Multidimensional Systems Theory

Progress, Directions and Open Problems in Multidimensional Systems

Edited by

N. K. Bose

School of Engineering, University of Pittsburgh, U.S.A.

With contributions by
N. K. Bose, J. P. Guiver, E. W. Kamen,
H. M. Valenzuela, and B. Buchberger

D. Reidel Publishing Company

A MEMBER OF THE KLUWER ACADEMIC PUBLISHERS GROUP

Dordrecht / Boston / Lancaster

Library of Congress Cataloging in Publication Data

CIP

Main entry under title:

Multidimensional systems theory.
(Mathematics and its applications)
Includes bibliographies and index.
1. System analysis. I. Bose, N. K. (Nirmal K.), 1940– . II. Guiver, J. P. III. Series: Mathematics and its applications (D. Reidel Publishing Company)
QA402.M83 1984 003 84–15060
ISBN 90–277–1764–8

Published by D. Reidel Publishing Company,
P.O. Box 17, 3300 AA Dordrecht, Holland

Sold and distributed in the U.S.A. and Canada
by Kluwer Academic Publishers,
190 Old Derby Street, Hingham, MA 02043, U.S.A.

In all other countries, sold and distributed
by Kluwer Academic Publishers Group,
P.O. Box 322, 3300 AH Dordrecht, Holland

Printed in The Netherlands.

Chapter 6

B. Buchberger

Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory

6.1. INTRODUCTION

Problems connected with ideals generated by finite sets F of multivariate polynomials occur, as mathematical subproblems, in various branches of systems theory, see, for example, [6.1]. The method of Gröbner bases is a technique that provides algorithmic solutions to a variety of such problems, for instance, exact solutions of F viewed as a system of algebraic equations, computations in the residue class ring modulo the ideal generated by F, decision about various properties of the ideal generated by F, polynomial solution of the linear homogeneous equation with coefficients in F, word problems modulo ideals and in commutative semigroups (reversible Petri nets), bijective enumeration of all polynomial ideals over a given coefficient domain etc.

For many years, the work of G. Hermann [6.2] was the only algorithmic method for tackling problems in polynomial ideal theory. Still, her paper is a rich source. However, as pointed out in [6.3] and [6.4], the solution of her main problem "is a multivariate polynomial f in the ideal generated by F?" does not yet give a feasible solution to the "simplification problem modulo an ideal" (i.e. the problem of finding unique representatives in the residue classes modulo the ideal) and to the problem of effectively computing in the residue class ring modulo an ideal.

The method of Gröbner bases, as its central objective, solves the simplification problem for polynomial ideals and, on this basis, gives easy solutions to a large number of other algorithmic problems including Hermann's original membership problem. Also, when compared with Hermann's algorithms, our algorithm that constructs Gröbner bases is of striking simplicity and, depending on the example considered, may get through with intermediate computations using polynomials of relatively low degree. On the other hand, as shown in [6.5] and [6.6], the decision of polynomial ideal congruence intrinsically is a complex problem. In the worst case, therefore, also the method of Gröbner bases may lead to

N. K. Bose (ed.), Multidimensional Systems Theory, 184–232

exploding computations. Much work is going on to analyze and predict these phenomena and to extend the applicability of the method.

The method of Gröbner bases was introduced 1965 by this author in [6.7], [6.8] and, starting from 1976, was further refined, generalized, applied and analyzed in a number of papers [6.9]–[6.35]. The basic idea of the method is the transformation of the given set of polynomials F into a certain standard form G, for which in [6.9] the author introduced the name 'Gröbner bases', because Prof. W. Gröbner, the thesis advisor of [6.7] stimulated the research on the subject by asking how a multiplication table for the associative algebra, which is formed by the residue ring modulo a polynomial ideal, can be constructed algorithimically and by presenting a first sketch of an algorithm: He proposed to 'complete' the basis F by adjoining the differences of different representations of power products (modulo the ideal). This, however, is no finite procedure. It was the author's main contribution to see and prove in [6.7], [6.8] that it suffices to adjoin the differences of (the reduced forms of) certain 'critical pairs' (or, equivalently, the reduced form of the 'S-polynomials' [6.7]), which are finite in number.

In retrospect, it seems that the concept of 'Gröbner bases' under the name "standard bases" appeared already one year earlier (1964) in Hironaka's famous paper [6.36]. However, Hironaka only gave an inconstructive existence proof for these bases, whereas in [6.7], together with the concept of such bases, we also presented an algorithm for constructing the bases and only this algorithm allows an algorithmic solution to the various problems shortly mentioned above. An inconstructive existence proof for Gröbner bases may also be found in [6.37]. Hilbert's basis theorem, then, follows as a corollary.

Later (1967) the two basic ideas of our method, critical pairs and completion, where also proposed by Knuth and Bendix [6.38] in the more general context of equations between first order terms. The Knuth-Bendix algorithm now plays an important role in various branches of computer science (abstract data type transformations, equational theorem proving and applications in automated program verification). Recently, the Knuth-Bendix algorithm and the author's own algorithm for constructing Gröbner bases were brought together under a common algorithm structure by R. Llopis de Trias [6.32] and, independently, by P. Le Chenadec [6.39]; see also [6.3] for a general introduction to the "critical-pair completion" algorithm type. On the other hand, the improvements of the author's algorithm were carried over to the Knuth-

Bendix algorithm, see [6.40]. A lot of challenging questions remain to be treated, which, in the near future, might also affect systems theory (for example, decision methods for boolean algebra based on the critical-pair/completion approach, see [6.41].)

In the present paper, a survey on the method of Gröbner bases is given. In Section 6.2, the concept of Gröbner bases is defined and, in Section 6.3, the basic form of the algorithm for constructing Gröbner bases is described. In Section 6.4 an improved version of the algorithm is presented. The improvements are important for the practical feasibility of the computations. In Section 6.5, the algorithm is applied to the simplification problem, the congruence problem and related problems in polynomial ideal theory. In Section 6.6, the algorithm is applied to the exact solution of systems of algebraic equations and related problems. In Section 6.7, it is demonstrated that the S-polynomials have also a significance as the generators of the module of solutions for linear homogeneous equations with polynomial coefficients and an algorithm for a systematic solution of such equations is presented. Gröbner bases for polynomial ideals with integer coefficients are treated in Section 6.8. Some other applications are summarized in Section 6.9. Finally, in Section 6.10, some remarks about specializations, generalizations, implementations and the computational complexity of the algorithm are made.

The emphasis of this paper is on explicit formulation of algorithms (in an easy notation) and on examples. With the exception of some sketches, no proofs of the underlying theorems can be given. However, complete references to the original publications are provided.

6.2. GRÖBNER BASES

Notation

K	*a field.*
$K[x_1, \ldots, x_n]$	ring of n-variate polynomials over K.

The following typed variables will be used:

f, g, h, k, p, q	polynomials in $K[x_1, \ldots, x_n]$.
F, G	finite subsets of $K[x_1, \ldots, x_n]$.
s, t, u	power products of the form $x_1^{i_1} \ldots x_n^{i_n}$.
a, b, c, d	elements in K.
i, j, l, m	natural numbers.

Let $F = \{f_1, \ldots, f_m\}$. By 'Ideal(F)' we will denote "the ideal generated by F' (i.e. the set

$$\{ \sum_{l \leq i \leq m} h_i \cdot f_i | h_i \epsilon K[x_1, \ldots, x_n] (i = 1, \ldots, m)\}).$$

Furthermore, we will write '$f \equiv_F g$' for "f is congruent to g modulo Ideal(F)" (i.e. f-$g\epsilon$Ideal(F)).

Before one can define the notion of Gröbner bases the notion of 'reduction' must be introduced. For this it is necessary to fix a total odering $<_T$ of the power products $x_1^{i_1} \ldots x_n^{i_n}$, for example, the 'total degree ordering' (which is $1 <_T x <_T y <_T x^2 <_T xy <_T y^2 <_T x^3 <_T x^2y <_T xy^2 <_T y^3 <_T \ldots$ in the case of two variables) or the 'purely lexicographical ordering' (which is $1 <_T x <_T x^2 <_T x^3 <_T \ldots y <_T xy <_T x^2y <_T \ldots <_T y^2 <_T xy^2 <_T \ldots$ in the case of two variables). In fact, any total ordering is suitable, which at least has the following two properties:

(T1) $1 <_T t$ *for all* $t \neq 1$,

(T2) if $s <_T t$ *then* $s \cdot u <_T t \cdot u$.

A total ordering satisfying (T1) and (T2) will be called 'admissible'. For the sequel, assume that an arbitrary $<_T$ has been fixed. With respect to the chosen $<_T$, we use the following notation.

Notation

Coefficient(g, t)	the coefficient of t in g.
LeadingPowerProduct(f)	the maximal power product (*w.r.t.* $<_T$) occurring with non-zero coefficient in f.
LeadingCoefficient(f)	the coefficient of the LeadingPowerProduct(f).

DEFINITION 6.1 [6.7], [6.8].
$g \rightarrow_F h$ (read: 'g *reduces* to h modulo F') iff there exists $f \epsilon F$, b and u such that

$$g \rightarrow_{f, b, u} \quad \text{and} \quad h = g - b \cdot u \cdot f.$$

$g \rightarrow_{f, b, u}$ (read: 'g is reducible using f, b, u') iff Coefficient(g, $u \cdot$ LeadingPowerProduct(f)) $\neq 0$, $b =$ Coefficient(g, $u \cdot$ LeadingPowerProduct(f))/LeadingCoefficient(f) ●

Hence, roughly, g reduces to h modulo F iff a monomial in g can be deleted by the subtraction of an appropriate multiple $b \cdot u \cdot f$ of a polynomial f in F yielding h. Thus, the reduction may be viewed as one step in a generalized division.

EXAMPLE 6.1. Consider $F: = \{f_1, f_2, f_3\}$, where

$$f_1: = 3x^2y + 2xy + y + 9x^2 + 5x - 3,$$
$$f_2: = 2x^3y - xy - y + 6x^3 - 2x^2 - 3x + 3,$$
$$f_3: = x^3y + x^2y + 3x^3 + 2x^2.$$

The polynomials f_1, f_2, f_3 are ordered according to the purely lexicographical ordering. The leading power products are x^2y, x^3y, x^3y, respectively, and the leading coefficients are 3, 2, and 1. Consider

$$g: = 5y^2 + 2x^2y + 5/2xy + 3/2y + 8x^2 + 3/2x - 9/2.$$

Modulo F, g reduces, for example, to

$$h: = 5y^2 + 7/6xy + 5/6y + 2x^2 - 11/6x - 5/2.$$

Namely,

$$g \rightarrow_{f,\, b,\, u} \quad \text{for} \quad f: = f_1, \qquad b: = 2/3, \qquad u: = 1$$

because Coefficient(g, $1 \cdot x^2y$) $= 2 \neq 0$ and $b =$ Coefficient(g, $1 \cdot x^2y$)/ LeadingCoefficient(f_1),
and

$$h = g - (2/3) \cdot 1 \cdot f_1.$$

DEFINITION 6.2.
h is in *normal form* (or reduced form) modulo F iff there is no h' such that $h \rightarrow_F h'$.
h is *a normal form of* g modulo F iff there is a sequence of reductions

$$g = k_0 \rightarrow_F k_1 \rightarrow_F k_2 \rightarrow_F \ldots \rightarrow_F k_m = h$$

and h is in normal form modulo F.
An algorithm S is called a *normal form algorithm* (or simplifier) iff for all F and g:

$$S(F,\ g) \text{ is a normal form of } g \text{ modulo } F.$$

LEMMA 6.1 [6.7] [6.9].
The following algorithm is a normal form algorithm:

ALGORITHM 6.1 (h: = NormalForm(F, g)).

```
$h: = g$
while exist $f \in F$, $b$, $u$ such that $h \rightarrow_{f,\, b,\, u}$ do choose $f \in R$, $b$, $u$ such
        that $h \rightarrow_{f,\, b,\, u}$ and $u \cdot$ LeadingPowerProduct($f$) is maximal
        (w.r.t. $<_T$)
    $h: = h - b \cdot u \cdot f$                                        ●
```

The correctness of this algorithm should be clear. For the correctness, the selection of the maximal product $u \cdot$ LeadingPowerProduct(f) is not mandatory. However, this choice is of crucial importance for efficiency. The termination of the algorithm is guaranteed by the following lemma.

LEMMA 6.2 [6.7], [6.9]. For all F: $\rightarrow_F$ is a noetherian relation (i.e. there is no infinite sequence $k_0 \rightarrow_F k_1 \rightarrow_F k_2 \rightarrow_F \ldots$).

EXAMPLE 6.2. h in the Example 6.1 is in normal form modulo F: no power product occurring in h is a multiple of the leading power product of one of the polynomials in F. Thus, no reduction is possible. Another example:

$$x^3y \rightarrow_{f_1} - 2/3x^2y - 1/3xy - 3x^3 - 5/3x^2 + x = :g_1.$$

g_1 can be further reduced:

$$g_1 \rightarrow_{f_1} 1/9xy + 2/9y - 3x^3 + 1/3x^2 + 19/9x - 2/3 = :g_1'.$$

g_1' is in normal form modulo F. g_1', hence, is a normal form of x^3y modulo F. Actually, g_1' may be the result of applying the algorithm 'NormalForm' to x^3y (depending on how the instruction 'choose $f \in F$, such that . . .' in the algorithm is implemented). In this example, a second reduction is possible:

$$x^3y \rightarrow_{f_2} 1/2xy + 1/2y - 3x^3 + x^2 + 3/2x - 3/2 = :g_2.$$

g_2 is already in normal form modulo F.

From the example one sees that, in general, it is possible that, modulo F, g_1 and g_2 are normal forms of a polynomial g, but $g_1 = g_2$. Those sets F, for which such a situation does not occur, play the crucial role for our approach to an algorithmic solution of problems in polynomial ideal theory:

DEFINITION 6.3 [6.7], [6.9]. F is called a *Gröbner basis* (or Gröbner set) iff for all g, h_1, h_2:

if h_1 and h_2 are normal forms of g modulo F then $h_1 = h_2$. ●

It is the central theme of this paper to show that

(a) for those sets F that are Gröbner bases, a number of important algorithmic problems (that are formulated in terms of Ideal(F)) can be solved elegantly and

(b) those sets F, which are not Gröbner bases, can be transformed into sets G, that are Gröbner bases and generate the same ideal.

Most of the algorithmic applications of Gröbner bases are based on the following fundamental property of Gröbner bases.

THEOREM 6.1 [6.7], [6.9], [6.22] (Characterization Theorem for Gröbner bases). Let S be an arbitrary normal form algorithm. The following properties are equivalent:

(GB1) F is a Gröbner basis.

(GB2) For all f, g: $f \equiv_F g$ iff $S(F, f) = S(F, g)$. ●

(GB1) is also equivalent to:

(GB3) $\rightarrow_F$ has the 'Church-Rosser' property.

(GB3) links Gröbner bases with analogous concepts for equations of first order terms and the Knuth-Bendix algorithm. For details see [6.3]. (GB3) is not needed in this paper. The following lemma is helpful in establishing this link.

LEMMA 6.3 [6.22], [6.30] (Connection between reduction and congruence): For all F, f, g:

$$f \equiv_F g \quad \text{iff} \quad f \overset{*}{\leftrightarrow}_F g.$$

(Here, $\leftrightarrow_F^*$ is the reflexive, symmetric, transitive closure of $\rightarrow_F$, i.e.

$$f \leftrightarrow_F^* g \quad \text{iff} \quad \text{there exists a sequence}$$

$$f = k_0 \leftrightarrow_F k_1 \leftrightarrow_F k_2 \leftrightarrow_F \ldots \leftrightarrow_F k_m = g,$$

where

$$f \leftrightarrow_F g \quad \mathit{iff} \quad (f \rightarrow_F g \quad \text{or} \quad g \rightarrow_F f)).$$ ●

(GB2) immediately shows that, for Gröbner bases F, the decision problem '$f \equiv_F g$' is algorithmically decidable (uniformly in F). For Gröbner bases, other computability problems will have similarly easy solutions: see Sections 5–9.

6.3. ALGORITHMIC CONSTRUCTION OF GRÖBNER BASES

Before we give the algorithmic applications of Gröbner bases we show how it may be decided whether a given set F is a Gröbner basis and how Gröbner bases may be constructed. For this the notion of an 'S-polynomial' is fundamental:

DEFINITION 6.4 [6.7], [6.8], [6.9].
The '*S-polynomial* corresponding to f_1, f_2' is
SPolynomial(f_1, f_2): $= u_1 \cdot f_1 - (c_1/c_2) \cdot u_2 \cdot f_2$,
where c_i = LeadingCoefficient(f_i),
u_i is such that $s_i \cdot u_i$ = the least common multiple of s_1, s_2 and
s_i = LeadingPowerProduct(f_i) $(i = 1, 2)$.

EXAMPLE 6.3. For f_1, f_2 as in Example 6.1, the SPolynomial(f_1, f_2) is

$$2x^2y + 5/2xy + 3/2y + 8x^2 + 3/2x - 9/2. \qquad \bullet$$

Note that the least common multiple of s_1 and s_2 is the minimal power product that is reducible both modulo f_1 and modulo f_2. The algorithmic criterion for Gröbner bases is formulated in the following theorem, which forms the core of the method:

THEOREM 6.2 (Buchberger [6.7], [6.8], [6.9], [6.22]; Algorithmic Characterization of Gröbner bases). Let S be an arbitrary normal form algorithm. The following properties are equivalent:
(GB1) F is a Gröbner basis.
(GB4) For all $f_1, f_2 \in F$: $S(F, S\text{Polynomial}(f_1, f_2)) = 0$. ●

(GB4), indeed, is a decision algorithm for the property 'F is a Gröbner basis': one only has to consider the finitely many pairs f_1, f_2 of polynomials in F, compute the corresponding S-polynomials and see whether they reduce to zero by application of the normal form algorithm S. In addition, Theorem 6.2 is the basis for the central Algorithm 6.2 of this paper for solving the following problem.

PROBLEM 6.1.
Given F.
Find G, such that Ideal(F) = Ideal(G) and G is a Gröbner basis.

ALGORITHM 6.2 (Buchberger [6.7], [6.8]) for Problem 6.1.

```
$G: = F$
$B: = \{\{f_1, f_2\} | f_1, f_2 \in G, f_1 \neq f_2\}$
*while* $B \neq \emptyset$ *do*
    $\{f_1, f_2\}: = a$ pair in $B$
    $B \quad : = B - \{\{f_1, f_2\}\}$
    $h \quad : =$ $S$Polynomial$(f_1, f_2)$.
    $h' \quad : =$ NormalForm$(G, h)$
    *if* $h' \neq 0$ *then*
        $B: = B \cup \{\{g, h'\} | g \in G\}$
        $G: = G \cup \{h'\}$.
```
●

The partial correctness of this algorithm, essentially, relies on Theorem 6.2. The termination can be shown in two ways, see [6.8], [6.17]. (Sketch of the first method [6.17]: One considers the sequence of ideals Ideal$(P_1) \subset$ Ideal$(P_2) \subset \ldots$, where P_i is the set of leading power products of polynomials in G_i and G_i is the value of G after G has been extended for the i-th time. It is easy to see, that the inclusions in this sequence are proper. Hence, by Hilbert's theorem on ascending chains of ideals in $K[x_1, \ldots, x_n]$, see [6.42], the sequence must be finite. Sketch of the second method [6.8]: One uses Dickson's lemma [6.43], which, applied to the present situation, shows that a sequence $t_1, t_2, \ldots$ of power products with the property that, for all j, t_j is not a multiple of any of its predecessors, must be finite. Actually, if t_i is the leading power product of the i-th polynomial adjoined to G in the course of the algorithm ($i = 1, 2, \ldots$), then the sequence $t_1, t_2, \ldots$ has this property and, hence, must be finite. This is the way, the termination of the algorithm was first proven in [6.8], where Dickson's lemma was reinvented. Hilbert's basis theorem can be obtained as a corollary in this approach, see [6.37].)

EXAMPLE 6.4. Starting from the set F of Example 6.1, we first choose, for instance, the pair f_1, f_2 and calculate

$$S\text{Polynomial}(f_1, f_2) =$$
$$2x^2y + 5/2xy + 3/2y + 8x^2 + 3/2x - 9/2.$$

Reduction of this polynomial to a reduced form yields

$$7/6xy + 5/6y + 2x^2 - 11/6x - 5/2.$$

We adjoin this polynomial to G in the form

$$f_4 := xy + 5/7y + 12/7x^2 - 11/7x - 15/7,$$

where we normalized the leading coefficient to 1. (This normalization is not mandatory. However, as a matter of computational experience, it may result in drastic savings in computations over the rationals. Theoretically, this phenomenon is not yet well understood. Investigations of the kind done for Euclid's algorithm should be worthwhile, see [6.44] for a survey on these questions.)

Now we choose, for example, the pair f_1 and f_4:

$$S\text{Polynomial}(f_1, f_4) = 1 \cdot f_1 - (3/1) \cdot x \cdot f_4 =$$
$$-1/7xy + y - 36/7x^3 + 96/7x^2 + 80/7x - 3.$$

Reduction of this polynomial, by subtraction of $-(1/7) \cdot f_4$ (and normalization), yields the new polynomial.

$$f_5 := y - 14/3x^3 + 38/3x^2 + 61/6x - 3.$$

Furthermore, SPolynomial$(f_4, f_5) = 1 \cdot f_4 - (1/1) \cdot x \cdot f_5$. By subtracting $(5/7) \cdot f_5$ and normalization we obtain

$$f_6 := x^4 - 2x^3 - 15/4x^2 - 5/4x.$$

Finally, the reduction of SPolynomial$(f_1, f_3) = x \cdot f_1 - (3/1) \cdot 1 \cdot f_3$ leads to

$$f_7 := x^3 - 5/2x^2 - 5/2x.$$

The reduction of the S-polynomials of all the remaining pairs yields zero and, hence, no further polynomials need to be adjoined to the basis. For example,

$$S\text{Polynomial}(f_6, f_7) = 1/2x^3 - 5/4x^2 - 5/4x$$

reduces to zero by subtraction of $1/2\, f_7$. Hence, a Gröbner basis corresponding to F is

$$G := \{f_1, \ldots, f_7\}.$$

DEFINITION 6.5 [6.10]. *F is a reduced Gröbner basis* iff F is a Gröbner basis and for all $f \epsilon F$: f is in normal form modulo $F - \{f\}$ and LeadingCoefficient$(f) = 1$.

EXAMPLE 6.5. G in Example 6.4 is not a reduced Gröbner basis: For example, f_1 reduces to zero modulo $\{f_2, \ldots, f_7\}$. By successively reducing all polynomials of a Gröbner basis modulo all the other polynomials in the basis and normalizing the leading coefficients to 1, one always can transform a Gröbner basis into a reduced Gröbner basis for the same ideal. We do not give a formal description of this procedure, because it will be automatically included in the improved version of the algorithm below. In the example, also f_2, f_3, f_4, and f_6 reduced to zero and f_5 reduces to

$$f_5' := y + x^2 - 3/2x - 3.$$

Hence, the reduced Gröbner basis corresponding to F is

$$G' := \{f_5', f_7\} = \{y + x^2 - 3/2x - 3, x^3 - 5/2x^2 - 5/2x\}.$$

THEOREM 6.3 (Buchberger [6.10]: Uniqueness of reduced Gröbner bases). If Ideal(F) = Ideal(F') and F and F' are both reduced Gröbner bases then $F = F'$.

DEFINITION 6.6. Let GB be the function that associates with every F a G such that Ideal(F) = Ideal(G) and G is a reduced Gröbner basis. ●

By what was formulated in Theorems 6.2, 6.3, Algorithm 6.2 and the remarks in Example 6.5 we, finally, obtain the following main theorem, which summarizes the basic algorithmic knowledge about Gröbner bases.

MAIN THEOREM 6.4 (Buchberger 1965, 1970, 1976).
GB is an algorithmic function that satisfies for all F, G:
(SGB1) Ideal(F) = Ideal($GB(F)$),
(SGB2) if Ideal(F) = Ideal(G) then $GB(F) = GB(G)$,
(SGB3) $GB(F)$ is a reduced Gröbner basis.

6.4. AN IMPROVED VERSION OF THE ALGORITHM

For the tractability of practical examples it is crucial to improve the algorithm. There are three main possibilities for achieving a computational speed-up:

(1) The order of selection of pairs $\{f_1, f_2\}$ for which the S-polynomials are formed, though logically insignificant, has a crucial influence on the

complexity of the algorithm. As a general rule, pairs whose least common multiple of the leading power products is minimal with respect to the ordering $<_T$ should be treated first. This, in connection with (2), may drastically reduce the computation time.

(2) Each time a new polynomial is adjoined to the basis, all the other polynomials may be reduced using also the new polynomial. Thereby, many polynomials in G may be deleted again. Such reductions may initiate a whole cascade of reductions and cancellations. Also, if this procedure is carried out systematically in the course of the algorithm, the final result of the algorithm automatically is a *reduced* Gröbner basis. The reduction of the polynomials modulo the other polynomials in the basis should also be performed at the beginning of the algorithm.

(3) Whereas (1) and (2) are strategies that need no new theoretical foundation, the following approach is based on a refined theoretical result [6.19], which has proven useful also in the general context of 'critical-pair/completion' algorithms, in particular for the Knuth-Bendix algorithm: The most expensive operations in the algorithm are the reductions of the h' modulo G in the *while*-loop. We developed a 'criterion' that, roughly, allows to detect that certain S-polynomials h can be reduced to zero, without actually carrying out the reduction. This can result in drastic savings. Using this criterion, in favourable situations, only $0(l)$ S-polynomials must be considered instead of $0(l^2)$, where l is the number of polynomials in the basis. (Of course, in general, l is dynamically changing and, therefore, the effect of the criterion is very hard to assess, theoretically).

Strategy 1. was already used in [6.7], [6.8]. Also, the correctness of the reduction and cancellation technique sketched in (2) was already shown in [6.7], [6.8]. The criterion described in (3) was introduced and proven correct in [6.19], details of the correctness proof may be found in [6.20].

Before we give the details of the improved version of the algorithm based on (1)–(3) we present a rough sketch:

In addition to G and B, we use two sets R and P. R contains polynomials of G, which can be reduced modulo the other polynomials of G. As long as R is non-empty, we reduce the polynomials in R and store the resulting reduced polynomials in P. Only when R is empty, we adjoin the reduced polynomials in P to G and determine the new pairs in B for which the S-polynomials have to be considered. If an S-polynomial for a pair in B is reduced with a non-zero result h', h' is put into P and, again, polynomials in G are sought that are reducible with respect to h'. Such polynomials are put into R and we continue with the systematic reduction of R. We now give the details.

PROBLEM 6.2.
Given: F.
Find: G, such that Ideal(F) = Ideal(G) and G is a reduced Gröbner basis.

ALGORITHM 6.3 (Buchberger [6.19]) for Problem 6.2.

```
$R:=F; P:=\emptyset; G:=\emptyset; B:=\emptyset$
Reduce All ($R, P, G, B$); New Basis ($P, G, B$)
while $B \neq \emptyset$ do
        $\{f_1, f_2\}:=$ a pair in $B$ whose $LCM(LP(f_1), LP(f_2))$ is minimal
                w.r.t. $<_T$
        $B:=B-\{\{f_1, f_2\}\}$
        if (not Criterion1($f_1, f_2, G, B$) and
           not Criterion2($f_1, f_2$)) then
              $h:=$ NormalForm($G$, SPolynomial($f_1, f_2$))
              if $h \neq 0$ then
                    $G_0:=\{g \in G | LP(h) \leq_M LP(g)\}$
                    $R:=G_0; P:=\{h\}; G:=G-G_0$
                    $B:=B-\{\{f_1, f_2\} | f_1 \in G_0 \text{ or } f_2 \in G_0\}$
                    ReduceAll($R, P, G, B$); NewBasis($P, G, B$).
```

```
Subalgorithm Reduce All (transient : $R, P, G, B$):
while $R \neq \emptyset$ do
        $h:=$ an element in $R$; $R:=R-\{h\}$;
        $h:=$ NormalForm($G \cup P, h$)
        if $h \neq 0$ then
            $G_0:=\{g \in G | LP(h) \leq_M LP(g)\}$
            $P_0:=\{p \in P | LP(h) \leq_M LP(p)\}$
        $G := G - G_0$
        $P := P - P_0$
        $R := R \cup G_0 \cup P_0$
        $B := B - \{\{f_1, f_2\} \in B | f_1 \in G_0$ or $f_2 \in G_0\}$
        $P := P \cup \{h\}$.
```

Subalgorithm New Basis (*transient*: P, G, B):

```
    $G := G \cup P$
    $B := B \cup \{\{g, p\} | g \in G, p \in P, g \neq p\}$
    $H := G$; $K := \emptyset$
    while $H \neq \emptyset$ do
        $h :=$ an element in $H$; $H := H - \{h\}$
        $k :=$ NormalForm$(G - \{h\}, h)$; $K := K \cup \{k\}$
    $G := K$.
```

Subalgorithm Criterion1(f_1, f_2, G, B): $\Leftrightarrow$ there exists a $p \in G$ such that

$$f_1 \neq p, p \neq f_2,$$

$$LP(P) \leqslant_M LCM(LP(f_1), LP(f_2)),$$

$\{f_1, p\}$ not in B and $\{p, f_2\}$ not in B.

Subalgorithm Criterion2(f_1, f_2): $\Leftrightarrow$

$$LCM(LP(f_1), LP(f_2)) = LP(f_1) \cdot LP(f_2).$$

Abbreviations

$LP(g)$	the leading power product of g,
$LCM(s, t)$	the least common multiple of s and t,
$s \leqslant_M t$	t is a multiple of s.

The correctness of this improved version of the algorithm is based on the following lemma and theorem.

LEMMA 6.4 [6.7], [6.8]. For arbitrary F, f_1, f_2:
If $LP(f_1) \cdot LP(f_2) = LCM(LP(f_1), LP(f_2))$, then SPolynomial (f_1, f_2) can always be reduced to zero modulo F.

THEOREM 6.5 (Buchberger 1979 [6.19]; detection of unnecessary reductions of S-polynomials). Let S be an arbitrary normal form algorithm. The following properties are equivalent:

(GB1) F is a Gröbner basis.

(GB5) For all $f, g \epsilon F$ there exist $h_1, h_2, \ldots, h_k \epsilon F$ such that

$$f = h_1, \qquad g = h_k,$$
$$LCM(LP(h_1), \ldots, LP(h_k) \leq_M LCM(LP(f), LP(g)),$$
$$S(F, S\mathrm{Polynomial}(h_i, h_{i+1})) = 0 \text{ (for } 1 \leq i < k).$$ ●

Lemma 6.4 guarantees that we need not consider the S-polynomial of two polynomials f_1 and f_2, whose leading power products satisfy the condition stated in the lemma (Criterion2). The iteration of Criterion1 in Algorithm 6.3 guarantees that, upon termination of the algorithm, condition (GB5) is satisfied for G and, hence, G is a Gröbner basis.

EXAMPLE 6.6. Let $F: = \{f_1, f_2, f_3\}$, where

$$f_1: = x^3yz - xz^2, \qquad f_2: = xy^2z - xyz, \qquad f_3: = x^2y^2 - z^2.$$

The total degree ordering of power products is used in this example: first order by total degree and, within a given degree, order lexicographically. We took an example with a particularly simple structure of the polynomials in order to make the reduction process simple and to emphasize the crucial point: the difference of the crude version of the algorithm and the improved version, which is reflected in the pairs of polynomials $\{f_1, f_2\}$, for which the S-polynomials have to be considered.

A trace of the crude form of the algorithm could be as follows (if the selection strategy 1. for pairs of polynomials is used: in the trace, we write $f_i, f_j \rightarrow f_k$ for indicating that the reduction of the S-polynomial of f_i and f_j leads to the polynomial f_k):

$$f_2, f_3 \rightarrow f_4: = x^2yz - z^3,$$
$$f_1, f_4 \rightarrow f_5: = xz^3 - xz^2,$$
$$f_2, f_4 \rightarrow f_6: = yz^3 - z^3,$$
$$f_3, f_4 \rightarrow 0,$$
$$f_5, f_6 \rightarrow f_7: = xyz^2 - xz^2,$$
$$f_4, f_7 \rightarrow f_8: = z^4 - x^2z^2,$$

$$f_2, f_7 \to 0,$$
$$f_5, f_7 \to 0,$$
$$f_6, f_7 \to 0,$$
$$f_5, f_8 \to f_9 := x^3z^2 - xz^2,$$
$$f_6, f_8 \to 0.$$

The S-polynomials of all the other pairs are reduced to zero. All together one has to reduce $(9.8)/2 = 36$ S-polynomials.

In the improved algorithm, first, by ReduceAll, f_1, f_2, f_3 are reduced with respect to each other. In this example, this reduction process leaves the original basis unchanged. Then, by NewBasis, f_1, f_2, f_3 are put into G. Simultaneously the set of pairs B for which the S-polynomial have to be considered is generated. The first pair, again, is

$$f_2, f_3 \to f_4.$$

In this phase, again a call to ReduceAll is made. It is detected that, modulo $\{f_2, f_3, f_4\}$, f_1 can be reduced to f_5, hence, f_1 can be deleted from G and, correspondingly, the pairs $\{f_1, f_2\}$ and $\{f_1, f_3\}$ can be deleted from B. By NewBasis, f_4 and f_5 are adjoined to G and B is updated. The consideration of the next pair in B yields

$$f_2, f_4 \to f_6.$$

ReduceAll has no effect in this case. Thus, f_6 is adjoined to the basis immediately and B is updated. The consideration of the next pair $\{f_3, f_4\}$ in B can be skipped by application of Criterion1: $LP(f_2) = xy^2z$ divides $LCM(LP(f_3), LP(f_4)) = x^2y^2z$ and $\{f_3, f_2\}$ and $\{f_2, f_4\}$ are not in B any more, because they already were considered. The consideration of the next pairs in B yields

$$f_5, f_6 \to f_7,$$
$$f_4, f_7 \to f_8,$$

with the corresponding updating of G and B (no reductions and cancellations of polynomials in G are possible!). The S-polynomials of the next pairs reduce to zero

$$f_2, f_7 \to 0,$$
$$f_5, f_7 \to 0.$$