$, which proves the proposition.

The principle of the algorithm is now easy.
Let S be a standard basis of $I \subseteq K[x]^r = \sum_{i=1}^r K[x]e_i$. We may assume that $L(s) \not| L(s')$ for different $s, s' \in S$.
Let $q := \#S$.

$W :=$ **Syz(S)**
INPUT: S, a standard basis
OUTPUT: W, the module of syzygies of S

```
W := ∅
T := initSyz (S)
L := initPairs (T)
WHILE L ≠ ∅ DO

     p := (a, b, s) the last element from L
     L := L\{(a, b, s)}
     s := spoly(a, b)
     h := NF(s|T)
     h := h(t = 1)
     W := W ∪ {h}

END
```

$T :=$ **initSyz**$(\mathbf{S})$

```
INPUT: S, a standard basis v_1, ..., v_s
OUTPUT: T, the set [v_1, 1, 0, ..., 0], ..., [v_s, , ..., 0, 1]

i := r; T := ∅
WHILE S ≠ ∅ DO

     s := the first element from S
     i := i + 1
     S := S\{s}
     s := s + e_i
     T := T ∪ {s}

END
T := T^h.
```

$h :=$ **NF**$(s|T)$

```
INPUT: S, a polynomial vector to reduce
       T, a set of polynomials with which to reduce
OUTPUT: h, the reduced ponyomial vector

h := s
WHILE exist f ∈ T such that L(f) | t^α L(h) for some α DO

     choose the first possible f with respect to the ordering
     in T such that α is minimal
     IF α > 0 THEN
          T := T ∪ {h}
     END
     h := spoly(t^α h, f)
     update (h)

END
```

Here we use on $K[t,x]^{r+q} = \sum_{i=1}^{r+q} K[t,x]e_i$ the following ordering $<_2$: If $i \le r$ and $j \ge r+1$ then $me_j <_2 ne_i$ for all monomials $m, n \in K[t,x]$.

On $\sum_{i=1}^{r} K[t,x]e_i$ respectively $\sum_{i=r+1}^{q} K[t,x]e_i$ we use the extension of $<$ (respectively $<_1$) described in Chapter 1.

The algorithm is extremely sensitive with respect to the ordering of the standard basis S of the input.

Assuming that the set L is ordered as in Chapter 1, then S should be ordered in an opposite manner to Chapter 1, that is it should be ordered by decreasing monomial ordering. This produces potentially more syzygies of smaller degree at the beginning.

Because of Proposition 4.1 we know the leading term of the s–polynomial of a pair in L without conputing the s–polynomial. If this leading term is divisible by another one, the pair can be cancelled from L.

The algorithm "Standard basis" of paragraph 1, together with repeated application of the algorithm "Syz", provides an effective way to construct finite $\mathrm{Loc}_< K[x]$–free resolutions and gives a sharpened version of Hilbert's syzygy theorem which generalizes Schreyer's proof (cf. [E], [S1], [S2]).

Lemma 4.2 *Let* $\{g_1, \dots, g_q\}$ *be a standard basis of* $I \subset K[x]^r = \sum_{i=1,\dots,r} K[x]e_i$. *We assume that the leading terms are a basis vector of* $K[x]^r$, *that is* $L(g_i) = e_{\nu_i}$. *We set* $J = \{\nu \mid \exists i \text{ s.t. } \nu = \nu_i\}$ *and for* $\nu \in J$ *we choose exactly one* g_{i_ν} *such that* $L(g_{i_\nu}) = e_\nu$. *Then* $I\mathit{Loc}_< K[x]$ *is a free* $\mathit{Loc}_< K[x]$*–module with basis* $\{g_{i_\nu} \mid \nu \in J\}$ *and* $(\mathit{Loc}_< K[x])^r / I\mathit{Loc}_< K[x]$ *is* $\mathit{Loc}_< K[x]$*–free with basis represented by the* $\{e_j \mid j \notin J\}$.

Proof: Let us renumber the g_i such that $g_{i_\nu} = g_\nu$ for $\nu \in J$. First of all, the subset $\{g_\nu \mid \nu \in J\} \subset \{g_1, \dots, g_q\}$ remains a standard basis of I since the set of leading terms is not changed. Hence, we may assume that all leading terms are different. By Proposition 1.4, $\{g_\nu \mid \nu \in J\}$ generates $I\mathrm{Loc}_< K[x]$. Now consider a relation

$$\sum_{j \notin J} \xi_j e_j = \sum_{j \in J} \xi_j g_j, \quad \xi_j \in \mathrm{Loc}_< K[x].$$

After clearing denominators we may assume that $\xi_j \in K[x]$. Since the leading terms involve different e_i on each side, we obtain $\xi_1 = \cdots = \xi_n = 0$. This shows that the $g_\nu, \nu \in J$ are linear independent and that the $e_j, j \notin J$ are independent modulo $I\mathrm{Loc}_< K[x]$. Since $\{L(g_j) \mid j \in J\} \cup \{e_i \mid i \notin J\}$ generate $L(K[x]^r) = (e_1, ..., e_r)K[x]$ $\{g_j \mid j \in J\} \cup \{e_i \mid i \notin J\}$ is a standard basis of $K[x]^r$ this set generates $(\mathrm{Loc}_< K[x])^r$ by Corollary 1.11. Therefore, $\{e_j \mid j \notin J\}$ generates $(\mathrm{Loc}_< K[x])^r / I\ \mathrm{Loc}_< K[x]$ over $\mathrm{Loc}_< K[x]$.

Theorem 4.3 *Let* $S = \{g_1, \dots, g_q\}$ *be a standard basis of* $I \subseteq K[x]^r$. *Order* S *in such a way that whenever* $L(g_i)$ *and* $L(g_j)$ *involve the same component, say* $L(g_i) = x^{\alpha_i} e_k$ *and* $L(g_j) = x^{\alpha_j} e_k$, *then* $\alpha_i \geq \alpha_j$ *in the lexicographical ordering if* $i < j$. *If* $L(g_1), \dots, L(g_q)$ *do not depend on the variables* $x_1, \dots, x_s$, *then the* $L(\tau_{ij})$ *do not depend on the variables* $x_1, \dots, x_{s+1}$ *and*

$$M := (\mathit{Loc}_< K[x])^r / I\ \mathit{Loc}_< K[x]$$

has a $\mathit{Loc}_< K[x]$*–free resolution of length* $\leq n - s$. *In particular,* M *always has a free resolution of length* $\leq n$ *and, by Serre's theorem,* $\mathit{Loc}_< K[x]$ *is a regular ring.*

Proof: For $i < j$ and $L(g_i) = x^{\alpha_i} e_k$, $L(g_j) = x^{\alpha_j} e_k$ we have $\alpha_i = (0, \dots, 0, \alpha_{i,s+1}, \dots), \alpha_j = (0, \dots, 0, \alpha_{j,s+1}, \dots)$ with $\alpha_{i,s+1} \geq \alpha_{j,s+1}$. Therefore, $L(\tau_{ij}) = m_{ji} e_{i+r}$ does not depend on $x_1, \dots, x_{s+1}$.

Let $q_1 := q$ and $\varphi_1 : K[x]^{q_1} \to K[x]^r$ the morphism given by $\{g_i\}$, $\sum w_i e_{i+r} \mapsto \sum w_i g_i$, and $\varphi_2 : K[x]^{q_2} \to K[x]^{q_1}$ the analog morphism given by the standard basis $\{\tau_{ij}\}$, $q_2 = \#\{\tau_{ij}\}$. Applying the same construction as above to $\mathrm{syz}^1(I) := \mathrm{syz}(I) = \ker(\varphi_1)$ and $\{\tau_{ij}\}$ we obtain a standard basis $\{\tau^2_{kl}\}$ of $\mathrm{szy}^2(I) := \mathrm{syz}(\mathrm{syz}(I)) = \ker(\varphi_2)$ such that the leading terms of τ^2_{kl} do not depend on $x_1, \dots, x_{s+2}$.

Continuing in the same way we obtain an exact sequence

$$0 \to K[x]^{q_{n-s}} / \ker(\varphi_{n-s}) \overset{\varphi_{n-s}}{\to} K[x]^{q_{n-s+1}} \to \dots \overset{\varphi_2}{\to} K[x]^{q_1} \overset{\varphi_1}{\to} K[x]^r \to K[x]^r / I \to 0.$$

Moreover, $\ker(\varphi_{n-s}) = \mathrm{syz}^{n-s}(I)$ has a standard basis $\{\tau_{k,l}^{n-s}\}$ such that none of the variables appear in $L(\tau_{k,l}^{n-s})$. Hence, by the preceding lemma, $K[x]^{q_{n-s}}/\ker(\varphi_{n-s})$ becomes free after tensoring with $\mathrm{Loc}_< K[x]$. If we tensor the whole sequence with $\mathrm{Loc}_< K[x]$ it stays exact (since $\mathrm{Loc}_< K[x]$ is $K[x]$–flat) and is the desired free resolution of M.

Remark 4.4 The above algorithm almost never gives a minimal free resolution (in the local or in the homogeneous case), on the contrary, every syzygy module is generated by a standard basis. In SINGULAR it is implemented with an option to cancel superfluous free factors in an additional step. Because of the non–minimality there are more reductions necessary than in the usual implementations of free resolutions. On the other hand, since we know explicitly a standard basis of the syzygy modules (for some ordering) by Proposition 4.1, we have an advantage. An effective implementation of the above algorithm requires avoiding comparisons of monomials with respect to the Schreyer ordering. We proceed in the following manner. Instead of computing directly the monomials of the syzygies we compute their product with the corresponding monomial defining the Schreyer ordering (weight monomial). This can be realized by an adapted initialization (just exchange the line $s := s + e_i$ in the procedure `initSyz` described above by $s := s + lm(s) \cdot e_i$ with $L(s) = lm(s) \cdot e_k$ for a suitable k). This allows use of the initial ordering during the whole resolution. At the end we have, of course, to divide by the weight monomial to obtain the resolution. The results of our testing of the above method, and its comparison to other methods, is given at the end of this chapter. We see there that it is usually faster to compute a minimal free resolution by fixing an ordering and minimizing the syzygy module step by step if we are closed to complete intersections and, if we are far away from complete intersections, Schreyer's method seems to be faster. SINGULAR offers both possiblities.

Example 4.5

1. the homogenization of Example 1 in Chapter 3 with respect to w (t, x, y, z, w the order of the variables)

2. the previous example with (w, t, x, y, z the order of the variables)

3. the homogenization of Example 3 in Chapter 3 with respect to w (w, x, y, z the order of the variables)

4. the homogenization of Example 5 in Chapter 3 with respect to w (w, x, y, z the order of the variables)

5. the homogenization of Example 12 in Chapter 3 with respect to w (t, x, y, z, w the order of the variables)

6. the homogenization of Example 8 in Chapter 3 with respect to w (w, x, y the order of the variables)

7. (max 5)^2
   the ideal $(f_1, \ldots, f_5)^2$ where $f_i = x_i(x_0^8 + x_1^2 \cdot \ldots \cdot \hat{x}_i \cdot \ldots \cdot x_5^2)$, $i = 1, \ldots, 5$.

8. Sparse
   $t^{18}x^2 - t^{19}z - t^{18}z^2$,
   $t^{26}xy - t^{17}z - t^{25}z^3$,
   $t^{38}y^2 - t^{37}xyw$;
   (t, x, y, z, w the order of the variables)

9. five homogeneous random polynomials of degree 3 in the variables a, b, c, d, e

10. cyclic roots 5
    $a + b + c + d + e,$
    $de + cd + bc + ae + ab,$
    $cde + bcd + ade + abe + abc,$
    $bcde + acde + abde + abce + abcd,$
    $abcde - h^5;$
    (a, b, c, d, e, h the order of the variables)

11. Kahn4
    $a^4 - b^4,$
    $b^4 - c^4,$
    $c^4 - d^4,$
    $d^4 - e^4,$
    $a^3 b + b^3 c + c^3 d + d^3 e + e^3 a;$
    (a, b, c, d, e the order of the variables)

12. Iarrobino
    $xy + y^2 + uz - wz - xz + yz,$
    $x^2 - y^2 + uz - wz - xz + z^2,$
    $wy,$
    $wx - uz + yz,$
    $w^2 - y^2 + uz - z^2,$
    $vz - yz - z^2,$
    $vy - y^2 - wz + xz + yz + z^2,$
    $vx - y^2 + uz + yz - z^2,$
    $vw - y^2 + uz + yz - z^2,$
    $v^2 + y^2 + uz - xz - yz - z^2,$
    $uy + y^2 + yz,$
    $ux - y^2 + wz - xz - z^2,$
    $uw - y^2 + uz + yz - z^2,$
    $uv - y^2 - xz + yz,$
    $u^2 + yz;$
    (u, v, w, x, y, z the order of the variables)

13. three homogeneous random polynomials of degree 5 in the variables a, b, c, d, e

14. Schreyer 1
    polynomials created in a certain random manner such that the ideal generates a 0–dimensional Gorenstein ring in characteristic 31991.

    $ab + 14766ac - 12713bc + 8997ad + 1878ae,$
    $ac + 9210bc + 9399ad + 13903bd - 9583ae,$
    $bc - 13988ad - 4712bd + 6771ae - 7341be,$
    $ad - 2340bd - 7515cd + 1575ae - 4211be,$
    $bd + 5023cd - 874ae + 4773be + 14016ce,$
    $cd + 1617ae - 14718be - 1384ce + 12060de,$
    $a^3 - 10731b^3 + 5818ae^2 + 13936be^2 + 11725ce^2 + 6401de^2,$
    $b^3 - 4862c^3 - 2334ae^2 + 9991be^2 + 14579ce^2 + 10173de^2,$
    $c^3 - 6087d^3 + 1135ae^2 + 4570be^2 + 5250ce^2 + 1393de^2,$
    $d^3 + 11392ae^2 + 7125be^2 - 15188ce^2 - 12706de^2 - 10957e^3;$
    (a, b, c, d, e the order of the variables)

15. Schreyer 2
    polynomials created as in 14

    $ab - 2302ac + 1504bc - 2175ad - 13468bd - 8487ae + 2984be,$
    $ac + 7021bd - 2524ad - 8459bd + 4807cd - 4372ae - 1908be,$

$bc - 1768ad + 12994bd + 7816cd - 7083ae + 2769be - 354ce,$
$ad + 8219bd - 13765cd + 14989ae + 36be - 4732ce + 8648de,$
$a^3 - 7986a^2e - 744b^2e + 2389c^2e - 7991bde + 11075cde - 15425d^2e +$
$7645ae^2 - 1871be^2 + 8698ce^2 - 9609de^2,$
$a^2e + 3582b^2e - 6012c^2e - 14933bde + 14203cde + 11560d^2e + 2294ae^2 +$
$12826be^2 - 12284ce^2 + 10870de^2,$

$b^2e+3097c^2e+10611bde+6826cde+2785d^2e-14231ae^2+13731be^2-1863ce^2+8227de^2$,
$c^2e-8559bde+13444cde+3372d^2e+5235ae^2+9139be^2-7685ce^2+3756de^2$,
$bde-14077cde-2960d^2e-4481ae^2-3950be^2-850ce^2-14832de^2$,
$cde-13603d^2e+4327ae^2+15554be^2+2054ce^2-12484de^2$,
$b^3+7339d^2e-3253ae^2+8720be^2+9224ce^2-11849de^2$,
$c^3-9222d^2e-12818ae^2-11519be^2+6703ce^2-10364de^2$,
$d^3+4730d^2e+7627ae^2+14168be^2+11428ce^2+8632de^2$,
$d^2e+9376ae^2+6688be^2-11914ce^2-295de^2-5576e^3$;
(a, b, c, d, e the order of the variables)

16. caprasse 4
$-2w^2x-w^2z+2xyt+y^2z$
$2w^4+4w^2x^2+4w^2xz-10w^2y^2-10w^2y^2-10w^2yt-x^3z+4x^2yt+4xy^2z+2y^3t$
$-w^2x-2w^2z+xt^2+2yzt$
$2w^4+4w^2xz-10w^2yt+4w^2z^2-10w^2t^2-xz^3+4yz^2t+2yt^3$;
(w, x, y, z, t the order of the variables)

17. Pellikaan, Jaworski S(9)
$S(m)$ denotes the set of polynomials (describing a minimal elliptic Gorenstein surface singularity), defined inductively:

$$\begin{array}{ll}
S(4): & x_4x_1-x_2^2,\ x_4x_2-x_3^2; \\
S(5): & S(4),\ f_{5,1}, f_{5,2}, x_5x_3-x_4^2+x_2x_1; \\
S(6): & S(5),\ f_{6,1}, f_{6,2}, f_{6,3}, x_6x_4-x_5^2-x_2^2; \\
S(m): & S(m-1),\ f_{m,1}, \dots, f_{m,m-3}, g_m;\ \text{ for } m \geq 7, \text{ where} \\
f_{m,i} & = x_mx_i - x_{m-1}x_{i+2}, \\
g_m & = x_mx_{m-2}-x_{m-1}^2-x_{m-3}x_{m-5},\ \text{ if } m \text{ is odd} \\
g_m & = x_mx_{m-2}-x_{m-1}^2-x_{m-1}^2-x_{m-2}x_{m-6};\ \text{ if } m \text{ is even}
\end{array}$$

($x_1, \dots, x_m$ the order of the variables)

18. Pellikaan, Jaworski S(10)
compare example 17

19. 9 random i–forms in 4 variables,
$i=2,\dots,10$

20. 6 random i–forms in 4 variables,
$i=5,\dots,10$.

For these examples we computed a minimal resolution with MACAULAY and SINGULAR using the regularity bound (cf. [E]) with the ordering degrevlex (in both systems realized as scripts; without this bound the computation is, at least for complicated examples, usually much slower) and computed a resolution with Schreyer's method (SING/SCHREYER).
vars denotes the number of variables, minbase the number of a minimal system of generators, stdbase the number of elements in the standard base and dim the dimension of the zero-set of the ideal.
For versions and notations cf. Chapter 3.

	MAC	SING	SING SCHREYER	vars	dim	minbase	stdbase
1.	552	96	13769	5	3	3	63
2.	2222	87	6293	5	3	3	82
3.	16	5	9	4	1	3	75
4.	20	5	7	4	1	3	44
5.	10	2	9	4	2	3	83
6.	12	2	23	5	2	3	22
7.	4	12	542	5	4	15	138
8.	14	1	1	5	4	3	38
9.	238	91	1228	5	1	5	76
10.	299	322	15	6	1	5	38
11.	950	166	2412	5	0	5	142
12.	49	24	3	6	0	15	22
13.	44	23	31	5	2	3	25
14.	101	60	1	5	0	10	19
15.	78	23	2	5	0	14	23
16.	1	1	12	5	2	4	23
17.	2	4	3	9	2	27	27
18.	20	29	22	10	2	35	35
19.	12	7	6	4	0	6	31
20.	825	446	7958	4	0	6	192

5 Zariski's question, Milnor numbers and multiplicities

The generalization of Buchberger's algorithm presented in this paper has many applications. We just mention the computation of Milnor numbers, Tjurina numbers, local multiplicities, Buchsbaum–Rim and Polar multiplicities, first and second order deformations of isolated singularities, projections of families with affine fibres onto a local base space and, of course, all the usual ideal theoretic operations in a factorring $\mathrm{Loc}_< K[x]/I$ such as intersection, ideal quotient and decision about ideal or radical membership. For further applications see [AMR]. Here we shall only explain how its implementation in SINGULAR helped to find a partial answer to Zariski's multiplicity question.

Let $f \in \mathbb{C}\{x_1, \dots, x_n\} = \mathbb{C}\{x\}$, $f = \sum c_\alpha x^\alpha$, $f(0) = 0$, be a not constant convergent powerseries and $\mathrm{mult}(f) = \min\{|\alpha| \mid c_\alpha \neq 0\}$ the multiplicity of f (for the general definition of multiplicity see the end of this chapter). Let $B \subseteq \mathbb{C}^n$ be a sufficiently small ball with centre 0 and $X = f^{-1}(0) \cap B$, the hypersurface singularity defined by f. If $g \in \mathbb{C}\{x\}$ is another powerseries and $Y = g^{-1}(0) \cap B$, then f and g (or X and Y) are called topologically equivalent if there exists a homeomorphism $h : (B, 0) \to (B, 0)$ such that $h(X) = Y$. The topological type of f is its class with respect to topological equivalence.

Zariski asked in 1971 (cf. [Z]) whether two complex hypersurface singularities f and g with the same topological type have the same multiplicity.

Zariski's question (usually called Zariski's conjecture) is, in general, unsettled but the answer is known to be yes in the following special cases:

- $n = 2$, that is for plane curve singularities (Zariski, Lê Dũng Trang),
- f is semiquasihomogeneous and g is a deformation of f (Greuel [Gr], O'Shea [OS]).

Recall that f is called semiquasihomogeneous if there exists an analytic change of coordinates and positive weights such that the sum of terms of smallest weighted degree has an isolated singularity.

The two series of examples $f_t^{a,b,c}$ and $g_t^{a,b,c,d,e}$ in Chapter 3 were actually constructed to find a counter example to Zariski's conjecture. The idea is as follows: let

$$f_t(x) = f(x) + t f_1(x) + t^2 f_2(x) + \dots$$

be a deformation of $f(x)$. Let

$$\mu(f_t) = \dim_{\mathbb{C}} \mathbb{C}\{x\} / \left(\frac{\partial f_t}{\partial x_1}, \dots, \frac{\partial f_t}{\partial x_n} \right)$$

denote the Milnor number of f_t which we assume to be finite for $t = 0$ (then it is finite for t close to 0). Then, if the topological type of f_t is independent of t, the Milnor number $\mu(f_t)$ is independent of t (for t sufficiently close to 0). The converse is also known to be true if $n \neq 3$. Hence, if $\mu(f_t)$ is constant but $\mathrm{mult}(f_t)$ is not, we get a counter example (at least if $n \neq 3$).

For the above mentioned series f_t^{abc} and g_t^{abcde} the multiplicity is not constant. For $t = 0$ both series have non–degenerate Newton diagram and one can show that $\mu(f_0^{abc}) - \mu(f_t^{abc}) \leq 12$ and $\mu(g_0^{abcde}) - \mu(g_t^{abcde}) \leq 4$ ($t \neq 0$ and small and some restrictions on a, b, c, d, e). Since μ is several hundred or even several thousand there seemed to be a good chance for $\mu(f_t)$ or $\mu(g_t)$ to be constant. Using SINGULAR we were able to compute many of these Milnor numbers but neither $\mu(f_t)$ nor $\mu(g_t)$ were constant. (Actually, in most cases we obtained $\mu(f_0) - \mu(f_t) = 6$ and $\mu(g_0) - \mu(g_t) = 2$.) None of the existing computer algebra systems were able to compute the standard basis of the ideal of partials of f_t respectively g_t for relevant cases. (Only the system Macaulay was able to do some cases with small a, b, c using Lazard's method but it needed hours or days, whereas SINGULAR needed seconds or minutes. In these cases the success of SINGULAR was mainly due to the HCtest.) The failure to find a counter example led to the following positive result which shows that the families f_t and g_t can never be a counter example.

Let $f_t(x)$ be a (1–parameter) holomorphic **family of isolated hypersurface singularities**, that is $0 \in \mathbb{C}^n$ is an isolated critical point of f_t for each t close to $0 \in \mathbb{C}$. The **polar curve** of such a family is the curve singularity in $\mathbb{C}^n \times \mathbb{C}$ defined by the ideal $(\partial f_t/\partial x_1, \dots, \partial f_t/\partial x_n) \subset \mathbb{C}\{x, t\}$.

Lemma 5.1 *Let f_t be a family of isolated hypersurface singularities. Let $H \cong \mathbb{C}^{n-1}$ be a hyperplane through 0 such that formation of the polar curve is compatible with restriction to H. That is: polar curve$(f_t \mid H)$ = polar curve$(f_t) \cap H$. Then*

$$\mu(f_t) = \textit{ constant } \Rightarrow \mu(f_t|H) = \textit{ constant.}$$

Proof: We may assume that $H = \{x_n = 0\}$ and then the polar curve$(f_t|H)$ is given by $(\partial f_t/\partial x_1, \dots, \partial f_t/\partial x_{n-1}, x_n)$ while polar curve$(f_t) \cap H$ is given by $(\partial f_t/\partial x_1, \dots, \partial f_t/\partial x_n, x_n)$. Hence, the assumption is equivalent to $\partial f_t/\partial x_n \in (\partial f_t/\partial x_1, \dots, \partial f_t/\partial x_{n-1}, x_n)$.

We shall use the valuation test for μ–constant by Lê and Saito ([LS]): $\mu(f_t)$ = constant $\Leftrightarrow$ for any holomorphic curve $\gamma : (\mathbb{C}, 0) \to (\mathbb{C}^n \times \mathbb{C}, 0)$ we have $\mathrm{val}(\partial f_t/\partial t(\gamma(s))) \geq \min\{\mathrm{val}(\partial f_t/\partial x_i(\gamma(s))), i = 1, \dots, n\}$. Moreover, this is equivalent to "$\geq$" replaced by "$>$". (val denotes the natural valuation with respect to s.)

Now let $\gamma(s)$ be any curve in $H = \{x_n = 0\}$.
Then $\partial f_t/\partial x_n \in (\partial f_t/\partial x_1, \dots, \partial f_t/\partial x_{n-1}, x_n)$ implies that $\mathrm{val}(\partial f_t/\partial x_n(\gamma(s)) \geq \min\{\mathrm{val}(\partial f_t/\partial x_i(\gamma(s))), i = 1, \dots, n-1\}$.

Applying the valuation test to f_t and to $f_t \mid H$, the result follows.

Proposition 5.2 *Let $f_t(x_1, \dots, x_n) = g_t(x_1, \dots, x_{n-1}) + x_n^2 h_t(x_1, \dots, x_n)$ be a family of isolated hypersurface singularities. Let g_0 be semiquasihomogeneous or let $n = 3$. If the topological type of f_t is constant then the multiplicity of g_t is constant (for t close to 0).*

Proof: Since f_t has an isolated singularity we may add terms of sufficiently high degree without changing the analytic type of f_t. If $n = 3$ we may replace g_t by $g_t(x_1, x_2) + x_1^N + x_2^N$, N sufficiently big, which has an isolated singularity and the same multiplicity as $g_t(x_1, x_2)$. Hence, in any case we may assume that g_t has an isolated singularity. Applying the preceding lemma to the hyperplane $\{x_n = 0\}$ we obtain $\mu(g_t)$ constant. But since Zariski's conjecture is true for plane curve singularities and for deformations of semiquasihomogeneous singularities ([Gr], [OS]), $\mathrm{mult}(g_t)$ is constant.

The Milnor number $\mu(f)$ of an isolated singularity can be computed as the number of monomials in $K[x_1, \dots, x_n]/L(I)$ where I is the leading ideal of $(\partial f/\partial x_1, \dots, \partial f/\partial x_n)$ with respect to any ordering $>$ such that $x_i < 1$, $i = 1, \dots, n$. This follows from the following Corollary 5.4.

The reason why standard bases can be applied to compute certain invariants of algebraic varieties or singularities (given in terms of submodules $I \subset K[x]^r$), is that for any monomial ordering on $K[x]^r$ we have:

a) $(\mathrm{Loc}_< K[x])^r/I$ is a (flat) deformation of $\mathrm{Loc}_< K[x]^r/L(I)$ (as we shall show below) which implies that certain invariants behave semicontinuously or even continuously during the deformation.

b) For a monomial ideal these invariants can be computed combinatorially (but one needs extra algorithms for the actual computation).

In order to show a) we make the following construction:
Let $g_1, \dots, g_q$ be a standard basis of $I \subset K[x]^r = \sum_{i=1,\dots,r} K[x]e_i$. Any monomial $x^\alpha e_k$ may be identified with the point $(\alpha_1, \dots, \alpha_n, 0, \dots, 1, \dots 0) \in \mathbb{N}^{n+r}$. For a weight vector $w = (w_1, \dots, w_{n+r}) \in \mathbb{Z}^{n+r}$ we define

$$w_1\alpha_1 + \cdots + w_n\alpha_n + w_{n+k}$$

to be the weighted degree of $x^\alpha e_k$. Let $\mathrm{in}_w(f)$ the initial term of $f \in K[x]^r$, that is the sum of terms (monomial times coefficient) of f with maximal weighted degree and $\mathrm{in}_w(I)$ the submodule generated by all $\mathrm{in}_w(f)$, $f \in I$.

It is not difficult to see that there exists a weight vector $w \in \mathbb{Z}^{n+r}$ (indeed almost all w will do) such that $\mathrm{in}_w(g_i) = c(g_i)L(g_i)$, $i = 1, \dots, q$, and, moreover, $\mathrm{in}_w(I) = L(I)$.

We choose such a w and shall now construct the deformation from $L(I)$ to I: For $f \in K[x]^r$ we can write $f = f_p + f_{p-1} + f_{p-2} + \cdots$ such that the weighted degree of each monomial of f_ν is ν. Let t be one extra variable and put

$$\tilde{f}(x,t) = f_p(x) + tf_{p-1}(x) + t^2 f_{p-2}(x) + \cdots \in K[x,t]^r.$$

Let $\tilde{I} \subset K[x,t]^r$ be the submodule generated by all $\tilde{f}$, $f \in I$. On $K[x,t]^r$ we choose the product ordering with lex^- on $K[t]$: $x^\alpha t^p e_k < x^\beta t^q e_l$ if $p > q$ or if $p = q$ and $x^\alpha e_k < x^\beta e_l$.

With respect to this ordering we have $L(\tilde{f}) = L(f)$ and, moreover, $L(\tilde{I}) = L(I)K[t]$. In particular, $\tilde{g}_1, \dots, \tilde{g}_q$ is a standard basis of $\tilde{I}$.

Let $R := \mathrm{Loc}_< K[x]$, $S := \mathrm{Loc}_< K[x,t]$ and $K(t)$ the quotient field of $K[t]$.

Proposition 5.3 *If $I \neq R^r$ then $S^r/\tilde{I}S$ is a faithfully flat $K[t]_{(t)}$–module with "special fibre"*

$$(S^r/\tilde{I}S) \otimes_{K[t]_{(t)}} K \cong R^r/L(I)R$$

and "generic fibre"

$$(S^r/\tilde{I}S) \otimes_{K[t]_{(t)}} K(t) \cong R^r/IR \otimes_K K(t).$$

Proof: The statements regarding the special and the generic fibres are easy. If $I \neq R^r$ then the support of $S^r/\tilde{I}S$ is surjective over Spec $K[t]_{(t)}$ and hence ([AK], V, Proposition 2.4) it remains to show that t is a non–zero divisor of $S^r/\tilde{I}S$. Let $f \in S^r$ and $tf \in \tilde{I}S$. By Corollary 1.11 we have

$$(1+g)tf = \sum \xi_i \tilde{g}_i, L((1+g)tf) = L(tf) = tL(f),$$

and $L(\xi_i\tilde{g}_i) \le L(tf)$ if $\xi_i \neq 0$. Hence, $tL(f)$ is equal to $L(\xi_i)L(g_i)$ for some i. Then $L(\xi_i)$ is divisible by t and therefore also ξ_i, by definition of $>$ on $K[x,t]^r$. Subtracting $\xi_i\tilde{g}_i$ on both sides and arguing as before, we see that all ξ_i are divisible by t. Therefore, $(1+g)f \in \tilde{I}$ and $f \in \tilde{I}S$.

Corollary 5.4 *Let either $<$ be a wellordering or R^r/IR a finite dimensional K–vector space. Then the monomials in $K[x]^r \setminus L(I)$ represent a K–basis of R^r/IR.*

Proof: If $<$ is a wellordering, the monomials not in $L(I)$ are a basis of the free module $S^r/\tilde{I}S$ (Theorem of Macaulay cf. [E]), hence the result. In general, it is easy to see that these monomial are linear independent modulo IR. (Use a standard basis of I and Corollary 1.11.)

Hence, if R^r/IR is finite dimensional, there are only finitely many monomials in $K[x]^r \backslash L(I)$. The proposition implies that $S^r/\tilde{I}S$ is $K[t]_{(t)}$–free with these monomials as basis, hence they also generate R^r/IR.

Remark 5.5 In general, the monomials not in $L(I)$ are not a basis of $\mathrm{Loc}_< K[x]/I$. Take, for example, $K[x]$ with lex^- and $I = (0)$. Then $\mathrm{Loc}_< K[x] = K[x]_{(x)}$ is not K–generated by monomials. If $<$ is a wellordering, then $S^r/\tilde{I}S$ is even free over $K[t]$ (cf. [E]).

Corollary 5.6 *For any module ordering* $\dim R^r/IR = \dim K[x]^r/L(I)$ *where dim denotes the Krull dimension.*

Proof: $I = R^r$ implies $L(I) = K[x]^r$, hence we may assume $I \neq R^r$. Faithful flatness implies that $\dim R^r/IR = \dim R^r/L(I)R$ ([AK], V, Proposition 2.10), hence the result.

Let us finish with a final remark about multiplicites in the local case: Consider the local ring $R = K[x]_{(x)}$ with maximal ideal $(x) = (x_1, \ldots, x_n)$ and $M = R^r/IR$ a finitely generated R–module, where I is given as a submodule of $K[x]^r$ by finitely many generators. Consider

$$\mathrm{gr}\, M = \sum_{i \geq 0} (x)^i M/(x)^{i+1} M,$$

which is a graded module over $\mathrm{gr}\, R = K[x]$. The formal power series

$$S_M(T) = \sum_{i \geq 0} \dim_K (M/(x)^{i+1} M) T^i,$$

is called the **Hilbert–Samuel series** and the function $s_M(i) = \dim_K(M/(x)^{i+1}M)$ the **Hilbert–Samuel function** of M (with respect to the maximal ideal).

If $h_{\mathrm{gr}\, M}$ denotes the Hilbert function of the graded module $\mathrm{gr}\, M$ we have (cf. [Ma]) $h_{\mathrm{gr}\, M}(i) = \dim_K (x)^i M/(x)^{i+1} M$ and a polynomial $hp_{\mathrm{gr}\, M} \in \mathbb{Q}[t]$ (the Hilbert polynomial of $\mathrm{gr}\, M$) of degree $d-1$ of the form

$$hp_{\mathrm{gr}\, M}(t) = \frac{e_{\mathrm{gr}\, M}}{(d-1)!} t^{d-1} + \text{ (terms of lower degree)}$$

such that $hp_{\mathrm{gr}\, M}(i) = h_{\mathrm{gr}\, M}(i)$ for i sufficiently big. $e_{\mathrm{gr}\, M}$ is called the **degree** of the graded module $\mathrm{gr}\, M$ whilst d is equal to the Krull dimension of $\mathrm{gr}\, M$.

It follows that

$$s_M(i) - s_M(i-1) = h_{\mathrm{gr}\, M}(i)$$

and there exists a polynomial $sp_M \in \mathbb{Q}[t]$ of degree d such that

$$sp_M(t) = \frac{e_M}{d!} t^d + \text{(terms of lower degree)},$$

where $d = \dim\, M = \dim\, \mathrm{gr}\, M$ and $e_M = e_{\mathrm{gr}\, M}$ is called the (Samuel) **multiplicity** of M. The following proposition follows now easily.

Proposition 5.7 *Let $<$ be a degree ordering (cf. Chapter 1) on the monomials of* $K[x]$ *such that* $w_i =$ *degree* $(x_i) = -1$ *for* $i = 1, \ldots, n$ *which is extended to a module ordering on* $K[x]^r$ *arbitrarily. Let* $M = R^r/IR$ *as above and* $L(I)$ *the leading ideal of* I. *Then the Hilbert function* $h_{\mathfrak{gr} M}$ *coincides with the Hilbert function* $h_{K[x]^r/L(I)}$ *of the graded module* $K[x]^r/L(I)$. *In particular,* $\dim M = \dim K[x]^r/L(I)$ *and* $e_M =$ *degree*$(K[x]^r/L(I))$. *Consequently,* $\dim M$ *and* e_M *can be computed from a standard basis of* I *with repect to an ordering as above and a Hilbert polynomial algorithm.*

Remark 5.8 If one is only interested in the dimension and the multiplicity it is not necessary to compute the full Hilbert series. A fast, direct algorithm for computing this is implemented in SINGULAR. Moreover, for computing dimensions we can use the fast lex^- ordering by Corollary 5.6 (but, of course, not to compute multiplicities or Hilbert functions).

6 The Weyl algebra and the exterior algebra

For applications to D–modules and to vector bundles over projective spaces the computation of standard bases and syzygies in two special non–commutative algebras is also implemented in SINGULAR. The advantage of SINGULAR having almost all possible monomial orderings implemented allows us to compute in the local and in the global case.

SINGULAR can compute the standard basis of modules over the (non–commutative) algebra $K\langle D_1, \ldots, D_n,\ X_1, \ldots, X_n\rangle$ with the relations

$$\begin{aligned} X_i X_j &= X_j X_i \text{ for all } i, j \\ D_i D_j &= D_j D_i \text{ for all } i, j \\ X_i D_j &= D_j X_i \text{ for all } i \neq j \\ X_i D_i + 1 &= D_i X_i. \end{aligned}$$

The D_i may be considered as differential operators.

The only restriction we have to make to the ordering is the assumption that $L([m, m']) < L(m, m')$ for all monomials m, m' with $[m, m'] \neq 0$. Especially a product ordering with the property that monomials in the D_i are always greater than monomials in the X_i and which is a wellordering on $K[D_i, \ldots, D_n]$, is admissible. Hence, SINGULAR can compute standard bases in $(\mathrm{Loc}_< K[X_1, \ldots, X_n])$ $[D_1, \ldots, D_n]$, in particular in the Weyl algebra $K[X_1, \ldots, X_n]\ [D_1, \ldots, D_n]$ and in the "local Weyl algebra" $K[X_1, \ldots, X_n]_{(X_1, \ldots, X_n)}\ [D_1, \ldots, D_n]$. Moreover, we have implemented standard basis and syzygies of modules over the algebra $(\mathrm{Loc}_< K[X_1, \ldots, X_n])\ \langle Y_1, \ldots, Y_m\rangle$ with the relations

$$\begin{aligned} Y_i^2 &= 0, \\ Y_i Y_j &= -Y_j Y_i, \end{aligned}$$

in particular over the tensor product of $K[X_1, \ldots, X_n]$ with the exterior algebra.

References

[AK] Altmann, A.; Kleiman, S.: Introduction to Grothendieck Duality Theory. Lect. Notes in Math. 146, Springer 1970.

[AMR] Alonso, M.E.; Mora, T.; Raimondo, M.: Local decomposition algorithms. In: Proc. AAECC Tokyo 1990, Lect. Notes Comp. Sci. **508** 208 – 221 (1991).

[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).

[E] Eisenbud, D.: Commutative Algebra with a view toward Algebraic Geometry. Book in preparation.

[G] Gräbe, H.-G.: The tangent cone algorithm and homogenization. To appear.

[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.

[Gr] Greuel, G.-M.: Constant Milnor number implies constant multiplicity for quasihomogeneous singularities. Manuscr. Math. **56**, 159-166 (1986).

[LS] La Scala, R.: An algorithm for complexes. Preprint Univ. Pisa (1993).

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

[LS] Lê D. T.; Saito, K: La constance du nombre de Milnor donne des bonnes stratifications. CRAS Paris, **277**, 793-795 (1973).

[Ma] Matsumura, H.: Commutative ring theory. Cambridge University Press (1989).

[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).

[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).

[MM] Möller, H.M.; Mora, T.: Computational aspects of reduction strategies to construct resolutions of monomial ideals. Proc. AAECC 2, Lecture Notes in Computer Science **228** (1986).

[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).

[OS] O'Shea, D.: Topologically trivial deformations of isolated quasihomogeneous hypersurface singularities are equimultiple. Proc. Amer. Math. Soc. **101**, 260-262 (1987).

[Po] Pohl, W.: About the weighted ecart and the weighted sugarMethod for computing standard bases. Preprint, Univ. Kaiserslautern, 1994.

[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).

[S1] Schreyer, F.-O.: Die Berechnung von Syzygien mit dem verallgemeinerten Weierstrassschen Divisionssatz. Diplomarbeit, Hamburg (1980).

[S2] Schreyer, F.-O.: A standard basis approach to syzygies of canonical curves. J. reine angew. Math. **421**, 83-123 (1991).

[Z] Zariski, O.: Some open questions in the theory of singularities. Bull. Amer. Math. Soc. **77**, 481-491 (1971).

[Zi] Zimnol, M.: Beispiele algebraischer Reduktionsstrukturen. Diplomarbeit, Kaiserslautern (1987).

List of papers published in the Reports on Computer Algebra series

[RCA:29] Greuel, G.-M. Computer Algebra and Algebraic Geometry – Achievements and Perspectives. Feb 2000.

[RCA:28] Siebert, T. Recursive Computation of Free Resolutions and a Generalized Koszul Complex. Jan 2000.

[RCA:27] Bachmann, O. and Gräbe, H.-G. The SymbolicData Project: Towards an Electronic Repository of Tools and Data for Benchmarks of Computer Algebra Software. Jan 2000.

[RCA:26] B. Reinert. Solving One-Sided Equations in Integer Monoid Rings. Jan 2000.

[RCA:25] B. Reinert and D. Zeckzer. Coset Enumeration using Prefix Gröbner Bases in Mrc – An Experimental Approach. Nov 1999.

[RCA:23] Birgit Reinert. Observations on coset enumeration. Nov 1998.

[RCA:22] K. Madlener and F. Otto. Some Applications Of Prefix-Rewriting In Monoids, Groups, And Rings. November 1998.

[RCA:21] G.-M. Greuel, G. Pfister, and H. Schönemann. Singular version 1.2 user manual. June 1998.

[RCA:20] B. Reinert and D. Zeckzer. MRC – A System for Computing Gröbner Bases in Monoid and Group Rings. July 1998.

[RCA:19] B. Reinert, K. Madlener, and T. Mora. A note on nielsen reduction and coset enumeration. February 1998.

[RCA:18] O. Bachmann and H. Schönemann. Monomial Representations for Gröbner Bases Computations. January 1998.

[RCA:17] Thomas Siebert. An algorithm for constructing isomorphisms of modules. January 1998.

[RCA:16] K. Madlener and B. Reinert. String Rewriting and Gröbner Bases – A General Approach to Monoid and Group Rings. October 1997.

[RCA:15] B. Martin and T. Siebert. Splitting Algorithm for vector bundles. September 1997.

[RCA:14] K. Madlener and B. Reinert. Relating rewriting techniques on monoids and rings: Congruences on monoids and ideals in monoid rings. September 1997.

[RCA:12] O. Bachmann, S. Gray, and H. Schönemann. MP Prototype Specification. Dec 1997.

[RCA:11] O. Bachmann. Effective simplification of cr expressions. January 1997.

[RCA:10] O. Bachmann, S. Gray, and H. Schönemann. A proposal for syntactic data integration for math protocols. January 1997.

[RCA:09] B. Reinert. Introducing reduction to polycyclic group rings – a comparison of methods. October 1996.

[RCA:08] T. Siebert. On strategies and implementations for computations of free resolutions. September 1996.

[RCA:02] H. Schönemann. Algorithms in singular. June 1996.

[RCA:01] H. Grassmann, G.-M. Greuel, B. Martin, W. Neumann, G. Pfister, W. Pohl, H. Schönemann, and T. Siebert. Standard bases, syzygies and their implementation in singular. July 1996.