The criterion does not detect this fact a priori! However, the consideration of the next pair $\{f_6, f_7\}$ can, again, be skipped by application of Criterion1: f_3 is a suitable p in the criterion. Then, the following pairs are considered:

$$f_5, f_8 \rightarrow f_9,$$
$$f_6, f_8 \rightarrow 0,$$
$$f_4, f_9 \rightarrow 0.$$

The next pair $\{f_7, f_9\}$ may, again, be skipped by application of Criterion1. Finally,

$$f_5, f_9 \rightarrow 0.$$

From now on, the application of Criterion1 detects a priori, without actually carrying out the reductions, that all the remaining pairs may be skipped. Hence, instead of 36 reductions, only 11 have to be carried out with the improved algorithm. The pair $\{f_3, f_8\}$ is an example of a pair, for which Criterion2 is successful. The gain by using the criteria, in particular Criterion1, becomes more drastic as the complexity of the examples, in terms of the number of variables, the degrees of polynomials and the number of polynomials, increases.

6.5. APPLICATION: CANONICAL SIMPLIFICATION, DECISION OF IDEAL CONGRUENCE AND MEMBERSHIP, COMPUTATION IN RESIDUE CLASS RINGS

In this section, it is shown how our algorithm for constructing Gröbner bases may be applied for algorithmic solutions to the canonical simplification problem modulo polynomial ideals, the decision problems '$f \equiv_F g$' and '$f \in \mathrm{Ideal}(F)$', and the problem of effectively computing in the associative algebra $K[x_1, \ldots, x_n]/\mathrm{Ideal}(F)$. Actually, the three problems are intimately connected with each other. This connection is summarized in the following definitions and lemmas whose proof may be found in [6.3]. The concepts involved in these lemmas have been developed and refined in various papers by B. Caviness, J. Moses, D. Musser, H. Lausch and W. Nöbauer, R. Loos, M. Lauer, and the author: see [6.3] for a detailed reference to the literature.

Let T be an arbitrary (decidable) set (for example, $T := K[x_1, \ldots, x_n]$) and $\sim$ an equivalence relation on T (for example, $\sim \, = \, \equiv_F$).

DEFINITION 6.7. An algorithm C with inputs and outputs in T is called a '*canonical simplifier*' (or 'ample function') for $\sim$ (on T)
iff for all objects f, g in T

(SE) $C(f) \sim f$ and

(SC) if $f \sim g$ then $C(f) = C(g)$,

(i.e. C singles out a unique representative in each equivalence class. $C(f)$ is called a *canonical form* of f).

LEMMA 6.5. *$\sim$ is decidable if there exists a canonical simplifier C for $\sim$.*

Proof. By (SE) and (SC): $f \sim g$ iff $C(f) = C(g)$. The converse of the lemma is true, also. However, the simplification algorithm constructed in the proof of the converse is of no practical value, see [6.3], [6.4].

LEMMA 6.6. Let R be a computable (binary) operation on T, such that $\sim$ is a congruence relation with respect to R. Assume we have a canonical simplifier C for $\sim$. Define:

$$\mathrm{Rep}(T) := \{f \epsilon T \mid C(f) = f\} \text{(set of '\textit{canonical representatives}', ample set)},$$

$$R'(f, g) := C(R(f, g)) \text{ (for all } f, g \epsilon \mathrm{Rep}(T)).$$

Then, $(\mathrm{Rep}(T), R')$ is isomorphic to $(T/\sim, R/\sim)$, $\mathrm{Rep}(T)$ is decidable, and R' is computable. (Here, $R/\sim([f], [g]) := [R(f, g)]$, where $[f]$ is the congruence class of f with respect to $\sim$). ●

Lemma 6.6 shows that, having a canonical simplifier for an equivalence relation that is a congruence with respect to a computable operation, one can *algorithmically* master the factor structure. The theorem is proven by realizing that $i(f) := [f]$ ($f \epsilon \mathrm{Rep}(T)$) defines an isomorphism between the two structures and by checking the computability properties. Applying these general concepts and facts to the case of polynomial ideals we first note:

COROLLARY 6.1 (to Theorem 6.1). *Let S be an arbitrary normal form algorithm in the sense of Definition 6.2 and F a Gröbner basis. Then $C := \lambda f.\ S(F, f)$, i.e. the algorithm, that takes the fixed F and a variable f as input and computes $S(F, f)$, is a canonical simplifier for $\equiv_F$.*

Proof. (SE) is fulfilled because, clearly, $f \equiv_F g$ if $f \rightarrow_F g$ (see *Definition*

6.1). By iteration, $f \equiv_F S(F, f)$. (SC), in case of $\equiv_F$, is just the content of Theorem 6.1. •

In addition, one can prove the following lemma.

LEMMA 6.7 [6.7], [6.8]. *Let F be a Gröbner basis. Then* $B := \{[u] \mid u$ *is a power product that is not a multiple of the leading power product of any of the polynomials in* ~~G~~ F *is a linearly independent vector space basis for the vector space* $K[x_1, \ldots, x_n]/Ideal(F)$ *(the residue class ring modulo* $Ideal(F)$*).*

Proof. Assume that there is a linear dependence

$$c_1 \cdot [u_1] + c_2 \cdot [u_2] + \ldots + c_l \cdot [u_l] = 0$$

for some $\boxed{u_i}$ in B. Then

$$f := c_1 \cdot u_1 + c_2 \cdot u_2 + \ldots + c_l \cdot u_l \in \text{Ideal}(F).$$

Hence, by Theorem 6.1, f must be reducible to 0 modulo F. However, f is already in normal form because, by definition of B, non of the u_i can be reduced modulo F. Thus, $f = 0$, i.e. $c_1 = \ldots = c_l = 0$. •

Based on the above lemmata, the following problems can be solved by the following methods (for S use the normal form algorithm NormalForm described in Algorithm 6.1):

PROBLEM 6.3.
Given F.

Find a canonical simplifier C for the congruence $\equiv_F$ modulo Ideal(F).

METHOD 6.1 [6.12], [6.9].
Compute $G := GB(F)$.

Then the normal form algorithm $S(G, f)$ is a canonical simplifier for $\equiv_F$.

PROBLEM 6.4.
Given F, f, g.

Decide, whether $f \equiv_F g$.

METHOD 6.2 [6.9].
Compute $G := GB(F)$.

Then: $f \equiv_F g$ iff $S(G, f) = S(G, g)$.

PROBLEM 6.5.
Given E, a finite set of equations between generators of a commutative semigroup and two words f, g.

Decide whether the equality $f = g$ is derivable from E.

METHOD 6.3 [6.19] [6.23].
Let $x_1, \ldots, x_n$ be the finitely many generators of the commutative semigroup. Conceive every equation $p = q$ in E as a polynomial $p - q$ in $Q[x_1, \ldots, x_n]$.
Compute $G := GB(E)$.

Then: $f = g$ is derivable from E iff $S(G, f) = S(G, g)$.

PROBLEM 6.6.
Given F, f.

Decide whether $f \in \text{Ideal}(F)$.

METHOD 6.4 [6.9].
Compute $G := GB(F)$.

Then: $f \in \text{Ideal}(F)$ iff $S(G, f) = 0$.

PROBLEM 6.7.
Given F_1, F_2:

Decide whether $\text{Ideal}(F_1) \subseteq \text{Ideal}(F_2)$.

METHOD 6.5 [6.9], [6.10].
Compute $G_2 := GB(F_2)$.

Then: $\text{Ideal}(F_1) \subseteq \text{Ideal}(F_2)$ iff for all $f \in F_1$: $S(G_2, f) = 0$.

PROBLEM 6.8.
Given F.

Find a linearly independent basis B for the vector space $K[x_1, \ldots, x_n]/\text{Ideal}(F)$ (the residue class ring modulo $\text{Ideal}(F)$) and,
for any two basis elements $[u]$ and $[v]$ in B find a linear representation of $[u] \cdot [v]$ in terms of the basis elements in B (i.e. find the 'multiplication table' for $K[x_1, \ldots, x_n]/\text{Ideal}(F)$).

METHOD 6.6 [6.7], [6.8].

Compute $G := GB(F)$.

Take $B := \{[u] | u$ is a power product that is not a multiple of the leading power product of any of the polynomials in $G\}$.

$S(G, u \cdot v)$ yields a linear representation of $[u] \cdot [v]$.

PROBLEM 6.9.

Given F, f, h (where $K[x_1, \ldots, x_n]/\mathrm{Ideal}(F)$ is assumed to be finite-dimensional as a vector space).

Find g, such that $f \cdot g \equiv_F h$ (if such a g exists).

METHOD 6.7.

Compute $G := GB(F)$.

Represent f and h as a linear combination of the elements in B (see Method 6.6) and represent g as a linear combination with unknown coefficients. Thus, one gets a linear system of equations for the unknown coefficients, which is solvable iff a solution g exists. ●

Note, that all the above methods are 'uniform' in the sense that F is a free parameter in the respective algorithms. Thus, for example, Method 6.3 is a solution to the uniform word problem for finitely generated commutative semigroups (which is equivalent, for example, to the reachability problem for reversible Petri nets). It has been proven [6.5], [6.6] that the uniform word problem for finitely generated commutative semigroups and, also, the uniform congruence problem for polynomial ideals in $Q[x_1, \ldots, x_n]$ is exponentially space complete, i.e. is an intrinsically hard problem. Method 6.2 shows that this problem can be 'easily' reduced to the problem of constructing Gröbner bases. Hence, the problem of constructing Gröbner bases must be an intrinsically hard problem. For practice, this means that the worst case behavior of the Algorithm 6.2 and 6.3 may be extremely bad. However, this does not mean that it is useless to construct Gröbner bases, because in the particular cases at hand, the algorithm may perform well (for example, if the input F is 'nearly' a Gröbner basis). Also, if for a given F the Gröbner basis G has been constructed, an infinite number of particular algorithmic problems of the kind '$f \in \mathrm{Ideal}(F)$?', 'compute a representation of $[u] \cdot [v]$' etc. can be solved extremely easily.

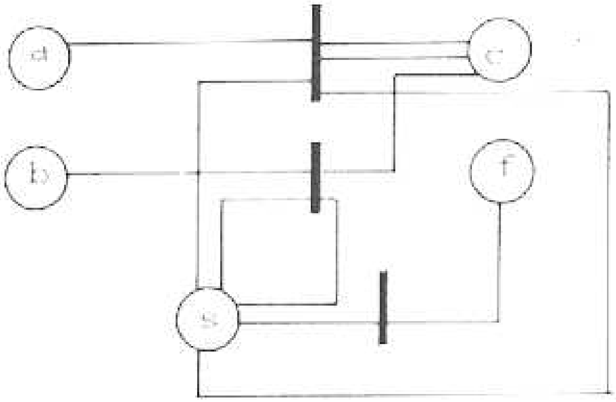
EXAMPLE 6.7. For F as in Example 6.1, $f := xy$ is not in Ideal(F), because

$$S(GB(F), xy) = -x^2 + 1/2x \neq 0.$$

$$f -_{I} g := x^2y + 3/2xy + 1/2y + 3x^2 + 3/2x - 3/2,$$

because $S(GB(F), g)$ is also $-x^2 + 1/2x$.

EXAMPLE 6.8. The following reversible Petri net



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

$$as \rightarrow c^2s,$$

$$bs \rightarrow cs$$

$$s \rightarrow f,$$

where it is implicitly assumed that the 'reverse' rules

$$c^2s \rightarrow as$$

etc. are also available. Let

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

Then: a configuration v is reachable from configuration w iff $v \equiv_F w$. For example, $a^5bc^3f^2s^3$ is reachable from $a^5b^2c^2s^5$ iff $a^5bc^3f^2s^3 =_F a^5b^2c^2s^5$. In order to answer such questions, we first compute (w.r.t the total degree ordering)

$$G := GB(F) = \{s - f, cf - bf, b^2f - af\}.$$

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

EXAMPLE 6.9. For F of Example 6.1,

$$B = \{[1], [x], [x^2]\}$$

is a linearly independent vector basis for $K[x, y]/\text{Ideal}(F)$, see the corresponding reduced Gröbner basis G in Example 6.5,

$$[x] \cdot [x^2] = 5/2[x^2] + 5/2[x],$$

because

$$S(GB(F), x^3) = 5/2x^2 + 5/2x.$$

EXAMPLE 6.10. As an application of the construction of inverses in polynomial residue class rings, we take the simplification of radical expressions. For the formulation of the problem see [6.45]. Consider, for example, 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 $Q(x)[2^{1/2}, 3^{2/3}]$, which is isomorphic to $Q(x)[y_1, y_2]/\text{Ideal}(y_1^2 - 2, y_2^3 - 3)$, i.e. the polynomial ring in the two indeterminates y_1, y_2 over the rational function field $Q(x)$ modulo the ideal generated by the polynomials $y_1^2 - 2$ and $y_2^3 - 3$. The application of the algorithm yields the equivalent Groebner-basis

$$G: = \{y_1^2 - 2, y_2^3 - 3\},$$

i.e. it is shown by the application of the algorithm that the given basis is already a Groebner-basis. (In fact, in this simple case, this can be shown by Criterion2 in Algorithm 6.3.) The residue classes of

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

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

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

By using the reductions $y_1^2 \rightarrow_G 2$, $y_2^3 \rightarrow_G 3$ this yields a *linear* system of equations in the unknowns $a_1, \ldots, a_6$ (by comparison of coefficients at the power products $1, y_1, \ldots, y_1 y_2^2$), 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 $d = x^6 - 6x^4 + 18x^3 + 12x^2 + 108x + 73$.

6.6. APPLICATION: SOLVABILITY AND EXACT SOLUTION OF SYSTEMS OF ALGEBRAIC EQUATIONS

In this section, it is shown how the algorithm for constructing Gröbner bases may be used for the exact solution of systems of algebraic equations and questions about the solvability of such systems. The significance of Gröbner bases for problems in this category stems from the fact that, for Gröbner bases, the explicit construction of all the elimination ideals is extremely simple. This is particularly true for Gröbner bases with respect to the purely lexicographical ordering of power products. It is not so easy for Gröbner bases with respect to other orderings, for example, the total degree ordering. Still, it is also reasonable to construct Gröbner bases with respect to the total degree ordering for solving algebraic systems because, in extensive computational experiments, it turned out recently [6.46] that the complexity of the algorithm for constructing Gröbner bases is extremely sensitive to a permutation of variables when the purely lexicographical ordering is used, whereas it is nearly stable, when the total degree ordering is used. Furthermore, the complexity with respect to the total degree ordering is approximately in the same range as the complexity with respect to the purely lexicographical ordering, when the most favorable permutation of variables is used. Since, for a given example, there is no a priori method to predict which permutation of the variables will give the best computation times, it, therefore, is also a good method to compute the Gröbner basis with respect to the total degree ordering and then accept the disadvantage that the computation of the elimination ideals is not so easy as in the case of the purely lexicographical ordering. In the sequel, we present the method with respect to both orderings of power products.

LEMMA 6.8 [6.15]. *Let F be a Gröbner basis with respect to the purely lexicographical ordering of power products. Without loss of generality let us assume* $x_1 <_T x_2 <_T \ldots <_T x_n$. *Then*

$$Ideal(F) \cap K[x_1, \ldots, x_i] = Ideal(F \cap K[x_1, \ldots, x_i])$$

(for $i = 1, \ldots, n$*), where the ideal on the right-hand side is formed in* $K[x_1, \ldots, x_i]$. •

This lemma shows that the '*i-th elimination ideal*' of F is generated by just those polynomials in F that depend only on the variables $x_1, \ldots, x_i$.

Proof. If $f \in \text{Ideal}(F) \cap K[x_1, \ldots, x_i]$, then f can be reduced to 0 modulo F (use Theorem 6.1). With respect to the purely lexicographical ordering determined by $x_1 <_T x_2 <_T \ldots <_T x_n$, this means that f can be reduced to zero by subtraction of appropriate multiples $b_j \cdot u_j \cdot f_j$ ($f_j \in F$) such that $LP(f_j)$ contains only indeterminates from the set $\{x_1, \ldots, x_i\}$ and, hence, all power products occurring in f_j contain only indeterminates in this set. Also u_j can contain only indeterminates in this set. Adding all these $b_j \cdot u_j \cdot f_j$, one gets a representation of f of the form

$$f = \sum a_j \cdot u_j \cdot f_j$$

which shows that f is in $\text{Ideal}(F \cap K[x_1, \ldots, x_i])$. •

PROBLEM 6.10.
Given F.

Decide, whether F has a solution (i.e. whether there exist $a_1, \ldots, a_n$ in an algebraic extension of K such that for all f in F: $f(a_1, \ldots, a_n) = 0$.)

METHOD 6.8 [6.7], [6.8].
Compute $G := GB(F)$.
Then: F *is unsolvable iff* $1 \in G$.

Proof. It is well known that F has a solution iff $1 \notin \text{Ideal}(F)$, see, for example, [6.47]. Now, $\text{Ideal}(F) = \text{Ideal}(G)$ and $1 \in \text{Ideal}(G)$ iff 1 is reducible w.r.t. G (by Theorem 6.1). The latter is true iff $1 \in G$.

PROBLEM 6.11.

Given F.

Decide, whether F has finitely oder infinitely many solutions.

METHOD 6.9 [6.7], [6.8].

Compute $G := GB(F)$.

Then: F *has finitely many solutions iff for all* i $(1 \leq i \leq n)$: *a power product of the form* x_i^j *occurs among the leading power products of the polynomials in* G.

Proof. It is well known that F has finitely many solutions iff the vector space $K[x_1, \ldots, x_n]/\mathrm{Ideal}(F)$ has finite vector space dimension, see, for example, [6.47]. Because of Lemma 6.7 this is true iff the set B considered in Lemma 6.7 is finite. It is easy to see from the definition of B that B is finite iff the condition stated in Method 6.9 is satisfied.

About the exact dimension of polynomial ideals, one can say more than is expressed above by using Gröbner bases for computing the Hilbert function of polynomial ideals. Many details are given in [6.33], [6.34].

PROBLEM 6.12.

Given F (solvable, with finitely many solutions).

Find all the solutions of the system F.

METHOD 6.10 [6.15].

Compute $G := GB(F)$ with respect to the purely lexicographical ordering of power products.

The polynomials in G, then, have there variables "separated" in the precise sense of Lemma 6.8 (G is 'triangularized'). G contains exactly one polynomial of $K[x_1]$ (actually, it is the polynomial in $\mathrm{Ideal}(G) \cap K[x_1]$ with smallest degree).

The successive elimination can, then, be carried out by the following process:

```
$p := $ the polynomial in $G \cap K[x_1]$
$X_1 := \{(a) | p(a) = 0\}$
for $i := 1$ to $n - 1$ do
    $X_{i+1} := \emptyset$
```

```
for all $(a_1, \ldots, a_i) \in X_i$ do
    $H: = \{g(a_1, \ldots, a_i, x_{i+1}) \mid$
        $g \in G \cap K[x_1, \ldots, x_{i+1}] - K[x_1, \ldots, x_i]\}$
    $p: =$ greatest common divisor of the polynomials in $H$
    (Actually, $\{p\} = GB(H)$; in the case of univariate
    polynomials the algorithm $GB$ specializes to Euclid's
    algorithm!)
    $X_{i+1}: = X_{i+1} \cup \{a_1, \ldots, a_i, a) \mid p(a) = 0\}$.
```

Upon termination, X_n will contain all the solutions. (Note that some of the p may be 1, i.e. the corresponding partial solution $(a_1, \ldots, a_i)$ can not be continued.) •

Of course, for the univariate polynomials p occurring in the algorithm, the 'exact' determination of all their zeros may not be possible effectively. However, of course, this is not a deficiency of the particular method but an intrinsic limitation of algorithmic solvability of polynomial equations. Still, Method 6.10 is an algorithmic method (using only arithmetic in K) for completely reducing the multivariate problem to the univariate one.

Before we can give a method for Problem 6.11 that is based on Gröbner bases with respect to arbitrary orderings of power products we must solve the following problem.

PROBLEM 6.13.

Given a Gröbner basis G, such that G, as a system of equations, has only finitely many solutions.

Find the $p \in \mathrm{Ideal}(G) \cap K[x_1]$ with minimal degree.

METHOD 6.11 [6.8].

(In case the purely lexicographical ordering with $x_1 <_T x_2 <_T \ldots <_T x_n$ is used, the solution of the problem is easy, see Method 6.10. In the other cases proceed by the following method.)

Determine $d_0, \ldots, d_1$ by the following process, which involves the solution of systems of linear equations in every step:

```
$i: = 0$
repeat $p_i: = S(G, x_1^i)$
       $i: = i + 1$
```

```
until there exists $(d_0, \ldots, d_{i-1}) \neq (0, \ldots, 0)$ such that $d_0 \cdot p_0 + \ldots + d_{i-1} \cdot p_{i-1} = 0$
$l := i - 1$
```

Then, $p = d_0 \cdot 1 + d_1 \cdot x_1 + \ldots + d_l \cdot x_1^l$.

METHOD 6.12 [6.8] for solving Problem 6.12.
Compute $G := GB(F)$.

The successive elimination can, then, be carried out by the following process:

```
$p :=$ the polynomial in Ideal$(G) \cap K[x_1]$ of minimal degree
      (see Method 6.11)
$X_1 := \{(a) | p(a) = 0\}$
for $i := 1$ to $n - 1$ do
    $X_{i+1} := \emptyset$
    for all $(a_1, \ldots, a_i) \in X_i$ do
        $H := \{g(a_1, \ldots, a_i, x_{i+1}, \ldots, x_n) | g \in G\}$
        $H := GB(H)$
        $p :=$ the polynomial in Ideal$(H) \cap K[x_{i+1}]$ of
              minimal degree
        $X_{i+1} := X_{i+1} \cup \{(a_1, \ldots a_i, a) | p(a) = 0\}$
```

Upon termination, X_n will contain all the solutions. (Note, again, that some of the p may be one, i.e. the corresponding partial solution $(a_1, \ldots, a_i)$ can not be continued. Also, of course, one will store the Gröbner basis H corresponding to a particular partial solution $(a_1, \ldots, a_i)$ and use it instead of G for construction of H corresponding to $(a_1, \ldots, a_i, a)$.)

EXAMPLE 6.11. The system F of Example 6.1 is solvable, because $G = GB(F)$ does not contain the polynomial 1 (see Example 6.5).

The system

$$F := \{x^2y - x^2, x^3 - x^2 + y, xy^2 - xy + 2\}$$

is unsolvable. Let us use the total degree ordering in this example.

$$SPolynomial(x^2y - x^2, x^3 - x^2 + y) = x^2y - x^3 - y^2 \rightarrow_F$$
$$\rightarrow_F - x^3 - y^2 + x^2 \rightarrow_F - y^2 + y.$$

Thus, we have to adjoin $y^2 - y$ to the basis.

$$SPolynomial(xy^2 - xy + 2, y^2 - y) = 2,$$

which can not be reduced further. Hence, we have to adjoin 1 to the basis. This is the signal that F is unsolvable.

EXAMPLE 6.12. F of Example 6.1 has only finitely many solutions, because x^3 and y appear as leading power products in $GB(F)$.

$$F := \{x^2y - y^2 - x^2 + y, x^2 - y\}$$

has infinitely many solutions. Actually, F is already a Gröbner basis (with respect to the total degree ordering of power products): check by applying Algorithm 6.3 which, in this case, does not adjoin any new polynomial to F. No power products of the form y^i occurs among the leading power products. Hence, F has infinitely many solutions.

EXAMPLE 6.13. For F of Example 6.1,

$$GB(F) = \{x^3 - 5/2x^2 - 5/2x, y + x^2 - 3/2x - 3\}.$$

The solutions a of the first (univariate!) polynomial are 0, $(5 + \sqrt{65})/4$, $(5 - \sqrt{65})/4$. Each of these solutions can be continued to a solution (a, b) of F by solving the second polynomial in the form $y + a^2 - 3/2a - 3$ for y. This yields $(0, 3)$, $((5 + \sqrt{65})/4, -(3 + \sqrt{65})/4)$, $((5 - \sqrt{65})/4, (-3 + \sqrt{65})/4)$ as the three solutions of the system.

EXAMPLE 6.14. The same example can also be treated by Method 6.12. With respect to the total degree ordering, $G := GB(F) = \{g_1, g_2, g_3\}$ where

$$g_1 := x^2 + y - 3/2x - 3,$$
$$g_2 := xy - y + x + 3,$$
$$g_3 := y^2 - 5/2y - 4x - 3/2.$$

We now compute the normal forms of $1, x, x^2, \ldots$:

$$S(G, 1) = 1,$$

$d_0 \cdot 1 = 0$ has no non-trivial solution.

$$S(G, x) = x,$$

$d_0 \cdot 1 + d_1 \cdot x = 0$ has no non-trivial solution.

$S(G, x^2) = - y + 3/2x + 3$,

$d_0 \cdot 1 + d_1 \cdot x + d_2 \cdot x^2 = 0$ has no non-trivial solution.

$S(G, x^3) = - 5/2y + 25/4x + 15/2$,

$d_0 \cdot 1 + d_1 \cdot x + d_2 \cdot x^2 + d_3 \cdot x^3 = 0$ leads to the following linear system of equations:

$$\begin{aligned} - 5/2d_3 - d_2 \qquad\qquad &= 0, \\ 25/4d_3 + 3/2d_2 + d_1 \qquad &= 0, \\ 15/2d_3 + 3d_2 \qquad + d_0 &= 0, \end{aligned}$$

which has (after normalization $d_3 = 1$) the unique solution $d_3 = 1$, $d_2 = -5/2$, $d_1 = -5/2$, $d_0 = 0$. This means that

$$p: = x^3 - 5/2x^2 - 5/2x$$

is the polynomial in Ideal(G) $\cap$ $K[x]$ with minimal degree (in accordance to what we already have seen in Example 6.13). p has the three solutions $a_1 = 0$, $a_2 = (5 + \sqrt{65})/4$, $a_3 = (5 - \sqrt{65})/4$. Substitution of a_1 yields

$$\begin{aligned} g_1(a_1) &= y - 3, \\ g_2(a_1) &= -y + 3, \\ g_3(a_1) &= y^2 - 5/2y - 3/2. \end{aligned}$$

The Gröbner basis corresponding to these three polynomials is

$$G': = \{y - 3\}.$$

By computing the normal forms $1, y, y^2, \ldots$ and looking at the corresponding systems of linear equations as above one detects that

$$p': = y - 3$$

is the polynomial in Ideal(G') $\cap$ $K[y]$ of minimal degree. Of course, in this particularly simple example, this can be seen immediately from the Gröbner basis. Hence, (a_1, b_1) with $b_1: = 3$ is the first solution of the system. Similarly, substitution of a_2 yields

$$\begin{aligned} g_1(a_2) &= y + (3 + \sqrt{65})/4, \\ g_2(a_2) &= (1 + \sqrt{65})/4y + (17 + \sqrt{65})/4, \\ g_3(a_2) &= y^2 - 5/2y - (13 + \sqrt{65})/2. \end{aligned}$$

The Gröbner basis corresponding to these three polynomials is

$$G'' := \{y + (3 + \sqrt{65})/4\} \quad \text{and} \quad p'' := y + (3 + \sqrt{65})/4$$

is the polynomial in $\mathrm{Ideal}(G'') \cap K[y]$ of minimal degree. Hence, (a_2, b_2) with $b_2 := -(3 + \sqrt{65})/4$ is the second solution of the system.

Finally, substitution of a_3 yields, again, three polynomials in $K[y]$ whose Gröbner basis consists of the polynomial $y + (3 - \sqrt{65})/4$. Hence, the third solution is (a_3, b_3) with $b_3 := (-3 + \sqrt{65})/4$.

EXAMPLE 6.15. Given F consisting of

$$\begin{aligned}
&4x^2 + xy^2 - z + 1/4,\\
&2x + y^2z + 1/2,\\
&x^2z - 1/2x - y^2,
\end{aligned}$$

the corresponding Gröbner basis G (with respect to the purely lexicographical ordering, where $z <_T y <_T x$) consists of

$$\begin{aligned}
&z^7 - 1/2z^6 + 1/16z^5 + 13/4z^4 + 75/16z^3 + 171/8z^2 +\\
&\qquad + 133/8z - 15/4,\\
&y^2 - 19188/497z^6 + 318/497z^5 - 4197/1988z^4 -\\
&\qquad - 251555/1988z^3 - 481837/1988z^2 +\\
&\qquad + 1407741/1988z - 297833/994,\\
&x + 4638/497z^6 - 75/497z^5 + 2111/3976z^4 +\\
&\qquad + 61031/1988z^3 + 232833/3976z^2 - 85042/497z +\\
&\qquad + 144407/1988.
\end{aligned}$$

Applying Method 6.10 for solving G, one first had to find all the solutions of the first polynomial, which is univariate. Each of these solution a_1, can be continued to two solutions (a_1, a_2) of the second polynomial and each of these (a_1, a_2) can be continued to a solution (a_1, a_2, a_3) of the third polynomial. The solutions of the first polynomial can be determined systematically with any guaranteed precision, see [6.48]. It has not yet been studied systematically how, numerically, the precision of the solutions of the first equation must be fixed in order to guarantee a given precision for all the solutions of the last equation. This is a near-at-hand important problem for future study.

EXAMPLE 6.16. Sometimes, it is necessary to solve systems of algebraic equations with 'symbolic' coefficients. For example consider F consisting of

$$\begin{aligned} f_1 &:= x_4 + (b - d), \\ f_2 &:= x_4 + x_3 + x_2 + x_1 + (-a - c - d), \\ f_3 &:= x_3x_4 + x_1x_4 + x_2x_3 + (-ad - ac - cd), \\ f_4 &:= x_1x_3x_4 + (-acd), \end{aligned}$$

where $x_1 <_T x_2 <_T x_3 <_T x_4$ are the polynomial indeterminates and a, b, c, d are 'symbolic' coefficients. One might like to solve this system for x_1, x_2, x_3, x_4. This is nothing else then saying that one conceives the polynomials as elements in $Q(a, b, c, d)[x_1, \ldots, x_4]$, where $Q(a, b, c, d)$ is the field of rational functions over Q. Our algorithm works over arbitrary fields and, hence, in particular also over $Q(a, b, c, d)$. Some steps of Algorithm 6.3 are:

Reduction of f_1 modulo f_2 (by subtraction of f_2 from f_1 and normalizing the coefficient of the leading power product to 1) yields

$$f'_1 := x_3 + x_2 + x_1 + (-a - b - c) \; (f_1 \text{ may be canceled}).$$

Reduction of f_2 modulo f'_1 yields

$$f'_2 := x_4 + (b - d) \; (f_2 \text{ may be canceled}).$$

Reduction of f_3 modulo the other polynomials (starting with the subtraction of $x_3 \cdot f'_2$ and, then executing several other reduction steps) yields

$$\begin{aligned} f'_3 := {} & x_2^2 + 2x_1x_2 - (a + 2b + c - d)\, x_2 + x_1^2 \\ & - (a + b + c)\, x_1 + (ab + ac + b^2 + bc - bd) \\ & (f_3 \text{ may be canceled}). \end{aligned}$$

Reduction of f_4 yields

$$f'_4 := x_1x_2 + x_1^2 - (a + b + c)\, x_1 - acd/(b - d) \; (\text{cancel } f_4).$$

(Note here that division in $Q(a, b, c, d)$ has to be performed. f'_3 can now be further reduced (using f'_4) yielding f''_3

$$\begin{aligned} f''_3 := {} & x_2^2 - (a + 2b + c - d)\, x_2 - x_1^2 + (a + b + c)\, x_1 + \\ & + (ab^2 + abc - abd + acd + b^3 + b^2c - \\ & - 2b^2d - bcd + bd^2)/(b - d). \end{aligned}$$

Cancel f'_3. No further reduction is possible. Therefore, we consider

$$\text{SPolynomial}(f''_3, f'_4) = x_1 \cdot f''_3 - x_2 \cdot f'_4.$$

Reduction of this polynomial yields

$$f_5: = x_2 + (b^2 - 2bd + d^2)/(acd)\ x_1^2 + \\ + (abc + abd - ad^2 + bcd - cd^2)/(acd)\ x_1 + (-b + d).$$

Now, again, a number of reductions are possible yielding, finally,

$$g_1: = x_3 + (-b^2 + 2bd - d^2)/(acd)\ x_1^2 + \\ + (-abc - abd + 2acd + ad^2 - bcd + cd^2)/(acd)\ x_1 + \\ + (-a - c - d).$$

$abc + abd$
$- 2acd + bcd$
$- ad^2 - cd^2$
$/(b^2 - 2bd + d^2)$

$$g_2: = x_4 + (b - d).$$

$$g_3: = x_1^3 + \cancel{(ac + ad + cd)/(b - d)}\ x_1^2 + \\ + \cancel{(a^2cd + ac^2d + acd^2)}/(b^2 - 2bd + d^2)\ x_1 + \\ + (a^2c^2d^2)/(b^3 - 3b^2d + 3bd^2 + d^3).$$

$$g_4: = x_2 + (b^2 - 2bd + d^2)/(acd)\ x_1^2 + \\ + (abc + abd - ad^2 + bcd - cd^2)/(acd)\ x_1 + (-b + d).$$

$acd-$

By Criterion1, the reduction of the S-polynomials of these polynomial may be skipped. Hence, $G: = \{g_1, \ldots, g_4\}$ is the reduced Gröbner basis. By Methods 6.8 and 6.9 it can be seen that the system has finitely many solutions. The system must contain a univariate polynomial in $Q(a, b, c, d)[x_1]$: g_3. A particular solution of g_3 is

$$a_1: = (-ad)/(b - d),$$

which can be extended to a solution (a_1, a_2, a_3, a_4) of the entire system, where

$$a_2: = (ab + b^2 - bd)/(b - d),$$

$$a_3: = c,$$

$$a_4: = -b + d.$$

Dividing g_3 by $(x_1 - a_1)$ one gets a quadratic polynomial whose solutions can be extended to solutions of the entire system in the same way as before.

6.7. APPLICATION: SOLUTION OF LINEAR HOMOGENEOUS EQUATIONS WITH POLYNOMIAL COEFFICIENTS

In this section, it is shown how the algorithm for constructing Gröbner bases may be used for determining a finite set of generators for all the polynomial solutions of a linear homogeneous equation with polynomial coefficients. Before the method can be described, it must be shown how one can find a linear representation of the polynomials in a basis F in terms of the polynomials in its corresponding Grobner basis G and vice versa.

PROBLEM 6.14.
Given a Gröbner basis $G = \{g_1, \ldots, g_m\}$ and some f.

Find $h_1, \ldots, h_m$ such that $f = h_1 \cdot g_1 + \ldots + h_m \cdot g_m$ (and $LP(h_i \cdot g_i) \leq_T LP(f)$ for $i = 1, \ldots, m$).

METHOD 6.13.
Roughly, reduce f to zero modulo G and collect the multiples of the g_i necessary in the reduction. In more detail: take Algorithm 6.1 (the normal form algorithm) and insert instructions that collect the multiples of the g_i used in the reduction.

```
$h_1 := \ldots := h_m := 0$
while $f \neq 0$ do
        choose $i, b, u$ such that $f \rightarrow_{g_i, b, u}$ and $u \cdot LP(g_i)$
        is maximal w.r.t. $<_T$
        $f := f - b \cdot u \cdot g_i$
        $h_i := h_i + b \cdot u$
```

PROBLEM 6.15.
Given $F = \{f_1, \ldots, f_l\}$ and $G = \{g_1, \ldots, g_m\}$ such that $G = GB(F)$.

Find Y such that Y is a matrix of polynomials with m rows and l columns and

$$f_j = \sum_{1 \leq i \leq m} g_i \cdot Y_{i,j} \qquad (\text{for } j = 1, \ldots, l).$$

METHOD 6.14.
The j-th column of Y consists of $h_1, \ldots, h_m$ that are obtained by the Method 6.13 for the representation of f_j ($j = 1, \ldots, l$).

PROBLEM 6.16.
Given: $F = \{f_1, \ldots, f_l\}$.

Find $G = \{g_1, \ldots, g_m\}$ and X such that $G = GB(F)$, X is a matrix of polynomials with l rows and m columns and

$$g_i = \sum_{1 \leq j \leq l} f_j \cdot X_{j,i} \qquad (\text{for } i = 1, \ldots, m).$$

METHOD 6.15.
Augment Algorithm 6.2 or Algorithm 6.3 by instructions that keep track of the multiples of f_j that are used in the reduction of those polynomials whose normal form is adjoined to the basis G (compare Method 6.13).

PROBLEM 6.17.
Given a reduced Gröbner basis $G = \{g_1, \ldots, g_m\}$.

Find a matrix R with m columns such that the finitely many rows of R constitute a set of generators for the linear homogeneous equation

$$h_1 \cdot g_1 + \ldots + h_m \cdot g_m = 0 \; (h_1, \ldots, h_m \in K[x_1, \ldots, x_n]),$$

i.e. R should consist of m-tupels $(k_{1,1}, \ldots, k_{1,m}), \ldots, (k_{r,1}, \ldots, k_{r,m})$ of polynomials such that

$$k_{j,1} \cdot g_1 + \ldots + k_{j,m} \cdot g_m = 0 \; (\text{for } j = 1, \ldots r)$$

and for all $(h_1, \ldots, h_m)$ for which

$$h_1 \cdot g_1 + \ldots + h_m \cdot g_m = 0$$

there exist polynomials $p_1, \ldots, p_r$ such that

$$(h_1, \ldots, h_m) =$$
$$= p_1 \cdot (k_{1,1}, \ldots, k_{1,m}) + \ldots + p_r \cdot (k_{r,1}, \ldots, k_{r,m}).$$

METHOD 6.16 [6.14], [6.18], [6.21], [6.28], [6.33], [6.34].
R: = empty matrix

for all pairs (i, j) $(1 \leqslant i < j \leqslant 1)$:

Consider h: = $SPolynomial(g_i, g_j) = u_i \cdot g_i - (c_i/c_j) \cdot u_j \cdot g_j$, where c_i is the leading coefficients of g_i, u_i is such that $s_i \cdot u_i$ is the $LCM(s_1, s_2)$, s_i is the leading power product of g_i $(i = 1, 2)$.

Reduce h to zero modulo G and store the multiples of the $g_1, \ldots, g_l$

necessary for this reduction. This gives a representation of h of the form

$$h = k_1 \cdot g_1 + \ldots + k_l \cdot g_l \text{ (compare Method 6.13!).}$$

$$\text{Add } (\ldots, \underset{\substack{\uparrow \\ \text{position } i}}{u_i}, \ldots, \underset{\substack{\uparrow \\ \text{position } j}}{-(c_i/c_j) \cdot u_j}, \ldots) - (k_1, \ldots, k_l) \text{ as last row in } R$$

PROBLEM 6.18.
Given $F = \{f_1, \ldots, f_l\}$ arbitrary.

Find a matrix Q with l columns such that the finitely many rows of Q constitute a set of generators for the linear homogeneous equation

$$h_1 \cdot f_1 + \ldots + h_l \cdot f_l = 0 \; (h_1, \ldots, h_l \in K[x_1, \ldots, x_n]).$$

METHOD 6.17 [6.18].
By Method 6.15, compute $G = GB(F) = \{g_1, \ldots, g_m\}$ and a matrix X with l rows and m columns such that

$$g_i = \sum_{1 \leq j \leq l} f_j \cdot X_{j,i} \qquad (\text{for } i = 1, \ldots, m).$$

By Method 6.14, compute a matrix Y with m rows and l columns such that

$$f_j = \sum_{1 \leq i \leq m} g_i \cdot Y_{i,j} \qquad (\text{for } j = 1, \ldots, l).$$

By Method 6.16 compute a matrix R with m columns such that the r rows of R constitute a set of generators for the linear homogeneous equations

$$h_1 \cdot g_1 + \ldots + h_m \cdot g_m = 0.$$

Then,

$$Q = \begin{pmatrix} I - Y^t \cdot X^t \\ \cdots\cdots\cdots\cdots \\ R \cdot X^t \end{pmatrix} \text{ (a block matrix)}$$

(I is the unit matrix with l rows and columns, X^t is the transposed of X).

EXAMPLE 6.17.
Let $F := \{f_1, f_2, f_3\}$, where

$$f_1 := x^2y - xy, \qquad f_2 := xy^2 - x^2,$$
$$f_3 := x^4y - x^2y + x^3 - x^2.$$

We use the total degree ordering. First, $G := GB(F)$ has to be computed with simultaneous determination of the matrix X. We start with a reduction of f_3:

$$f_3 - x \cdot f_1 = x^3 - x^2 =: f'_3.$$

The representation

$$f'_3 = (-x) \cdot f_1 + 0 \cdot f_2 + 1 \cdot f_3$$

must be stored. Then we reduce the S-polynomial of f_1 and f_2:

$$h := \text{SPolynomial}(f_1, f_2) = y \cdot f_1 - x \cdot f_2,$$
$$h + f_2 - f'_3 = 0 =: f_4.$$

If f_4 was not zero, the following representation of f_4 in terms of f_1, f_2 and f_3 could be obtained from this reduction:

$$f_4 = y \cdot f_1 - x \cdot f_2 + f_2 - f_3 + x \cdot f_1 =$$
$$= (y + x) \cdot f_1 + (-x + 1) \cdot f_2 + (-1) \cdot f_3.$$

This example of a reduction should suffice to demonstrate how the linear representations of the new polynomials in G in terms of the polynomials in F can be obtained in general. Since, however, f_4 is zero, nothing has to be adjoined to G in this stage of the algorithm. The S-polynomial of f_1 and f'_3 and also the S-polynomial of f_2 and f'_3 reduce to zero. Hence,

$$G := \{g_1, g_2, g_3\},$$

where

$$g_1 := f_1, g_2 := f_2, g_3 := x^3 - x^2,$$

is the reduced Gröbner basis corresponding to F and

$$X := \begin{pmatrix} 1 & 0 & -x \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

is the transformation matrix.

The matrix Y for the reverse transformation (i.e. the linear representation of the elements of F in terms of the elements in G) is obtained by Method 6.14:

f_1 reduces to zero modulo G by subtraction of g_1.
f_2 reduces to zero modulo G by subtraction of g_2.
f_3 reduces to zero modulo G by subtraction of $x \cdot g_1$ and g_3.

Hence,

$$Y := \begin{pmatrix} 1 & 0 & x \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

For getting R, we have to reduce the S-polynomials of the pairs (g_i, g_j):

$$h_{1,2} := \text{SPolynomial}(g_1, g_2) = y \cdot g_1 - x \cdot g_2.$$

$$h_{1,2} + g_2 - g_3 = 0.$$

$$h_{1,3} := \text{SPolynomial}(g_1, g_3) = x \cdot g_1 - y \cdot g_3.$$

$$h_{1,3} = 0.$$

$$h_{2,3} := \text{SPolynomial}(g_2, g_3) = x^2 \cdot g_2 - y^2 \cdot g_3.$$

$$h_{2,3} - y \cdot g_1 + x \cdot g_3 - g_2 + g_3 = 0.$$

From the first reduction:

$$y \cdot g_1 - x \cdot g_2 + g_2 - g_3 = \\ (y) \cdot g_1 + (-x + 1) \cdot g_2 + (-1) \cdot g_3 = 0.$$

Hence, the first row in R is the solution (the 'syzygy')

$$(y, -x + 1, -1).$$

The other rows of R are obtained analogously:

$$R = \begin{pmatrix} (y) & (-x + 1) & (-1) \\ (x) & (0) & (-y) \\ (-y) & (x^2 - 1) & (-y^2 + x + 1) \end{pmatrix}.$$

Finally, the computation of Q requires only some matrix multiplications: First, we note that $Y^t \cdot X^t = I$ in this particular example. Hence,

$$Q = \begin{pmatrix} I - Y^t \cdot X^t \\ \cdots\cdots\cdots\cdots \\ R \cdot X^t \end{pmatrix} = \\ \begin{pmatrix} (0) & (0) & (0) \\ (0) & (0) & (0) \\ (0) & (0) & (0) \\ (y + x) & (-x + 1) & (-1) \\ (x + xy) & (0) & (-y) \\ (xy^2 - x^2 - y - x) & (x^2 - 1) & (-y^2 + x + 1) \end{pmatrix}$$

Of course, the first three rows can be canceled in this particular example, the last three rows constitute a complete set of generators for the solutions (h_1, h_2, h_3) to the equation $h_1 \cdot f_1 + h_2 \cdot f_2 + h_3 \cdot f_3 = 0$. •

For $K[x_1, \ldots, x_n]$-modules, as for example the module of all the solutions to the above linear equation, a notion of 'Gröbner bases' and 'reduced Gröbner bases' can be introduced, see [6.28], [6.33], [6.34]. Then the matrices Q can be reduced to a minimal set of generators and the construction can be carried over to obtain the whole 'chain of syzygies' or the 'free resolution' of a polynomial ideal.

6.8. GRÖBNER BASES FOR POLYNOMIAL IDEALS OVER THE INTEGERS

The concept of Gröbner bases, the essential properties of Gröbner bases and the algorithm for constructing Gröbner bases as reflected by Definitions 6.2, 6.3, 6.5, 6.6, Lemmata 6.1, 6.2, 6.3, Theorems 6.1, 6.2, 6.3, 6.4, Algorithms 6.1, 6.2, 6.3 and most of the applications in Sections 6.5 and 6.7 can be carried over to polynomial ideals in $Z[x_1, \ldots, x_n]$ and, in fact, to ideals in certain other rings, see [6.30]. However, a subtle analysis of the notion of reduction and, more essentially, of the notion of 'S-polynomial' must be carried out for this purpose. We can not go into the details of the theoretical foundations of the algorithm for integer polynomials. Rather, we explain the steps of the generalized algorithm in the style of the preceding sections.

The problem of deciding ideal membership for ideals in $Z[x_1, \ldots, x_n]$, the simplification problem for these ideals and related problems have a long and interesting history. For some of the details of the history, see [6.49]. The first general solution of both the simplification and (hence,) the membership problem, was given by Lauer [6.11] based on the Gröbner bases approach but needing two different types of 'S-polynomials'. Other solutions based on the Gröbner bases approach, but destroying the simple structure of the algorithm, were given in [6.15], [6.18], [6.21]. The first general solution based on a different approach was given only in [6.49]. Our own solution [6.30], which will be presented here, seems to be much more concise than the solutions given so far and leaves the simple structure of the algorithm untouched.

In addition to some ordering of the power products, in the case of $Z[x_1, \ldots, x_n]$, one also must fix some ordering of the integers, for example, $0 < -1 < 1 < -2 < 2 < -3 < 3 < \ldots$ (An axiomatic characterization of the admissible orderings is possible but will not be used in this paper). The crucial difference, then, to the case of polynomials with field coefficients is that, in the definition of 'reduction' (Definition 6.1) it is not possible to totally cancel Coefficient(g, t), where $t = u \cdot$ LeadingPowerProduct(f), because the element Coefficient(g, t)/LeadingCoefficient(f), in general, will not be in Z. In the following, the typed variables a, b, c, d will be used for integers instead of field elements, f, g, h, k, p, q will be used for polynomials in $Z[x_1, \ldots, x_n]$, and F, G for finite sets in $Z[x_1, \ldots, x_n]$.

DEFINITION 6.8 [6.30].
$g \rightarrow_F h$ (read: 'g *reduces* to h modulo F') iff there exists $f \in F$, 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

$$a \neq 0 \quad \text{and} \quad a - b \cdot c < a,$$

where,

$$a = \text{Coefficient}(g,\ u \cdot \text{LeadingPowerProduct}(f)), \text{ and}$$
$$c = \text{LeadingCoefficient}(f)$$

EXAMPLE 6.18. The b in Definition 6.8 can be determined by the following algorithm $M(a, c)$, for example:

```
M(a, c): = if a and c have the same sign
           then if $a - c < a$ then $M(a - c, c) + 1$
                                else 0
           else  if $a + c < a$ then $M(a + c, c) - 1$
                                else 0
```

In practice, M may be realized by a modified integer division. •

The definitions, theorems, algorithms and lemmata of Section 2 can now be carried over without any change: In particular, we have again the algorithm NormalForm that produces a normal form for every polynomial, we have the notion of a Gröbner basis, the characterizations (GB2) and (GB3) of Gröbner bases and the connection between reduction and ideal congruence stated in Lemma 6.3. For the formulation of the algorithm that constructs Gröbner bases, however, we need some additional preparation.

DEFINITION 6.9 [6.30].
The *least common reducible* of c_1, c_2 is defined as follows:

$$LCR(c_1, c_2) := \max(L(c_1), L(c_2)) \text{ (max taken w.r.t. } \prec\text{)},$$

where

$$L(c) := \begin{cases} abs(c)/2, & \text{if } c \text{ is even} \\ (abs(c) + 1)/2, & \text{if } c \text{ is odd.} \end{cases}$$

DEFINITION 6.10 [6.30].
p_1 and p_2 constitute the *critical pair* corresponding to f_1 and f_2 iff

$$\begin{aligned} p_i &= a \cdot U - M(a, c_i) \cdot u_i \cdot f_i, \text{ where} \\ U &= LCM(s_1, s_2), \\ a &= LCR(c_1, c_2), \\ s_i &= \text{LeadingPowerProduct}(f_i), \\ c_i &= LeadingCoefficient(f_i), \\ u_i &\text{ is such that } u_i \cdot s_i = U \qquad (i = 1, 2). \end{aligned}$$

●

The difference of the two components of a critical pair is the analogue to the S-polynomial in the case of field coefficients. We formulate the algorithm for critical pairs instead of S-polynomials, because, at present, we do not have a formal proof that, in fact, the algorithm below is correct with S-polynomials instead of critical pairs, although it is very likely. Also, we would like to introduce the concept of a critical pair to the reader, because this concept may be applied to domains without any operation of subtraction also. See [6.3] for an introduction to 'critical-pair/completion' algorithms.

EXAMPLE 6.19.
$0, -1, 1, -2, 2, -3, 3, -4, 4$ are the values of L for the arguments $0, 1, 2, 3, 4, 5, 6, 7, 8$, respectively, and $LCR(3, 1) = -2$, $LCR(7, 8) = 4$. Note that $L(c) = L(-c)$. ●

The main theorem of Section 3, which gives an algorithmic characterization of Gröbner bases, and the main algorithm for the main problem can now be carried over in the following form:

THEOREM 6.6 (Buchberger [6.30]).
Let S be an arbitrary normal form algorithm. The following properties are equivalent:

(GB1) F is a Gröbner basis.

(GB3) For all $f_1, f_2 \in F$, p_1, p_2:
if p_1 and p_2 constitute the critical pair corresponding to f_1, f_2, then $S(F, f_1) = S(F, f_2)$.

PROBLEM 6.19.
Given F.
Find G, such that $\mathrm{Ideal}(F) = \mathrm{Ideal}(G)$ and G is a Gröbner basis.

ALGORITHM 6.4 (Buchberger [6.30]) for solving Problem 6.19.

```
        $G := F$
        $B := \{\{f_1, f_2\} | f_1, f_2 \in G\}$
while $B \neq \emptyset$ do
        $\{f_1, f_2\} :=$ a pair in $B$
        $(p_1, p_2) :=$ the critical pair corresponding to $f_1, f_2$
        $(p'_1, p'_2) := (S(G, p_1), S(G, p_2))$
        $h' := p'_1 - p'_2$
        if $h' \neq 0$ then
            $B := B \cup \{\{g, h'\} | g \in G\}$
            $G := G \cup \{h'\}$.
```
●

Also the various improvements of the algorithm, the notion of reduced Gröbner bases and the theorem on the uniqueness of the reduced Gröbner bases (Section 3) can be carried over. We do not explicitly state the details.

EXAMPLE 6.20. Take F as in Example 6.1. Note that the leading coefficients of the polynomials in F can not be simply set to 1 by dividing the whole polynomial: the ideal would change! We fix the 'purely lexicographical' ordering for the bivariate power products with the ordering $x <_T y$ of the two indeterminates. In order to 'complete' F by Algorithm 6.4, one has to consider the 'critical pairs' of polynomials in F. We start with f_2, f_3: $LC(f_2) = 2$, $LC(f_3) = 1$, $LCR(2, 1) = 1$, $LCM(LP(f_2), LP(f_3)) = x^3y$. Thus, x^3y is the monomial that has to be reduced in one step modulo f_2 and f_3 in order to get the critical pair corresponding to f_2, f_3. The polynomial x^3y may be reduced by f_2 in the following way:

$$x^3y \rightarrow_{f_2} -x^3y + xy + y - 6x^3 + 2x^2 + 3x - 3 =: p.$$

p may be further reduced modulo f_3:

$$p \rightarrow_{f_3} x^2y + xy + y - 3x^3 + 4x^2 + 3x - 3 =: p'.$$

p' is irreducible with respect to F. The polynomial x^3y may also be reduced by f_3:

$$x^3y \rightarrow_{f_3} -x^2y - 3x^3 - 2x^2 =: q.$$

Also q is irreducible with respect to F. $p' \neq q$ and, hence,

$$f_4 := p' - q = 2x^2y + xy + y + 6x^2 + 3x - 3$$

must be adjoined to the basis.

Similarly, one now has to consider the next critical pair, for example, the one corresponding to f_1, f_4: $-2x^2y$ is the 'least common reducible' of f_1 and f_4, which has to be reduced in one step modulo f_1 and f_4, yielding

$$p := x^2y + 2xy + y + 9x^2 + 5x - 3 \quad \text{and}$$

$$q := xy + y + 6x^2 + 3x - 3,$$

respectively. Reduction to normal forms yields

$$p' := -x^2y + xy + 3x^2 + 2x \text{ (using } f_4\text{) and}$$

$$q' := xy + y + 6x^2 + 3x - 3.$$

Thus, the difference of these two polynomials must be adjoined to the basis:

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

Similarly, the consideration of the critical pair of f_4 and f_5 leads to

$$f_6 := -xy + y - x - 3.$$

The consideration of the critical pair of f_5 and f_6 leads to

$$f_7 := 2y + 2x^2 - 3x - 6.$$

Finally, the consideration of the critical pair of f_6 and f_7 leads to

$$f_8 := 2x^3 - 5x^2 - 5x.$$

The consideration of all the other critical pairs leads to identical normal forms. Hence, $G := \{f_1, \ldots, f_8\}$ is a Grobner basis corresponding to F. Actually, the consideration of most of these critical pairs can be avoided a prior by the improved version of the algorithm. Furthermore, some of the polynomials in the basis can also be canceled in the course of the algorithm. Reduction of all the f_i modulo $G - \{f_i\}$ leaves us with the reduced Gröbner basis $G' := \{f'_6, f_7, f_8\}$, where

$$f'_6 := -xy - y - 2x^2 + 2x + 3.$$

Note that the reduced Gröbner bases corresponding to F are different depending on whether we work in $Q[x_1, \ldots, x_n]$ or in $Z[x_1, \ldots, x_n]$.

6.9. OTHER APPLICATIONS

A number of other applications of Gröbner bases have been reported in the literature: decision, whether a given polynomial ideal is principal [6.8], Hilbert functions of polynomial ideals [6.7], [6.28], [6.33], [6.34], Lasker-Noether decomposition of polynomial ideals [6.13], free resolutions of polynomial ideals and syzygies (a generalization of the above linear equation problem with polynomial coefficients) [6.28], [6. 34], multidimensional integration [6.50] and bijective enumeration of polynomial ideals. The latter problem asks for an algorithm that enumerates bases for ideals in $R[x_1, \ldots, x_n]$ (R a ring) such that every ideal is represented exactly once in the enumeration. By Theorem 6.4, it is clear that a bijective enumeration of all ideals in $K[x_1, \ldots, x_n]$ and $Z[x_1, \ldots, x_n]$ can be achieved by bijectively enumerating all Gröbner bases in these polynomial rings, which is easily possible (see [6.37]). The applicability of Gröbner bases to other problems is investigated, for example, to the construction of Hensel codes for rational functions [6.51].

6.10. SPECIALIZATIONS, GENERALIZATIONS, IMPLEMENTATIONS, COMPLEXITY

The algorithm for constructing Gröbner bases *specializes* to Gauß' algorithm in case F consists only of linear polynomials, it specializes to Euclid's algorithm in case F consists only of univariate polynomials, it specializes to an algorithm for the word problem for finitely generated commutative semigroups in case F consists only of polynomials of the form $u - v$ (differences of power products) [6.19], [6.23]. The algorithm for $Z[x_1, \ldots, x_n]$ specializes to Euclid's algorithm in Z in case $n = 0$, [6.30].

The algorithm has been *generalized* for polynomials over various rings, in particular, over Z [6.11], [6.15], [6.18], [6.21], [6.30], and for associative algebras [6.17]. The Knuth-Bendix generalization [6.38] was already discussed in the introduction. Recently, an interesting generalization was also undertaken by G. Bauer [6.24], who gives an axiomatic definition of the concept of 'substitution' and is able to define the notion of 'critical pair' in this general context.

The algorithm has been *implemented* various times, [6.7], [6.13], [6.16], [6.21]. [6.16] is an implementation in SAC–1. R. Gebauer and H. Kredel [6.46], Univ. of Heidelberg, F.R.G., work on the implementation of the algorithm in SAC–2, which will be included in the next release of SAC–2 (announced for December 1983). SAC–2 is a large software system for symbolic computation in algebraic domains, in particular in polynomial domains. It is written in the ALDES language, whose compiler is written in FORTRAN. Thus, SAC–2 is installed easily whenever FORTRAN is available. G. E. Collins (University of Wisconsin-Madison, Departments of Computer Science) and R. Loos (Universität Karlsruhe, Institut für Informatik I) are the authors of the SAC–2 system. The implementation of our algorithm in SAC–2 by R. Gebauer and H. Kredel gives the user the choice to use various orderings of power products, to work over various coefficient domains (including the field of rational functions over Q) and to communicate in convenient input and output format with the computer.

Various analyses of the *complexity* of the algorithm have been carried out: [6.7], [6.19], [6.29], [6.6], [6.31]. Summarizing, these analyses show that the degrees of the polynomials in the reduced Gröbner bases, with probability 1, stay below $d_1 + \ldots + d_l - n + 1$, where the d_i are the degrees of the input polynomials. In exceptional cases, this bound does not hold. Many theoretical questions remain open. Typical running times in SAC–2 on an IBM 370/168: several seconds for F with 3 polynomials of degree 3 in 3 variables, 20 sec for the example in [6.15] with 6 polynomials of degree 3 in 6 variables. However, this computing time may drastically change if a different permutation of the variables and purely lexicographical ordering is used. For the worst permutation, the computation was as high as 10 000 sec, whereas in the total degree ordering the computation time for the same example was always in the range 20–30 sec independent of the permutation of variables. See Section 6 for the consequences of these obsevations.

ACKNOWLEDGEMENT

The work described in this paper is supported by the Austrian Research Fund, Project No. 4567. I am indebted to R. Gebauer, H. Kredel and F. Winkler for valuable support in the preparation of the examples.

REFERENCES

[6.1] N. K. Bose, *Applied Multidimensional System Theory*, Van Nostrand Reinhold Co., New York, 1982.

[6.2] G. Hermann, 'The Question of Finitely many Steps in Polynomial Ideal Theory (German)', *Mathematische Annalen*, vol. 95, 1926, pp. 736–788.

[6.3] B. Buchberger and R. Loos, 'Algebraic Simplification', in *Computer Algebra Symbolic and Algebraic Computation*, (B. Buchberger, G. Collins, R. Loos (eds.), Springer, Wien–New York, 1982, pp. 11–43.

[6.4] A. Blass and Yu. Gurevich, 'Equivalence Relations, Invariants, and Normal Forms', *Technical Report*, Dpt. of Math. and Dpt. of Comp. and Commun. Scie., The University of Michigan, Ann Arbor, Michigan.

[6.5] E. Cardoza, R. Lipton, and A. R. Meyer, 'Exponential Space Complete Problems for Petri Nets and Commutative Semigroups', *Conf. Record of the 8th Annual ACM Symp. on Theory of Computing*, 1976, pp. 50–54.

[6.6] E. W. Mayr and A. R. Meyer, 'The Complexity of the Word Problems for Comutative Semigroups and Polynomial Ideals', *Report LCS/TM–199*, M.I.T. Laboratory of Computer Science, 1981.

[6.7] B. Buchberger, 'An Algorithm for Finding a Basis for the Residue Class Ring of a Zero dimensional Polynomial Ideal (German)', Ph.D. Thesis, Univ. of Innsbruck (Austria), Math. Inst., 1965.

[6.8] B. Buchberger, 'An Algorithmical Criterion for the Solvability of Algebraic Systems of Equations (German)', *Aequationes Mathematicae*, Vol. 4, No. 3, 1970, pp. 374–383.

[6.9] B. Buchberger, 'A Theoretical Basis for the Reduction of Polynomials to Canonical Form', *ACM SIGSAM Bull.* Vol. 10, No. 3, 1976, pp. 19–29.

[6.10] B. Buchberger, 'Some Properties of Gröbner Bases for Polynomial Ideals', *ACM SIGSAM Bull.*, Vol. 10, No. 4, 1976, pp. 19–24.

[6.11] M. Lauer, 'Canonical Representatives for the Residue Classes of a Polynomial Ideal (German)', Diploma Thesis, University of Kaiserslautern (F.R.G.), Dept. of Mathematics, 1976.

[6.12] M. Lauer, 'Canonical Representatives for Residue Classes of a Polyomial Ideal', *Proc. of the 1976 ACM Symp. on Symbolic and Algebraic Computation*, Yorktown Heights, N.Y., August 1976, R. D. Jenks (ed.), pp. 339–345.

[6.13] R. Schrader, 'Contributions to Constructive Ideal Theory (German)', Diploma Thesis, Univ. of Karlsruhe (FRG), Math. Inst., 1976.

[6.14] D. Spear, 'A constructive Approach to Commutative Ring Theory', *Proc. of the MACSYMA Users' Conf.*, Berkeley, July 1977, R. J. Fateman (ed.), published by M.I.T., pp. 369–376.

[6.15] W. Trinks, 'On B. Buchberger's Method for Solving Systems of Algebraic Equations', *J. Number Theory*, Vol. 10, No. 4, 1978, pp. 475–488.

[6.16] F. Winkler, 'Implementation of an Algorithm for Constructing Gröbner Bases (German)', Diploma Thesis, Univ. of Linz (Austria), Dept. of Math., 1978.

[6.17] G. M. Bergman, 'The Diamond Lemma for Ring Theory', *Advances in Math.*, Vol. 29, 1978, pp. 178–218.

[6.18] G. Zacharias, 'Generalized Gröbner Bases in Commutative Polynomial Rings', Bachelor Thesis, M.I.T., Dept. Comp. Scie., 1978.

[6.19] B. Buchberger, 'A Criterion for Detecting Unnecessary Reductions in the Construction of Gröbner Bases', Proc. EUROSAM 79, Marseille, June 1979, W. Ng, (ed.), *Lecture Notes in Computer Science*, Vol. 72, 1979, pp. 3–21.

[6.20] B. Buchberger and F. Winkler, 'Miscellaneous Results on the Construction of Grobner Bases for Polynomial Ideals I', Techn. Rep. No. 137, University of Linz, Math. Inst., 1979.

[6.21] S. Schaller, 'Algorithmic Aspects of Polynomial Residue Class Rings', Ph.D. Thesis, Techn. Rep. No. 370, Univ. of Wisconsin-Madison, Comp. Scie. Dept., 1979.

[6.22] L. Bachmair and B. Buchberger, 'A Simplified Proof of the Characterization Theorem for Grobner Bases', *ACM SIGSAM Bull.*, Vol. 14, No. 4, 1980, pp. 29–34.

[6.23] A. M. Ballantyne and D. S. Lankford, 'New Decision Algorithms for Finitely Presented Commutative Semigroups', *Computers and Maths. with Appls.*, Vol. 7, 1981, pp. 159–165.

[6.24] G. Bauer, 'The Representation of Monoids by Confluent Rule Systems', Ph.D. Thesis, University of Kaiserslautern (F.R.G.), Dept. of Comp. Scie., 1981.

[6.25] F. Mora, 'An Algorithm to Compute the Equations of Tangent Cones', *Proc. EUROCAM 82*, Marseille, April 1982, J. Calmet (ed.), Lecture Notes in Comp. Scie., Vol. 144, pp. 158–165.

[6.26] M. Pohst and D. Y. Y. Yun, 'On Solving Systems of Algebraic Equations via Ideal Bases and Elimination Theory', *Proc. of the 1981 ACM Symposium on Symbolic and Algebraic Computation*, Snowbird (Utah), August 1981, P. S. Wang (ed.), published by ACM, pp. 206–211.

[6.27] J. P. Guiver, 'Contributions to Two-dimensional Systems Theory', Ph.D. Thesis, Univ. of Pittsburgh, Math. Dept., 1982.

[6.28] D. Bayer, 'The Division Algorithm and the Hilbert Scheme', Ph.D. Thesis, Harvard University, Cambridge, Mass., Math. Dept., 1982.

[6.29] B. Buchberger, 'A note on the Complexity of Constructing Gröbner bases', *Proc. of the EUROCAL 83*, London, March 1983, H. van Hulzen (ed.), Lecture Notes in Computer Science 162, Springer, 1983, pp. 137–145.

[6.30] B. Buchberger, 'A Critical-pair/completion Algorithm for Finitely Generated Ideals in Rings', *Proc. of the Conf. "Rekursive Kombinatorik"*, Münster, May, 1983, E. Börger, G. Hasenjäger, and D. Rödding (eds.), Lecture Notes in Computer Science 171, Springer, 1983, pp. 137–161.

[6.31] D. Lazard, 'Gröbner Bases, Gaussian Elimination, and Resolution of Systems of Algebraic Equations', *Proc. of the EUROCAL 83*, London, March 1983, H. van Hulzen (ed.), Lecture Notes in Computer Science 162, Springer, 1983, pp. 146–156.

[6.32] R. Llopis de Trias, 'Canonical Forms for Residue Classes of Polynomial Ideals and Term Rewriting Systems', Univ. Aut. de Madrid, Division de Matematicas, submitted to publication, also: Rep. 84–03, Univ. Bolivar, Venezuela.

[6.33] F. Mora and H. M. Möller, 'The Computation of the Hilbert Function', *Proc. of the EUROCAL 83*, London, March, 1983, H. van Hulzen (ed.), Lecture Notes in Computer Science 162, Springer, 1983, pp. 157–167.

[6.34] F. Mora, and H. M. Moller, 'New Constructive Methods in Classical Ideal Theory', Univ. of Genova (Italy), Math. Dept., submitted to publication.

[6.35] A. Galligo, 'The Division Theorem and Stability in Local Analytic Geometry (French)', Extrait des Annales de l'Institut Fourier, Univ. of Grenoble, Vol. 29, No. 2, 1979.

[6.36] H. Hironaka, 'Resolution of Singularities of an Algebraic Variety over a Field of Characteristic Zero: I, II', *Annals of Math.*, Vol. 79, 1964, pp. 109–326.

[6.37] B. Buchberger, 'Miscellaneous Results on Gröber bases for Polynomial Ideals II', *Techn. Rep.* 83–1, University of Delaware, Dept. of Comp. and Inform. Scie., 1983.

[6.38] D. E. Knuth and P. B. Bendix, 'Simple Word Problems in Universal Algebras', *Proc. of the Conf. on Computational Problems in Abstract Algebra*, Oxford, 1967, J. Leech, (ed.), Pergamon Press, Oxford, 1970.

[6.39] P. Le Chenadec, 'Canonical Forms in Finitely Presented Algebras (French)', Ph.D. Thesis, Univ. of Paris Sud, Centre d'Orsay, 1983.

[6.40] F. Winkler and B. Buchberger, 'A Criterion for Eliminating Unnecessary Reductions in the Knuth-Bendix Algorithm', *Proc. of the Coll. on Algebra, Combinatorics and Logic in Comp. Scie.*, Gyor, Sept. Coll. Math. Soc. J. Bolyai 42, 1985.

[6.41] J. Hsiang, 'Topics in Automated Theorem Proving and Program Generation', Ph.D. Thesis, Univ. of Illinois at Urbana-Champaign, Dept. of Comp. Scie., 1982.

[6.42] B. L. Van der Waerden, *Modern Algebra: I, II*, New York, Frederick Ungar, 1953.

[6.43] L. E. Dickson, 'Finiteness of the Odd Perfect and Primitive Abundant Numbers with *n* Distinct Prime Factors', *Am. J. of Math.*, Vol. 35, 1913, pp. 413–426.

[6.44] R. Loos, 'Generalized Polynomial Remainder Sequences', in *Computer Algebra – Symbolic and Algebraic Computation*, B. Buchberger, R. Loos, and G. E. Collins, (eds.), Springer, Wien–New York, 2nd edition, 1983, pp. 115–138.

[6.45] B. F. Caviness and R. Fateman, 'Simplification of Radical Expressions', *Proc. 1976 ACM Symposium on Symbolic and Algebraic Computation*, Yorktown Heights, N.Y., August 1976, R. D. Jenks (ed.), published by ACM, pp. 329–338.

[6.46] R. Gebauer and H. Kredel, 'Buchberger's Algorithm for Constructing Canonical Bases (Gröbner bases) for Polynomial Ideals', Program documentation, Univ. of Heidelberg, Dept. for Applied Math., 1983.

[6.47] W. Grobner, *Modern Algebraic Geometry* (German), Springer, Wien–Innsbruck, 1949.

[6.48] G. E. Collins and L. E. Heindel, 'The SAC-1 Polynomial Real Zero System', *Techn. Rep.* No. 93, Comp. Scie. Dept., Univ. of Wisconsin-Madison, 1970.

[6.49] C. W. Ayoub, 'On Constructing Bases for Ideals in Polynomial Rings over the Integers', *Techn. Rep.* No. 8184, Pennsylvania State Univ., Univ. Park, Dept. of Math., 1981.

[6.50] H. M. Moller, 'Multi-dimensional Hermite Interpolation and Numerical Integration (German)', *Math. Zeitschrift*, Vol. 148, 1976, pp. 107–118.
[6.51] B. Buchberger, V. E. Krishnamurthy, and F. Winkler, 'Gröbner Bases, Polynomial Remainder Sequences and Decoding of Multivariate Hensel Codes', (this volume).

Note added in proof: Meanwhile a number of new papers on Gröbner bases (complexity and applications) have appeared in the literature. Some of them are collected in the following two proceedings [6.52], [6,53]. Some will appear in the new Journal of Symbolic Computation (Academic Press).

[6.52] *Proc. of the EUROSAM 84 Symposium*, Cambridge, J. Fitch (ed.), Springer Lecture Notes in Computer Science 174, 1984.
[6.53] *Proc. of the EUROCAL 85 Symposium*, Linz, April 1985 (to appear.)