UNIVERSITÄT KAISERSLAUTERN
Zentrum für Computeralgebra

REPORTS ON COMPUTER ALGEBRA
NO. 17

An algorithm for constructing isomorphisms of modules

by

Thomas Siebert

January 1998

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/`

An algorithm for constructing isomorphisms of modules

Thomas Siebert

March 17, 1998

Contents

1 Introduction

This paper is a continuation of a joint paper with B. Martin [MS] dealing with the problem of direct sum decompositions. The techniques of that paper are used to decide wether two modules are isomorphic or not. An positive answer to this question has many applications - for example for the classification of maximal Cohen-Macaulay module over local algebras as well as for the study of projective modules. Up to now computer algebra is normally dealing with equality of ideals or modules which depends on chosen embeddings. The present algorithm allows to switch to isomorphism classes which is more natural in the sense of commutative algebra and algebraic geometry.

Let R be a finitely generated (local) k-algebra without zerodivisors. Let M and M' be two modules given via minimal representation matrices A and A'. Then $M \simeq M'$ if and only if there are matrices $U, V \in Gl(R)$ such that $UAV = A'$. We shall descibe a finite algorithm to either compute the matrices U and V or to disprove isomorphism.

2 The space of transformation matrices

Throughout this paper we assume R to be a finitely generated, local (or graded) k-algebra without zerodivisors and any (any graded) module M as given as cokernel of a map of free modules. The presentation matrix of this map is assumed to represent a minimal system of generators of its image and is denoted by $A(M)$ or simply A.

The case of a graded module is handled analogously. Therefore, we restrict ourself to the local settings.

Let two modules M and M' be given by their representation matrices A and A'. Assume the modules to be isomorphic. There are two quadratic matrices U and V such that $UAV = A'$. By the above properties of the representations both matrices have to be invertible, $U \in Gl_m(R)$ and $V \in Gl_m(R)$.

Now, let's look from the other side: Starting with two representation matrices A and A' a necessary condition for an isomorphism of the represented modules M and M' is an identical size of both matrices. Denote this size by $mxn,\ m, n \in \mathbf{N}$. The sufficient condition for an isomorphism is again the existence of matrices $U \in Gl_m(R)$ and $V \in Gl_n(R)$ with $UAV = A'$. That means, we have to resolve the equation

$$XA - A'Y = 0,\ X \in M_m(R),\ Y \in M_n(R) \tag{1}$$

and to look for a pair (X_0, Y_0) of regular matrices among all solutions of (1).

First, we determine the module of all possible transformations $(X, Y) \in M_m(R)xM_n(R)$. Let us consider the matrix A as map

$$\phi(A) : Hom(R^m, R^m) \longrightarrow Hom(R^n, R^m)$$

just by multiplying with A from the right. Analogously, we set

$$\psi(A') : Hom(R^n, R^n) \longrightarrow Hom(R^n, R^m)$$

to be the map induced by the multiplication with A' from the left. The map

$$\Phi_{(A,A')} : Hom(R^m, R^m)xHom(R^n, R^n) \longrightarrow Hom(R^n, R^m)$$

given by $\Phi_{(A,A')}(X, Y) = XA - A'Y$ is a well defined module homomorphism and its kernel is the module $Tr_{(A,A')}$ of possible transformations. $Tr_{(A,A')}$ can be computed in a single syzygy computation.

3 Finding a regular transformation

Let $Tr_{(A,A')}$ be given by generators $(X_i, Y_i),\ i = 1, \ldots, r$. We are searching for linear combinations $X(\underline{f}) = \sum_{i=1}^{r} f_i X_i$ and $Y(\underline{f}) = \sum_{i=1}^{r} f_i Y_i$ such that $det(X(\underline{f})) \neq 0$ and $det(Y(\underline{f})) \neq 0$ simultaniously.

3.1 The local case

At this point the local case is much easier to handle. For the invertibility of $X(\underline{f})$ and $Y(\underline{f})$ it is sufficient that its constant parts have full rank. Hence, we can restrict ourself to $X^c(\underline{c}) = \sum_{i=1}^r c_i X_i^c$ and $Y^c(\underline{c}) = \sum_{i=1}^r c_i Y_i^c$ with X^c and Y^c denoting the constant parts of the corresponding matrices and $c_i \in k$. It follows that $det(X^c(\underline{c}))$ and $det(Y^c(\underline{c}))$ are homogeneous polynomials of degree m and n respectively in the indeterminates $c_i,\ i = 1, \ldots, r$. We have to find a point $P_{\underline{c}}$ not lying on the projective hypersurface determined by $F(\underline{c}) = det(X^c(\underline{c}))det(Y^c(\underline{c}))$.

Lemma 3.1 *M and M' are isomorphic if and only if $F(\underline{c}) \neq 0$.*

Proof: This statement is obvious.

Now, assuming that $F(\underline{c}) \neq 0$, we can recursively insert $m+n+1$ different integer values for any c_i on which F depends. As $deg(F) \leq m+n$ for one of these values F does not vanish and we can repeat the procedure with $F(c_i = p_i)$ instead of F. Choosing at the end arbitrary (for example 0) values for the free c_i (Those on which F does not depend!) we obtain the desired point P.

3.2 The global homogeneous case

Here, the algorithm is in principle the same, but, we had to apply it to the set of purely constant matrices. Thus, we had to determine this vector space first.

Denote by (X_i^m, Y_i^m) the non-constant parts of the generators of $Tr_{(A,A')}$. To eliminate them we had to compute their syzygies with the generators (X_i, Y_i). That means, if $M =< (X_i^m, Y_i^m) > +Tr_{(A,A')}$ then $V =< M, M > Syz(M)$ is just the vector space of constant transformations. The algorithm of the local case completes the computation.

4 The algorithm

Here we give the "pseudo"-code of the algorithm.

$M :=$ **iso_modules**(A, A')

INPUT: (A, A') - a pair of representation matrices of modules M, M'

OUTPUT: (X_0, Y_0) - a pair of transformation matrices if M, M' are isomorphic and FALSE otherwise

```
A  := minimize(A)
A' := minimize(A')
IF (size(A)≠size(A')) THEN return FALSE END
M  := transformation(A, A')
M^c := constant_part(M)
F  := det_of_linear_combination(M^c)
IF (F == 0) THEN return FALSE END
```

```
P := point_outside_surface(F)
IF (P == ∅) THEN return FALSE END
(X_0, Y_0) := linear_comb(M, P)
return (X_0, Y_0)
```

The *constant_part* procedure computes the vector space of all constant transformations depending on the ordering as described above. Note, that in case *char* $k = 0$ a point P exists whenever $F \neq 0$ whereas in *char* $k = p$ this must not be true. It follows the code of the main subprocedure - the other are selfevident.

$M :=$ **transformation**(A, A')

INPUT: as above

OUTPUT: M - the module of all solutions of $XA - A'Y = 0$, where every column is of dimension $m^2 + n^2$ and represents a pair of matrices (X, Y)

```
Ã := kontra_hom(A)
Ã' := ko_hom(A')
C := concat(Ã, -Ã')
M := syz(C)
return M
```

References

[MS] B. Martin and Th. Siebert, A splitting algorithm for vector bundles, Preprint (1997), Centre for computer algebra, University Kaiserslautern.

List of papers published in the Reports on Computer Algebra series

[1] 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. 1996.

[2] H. Schönemann. Algorithms in singular. 1996.

[3] R. Stobbe. FACTORY: a C++ class library for multivariate polynomial arithmetic. 1996.

[4] O. Bachmann and H. Schönemann. A Manual for the MPP Dictionary and MPP Library. 1996.

[5] O. Bachmann, S. Gray, and H. Schönemann. MPP: A Framework for Distributed Polynomial Computations. 1996.

[6] G.M. Greuel and G. Pfister. Advances and improvements in the theory of standard bases and szyzygies. 1996.

[7] G.M. Greuel. Description of SINGULAR: A computer algebra system for singularity theory, algebraic geometry, and commutative algebra. 1996.

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

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

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

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

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

[13] O. Bachmann. MPT – a library for parsing and Manipulating MP Trees. January 1997.

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

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

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

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

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

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