ALGORITHM 628
An Algorithm for Constructing Canonical Bases of Polynomial Ideals

F. WINKLER, B. BUCHBERGER, F. LICHTENBERGER
Johannes Kepler Universität
H. ROLLETSCHEK
Kent State University

Categories and Subject Descriptors: G.4 [**Mathematics of Computing**]: Mathematical Software; I.1.1 [**Algebraic Manipulation**]: Expressions and Their Representation; I.1.2 [**Algebraic Manipulation**]: Algorithms; J.2 [**Computer Applications**]: Physical Sciences and Engineering

General Categories: None

Additional Key Words and Phrases: Church–Rosser property, computer algebra, Gröbner bases, polynomial ideals, simplification

---

1. THE PROBLEM OF CONSTRUCTING GRÖBNER BASES FOR POLYNOMIAL IDEALS

The notion of Gröbner bases for polynomial ideals, which is central to this paper, is given by the following:

Definition. A finite set F of polynomials in $K[x_1, \ldots, x_n]$ is called a *canonical basis* or *Gröbner basis* (for the ideal generated by F) if and only if (GB) for arbitrary polynomials $f, g, h \in K[x_1, \ldots, x_n]$:

if $f \to_F g$, $f \to_F h$, and g, h are irreducible modulo $\to_F$, then $g = h$.

Here, "$f \to_F g$" means that "f may be reduced to g modulo F" by applying a certain reduction process that may be considered as a "generalized division." For the exact definition of this reduction relation and examples, we refer to [4] and [6]. (GB) is equivalent to the assertion that $\to_F$ has the Church–Rosser property, whose fundamental importance in rewrite systems is well known (see, e.g., [22]).

The problem of constructing Gröbner bases for polynomial ideals is characterized by the following:

Given a finite set F of polynomials in $K[x_1, \ldots, x_n]$,

---

Received December 1980; revised August 1984; accepted November 1984
Sponsored by the Austrian Research Fund under Grant No. 3877.
Authors' addresses: F. Winkler, Department of Computer and Information Sciences, University of Delaware, Newark DE 19711; B. Buchberger and F. Lichtenberger, Institut für Mathematik, Johannes Kepler Universität, A-4040 Linz, Austria; H. Rolletschek, Department of Mathematics, Kent State University, Kent OH 44242.

find a finite set G of polynomials in $K[x_1, \ldots, x_n]$
such that
ideal(F) = ideal(G) and
G is a Gröbner basis.

Here we assume that for any finite set of polynomials H, ideal(H) denotes the ideal generated by the polynomials in H.

In [3], [4], [6], [9], [20], [22], [23], and [27]–[30] the relevance of the construction of Gröbner bases for constructive polynomial ideal theory and computer algebra has been explained in detail. Briefly summarizing, Gröbner bases are important because many decision and computation problems for polynomial ideals may be solved easily for ideals given by a Gröbner basis, whereas these problems may be extremely difficult for ideals given by arbitrary bases. Among these problems are the problem of deciding whether a given polynomial belongs to a given ideal, the construction of a vector space basis for the residue class ring modulo a given ideal, the question of whether a set of algebraic equations has a solution, the problem of deciding whether a given ideal is zero dimensional, the computation of the elimination ideals for a given ideal, the reduction of polynomials to canonical forms in the presence of polynomial side relations, and the problem of deciding whether a given ideal is principal (see also the examples in Section 3).

In [2–6], [10], and [18] an algorithm for constructing Gröbner bases for polynomial ideals and gradually refined versions of this algorithm have been developed. In this paper we present a FORTRAN implementation of this algorithm for the case of polynomials over the rational number field (however, the coefficient domain can easily be changed to any other field, as long as the user provides programs for the basic arithmetical operations). In order to avoid rewriting standard algorithms, we use the SAC-1 FORTRAN subroutines for operations with integers of arbitrary length [14], which are based on the SAC-1 FORTRAN subroutines for list processing [13]. All identifiers of subroutines and functions not explicitly declared in the program listing are from the SAC-1 system. All these portions of the SAC-1 system are supplied on the tape; that is, the software supplied is self-contained, and so the user can ignore this fact if desired. Out of all the various computer algebra systems we chose the SAC system because it is written in FORTRAN and therefore it is easily available on a great variety of machines.

In the algorithm any ideal basis F is treated as a sequence rather than a set of polynomials. Roughly, this algorithm has the following structure:

```
G := F
B := {{i, j} : 1 ≤ i < j ≤ length of F}
while B not empty do
  {I, J} := an element of B
  B := B − {{I, J}}
  if Criterion(G, B, I, J) then
    h := S-polynomial(G_I, G_J)
    h = Normalform(h, G)
    if h ≠ 0 then
      G := (G, h)
      B := B ∪ {{i, length of G} : 1 ≤ i < length of G}
G := Minor(B)
```

The specifications of the subroutines Criterion, S-polynomial, Normalform, and Minor are given in the listing of the program. Roughly, Normalform reduces the input polynomial with respect to the given basis, Criterion checks whether the pair of indices I and J might possibly lead to a new basis polynomial, S-polynomial computes a polynomial whose reduced form modulo G is a candidate for a new basis polynomial, and Minor eliminates unnecessary portions of the result after a Gröbner basis has been computed.

2. COMPUTATIONAL EXPERIENCE

In the case of univariate polynomials $F_1, \ldots, F_m$ of arbitrary degree the algorithm specializes to Euclid's algorithm. In the case of multivariate linear polynomials the algorithm specializes to Gauss' algorithm. Thus, the behavior of the algorithm in these special cases is well known.

In the case of bivariate polynomials $F_1, \ldots, F_m$ of arbitrary degree an upper bound for the number of steps of the algorithm may be found in [8]. This bound is $\frac{3}{2} \cdot (m + 2 \cdot (D + 2)^2)^4$, where D is the maximum degree of the polynomials in $F_1, \ldots, F_m$. For the case of three variables it is shown in [33] and [34] that $(8D + 1) \cdot 2^d$ is an upper bound for the degrees of the polynomials which are generated during the execution of the Gröbner basis algorithm, where D is as above and d denotes the minimum degree of the polynomials $F_1, \ldots, F_m$.

No reasonable theoretical upper bound for the time complexity of the algorithm in the general case is known so far. The work of Cardoza et al. [11] shows that the problem is intrinsically difficult: A special case of it, namely the uniform word problem for commutative semigroups, is complete in exponential space under log-space transformability. The ordering of the power products plays an essential role in the complexity behavior of the algorithm [17]. For complexity considerations under certain restrictions (generic case, homogeneity) we refer to [21] and [25]. A number of test computations [32] have been performed.

The execution times of two typical examples are 12.16 seconds for the basis with three polynomials in three variables given in [3] and 42 minutes for the basis with six polynomials in six variables given in [30] (measurements for the IBM 370/155).

Besides the implementation described in this paper, several other implementations of earlier versions of the algorithm have been carried out so far [2, 19, 28–30]). Our new implementation contains the refinements of the algorithm derived in [6], [10], and [18]. In [20], [28–30], [35], and [36] the algorithm is applied to various problems in algebraic geometry and computer algebra. In particular, in [30] the algorithm has been successfully applied in solving a system of algebraic equations for which no solution had been known (see [23]). Recently Gebauer and Kredel [15] implemented the algorithm in the SAC-2 system and successfully applied it to problems that had been unsolved so far.

In the next section we give some examples that show the reader how to use the algorithm for effectively solving practical problems of the above kind. The examples are sufficiently simple to allow description in this limited space and yet show the versatility and broad applicability of the algorithm.

3. SAMPLE APPLICATIONS

Example 1: Exact Solution of Systems of Algebraic Equations. Consider the following system of algebraic equations:

$$\begin{aligned} 4x^2 + xy^2 - z + \tfrac{1}{4} &= 0, \\ 2x + y^2 z + \tfrac{1}{2} &= 0, \\ -x^2 z + \tfrac{1}{2}x + y^2 &= 0. \end{aligned} \tag{1}$$

The application of the algorithm (with respect to the lexicographical term ordering) yields the equivalent system

$$\begin{aligned} &z^7 - \frac{1}{2}z^6 + \frac{1}{16}z^5 + \frac{13}{4}z^4 + \frac{75}{16}z^3 - \frac{171}{8}z^2 + \frac{133}{8}z - \frac{15}{4} = 0, \\ &y^2 - \frac{19{,}188}{497}z^6 + \frac{318}{497}z^5 - \frac{4197}{1988}z^4 - \frac{251{,}555}{1988}z^3 - \frac{481{,}837}{1988}z^2 \\ &\qquad + \frac{1{,}407{,}741}{1988}z - \frac{297{,}833}{994} = 0, \\ &2x + \frac{9276}{497}z^6 - \frac{150}{497}z^5 + \frac{2111}{1988}z^4 + \frac{61031}{994}z^3 + \frac{232833}{1988}z^2 \\ &\qquad - \frac{170084}{497}z + \frac{144407}{994} = 0. \end{aligned} \tag{2}$$

In this system the variables are "separated"; that is, the elimination process has been carried out in an effective manner by the algorithm. The numerical or symbolic computation of the roots may now be achieved by known techniques (see, e.g., [35]). From the Gröbner basis with respect to the lexicographical term ordering all the elimination ideals can be read off immediately. This elimination property of Gröbner bases stems from the fact that for a Gröbner basis G (w.r.t. the lexicographical term ordering)

$$\text{ideal}(G) \cap K[x_i, \ldots, x_n] = \text{ideal}(G \cap K[x_i, \ldots, x_n]) \qquad \text{for} \quad 1 \le i \le n$$

(compare [30]).

For a more complex example we refer to [30], where a system of six algebraic equations in six unknowns is solved by transforming them to a Gröbner basis for the corresponding polynomial ideal. The solution was needed by Matzat [23] for constructing certain number fields having Galois group M_{11} over $\mathbf{Q}(\sqrt{-11})$.

Example 2: Simplification of Radical Expressions. Simplification of symbolic expressions is one of the fundamental issues in "symbolic and algebraic computation" (computer algebra). Simplification algorithms for radical expressions, for instance, involve the construction of the residue class ring modulo polynomial ideals (see [12]).

As an easy example consider the problem of rationalizing the denominator of

$$\frac{1}{x + 2^{1/2} + 3^{2/3}}.$$

This problem may be solved by considering the given expression as an element in $\mathbf{Q}(x)[2^{1/2}, 3^{1/3}]$, which is isomorphic to $\mathbf{Q}(x)[y_1, y_2]/\text{ideal}(y_1^2 - 2, y_2^3 - 3)$, that is, the polynomial ring in the two indeterminates y_1, y_2 over the rational function field $\mathbf{Q}(x)$ modulo the ideal generated by the polynomials $y_1^2 - 2$ and $y_2^3 - 3$.

The application of the algorithm yields the equivalent Gröbner basis

$$y_1^2 - 2, \qquad y_2^3 - 3;$$

that is, it is shown by the application of the algorithm that the given basis is already a Gröbner basis. (In fact, in this simple case this can be shown by the theoretical criterion 4; see [10, p. 46], implemented as the subroutine CRIT4 in the algorithm.)

In residue class rings modulo ideals generated by Gröbner bases, arithmetic can be carried out effectively because a linearly independent vector space basis for these rings is readily available by taking the residue classes of the power products in normal form (see [3]). In particular, inverses may be computed effectively if they exist. We demonstrate this procedure for the above example: The residue classes of

$$1, \quad y_1, \quad y_2, \quad y_1y_2, \quad y_2^2, \quad y_1y_2^2$$

form a vector space basis for $\mathbf{Q}(x)[y_1, y_2]/\text{ideal}(y_1^2 - 2, y_2^3 - 3)$. In order to obtain the inverse of $x + 2^{1/2} + 3^{2/3}$, we merely have to solve the equation

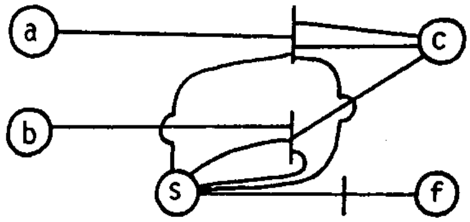
$$(x + y_1 + y_2^2) \cdot (a_1 + a_2y_1 + a_3y_2 + a_4y_1y_2 + a_5y_2^2 + a_6y_1y_2^2) = 1.$$

By using the reductions $y_1^2 \to 2$, $y_2^3 \to 3$, this yields a linear system of equations in the unknowns $a_1, \ldots, a_6$, whose solution is

$$\begin{aligned}
a_1 &= (x^5 - 4x^3 + 9x^2 + 4x + 18)/d,\\
a_2 &= (-x^4 + 4x^2 + 18x - 4)/d,\\
a_3 &= (3x^3 + 18x + 27)/d,\\
a_4 &= (-9x^2 - 6)/d,\\
a_5 &= (-x^4 - 9x + 4)/d,\\
a_6 &= (2x^3 - 4x - 9)/d,
\end{aligned}$$

where $d = x^6 - 6x^4 + 18x^3 + 12x^2 + 108x + 73$.

Example 3: Reachability Problem for Reversible Petri nets. A reversible Petri net consists of places and transitions, whose firing behavior is determined by a



set of rules (see [11]). For instance,

is a Petri net with places a, b, c, f, s and three transitions that may be described by the rules

$$as \longrightarrow ccs,$$

$$bs \longrightarrow cs,$$

$$s \longrightarrow f,$$

where it is implicitly assumed that the "reverse" rules $ccs \longrightarrow as$, etc. are available too.

Our algorithm may be applied in the following way for solving the reachability problem for such Petri nets: Take the rules as a set F of polynomials in the indeterminates a, b, c, f, s and construct the corresponding Gröbner basis G. Then, in the Petri net the marking $a^i b^j c^k f^l s^m$ is reachable from the marking $a^{i'} b^{j'} c^{k'} f^{l'} s^{m'}$ if and only if their normal forms with respect to G are the same.

In our example,

$$F = \{as - c^2 s, \ bs - cs, \ s - f\}.$$

Application of the algorithm (with respect to the graduated lexicographical term ordering) yields

$$G = \{s - f, \ cf - bf, \ b^2 f - af\}.$$

$a^5 b c^3 f^2 s^3$ is reachable from $a^5 b^2 c^2 s^5$ because the normal forms of both markings are $a^7 f^5$ (with respect to G), whereas cs^2 is not reachable from $c^2 s$ because their respective normal forms are distinct, namely bf^2 and af.

Example 4: Cubature Formulas for Multiple Integrals. A cubature formula of degree d for a multiple integral operator I is an equality of the form

$$I(f) = \sum_{j=1}^{N} c_j \cdot f(x^{(j)}) + R(f),$$

where the points $x^{(j)} \in \mathbf{R}^n$ are the "knots" and the $c_j \in \mathbf{R}$ are the "coefficients." $R(f)$ is the "rest," which should be zero for polynominals f of degree less than d (see [26] for an overview on cubature formulae of the above and also more general types). Among the questions intensively studied in this theory within the last decade are the specification of bounds for the number of knots and the generation of cubature formulas with knots being roots of polynomials.

We cannot go into the details of these investigations here. For showing the relevance of Gröbner bases for the computational aspects of this theory, we quote one of the typical results (Theorem of H. M. Möller; see [26, p. 225]):

> If $f_1, \ldots, f_s$ are a canonical basis of a zero-dimensional polynomial ideal and $f_1, \ldots, f_s$ are d-orthogonal with respect to I,
> then there exists a (generalized) cubature formula for I of degree d (with the common roots of $f_1, \ldots, f_s$ as knots).

The number of knots may be bounded bv $H(d;\ (f_1, \ldots, f_s))$, the value of the Hilbert function (see [24]).

From the theorem quoted we see that for the practical application of the theory it is essential to have a computational procedure for

— checking whether the dimension of a polynomial ideal is zero,
— checking whether an ideal basis is canonical, and
— determining the values of the Hilbert function.

For polynomials $f_1, \ldots, f_s$ of special types these questions may be answered by using theoretical results from polynomial ideal theory, for instance, M. Noether's theorem (or our Criterion 4 in the algorithm). For arbitrary polynomials these questions can be effectively answered by using the algorithm under discussion. Given $F := \{f_1, \ldots, f_s\}$, one can compute the corresponding Gröbner basis $G := \{g_1, \ldots, g_t\}$. A Gröbner basis is always canonical (see [7]). The criterion for zero dimensionality for Gröbner bases is simply the appearance of power products of the form $x_1^{i_1}, \ldots, x_n^{i_n}$ among the leading terms of $g_1, \ldots, g_t$. The value of the Hilbert function $H(d;\ (g_1, \ldots, g_t))$ for Gröbner bases (with respect to the graduated lexicographical term ordering) is the number of power products of degree $\leq d$, which are in normal form with respect to G.

For instance, if

$$f_1 = x_3^2 - \tfrac{1}{2}x_2^2 - \tfrac{1}{2}x_1^2,$$

$$f_2 = x_1x_3 + x_1x_2 - 2x_3,$$

$$f_3 = x_1^2 - x_2,$$

the application of the algorithm yields

$$g_1 = x_1^2 - x_2$$

$$g_2 = x_1x_3 + x_1x_2 - 2x_3$$

$$g_3 = x_2x_3 + x_2^2 + 2x_1x_2 - 4x_3$$

$$g_4 = x_3^2 - \tfrac{1}{2}x_2^2 - \tfrac{1}{2}x_2$$

$$g_5 = x_1x_2^2 + 6x_2^2 + 7x_1x_2 - 16x_3 + 2x_2$$

$$g_6 = x_2^3 - 29x_2^2 - 24x_1x_2 + 64x_3 - 12x_2.$$

$G := \{g_1, \ldots, g_6\}$ is a canonical basis; ideal$(g_1, \ldots, g_6)$(= ideal(f_1, f_2, f_3)) is zero dimensional because x_1^2, x_2^3 and x_3^2 appear among the leading terms of $g_1, \ldots, g_6$. The power products that are in normal form with respect to G are

$$1,$$
$$x_1,\quad x_2,\quad x_3,$$
$$x_1x_2,\quad x_2^2.$$

Hence

$$H(0; G) = 1,$$
$$H(1; G) = 4,$$
$$H(2; G) = H(3; G) = \cdots = 6.$$

(Thus the "order" $h_0(G)$ of G, i.e., the vector space dimension of $\mathbf{R}[x_1, x_2, x_3]$ modulo ideal(f_1, f_2, f_3), is 6.)

Example 5: Analysis of Algebraic Varieties. The Lasker-Noether representation theorem (see, e.g., [16]) for polynomial ideals I generated by polynomials $f_1, \ldots, f_s \in K[x_1, \ldots, x_n]$ is a means for analyzing the algebraic variety formed by the common roots of $f_1, \ldots, f_s$. Essentially, it yields the irreducible components of the variety and the multiplicity of the roots. The fundamental notions in this theory are those of the prime ideals and primary ideals associated with I. If I is zero dimensional, by a theorem of van der Waerden (see [31, p. 146]), the primary ideal q corresponding to a prime ideal p associated with I may be computed by

$$q = (I, p^\rho),$$

where ρ is the least natural number, such that $p^\rho \subseteq (I, p^{\rho+1})$ (= the "exponent" of p).

This procedure as it stands is not yet computationally effective because the containment $p^\rho \subseteq (I, p^{\rho+1})$ cannot be checked effectively, in general. Our algorithm, however, yields a solution for this subproblem: The basis of $(I, p^{\rho+1})$ is transformed to an equivalent Gröbner basis G. For Gröbner bases, the problem "$f \in$ ideal(G)" (the so-called "Hauptproblem" of polynomial ideal theory) may be decided by

$$f \in \text{ideal}(G) \qquad \text{if and only if} \quad f \rightarrow_G 0.$$

For Gröbner bases, $f \rightarrow_G 0$ may be decided by reducing f to a normal form f' with respect to G and checking whether $f' = 0$.

The generation of prime ideals associated with I may be effected by computing common roots of $f_1, \ldots, f_s$, a task for which our algorithm is relevant again (see Example 1).

On the basis of the above ideas, Schrader [28] implemented the van der Waerden procedure. One of his examples is

$$f_1 = x^4y^4 + y^6 - x^2y^4 - x^4y^2 + x^6 - y^4 + 2x^2y^2 - x^4,$$
$$f_2 = x^3y^4 - \tfrac{1}{2}xy^4 - x^3y^2 + \tfrac{3}{2}x^5 + xy^2 - x^3,$$
$$f_3 = x^4y^3 + \tfrac{3}{2}y^5 - x^2y^3 - \tfrac{1}{2}x^4y - y^3 + x^2y.$$

(The variety of this system is the center of a Lissajou curve.) The corresponding Gröbner basis with respect to the graduated lexicographic term ordering has the form

$$g_1 = x^4y^2 - x^6 + y^4 - 2x^2y^2 + x^4,$$

$$g_2 = x^2y^4 - x^6 - 2x^2y^2 + 2x^4,$$

$$g_3 = xy^5 - x^5y - 2xy^3 + 2x^3y,$$

$$g_4 = y^6 - x^6 - y^4 + x^4,$$

$$g_5 = x^7 - \tfrac{1}{2}xy^4 + x^3y^2 - \tfrac{1}{2}x^5 + xy^2 - x^3,$$

$$g_6 = x^6y + \tfrac{1}{2}y^5 + x^2y^3 - \tfrac{3}{2}x^4y - y^3 + x^2y.$$

This shows that ideal(f_1, f_2, f_3) is zero dimensional, the residue class ring having vector space dimension 24. One solution of the system is (0, 0), ideal(x, y) therefore is a prime ideal associated with ideal(f_1, f_2, f_3). Its exponent ρ may be computed by the van der Waerden procedure based on our algorithm yielding $\rho = 8$. Hence, (0, 0) has multiplicity 8.

4. FORMAT OF THE PARAMETERS FOR THE FORTRAN SUBROUTINE GROEB

The name of the FORTRAN subroutine that implements the algorithm is GROEB. Corresponding to the problem specification in Section 1, the subroutine GROEB has three parameters: F and G for the input and output sequences of polynomials, respectively, and N for the number of variables.

F and G are linear lists of polynomials. A polynomial is a linear list of terms (power products). Each term is a list of three elements

exp, num, den,

where

exp = a linear list of exponents (each exponent being a SAC-1 atom),
num = the numerator of the coefficient of the term (num is a SAC-1 infinite precision integer),
den = the denominator of the coefficient of the term (den is a SAC-1 infinite precision integer).

For example, the sequence F of the two polynomials (over $\mathbf{Q}[x_1, x_2]$)

$$F_1 := x_1x_2^2 - \tfrac{3}{2}x_1x_2 + 2x_1,$$

$$F_2 := x_1^2 - x_2$$

has to be represented as the list

((((1, 2), (1), (1)),
((1, 1), (− 3), (2)),
((1, 0), (2), (1))),
(((2, 0), (1), (1)),
((0, 1), (− 1), (1)))).

The user must guarantee that the terms of each polynomial in F are ordered in decreasing order according to the linear term ordering used in the algorithm. Every "admissible" linear ordering in the sense of [30], [16, Definition (1.1)] may be used. In the present implementation we provide the subroutines for handling the admissible orderings used in the examples of Section 3, namely the "graduated lexicographical ordering" of [3] and the normal lexicographical ordering used in [30] (see also Section 5). Only one of the two subroutines LINORD may be linked to the program at a time.

5. POSSIBLE MODIFICATIONS OF THE PROGRAM

The structured design of GROEB and its subroutines makes it easy to modify the program with respect to the following features:

(1) change of the term ordering,
(2) change of the underlying field,
(3) change of the criterion,
(4) packing of exponents,
(5) defining the available space.

The modification may be carried out by suitable replacements of a few subroutines (see commentaries).

6. MAIN PROGRAM FOR SAMPLE CALCULATIONS

In order to test the subroutine GROEB on a specific installation, we include a main program that produces sample output for a number of input sequences F of polynomials and shows how the subroutine GROEB has to be embedded into the SAC-1 system.

The structure of the main program is as follows:

(1) Initiate the SAC-1 system.
(2) Read a sequence F of polynomials of N variables until N is set to 0.
(3) Compute a Gröbner basis G for F by applying the subroutine GROEB.
(4) Print the resulting basis G.
(5) goto 2.

We apply the main program to the following sequences of polynomials:

First sequence:

$$F_1 := x^2y - xy,$$

$$F_2 := xy^2 + xy + y.$$

This sequence has to be presented to the main program in the following format (b denoting a blank):

b2	number of variables
b2	number of polynomials
b2	number of terms in the polynomial F_1
+1	numerator of the coefficient of x^2y (the sign is obligatory)
+1	denominator of the coefficient of x^2y
bb2bb1	exponents of x^2y (each exponent covers three positions)
−1 +1	coefficient of xy

```
bb1bb1      exponents of xy
b3          number of terms in the polynomial F2
+1     }
+1     }    representation of xy^2
bb1bb2 }
+1     }
+1     }    representation of xy
bb1bb1 }
+1     }
+1     }    representation of y(= x^0 y^1)
bb0bb1 }
```

For this input sequence the main program (with the graduated lexicographic term ordering) will produce the following output:

$$G_1 = xy - y,$$

$$G_2 = y^2 + 2y.$$

Given a second sequence (see Example 4 in Section 3),

$$F_1 := z^2 - \tfrac{1}{2}y^2 - \tfrac{1}{2}x^2,$$

$$F_2 := xz + xy - 2z,$$

$$F_3 := x^2 - y,$$

the program will return the Gröbner basis

$$G_1 = x^2 - y,$$

$$G_2 = xz + xy - 2z,$$

$$G_3 = yz + y^2 + 2xy - 4z,$$

$$G_4 = z^2 - \tfrac{1}{2}y^2 - \tfrac{1}{2}y,$$

$$G_5 = xy^2 + 6y^2 + 7xy - 16z + 2y,$$

$$G_6 = y^3 - 29y^2 - 24xy + 64z - 12y.$$

7. IMPLEMENTATION OF THE ALGORITHM

As already stated in the previous section, our algorithm relies on the SAC-1 system for list processing and integer arithmetic (handling "arbitrarily" long integers). Thus the according SAC-1 subsystems [13, 14] have to be implemented first together with the SAC-1 basic system. Once this is done, the user may start submitting a main program (e.g., the one provided on the distribution tape) together with GROEB and its subroutines.

The program is completely self-contained, unless the word size on the host computer is less than 32 bits. In this case the value of the variables BETA and THETA in the main program should be decreased (but THETA should always be the greatest power of 10, which is less than BETA).

In order to run the program on a machine with 16-bit integer variables the variable BETA should be set to 2**14 and the variable THETA to 10**4.

REFERENCES

1. BACHMAIR, L., AND BUCHBERGER, B. A simplified proof of the characterization theorem for Gröbner-bases. *ACM SIGSAM Bull. 14/4*, (1980), 29–34.
2. BUCHBERGER, B. Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal. Ph.D. dissertation, Universität Innsbruck, Innsbruck, Austria, 1965.
3. BUCHBERGER, B. Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems. *Aequ. Math. 4* (1970), 374–383.
4. BUCHBERGER, B. A theoretical basis for the reduction of polynomials to canonical forms. *ACM SIGSAM Bull. 10/3* (1976), 19–29.
5. BUCHBERGER, B. Some properties of Gröbner-bases for polynomial ideals. *ACM SIGSAM Bull. 10/4* (1976), 19–24.
6. BUCHBERGER, B. A criterion for detecting unnecessary reductions in the construction of Gröbner-bases. In *Proceedings of EUROSAM 79*, Lecture Notes in Computer Science, vol. 72. Springer-Verlag, Berlin, Heidelberg, New York, 1979, pp. 3–21.
7. BUCHBERGER, B. *H*-bases and Gröbner-bases for polynomial ideals. Tech. Report CAMP 81-2.0, Institut für Mathematik, Univ. of Linz, Linz, Austria (1981).
8. BUCHBERGER, B. A Note on the complexity of constructing Gröbner-bases. In *Proceedings of the European Computer Algebra Conference* (*EUROCAL* '83), Lecture Notes in Computer Science, vol. 162. Springer-Verlag, Berlin, Heidelberg, New York, 1983, pp. 137–145.
9. BUCHBERGER, B. Gröbner bases: An algorithmic method in polynomial ideal theory. In *Recent Trends in Multidimensional Systems Theory*, N. K. Bose, Ed. Reidel, Hingham, Mass., 1985.
10. BUCHBERGER, B., AND WINKLER, F. Miscellaneous results on the construction of Gröbner-bases for polynomial ideals. Tech. Rep. 137, Institut für Mathematik, Univ. of Linz, Linz, Austria (1979).
11. CARDOZA, E., LIPTON, R., AND MEYER, A. R. Exponential space complete problems for Petri nets and commutative semigroups: Preliminary report. In *Proceedings of the 8th Annual ACM Symposium on Theory of Computing.* ACM, New York, 1976, pp. 50–54.
12. CAVINESS, B. F., AND FATEMAN, R. J. Simplification of radical expressions. In *Proceedings of the 1976 ACM Symposium on Symbolic and Algebraic Computation* (Yorktown Heights, N.Y., R. D. Jenks, ed. ACM, New York, 1976, pp. 329–338.
13. COLLINS, G. E. The SAC-1 list processing system. Tech. Rep. 129, Computer Science Dept., Univ. of Wisconsin—Madison, 1971.
14. COLLINS, G. E. The SAC-1 integer arithmetic system. Tech. Rep. 156, Computer Science Dept., Univ. of Wisconsin—Madison, 1971.
15. GEBAUER, R., AND KREDEL, H. An algorithm for constructing canonical bases (Groebner-bases) of polynomial ideals. Tech. Rep., Institute for Applied Mathematics, Univ. of Heidelberg, Heidelberg, W. Germany (1984).
16. GRÖBNER, W. Moderne Algebraische Geometrie. Springer-Verlag, Wien-Innsbruck, 1949.
17. KOLLREIDER, C. Polynomial reduction: The influence of the ordering of terms on a reduction algorithm. Tech. Rep. 124, Institut für Mathematik, Univ. of Linz, Linz, Austria (1978).
18. KOLLREIDER, C., AND BUCHBERGER, B. An improved algorithmic construction of Gröbner-bases. *ACM SIGSAM Bull. 12/2* (1978), 27–36.
19. LAUER, M. Kanonische Repräsentanten für die Restklassen nach einem Polynomideal. Diplomarbeit, Institut für Informatik, Univ. Kaiserslautern, Kaiserslautern, W. Germany (1976).
20. LAUER, M. Canonical representatives for residue classes of a polynomial ideal. In *Proceedings of the ACM Symposium on Symbolic and Algebraic Computation* (Yorktown Heights, N.Y.) ACM, New York, 1976, pp. 339–345.
21. LAZARD, D. Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations. In *Proceedings of the European Computer Algebra Conference* (*EUROCAL* '83) (London), J. A. van Hulzen, Ed., 146–156 (1983).
22. LOOS, R. Toward a formal implementation of computer algebra. In *Proceedings of the EUROSAM '74, ACM SIGSAM Bull. 8/3* (1974), 9–16.
23. MATZAT, B. H. Konstruktion von Zahlkörpern mit der Galoisgruppe M_{11} über $\mathbf{Q}(\sqrt{-11})$. *Manuscr. Math. 27* (1979), 103–111.

24. MÖLLER, H. M. Mehrdimensionale Hermite-Interpolation und numerische Integration. *Math. Z. 148* (1976), 107–118.
25. MÖLLER, H. M., AND MORA, F. Upper and lower bounds for the degree of Groebner bases. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation (EUROSAM '84)* (Cambridge, England), pp. 172–183 (1984).
26. MYSOVSKIKH, I. P. The approximation of multiple integrals by using interpolatory cubature formulae. In *Quantitative Approximation*, R. A. De Vore and K. Scherer, Eds. Academic Press, New York, (1980), pp. 217–243.
27. SCHALLER, S. C. Algorithmic aspects of polynomial residue class rings. Ph.D. dissertation, Tech. Rep. 370, Computer Science Dept., Univ. of Wisconsin—Madison, 1979.
28. SCHRADER, R. Zur konstruktiven Idealtheorie. Diplomarbeit, Institut für Mathematik, Univ. Karlsruhe, Karlsruhe, W. Germany, 1976.
29. SPEAR, D. A. A constructive approach to commutative ring theory. In *Proceedings of the 1977 MACSYMA User's Conference* (1977), pp. 369–376.
30. TRINKS, W. Über B. Buchbergers Verfahren, Systeme algebraischer Gleichungen zu lösen. *J. Number Theory 10* (1978), 475–488.
31. VAN DER WAERDEN, B. L. *Algebra II*. Springer-Verlag, Berlin, Heidelberg, New York, 1967.
32. WINKLER, F. Implementierung eines Algorithmus zur Konstruktion von Gröbner-Basen. Diplomarbeit, Institut für Mathematik, Univ. of Linz, Linz, Austria (1978).
33. WINKLER, F. On the complexity of the Gröbner-bases algorithm over $K[x, y, z]$. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation (EUROSAM '84)* (Cambridge, England), pp. 184–194 (1984).
34. WINKLER, F. The Church–Rosser property in computer algebra and special theorem proving: An investigation of critical-pair/completion algorithms. Ph.D. dissertation, Institut für Mathematik, Univ. Linz, Linz, Austria (1984).
35. YUN, D. Y. Y. On algorithms for solving systems of polynomial equations. *ACM SIGSAM Bull.* No. 27 (1973).
36. ZACHARIAS, G. Generalized Gröbner bases in commutative polynomial rings. B.S. thesis, Massachusetts Institute of Technology (1978).