UNIVERSITÄT KAISERSLAUTERN
Zentrum für Computeralgebra

REPORTS ON COMPUTER ALGEBRA
NO. 28

Recursive Computation of Free Resolutions and a Generalized Koszul Complex

by

Siebert, T.

Jan 2000

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/`

Recursive Computation of Free Resolutions and a Generalized Koszul Complex

Thomas Siebert
Brandenburgische Technische Universität Cottbus
Institut für Mathematik
PF 101344, D-03046 Cottbus
siebert@mathematik.uni-kl.de

February 2, 2000

1 Introduction

The computation of syzygies and free resolutions of ideals and modules is one of the main applications of computer algebra. It is common folklore that a Groebner basis computation produces also all information about the module of syzygies of a set of generators. First implementations of free resolutions were based on this fact. In the last years several improvements of this approach were developped and implemented (see [5], [1] and [7]). The main ideas of these improvements are special reduction strategies based on F.-O. Schreyers monomial orderings and the usage of the Hilbert function to avoid useless pairs within the reduction process. All these algorithms use the traditional process of creating s-polynomials of pairs of polynomials and reducing them.

In abstract algebra there is another idea to compute free resolutions of ideals – the Koszul complex (see J.L. Koszul [4]). Its commutative variant is based on the concept of regular sequences of elements of a ring R (see Chapter 17 of [3]).

In this paper we connect the ideas used in the Koszul complex with the computation of free resolutions based on Buchberger's algorithm. For this purpose the notion of a sequential algorithm is introduced in chapter 2. This kind of algorithm constructs the free resolution of an ideal via a sequence of subideals (which differ by one generator each time) and the extension of the associated free resolutions. Along this line we decide first the I-regularity of a new generator and use a tensor product of complexes in the positive case. The key-point of the algorithm is the handling of the non-I-regular generators. In Chapter 3.3 we construct a natural generalization of the Koszul complex for non-I-regular generators (which should not be confused with the generalized Koszul complex introduced by D. Buchsbaum [2]). This construction might give a deeper insight into the nature of free resolutions.

The algorithmic part of this new construction is mainly done by a special choise of orderings on the resolution which allows to put the different cases in a natural framework (see Chapter 4). Finally, the pseudo code of such an algorithm is presented and we give some timings based on an implementation within the computer algebra system **SINGULAR**.

2 Sequential Algorithms

On the algorithmic side, our new approach differs from the former in the following point.

For any Groebner or standard basis computation the efficiency of an algorithm depends essentially on the strategy how to chose the next pair from the pairset constructed so far. The major criteria to sort (and select from) a pairset are usually order–dependend (**O**) (the order of the leading term of the s-polynomial or of the syzygy of the initial monomials) or the module index (**M**) for implementations of free resolutions. Moreover, there are some secondary criteria like the length of polynomials etc.

Of course, the organization of the pairset depends also on the sequence (**S**) with which new generators are added to the set of known generators. The following table gives an overview about the use of the major types of known criteria within the different algorithms:

	Naive	Schreyer	LaScala	Hilbert-driven	Sequential
1. criterion	M	M	O	O	S
2. criterion	O	O	M	M	
3. criterion	S	S	S	S	

For a description of these algorithms see [7] (for more details also [1] and [5]).

Definition 2.1 *An algorithm for the computation of a free resolutions is called sequential if it uses the sequence of generators as first criterion for ordering the pairsets. This means, it extends recursively a resolution for the first $n-1$ generators to a resolution of n generators of a given ideal.*

3 The Extension of free Resolutions

Let R be a polynomial ring over some base field k and $I = (f_1, \ldots, f_n)$ an ideal in R. Denote by $I_i = (f_1, \ldots, f_i)$, $i = 0, 1, \ldots, n$ the (sub-)ideal generated by the first i elements of $\{f_1, \ldots, f_n\}$. We obtain a chain of ideals

$$0 = I_0 \subset I_1 \subset \ldots \subset I_{n-1} \subset I_n = I,$$

where each ideal differs from the preceeding by just one generator. We may further assume that $f_i \in R \setminus I_{i-1}$.

Definition 3.1 *An element $f_i \in R$ is called I_{i-1}-regular if f_i does not represent a zero divisor of R/I_{i-1}. A sequence of elements $\{f_1, \ldots, f_n\}$ is called regular if each f_i is I_{i-1}-regular.*

3.1 Criteria for Regularity

In the case of a graded ring R and a graded ideal I we obtain a first necessary und sufficient condition for the regularity of f_i over I_{i-1} using on the computation of Hilbert series.

Consider the exact sequence

$$R/I_{i-1} \to R/I_{i-1} \to R/I_i \to 0,$$

where the left most mapping is the multiplication by the class of f_i in R/I_{i-1}. We denote this mapping by f_i, too. Assume that f_i is regular over I_{i-1}, i.e., f_i is not a zero divisor in R/I_{i-1}. Then the mapping f_i is a monomorphism and the Hilbert series of R/I_i is the difference of those of R/I_{i-1} and $(R/I_{i-1})[deg(f_i)]$, where $(R/I_{i-1})[deg(f_i)]$ denotes the module R/I_{i-1} with the degree shifted by $deg(f_i)$. If f_i is not regular the Hilbert series of R/I_i is different from this difference. Hence, the knowledge of the standard base of I_{i-1} and I_i provide us with a criterion for the regularity of f_i.

Let $h_M(t)$ denote the Hilbert serie of the R-module M.

Lemma 3.2 *The element $f_i \in R$ is I_{i-1}-regular if and only if*

$$h_{R/I_i}(t) = h_{R/I_{i-1}}(t) - t^{-deg(f_i)} h_{R/I_{i-1}}(t).$$

There is a second, simple, but, nevertheless very usefull criterion (not assuming gradedness).

Lemma 3.3 *The element $f_i \in R$ is I_{i-1}-regular if and only if $(I_{i-1} : f_i) = I_{i-1}$.*

PROOF: We have $I_{i-1} \subset (I_{i-1} : f_i)$. If f_i is not a zero divisor over I_{i-1} then any element of $(I_{i-1} : f_i)$ lies in I_{i-1}. □

Consider the module $M_i = Syz(I_i) \subset R^i$ of all syzygies of I_i. Then the i-th component of M_i represents the relative syzygies of f_i over R/I_{i-1}. The ideal generated by the i-th component of M_i is just $(I_{i-1} : f_i)$.

Definition 3.4 *We define*

$$J_i := J(f_i, I_{i-1}) := (I_{i-1} : f_i)$$

and call J_i the i-th extension ideal of I with respect to the sequence $\{f_1, \ldots, f_n\}$.

3.2 Regular Extensions

These are the computationally important extensions since they could be computed without Groebner techniques.

Assume the complex K_{i-1} to be a free resolution of R/I_{i-1}:

$$K_{i-1} : 0 \to R^{n_l} \to \ldots \to R^{n_2} \to R^{i-1} \to R^1 \to 0.$$

The multiplication by f_i induces a complex

$$K : 0 \to R^1 \to R^1 \to 0.$$

Now, the tensor product (see Chapter 17.3 in [3]) of the two complexes is a double complex $\hat{K}_i$:

$$\begin{array}{ccccccccccccc} 0 & \to & R^{n_l} & \to & \ldots & \to & R^{n_2} & \to & R^{i-1} & \to & R^1 & \to & 0 \\ & & \downarrow & & & & \downarrow & & \downarrow & & \downarrow & & \\ 0 & \to & R^{n_l} & \to & \ldots & \to & R^{n_2} & \to & R^{i-1} & \to & R^1 & \to & 0 \end{array}$$

where the vertical mappings are the multiplications with $(-1)^j f_i$ for $j = l, \ldots, 0$.

Lemma 3.5 *The total complex K_i associated with the double complex $\hat{K}_i$ is a free resolution for R/I_i.*

PROOF: This is easily deduced from the spectral sequence $E_1^{p,q} = H^q(\hat{K}_i^{*,p})$ converging to the homology of the total complex K_i (see Chapter A3.13.4, [3]). □

According to this lemma we construct the extension K_i from 2 copies of the resolution K_{i-1} and the homomorphism of complexes induced by f_i. As arithmetical operations this procedure requires only duplication and addition of polynomials. The number of them depends on the size of K_{i-1}. Thus, the involved operations are of polynomial complexity w.r.t. to the input of K_{i-1} and f_i.

3.3 Non-regular Extensions

Assume f_i to be an arbitrary element in $R \setminus I_{i-1}$. Let K_{i-1} be again a free resolution of R/I_{i-1} and denote by L_i a resolution of R/J_i:

$$L_i : 0 \to R^{m_l} \to \ldots \to R^{m_2} \to R^{m_1} \to R^1 \to 0.$$

Lemma 3.6 *The multiplication by f_i on the 0-level induces a homomorphism of complexes:*

$$F\_ : L_i \to K_{i-1}.$$

PROOF: The situation is the following:

$$\begin{array}{ccccccccccccc} 0 & \to & R^{m_l} & \xrightarrow{d'_{l-1}} & \ldots & \to & R^{m_2} & \xrightarrow{d'_1} & R^{m_1} & \xrightarrow{d'_0} & R^1 & \to & 0 \\ & & F_l \downarrow & & & & F_2 \downarrow & & F_1 \downarrow & & f_i \downarrow & & \\ 0 & \to & R^{n_l} & \xrightarrow[d_{l-1}]{} & \ldots & \to & R^{n_2} & \xrightarrow[d_1]{} & R^{i-1} & \xrightarrow[d_0]{} & R^1 & \to & 0 \end{array},$$

where R^{m_s} or R^{n_t} can be zero for $s > s_0$ or $t > t_0$ if the resolutions have different length. The mapping $F_0 := f_i$ is given. We construct the mappings F_j by induction on i: For $i = 1, \ldots, l-1$ there are mappings F_j such that

$$d_{j-1} F_j = F_{j-1} d'_j. \tag{1}$$

This condition means that

$$F_j \oplus d'_j : R^{m_j} \longrightarrow R_{n_j} \oplus R_{m_{j-1}}$$

maps into the kernel of

$$d_{j-1} - F_{j-1} : R_{n_j} \oplus R_{m_{j-1}} \longrightarrow R_{n_{j-1}}.$$

Now, we start with the indices 1 and 0. The image of d'_0 is the ideal $J_i = (I_{i-1} : f_i) = (g_1, \ldots, g_l) \subset R^1$. Thus, a standard generator e_s of R^{m_1} is mapped into I_{i-1} by $f_i d'_0$. We chose a representative $E_s \in R^{i-1}$ of $f_i d'_0(e_s) = f_i g_s$ as an element of I_{i-1} and define the map F_1 by $F_1(e_s) = E_s$. Then $(d_0 F_1 - F_0 d'_1) e_s = f_i g_s - f_i g_s = 0$ as required.

Assume F_j to be constructed. Let e_s be a standard generator of $R^{m_{j+1}}$. From $F_{j-1} d'_j d'_{j+1}(e_s) = 0$ we deduce $F_j d'_{j+1}(e_s) \in ker(d_{j-1})$. Thus, we can chose again a representative E_s of $F_i d'_{j+1}(e_s)$ in $R_{n_{j+1}}$ and we define F_{j+1} by $F_{j+1}(e_s) = E_s$. The equation (1) follows as in the beginning of the induction. □

Proposition 3.7 *The diagram $\hat{K}_i$ constructed form the homomorphism $F\_$ by reversing the sign of every second map F_j:*

$$\begin{array}{ccccccccccccc}
 & & & d'_{l-1} & & & & d'_1 & & d'_0 & & & \\
0 & \rightarrow & R^{m_l} & \rightarrow & \ldots & \rightarrow & R^{m_2} & \rightarrow & R^{m_1} & \rightarrow & R^1 & \rightarrow & 0 \\
 & (-1)^l F_l & \downarrow & & & & F_2 \downarrow & & -F_1 \downarrow & & f_i \downarrow & & \\
0 & \rightarrow & R^{n_l} & \rightarrow & \ldots & \rightarrow & R^{n_2} & \rightarrow & R^{i-1} & \rightarrow & R^1 & \rightarrow & 0 \\
 & & & d_{l-1} & & & & d_1 & & d_0 & & &
\end{array}$$

constitutes a double complex the total complex of which is a resolution of R/I_i.

PROOF: The first assertion follows from the construction in Lemma 3.6. The second follows again from the spectral sequence $E_1^{p,q} = H^q(\hat{K_i}^{*,p})$ converging to the homology of the total complex. For a fixed p the homology of $\hat{K_i}^{*,p}$ is concentrated in the zeroth term and

$$H^0(\hat{K_i}^{*,-}) : 0 \rightarrow R/J_i \rightarrow R/I_{i-1} \rightarrow 0.$$

The map in the middle is the multiplication by f_i and, consequently, a monomorphism. Thus, this complex is a resolution of R/I_i and the second statement follows. □

REMARK: For an I_{i-1}-regular generator f_i the upper resolution L_i could be chosen as identical copy of the lower K_{i-1} and, hence, the double complex $\hat{K}_i$ is the natural generalization of the Koszul complex.

Note, that the whole construction works over arbitrary rings: The only assumption is the existence of the free resolutions L_i and K_{i-1}.

Definition 3.8 *The total complex associated to $\hat{K}_i$ is denoted by K_i and is called the i-th subresolution of I in R.*

Now, asssume that I is replaced by an R-module $M = (m_1, \ldots, m_n) \subset R^p$. We define again the i-th extension ideal by $J_i := (M_{i-1} : m_i) \subset R$. Here, J_i and M_{i-1} are non-compatible objects. Therefore, there is no concept analogously to the I_{i-1}-regularity in the ideal case. Nevertheless, the non-regular extension of a resolution applies, too.

Theorem 1 *Let $M = (m_1, \ldots, m_n)$ be a module (or an ideal) in R^p (or R resp.). A resolution of M could be computed from a sequence of resolutions of the extension ideals $J_i = (M_{i-1} : m_i)$ for $i = 1, \ldots, n$.*

REMARK: The resolution constructed via Proposition 3.7 is not minimal in general: Going back to the proof of Lemma 3.6 we see that a chosen representation E_s of $F_i d'_{j+1}(e_s)$ can contain absolute entries for certain components.

EXAMPLE: Let us consider $I = (x^2, y^2, xy) \subset k[x, y]$. We start with $I_2 = (x^2, y^2)$. y^2 is regular over (x^2) and a resolution of R/I_2 is given by the usual Koszul complex:

$$K_2 : 0 \to R^1 \to R^2 \to R^1 \to 0.$$

xy is a zero divisor in $R/(x^2, y^2)$ and we obtain $J_3 = (x, y)$. Again the Koszul complex is a resolution of R/J_3.

$$L_3 : 0 \to R^1 \to R^2 \to R^1 \to 0.$$

Now d'_0 sends e_1 to $x \in R$ and $xyd'_0(e_1) = x^2y$ has a representation by ye_1 in the lower R^2. A representation for $xyd'_0(e_2) = xy^2$ is given by xe_2. Thus, we define F_1 by $F_1(e_1) = ye_2$ and $F_1(e_2) = xe_2$.

$d'_1(e_1) = ye_1 - xe_2$ by definition of the Koszul complex. We conclude $F_1 d'_1(e_1) = y^2 e_1 - x^2 e_2$ which is just the image of e_1 under d_1 in the lower Koszul complex. Hence, F_2 is defined by $F_2(e_1) = e_1$.

We see that the total complex K_3 of the double complex $\hat{K}_3$ contains absolute entries, i.e., it is not minimal. Indeed, this property reflects the fact that the syzygy $y^2 e_1 - x^2 e_2$ of (x^2, y^2) is not a minimal generator of

$$Syz((x^2, y^2, xy)) = < ye_1 - xe_3, xe_2 - ye_3 > .$$

4 The Algorithm

The implementation of the ideas of the preceding Chapter is realized as SMO-algorithm in the terminology of Chapter 2. After the sequence of generators we use secondly the module index and than, thirdly, the degree of the lowest common multiples of the pairs as ordering criteria on the pair sets.

Within the computation of free resolutions each new generator (which is either a polynomial from the input or a syzygy of certain generators) is equipped with a unique module component which stores the reduction performed with this generator. Moreover, the assigned module components are the basing units for the syzygies. The relation of the ordering of generators and that of their corresponding components is very sensible. Our practical experience indicates that the orderings introduced by F.-O. Schreyer (see [6]) are the best choice for computations of free resolutions.

Here, we have to use another class of orderings:

Definition 4.1 *Assume that the ordering $<$ on R^p fulfils either:*

$$i < j \longrightarrow me_i < m'e_j$$

or

$$i < j \longrightarrow me_i > m'e_j$$

for any $\{me_i, m'e_j\} \subset R^p$. Such an ordering is called a sequential monomial ordering if it agrees with the addition of new elements within an algorithm to compute free resolutions, i.e., module components assigned to new generators are greater w.r.t. $<$ than all the former components.

REMARK: The definition requires to give new generators greater components for the first and smaller components for the second variant. Note that the definition is restricted to only 2 possible variants. In general any ordering for which new generators have components greater than all former are working.
In the sequel we always assume to work with the first sequential ordering from the definition.

Now, we describe the several steps of a sequential algorithm for an ideal $I = (f_1, \ldots, f_n) \subset R$:

1. We start with index $i = 1$.
2. We increase i and return the result if $i = n + 1$.
3. We normalize the generator f_i with respect to the ideal I_{i-1}. If $Normalform(f_i) \neq 0$ we put it to the set of generators and compute a standardbasis of I_i as well as of the syzygies of I_i.
4. We decide wether $Normalform(f_i)$ yields a regular extension by the criteria of the Chapter 3.1. If so, we go to the next step. If not, we skip the next step.

5. We are in the case of a regular extension. We copy the resolution of I_{i-1} and the mappings F_i as multiplications with $Normalform(f_i)$. Now, we are going back to step 2.

6. We are in the case of a non-regular extension. We initialize an inner loop over the module index starting with 0.

7. We are going to the next module. If there are no new generators we return to step 2. If there are new generators, we assign new module components to the new generators and add them to the known generators.

8. We perform a standard basis computation on all pairs build from the new generators. With the new syzygies we go back to the step 7.

The recursion step of the algorithm can be visualized by the following figures (for a proof see Lemma 5.1): The first figure shows the resolution after the $(i-1)$-th step.

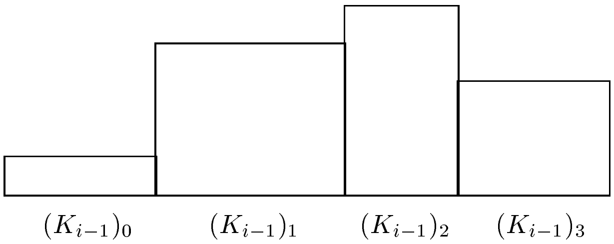

Figure 1: (i-1)–th subresolution

The j-th block of this figure shows the size of the matrix of the j-th map in the $(i-1)$-th subresolution. Therefore, the hight of the j-th block is equal to the length of the $(j-1)$-th block. The situation after the computation of the i-th subresolution is shown in Figure 2.

The old figure corresponds to the empty blocks. In the first matrix a new column is added for the new generator f_i. Corresponding to this the second matrix has one new row (vertically hatched). Analogously, any higher matrix contains as many more new generators (horizontally hatched) as the next matrix new rows (vertically hatched). Further, the vertically hatched blocks of index > 0 correspond to a free resolution of the i-th extension ideal J_i, whereas, the horizontally hatched blocks correspond to the representations chosen in Lemma 3.6.

If the i-th extension is regular the horizonally hatched blocks can be chosen as identical copies of the $(i-1)$-th subresolution with index shifted by 1. Thus,

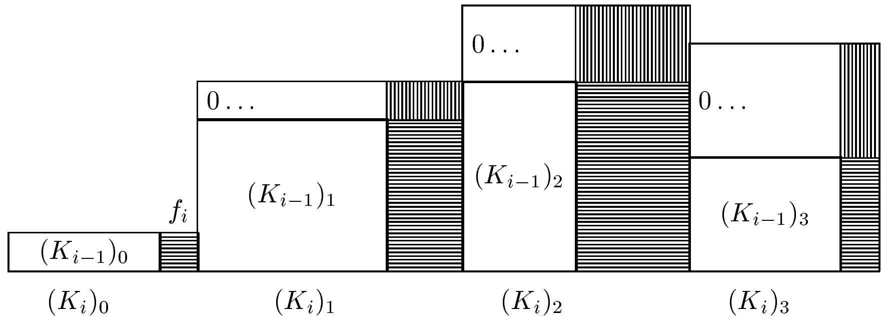

Figure 2: i–th subresolution

if all matrices of the $(i-1)$-th subresolution are given as standard bases the i-th subresolution consists of standard bases, too.

The algorithm is implemented in **SINGULAR** realizing the following pseudo code: $L_* := \mathbf{Res}(\mathbf{I})$
INPUT: $I = (f_1, \ldots, f_n)$, a basis of an ideal
OUTPUT: L_* a resolution of R/I;

> $i := 0$;
> $I_0 := (0)$;
> `WHILE` $i < n$ `DO`
>
> > $i := i + 1$;
> > f_i`:=Normalform(`f_i`,`I_{i-1}`);`
> > `IF` $f_i \neq 0$ `DO`
> >
> > > $I_i := I_{i-1} + (f_i)$;
> > > `IF` $is\_regular\_extension(f_i, I_{i-1}, I_i)$ `DO`
> > >
> > > > K_i`:=Koszul_extension(`K_{i-1}`,`f_i`);`
> > >
> > > `ELSE`
> > >
> > > > `WHILE` $(K_i)_j \neq (K_{i-1})_j$ `DO`
> > > > . $(K_i)_{j+1} := reduce\_new\_generators((K_i)_j)$
> > > > `OD`
> > >
> > > `FI`
> >
> > `FI`
>
> `OD`
> $L_* := K_n$

5 Special Improvements

The sequential algorithm allows some improvements of the reduction procedure in comparison with other implementations of free resolutions.

Lemma 5.1 *The computation of a standard basis of any higher syzygy modules, i.e. within $(K_i)_j$ for any $i = 0, \ldots, n$ and $j > 0$, is devided into a standard basis computation on a partial module and the computation of normal forms.*

Proof: Let us look at Figure 4 The module $(K_i)_j$ contains $(K_{i-1})_j$ via a subset of generators which is, moreover, a standardbasis of $(K_{i-1})_j$. Further, there is a $k > 0$ such that the first k module components (w.r.t. the ordering) correspond exactly to the new generators of $(K_i)_{j-1}$ in the i-th subresolution compared with the $(i-1)$-th. Finally, any generator of $(K_i)_j$ is a syzygy of $(K_i)_{j-1}$.

The syzygies of the subset $(K_{i-1})_j$ are computed in the $(i-1)$-th subresolution. Thus, we have to consider only pairs of module elements whose leading term is in the first k components. Let us assume that a reduction of such a pair leads to an element sp with leading term not in these k components. Then it remains, of course, a syzygy of $(K_i)_{j-1}$. Indeed, because of its leading term it is a syzygy of $(K_{i-1})_{j-1}$ and, hence, it lies in $(K_{i-1})_j$. As $(K_{i-1})_j$ is given as a standard basis the reduction could be completed by the computation of the normal form of sp w.r.t. $(K_{i-1})_j$. $\square$

Remark: Compared with Chapter 3.3 the computation of the standard basis of the partial module is exactly the computation of the next syzygy module of the resolution of the extension ideal J_i. The normal form correspond to the choice of representation of $F_i d'_{j+1}(e_s)$.

For the original ideal (or, module) the situation is different: The leading term of the new generators f_i may ly within the given leading ideal (or, module). Thus, generators of the standard basis of I_{i-1} might be reduced by the extension coming from f_i. In this case, the generator of the standard basis of I_{i-1} is replaced by its reductum if it is not contained in the set $\{f_1, \ldots, f_n\}$ (which may happen only in the non-homogeneous case).

Moreover, there is a new criterion concerning the dependence from sets of generators for the computation of the standard basis of an ideal I:

Lemma 5.2 *Let $e_i \in R^i$ denote the component assigned to the new generator f_i. As soon as the reductum of the s-polynomial sp of an arbitrary pair of elements of I_i has a representation whose leading term is not a multiple of e_i, the pair can be skipped from the reduction.*

Proof: When the leading term of the representaion of sp is not a multiple of e_i this means simply $sp \in I_{i-1}$. But, the standard basis of I_{i-1} as well as its syzygies are just computed in the $(i-1)$-th subresolution. $\square$

6 Timings and Choice of Examples

We compare the implementation of the sequential algorithm *Seq* with the standard algerithm *Res*, Schreyer's algorithm *Schreyer*, LaScala's algorithm *LaScala* and the plain Groebner basis computation *Groebner*.

The timings of Singular are always greater than 0.5 sec, whence, the times smaller than that are set to be zero. The examples were computed on a Pentium III 500 MHz with 1GB RAM running under linux, kernel version 2.2.10 As some of the examles are very extensive (random cubics and quintics) we do not give a complete list of the examples. They can be downloaded (as well as Singular itself) via anonymous ftp from
www.mathematik.uni-kl.de:/pub/Math/Singular
There is a file test_syz.sing containing a Singular version of the tested rings and ideals. All examples are global, homogeneous and computed over the base field $\mathbf{F}_{32003}$.

At first we give an overview over the sequence of extension of our examples:

Example	Number of generators	Regularity sequence
Alex4	3	rrn
standard	3	rrr
f(11,10,3,1)	3	rrr
h(6)	3	rrr
h(7)	3	rrr
g(6,8,10,5,5;0)	3	rrr
f(19,19,4,1)	3	rrr
(max5)$\hat{2}$	15	rnnnnnnnnnnnnnn
randomSyz1	4	rrrr
randomSyz2	5	rrrrr
cyclic roots 5	5	rrrrn
cyclic roots 5 homog	5	rrrrr
Schwarz6	6	rrrrrr
Kahn4	5	rrrrr
Iarrobino1	15	rrrnnnnnnnnnnnn
Random5,3	3	rrr
Alex1 Mora	3	rrn
Alex 6	3	rrn
katsura5	6	rrrrrr
mac1	3	rrn
mac2	3	rnn
mac3	3	rrn
cyclic roots 6 homog	6	rrrrnr
cyclic6 reordered	6	rrrrnr

Here "r" stands for a regular while "n" stands for a non-regular extension. It can be seen that especially the cyclic examples have an untypical defect compared with other: The last extension is always I_{n-1}-regular whereas there are non-regular extensions before. This behaviour results from the homogenization of a constant monomial with a new variable in the last generator. Therefore, it is questionable to use them for more than a computational challenge.

The table of timings looks like follows:

Example	Groebner	Res	Schreyer	LaScala	Seq
Alex4	-	2.1	-	-	-
standard	-	0.7	-	-	-
f(11,10,3,1)	-	0.8	-	-	-
h(6)	-	0.6	-	-	-
h(7)	-	1.0	-	-	-
g(6,8,10,5,5;0)	-	-	0.6	-	-
f(19,19,4,1)	-	17.7	1.6	-	-
(max5)$\hat{2}$	-	0.5	12.5	-	-
randomSyz1	-	1.1	1.0	0.6	0.8
randomSyz2	-	15.9	25.7	8.2	4.5
cyclic roots 5	-	5.2	-	-	-
cyclic roots 5 homog	-	2000.9	0.6	-	-
Schwarz6	-	0.7	1.6	-	-
Kahn4	0.8	27.8	67.1	52.2	28.6
Iarrobino1	-	4.6	-	1.0	6.5
Random5,3	1.4	3.4	3.4	0.7	2.7
Alex1 Mora	0.9	18.0	16.7	-	2.2
Alex 6	1.1	25.8	9.2	0.5	2.4
katsura5	-	9.5	3.5	5.7	0.7
mac1	-	2.6	1.8	-	-
mac2	7.5	58.4	86.8	-	6.9
mac3	9.2	86.1	39.8	-	-
cyclic roots 6 homog	-	29162.5	116.4	501.4	2.1
cyclic6 reordered	-	2482.3	1344.6	103.0	10.7

As it is expected the sequential algorithm works especially well on the cyclic examples. Besides this, the algorithm seems to benefit from a great amount of computations in the higher modules of syzygies, whereas it is disadvantageously to have the Groebner basis as the major part of the computation of a free resolution. Our experience shows that the efficiency of the sequential algorithm depends mainly on the fast computation of the extending resolutions L_i. Therefore, it can be expected that a hybrid of the sequential algorithm and the Hilbert-driven algorithm for the extending resolution L_i is the best implementation of the sequential algorithm.

References

[1] A.Capani, G. DeDominicis, G. Niesi, L. Robbiano, Computing minimal finite free resolution, Preprint (1996), Universita di Genova, 1-15.

[2] D. Buchsbaum, A generalized Koszul complex I, Trans. Amer. Math. Soc. 111 (1964),183-196.

[3] D. Eisenbud, Commutative Algebra with a View Toward Algebraic Geometry, GTM 150, Springer (1995).

[4] J.L. Koszul, Homologie et cohomologie des algebres de Lie, Bull. Soc. Math. de France, 78 (1950), 65-127.

[5] R. LaScala and M. Stillman, Strategies for computing minimal free resolutions, J. Symb. Comput. 26 (1998), No.4, 409-431.

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

[7] Th. Siebert, Algorithms for the computation of free resolutions, in Algorithmic Algebra and Number Theory, B.H. Matzat, G.-M. Greuel, G. Hiss (Eds.), Springer (1998), 295-310.

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.