daisy
Kogeddes@uwaterloo.edu
~~daisy.edu~~

CHAPTER 10

GROEBNER BASES FOR POLYNOMIAL IDEALS

10.1. INTRODUCTION

We have already seen that, among the various algebraic objects we have encountered, polynomials play a central role in symbolic computation. Indeed, many of the (higher-level) algorithms discussed in Chapter 9 (and later in Chapter 11) depend heavily on computation with multivariate polynomials. Hence, considerable effort has been devoted to improving the efficiency of algorithms for arithmetic, GCD's and factorization of polynomials. It also happens, though, that a fairly wide variety of problems involving polynomials (among them, simplification and the solution of equations) may be formulated in terms of *polynomial ideals*. This should come as no surprize, since we have already used particular types of ideal bases (i.e. those derived as kernels of homomorphisms) to obtain algorithms based on interpolation and Hensel's Lemma. Still, satisfactory *algorithmic* solutions for many such problems did not exist until the fairly recent development of a special type of ideal basis, namely the Groebner basis.

We recall that, given a commutative ring with identity R, a non-empty subset $\mathrm{I} \subseteq \mathrm{R}$ is an *ideal* when

(i) $p, q \in \mathrm{I} \quad \Rightarrow \quad p - q \in I$,

(ii) $p \in \mathrm{I}, r \in \mathrm{R} \quad \Rightarrow \quad r p \in \mathrm{I}$.

Every (finite) set of polynomials $P = \{p_1, \ldots, p_k\} \subset \mathrm{F}[x_1, \ldots, x_n]$ generates an ideal

$$<P> \; = \; <p_1, \ldots, p_k> \; = \; \{\sum_{i=1}^{k} a_i p_i \mid a_i \in \mathrm{F}[x_1, \ldots, x_n]\} .$$

The set P is then said to form a *basis* for this ideal. Unfortunately, while P generates the (infinite) set $<P>$, the polynomials p_i in P may not yield much insight into the nature of this ideal. For example, a set of simple polynomials over **Q** such as

$$p_1 = x^3yz - xz^2\,,\ \ p_2 = xy^2z - xyz\,,\ \ p_3 = x^2y^2 - z^2$$

generates a polynomial ideal in Q[x,y,z]; namely,

$$<p_1, p_2, p_3> = \{a_1 \cdot p_1 + a_2 \cdot p_2 + a_3 \cdot p_3 \mid a_1, a_2, a_3 \in Q[x,y,z]\}.$$

It is not difficult to show that $q = x^2yz - z^3$ is a member of this ideal since one can find polynomials a, b, c such that

$$q \ = \ ap_1 + bp_2 + cp_3\ .$$

In this case, one could eventually determine these a, b, c by trial-and-error. However, it is generally a difficult problem to decide whether a given q is in the ideal $<p_1, \ldots, p_k>$ for arbitrary polynomials p_i. We mention that the "ideal membership" problem (which was considered, but not fully solved by Hermann [22] in 1926) may be viewed as an instance of the "zero-equivalence" problem studied in Chapter 3. For example, deciding if $q \in <p_1, p_2, p_3>$ in the previous problem is the same as deciding if q simplifies to 0 with respect to the side relations

$$x^3yz - xz^2 \ = \ 0\,,\ \ xy^2z - xyz \ = \ 0\,,\ \ x^2y^2 - z^2 \ = \ 0\ .$$

It is easy to show that for a fixed set of polynomials P, the relation $\sim$ defined by

$$q_1 \sim q_2 \quad \Leftrightarrow \quad q_1 - q_2 \in \ <P>$$

is an equivalence relation. Hence, both of these problems will be solved *if* we can find a normal function (i.e. a zero-equivalence simplifier) for F[$x_1, \ldots, x_n$] with respect to $\sim$.

Consider also the problem of solving a system of nonlinear equations

$$p_1 \ = \ 0\,,\ p_2 \ = \ 0\,,\ \ldots,\ p_k \ = \ 0\,,$$

where each $p_i \in$ F[$x_1, \ldots, x_n$] and F is a field. In the previous chapter we used resultants to transform a set of polynomials $P = \{p_1, \ldots, p_k\}$ into an equivalent set (i.e. one with all of the original common zeros) from which the roots could be more easily obtained. For example, the nonlinear system of equations

$$\{x^2y - x^2 + 5xy - 2y + 1 \ = \ 0\ ,\ \ xy^2 - 2xy + x - 4y^3 - 7 \ = \ 0\}$$

may be "reduced" into the system

$$\{x^2y - x^2 + 5xy - 2y + 1 \ = \ 0\ ,$$
$$xy^2 - 2xy + x - 4y^3 - 7 \ = \ 0\ ,$$
$$16y^7 + 4y^6 - 42y^5 + 85y^4 - 37y^3 - 56y^2 + 78y - 48 \ = \ 0\}\ .$$

which is then solved. However, we noted in Chapter 10 that such a reduced system will not always exist; moreover one cannot always tell from a reduced system whether a given system of equations is solvable or not. In hindsight, it should be clear that a reduced system for P is simply an alternate (but more useful) basis for the ideal $<P>$. What we would like, however, is an alternate ideal basis which always exists and from which the existence and uniqueness of solutions (as well as the solutions themselves) may easily be determined.

It is reasonable to wonder if the above problems might be solvable, if only an arbitrary ideal basis could be transformed into a sufficiently potent form. In fact, Hironaka [23] established the *existence* of such a basis (which he called a "standard basis") for ideals of formal power series in 1964. However it was Buchberger [5] who, in his Ph.D. thesis, first presented an *algorithm* to perform the required transformation in the context of polynomial ideals. He soon named these special bases *Groebner bases* (after his supervisor, W. Groebner), and refined both the concept and algorithm further. Hence, most of the concepts (and, in fact, many of the proofs) we present are due to Buchberger. Today, most modern computer algebra systems include an implementation of Buchberger's algorithm.

In this chapter, we will first present the concepts of reduction and Groebner bases, in terms of the ideal membership problem. We develop Buchberger's algorithm for computing Groebner bases, and consider its practical improvement. Various extensions of the algorithm, and its connection with other symbolic algorithms are (briefly) discussed. Finally, we examine some of the applications of Groebner bases, including solving systems of algebraic equations.

10.2. TERMS ORDERINGS AND REDUCTION

For univariate polynomials, the zero-equivalence problem is easily solved since F[x] is a Euclidean domain. Hence, we can simplify with respect to univariate polynomials using ordinary polynomial division (i.e. the rem function). For multivariate domains, however, the situation is much less clear, as our previous example shows. Still, it was pointed out in Chapter 5 that a multivariate polynomial domain over a field (while not a Euclidean domain, or even a principal ideal domain) is a Noetherian ideal domain; that is, every ideal in such a domain has a finite basis. Fortunately, this is almost enough to allow us to solve the above problems (and more). The missing (but easily supplied) element is a small amount of additional structure on the polynomial ring, which will permit a more algorithmic treatment of multivariate polynomials. As in earlier chapters, we will denote the polynomial ring by F[x] when the (ordered) set of variables $x = (x_1, x_2, \ldots, x_n)$ is understood.

Orderings of Multivariate Terms

We begin by defining the set of *terms* in **x** by

$$T_{\mathbf{x}} = \{ x_1^{i_1} \cdots x_n^{i_n} \mid i_1, \ldots, i_n \in \mathrm{N} \},$$

where N is the set of non-negative integers. Note that this constitutes a (vector space) basis for F[**x**] over the field (coefficient domain) F. We will require that these terms be ordered as follows.

Definition 10.1. An *admissible total ordering* $<_T$ for the set $T_{\mathbf{x}}$ is one which satisfies:

(i) $1 <_T t$,

(ii) $s <_T t \implies su <_T tu$

for all $s, t, u \in T_{\mathbf{x}}$, where $1 = x_1^0 \cdots x_n^0$.

■

A wide variety of admissible orderings are possible. (See, for example, Exercise 10.17.) However, we will discuss the two which are most common in the literature (and which seem to be the most useful in practice).

Definition 10.2. The *(pure) lexicographic* term ordering is defined by

$$s = x_1^{i_1} \cdots x_n^{i_n} <_L x_1^{j_1} \cdots x_n^{j_n} = t \iff$$

$$\exists\, l \text{ such that } i_l < j_l \text{ and } i_k = j_k,\ 1 \le k < l .$$

■

Note that by specifying the polynomial ring as $\mathrm{F}[x_1, \ldots, x_n]$, the precedence

$$x_1 >_L x_2 >_L \cdots >_L x_n$$

is implied.

Example 10.1. The trivariate terms in x, y, z are lexicographically ordered

$$1 <_L z <_L z^2 <_L \cdots <_L y <_L yz <_L yz^2 <_L \cdots$$
$$<_L y^2 <_L y^2z <_L \cdots <_L x <_L xz <_L \cdots <_L xy <_L \cdots .$$

■

Definition 10.3. The *(total) degree* (or *graduated*) term ordering is defined by

$$s = x_1^{i_1} \cdots x_n^{i_n} <_D x_1^{j_1} \cdots x_n^{j_n} = t \iff$$

$$\deg(s) < \deg(t)\ , \text{ or}$$

$$\{\deg(s) = \deg(t) \text{ and}$$
$$\exists\, l \text{ such that } i_l > j_l \text{ and } i_k = j_k,\ l < k \le n\} .$$

■

We note that terms of equal total degree are ordered using an *inverse* lexicographic ordering, which is admissible within these graduations. Obviously, a different term ordering results from using the regular lexicographic ordering for this purpose. Both types are referred to as "total degree" orderings in the literature; however, we will use Definition 10.3 exclusively.

Example 10.2. The trivariate terms in x, y, z are degree-ordered

$$1 <_D z <_D y <_D x <_D$$
$$<_D z^2 <_D yz <_D xz <_D y^2 <_D xy <_D x^2$$
$$<_D z^3 <_D yz^2 <_D xz^2 <_D y^2z <_D xyz <_D \cdots .$$

■

Clearly, any polynomial in F[x] contains a monomial whose term is maximal with respect to a given term ordering $<_T$. We will adopt the following notation.

Definition 10.4. The *leading monomial* of $p \in$ F[x] with respect to $<_T$ is the monomial appearing in p whose term is maximal among those in p. We denote this by $\mathrm{M}_T(p)$, or simply by $\mathrm{M}(p)$ if the term ordering $<_T$ is understood. Also define hterm(p) to be the maximal ("head") term, and hcoeff(p) to be the corresponding coefficient, so that

$$\mathrm{M}(p) = \mathrm{hcoeff}(p)\ \mathrm{hterm}(p) .$$

We adopt the convention that hcoeff(0) = 0 and hterm(0) = 1.

■

Example 10.3. Suppose we consider

$$p = -2x^2yz + x^2y^2 + x^2z^2 + x^2y + 2xy^2z^2 - 3xyz^3 - xy + yz + z^2 + 5$$

as an element of $Q[x,y,z]$. We may write p so that its terms are in descending order with respect to $<_D$, as

$$p = 2xy^2z^2 - 3xyz^3 + x^2y^2 - 2x^2yz + x^2z^2 + x^2y - xy + yz + z^2 + 5 .$$

Clearly, then, we have

$$\mathrm{M}(p) = 2xy^2z^2 ,\quad \mathrm{hterm}(p) = xy^2z^2 ,\quad \mathrm{hcoeff}(p) = 2 .$$

If p is considered as an element of $Q(z)[x,y]$, then we write

$$p = x^2y^2 - (2z-1)x^2y + (2z^2)xy^2 + (z^2)x^2 - (3z^3+1)xy + (z)y + (z^2+5) \; ;$$

hence

$$\mathrm{M}(p) = x^2y^2 \, , \;\; \mathrm{hterm}(p) = x^2y^2 \, , \;\; \mathrm{hcoeff}(p) = 1 \, .$$

We note finally that under the *lexicographic* ordering for $\mathrm{T}_{[x,y]}$ we would write the terms in descending order as

$$p = x^2y^2 - (2z-1)x^2y + (z^2)x^2 + (2z^2)xy^2 - (3z^3+1)xy + (z)y + (z^2+5) \, .$$

■

Reduction in Multivariate Domains

The above structure on F[x] now permits a certain type of simplification.

Definition 10.5. For p, $q \in \mathrm{F}[\mathrm{x}]$ we say that *p reduces with respect to q* (and with respect to a fixed term ordering) if there exists a monomial in p which is divisible by hterm(q). If $p = \alpha t + r$ where $\alpha \in \mathrm{F}$, $t \in \mathrm{T}_{\mathrm{x}}$, $r \in \mathrm{F}[\mathrm{x}]$ and

$$\frac{t}{\mathrm{hterm}(q)} = u \in \mathrm{T}_{\mathrm{x}} \, ,$$

then we write

$$p \mapsto_q p - \frac{\alpha t}{\mathrm{M}(q)} q = p - \frac{\alpha}{\mathrm{hcoeff}(q)} u \, q = p'$$

to signify that *p reduces to p'* (*with respect to q*). If p reduces to p' with respect to some polynomial in $Q = \{q_1, q_2, \ldots, q_m\}$, we say that *$p$ reduces modulo Q* and write $p \mapsto_Q p'$; otherwise, we say that p is *irreducible* (or *reduced*) *with respect to Q*. We adopt the convention that 0 is always irreducible.

■

It is apparent that the process of reduction involves subtracting an appropriate multiple of one polynomial from another, to obtain a result which is (in a sense) smaller. As such, it may be viewed as one step in a generalized division.

Example 10.4. For the polynomials

$$p = 6x^4 + 13x^3 - 6x + 1 \, , \quad q = 3x^2 + 5x - 1$$

we have

$$p \mapsto_q p - 2x^2 \cdot q = 3x^3 + 2x^2 - 6x + 1$$

if we reduce the leading term. We might also compute

$$p \mapsto_q p - \frac{13}{3} x \cdot q = 6x^4 - \frac{65}{3} x^2 - \frac{5}{3} x + 1$$

if we instead reduce the term of degree 3. We note that in either case, we could continue reducing to eventually obtain 0, since in fact $q \,|\, p$. (Note that in this case, reduction and polynomial division are equivalent.)

■

Example 10.5. Consider the polynomials

$$p \;=\; 2y^2z - xz^2 \;, \quad q \;=\; 7y^2 + yz - 4 \;,$$

and impose the ordering $<_D$ on $T_{[x,y,z]}$. Note that we have written the terms of these polynomials in descending order with respect to $<_D$.) Then we have

$$p \;\mapsto_q\; p - \frac{2}{7}z \cdot q \;=\; -xz^2 - \frac{2}{7}yz^2 + \frac{8}{7}z \;,$$

which is then irreducible modulo q.

■

A fundamental property of reduction is the following.

Theorem 10.1. For a fixed set Q and ordering $<_T$, there is no infinite sequence of reductions

$$p_0 \;\mapsto_Q\; p_1 \;\mapsto_Q\; p_2 \;\mapsto_Q\; \cdots \;. \tag{10.1}$$

Proof: We proceed by induction on i, the number of variables in p_0. It is clear that there is no infinite sequence for $i=0$, since either Q contains elements of headterm 1 ($p_0 \mapsto_Q 0$), or p_0 is irreducible modulo Q. (We may now ignore the possibility that $<Q> \;=\; <1>$.)

Now consider $i=1$: assume there is no infinite sequence of reductions of a polynomial in $F[x_1]$ of degree $k-1$, and suppose that $p_0 \in F[x_1]$ is of degree k. (The previous point treats the case $k=0$.) By assumption, there is no infinite sequence of reductions on the lower order terms of p_0. However, if reduction of the term of degree k is possible, we obtain a polynomial of lower degree.

Similarly, suppose that there is no infinite sequence of reductions of a polynomial in $F[x_1,x_2]$ of degree $l-1$ in x_2, and let $p_0 \in F[x_1,x_2]$ have $\deg_2(p_0) \;=\; l$. As before, consider the terms in p_0 of degree l in x_2; together, these constitute a polynomial of fixed degree (say, m) in x_1 which (by a previous argument) must eventually be reduced as part of p_0. By another induction, we see that the maximal term $x_1^m x_2^l$ must eventually be reduced; i.e., the process must terminate. The argument may be extended to an arbitrary $i \;=\; n$ variables.

■

Let $\overset{*}{\mapsto}_Q$ denote the associative closure of $\mapsto_Q$. That is, $p \overset{*}{\mapsto}_Q q$ if and only if there is a sequence

$$p \;=\; p_0 \;\mapsto_Q\; p_1 \;\mapsto_Q\; \cdots \;\mapsto_Q\; p_n \;=\; q \;.$$

If $p \overset{*}{\mapsto}_Q q$ and q is irreducible, we will write $p \overset{!}{\mapsto}_Q q$. By Theorem 10.1, we may construct an algorithm which, given a polynomial p, finds q such that

$p \overset{*}{\mapsto}_Q q$. For the sake of efficency, it makes the most sense to organize this algorithm so that the *largest* monomials are reduced first, since these reductions affect the lower order monomials anyway. (Compare, for example, this and the opposite strategy on Example 10.4.) Therefore, we formulate our scheme to first reduce the leading monomial $\mathrm{M}(p)$ (as part of p), and then $p - \mathrm{M}(p)$ (as a distinct polynomial). Since we will only need to find reducers for leading monomials, it is convenient to define $R_{0,Q} = \emptyset$, and (for non-zero p)

$$R_{p,Q} \;=\; \{q \in Q \text{ such that } \operatorname{hterm}(q) \,|\, \operatorname{hterm}(p)\}\,.$$

We note that if several reducers exist for $\mathrm{M}(p)$, *any* one may be chosen. However, this choice will affect the efficiency of the algorithm. In practice, the optimal selection depends on the term ordering used. (See Exercise 10.3, for example.) We will therefore write selectpoly($R_{p,Q}$) to denote that some reducer (e.g. the first one) is chosen. A possible reduction algorithm is presented below as Algorithm 10.1.

Algorithm 10.1 Full Reduction of p Modulo Q

```
procedure Reduce($p$, $Q$)

    # Given a polynomial $p$ and a set of polynomials $Q$
    # from the ring F[x], find a $q$ such that $p \overset{*}{\mapsto} q$.

    # Start with the whole polynomial.

    $r \leftarrow p$ ; $q \leftarrow 0$

    # If no reducers exist, strip off the leading monomial;
    # otherwise, continue to reduce.

    while $r \neq 0$ do {
        while $R_{r,Q} \neq \emptyset$ do {
            $f \leftarrow$ selectpoly($R_{r,Q}$)
            $r \leftarrow r - \frac{\mathrm{M}(r)\,f}{\mathrm{M}(f)}$ }
        $q \leftarrow q + \mathrm{M}(r)$ ; $r \leftarrow r - \mathrm{M}(r)$ }
    return( $q$ )
end
```

There are several ways in which the efficiency of this procedure may be further improved. For example, it should actually terminate when all terms in "r" as large as the smallest headterm in Q have been reduced. This is a small point, however, since no significant amount of arithmetic is performed in this phase. It is far more important to economize, where possible, on the amount of (coefficient) arithmetic performed in the innermost loop. One approach is to first divide each of the polynomials in Q by its leading coefficient. Another approach is possible when the coefficient field is the fraction field of some integral domain D. Namely, as in the previous chapter it is possible to (temporarily) perform most of the computations in the domain D (essentially, in the manner of the primitive PRS). (See Czapor [17] for the details.)

Example 10.6. Consider the set $P = \{p_1, p_2\} \subset \mathbf{Q}[x,y]$, where

$$p_1 = x^2y + 5x^2 + y^2 \ , \ \ p_2 = 7xy^2 - 2y^3 + 1 \ ;$$

impose the lexicographic term ordering (where $x >_L y$). Then for the polynomial

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

we have

$$\begin{aligned} q &\longmapsto_{p_1} q - 3x \cdot p_1 \\ &= -15x^3 + 2x^2y^2 - 3xy^2 - 3xy + 5x \\ &\longmapsto_{p_1} -15x^3 - 10x^2y - 3xy^2 - 3xy + 5x - 2y^3 \\ &\longmapsto_{p_1} -15x^3 + 50x^2 - 3xy^2 - 3xy + 5x - 2y^3 + 10y^2 \\ &\longmapsto_{p_2} -15x^3 + 50x^2 - 3xy + 5x - \frac{20}{7}y^3 + 10y^2 + \frac{3}{7} \ . \end{aligned}$$

The final result is the fully reduced form of q modulo P. Note that we have written the terms of each polynomial in descending order with respect to $<_L$.

■

Example 10.7. Suppose we adopt the degree ordering for $\mathbf{Q}[x,y,z]$, and consider again the set of polynomials $P = \{p_1, p_2, p_3\}$ where

$$p_1 = x^3yz - xz^2 \ , \ \ p_2 = xy^2z - xyz \ , \ \ p_3 = x^2y^2 - z^2 \ .$$

Also, let

$$q = x^2y^2z - z^3 \ , \ \ r = -x^2y^2z + x^2yz \ .$$

Then

$$q \ \longmapsto_{p_3} \ x^2y^2z - z^3 \ - \ z(x^2y^2 - z^2) \ = \ 0 \ ,$$

and similarly $r \mapsto_{p_2} 0$. Note, however, that $q+r = x^2yz - z^3$ is irreducible modulo P.

■

The fact that, in the above example, $q+r$ is irreducible when q, r each reduce to 0 suggests that useful theorems on reduction may be difficult to prove. The following theorems (which we will use in the next section) illustrate that in general only modest results exist.

Theorem 10.2. Consider $p,\ q,\ r \in \mathrm{F}[\mathrm{x}]$ and $S \subset \mathrm{F}[\mathrm{x}]$. If $p-q \mapsto_S r$, then there exist $\tilde{p}$, $\tilde{q}$ such that

$$p \mapsto_S \tilde{p}\ ,\ q \mapsto_S \tilde{q}\ ,\ r = \tilde{p} - \tilde{q}\ .$$

Proof: Let $s \in S$, $\alpha \in \mathrm{F}$, $\nu \in \mathrm{T_x}$ be such that

$$r = (p-q) - \alpha{\cdot}\nu{\cdot}\frac{s}{\mathrm{M}(s)}\ .$$

(Then ν is the term eliminated in the reduction.) Suppose that ν has coefficient β_1 in p, and coefficient β_2 in q. Assume that $\beta_1 \neq \beta_2$, since the result is only interesting if ν actually appears in $p-q$ (with coefficient $\alpha = \beta_1 - \beta_2$) and either p or q. Now let $u = \nu/\mathrm{hterm}(s)$, and choose

$$\tilde{p} = p - \frac{\beta_1}{\mathrm{hcoeff}(s)}{\cdot}u{\cdot}s\ ,\ \tilde{q} = q - \frac{\beta_2}{\mathrm{hcoeff}(s)}{\cdot}u{\cdot}s\ .$$

■

Theorem 10.3. Suppose $p,\ q \in \mathrm{F}[\mathrm{x}]$ are such that $p-q \overset{+}{\mapsto}_S 0$ for $S \subset \mathrm{F}[\mathrm{x}]$. Then there exists $r \in \mathrm{F}[\mathrm{x}]$ such that $p \overset{*}{\mapsto}_S r$ and $q \overset{*}{\mapsto}_S r$, i.e. p, q have a "common successor" when reduced with respect to S.

Proof: As with many of the results of this chapter, we proceed by induction (on the number of steps necessary to reduce $p-q$ to 0). Clearly, if $p = q$ the result is true. Now assume that the result holds for $n-1$ reduction steps, and suppose that

$$p-q \mapsto_S h_1 \mapsto_S h_2 \mapsto_S \cdots \mapsto_S h_n = 0\ .$$

By Theorem 10.2, $\exists\ \tilde{p}$, $\tilde{q}$ such that $p \mapsto_S \tilde{p}$, $q \mapsto_S \tilde{q}$, and $\tilde{p} - \tilde{q} = h_1$. But then, by hypothesis, $\tilde{p}$ and $\tilde{q}$ (and hence $p,\ q$) have a common successor.

■

Theorem 10.4. If p_1, p_2 are polynomials such that $p_1 \mapsto_Q p_2$, then for any polynomial r, there exists s such that

$$p_1 + r \mapsto_Q s \ , \ p_2 + r \mapsto_Q s \ .$$

Proof: Let $\alpha \in \mathrm{F}$, $u \in \mathrm{T_x}$, $q \in Q$ be such that $p_2 = p_1 - \alpha u \cdot q/\mathrm{hcoeff}(q)$, and let $t = u \cdot \mathrm{hterm}(q)$ be the term cancelled in the reduction. For arbitrary r, suppose that t has coefficient β in r (or in $p_2 + r$); then t has coefficient $\alpha + \beta$ in $p_1 + r$. Now, for $\tilde{q} = q/\mathrm{hcoeff}(q)$ we have

$$p_1 + r \mapsto_q s_1 = (p_1 + r) - (\alpha + \beta) u \cdot \tilde{q} \ ,$$

$$p_2 + r \mapsto_q s_2 = (p_2 + r) - \beta u \cdot \tilde{q} \ ,$$

and

$$s_1 - s_2 = [\alpha - (\alpha + \beta) + \beta] u \cdot \tilde{q} = 0 \ .$$

Therefore, $s = s_1 = s_2$ is the required polynomial.

■

10.3. GROEBNER BASES AND BUCHBERGER'S ALGORITHM

While it is certainly true that $p \in \langle Q \rangle$ if $p \overset{*}{\mapsto}_Q 0$, Example 10.7 shows that the converse is not true. Hence, the process of reduction will not solve the zero-equivalence problem as it stands. It turns out that this is not, strictly speaking, due to a deficiency of Algorithm 10.1, but rather the structure of the ideal basis Q. We therefore propose the following:

Definition 10.6. An ideal basis $G \subset \mathrm{F[x]}$ is called a *Groebner basis* (with respect to a fixed term ordering $<_T$ and the implied permutation of variables) if

$$p \in \langle G \rangle \iff \mathrm{Reduce}(p, G) = 0 \ .$$

■

Since $\mathrm{Reduce}(p, G) - p \in \langle G \rangle$, this states that G is a Groebner basis precisely when its reduction algorithm is a normal simplifier for $\mathrm{F[x]}/\langle G \rangle$.

Example 10.8. For the polynomials $P = \{p_1, p_2, p_3\}$ and q, r of the previous example,

$$G = \{p_1, p_2, p_3, \ x^2yz - z^3, \ xz^3 - xz^2, \\ yz^3 - z^3, \ xyz^2 - xz^2, \ x^2z^2 - z^4, \ z^5 - z^4\}$$

is a Groebner basis (with respect to the degree ordering for $\mathrm{T}_{[x,y,z]}$) such that $\langle P \rangle = \langle G \rangle$. Note that $q \overset{*}{\mapsto}_G 0$, $r \overset{*}{\mapsto}_G 0$, and $q + r \overset{*}{\mapsto}_G 0$, irrespective of the sequence of reductions that is followed.

■

Unfortunately, we do not yet have a means to actually prove that the above set is a Groebner basis. Thus we require an algorithm for their construction.

Alternate Characterizations of Groebner Bases

We have already seen that an arbitrary ideal basis P does not, in general, constitute a Groebner basis for $<P>$. The idea behind Buchberger's method is to "complete" the basis P by adding (a finite number of) new polynomials to it. Buchberger's primary contribution was to show that this completion only requires consideration of the following quantity, for finitely many pairs of polynomials from P.

Definition 10.7. The *S-polynomial of* p, $q \in \mathrm{F}[\mathbf{x}]$ is

$$\mathrm{Spoly}(p, q) = \mathrm{LCM}(\mathrm{M}(p), \mathrm{M}(q)) \left[\frac{p}{\mathrm{M}(p)} - \frac{q}{\mathrm{M}(q)} \right] . \qquad (10.2)$$

■

Example 10.9. For the polynomials $p_1, p_2 \in \mathrm{Q}[x,y]$ defined by

$$p_1 = 3x^2y - y^3 - 4 , \quad p_2 = xy^3 + x^2 - 9 ,$$

using the degree ordering on $\mathrm{T}_{[x,y]}$, we have

$$\begin{aligned} \mathrm{Spoly}(p_1, p_2) &= y^2(3x^2y - y^3 - 4) - 3x\,(xy^3 + x^2 - 9) \\ &= -y^5 - 3x^3 - 4y^2 + 27x \ . \end{aligned}$$

■

It is useful to view the S-polynomial (which generalizes the operation of reduction) as the difference between reducing $\mathrm{LCM}(\mathrm{M}(p), \mathrm{M}(q))$ modulo p and reducing it modulo q. This plays a crucial role in the following (fundamental) theorem of Buchberger [8] , which leads almost directly to an algorithm for computing Groebner bases.

Theorem 10.5. (Alternate Characterizations of Grobner Bases) The following are equivalent:

(i) G is a Groebner basis;

(ii) Reduce(Spoly(p, q), G) = 0 for all $p, q \in G$;

(iii) If Reduce(p, G) = q and Reduce(p, G) = r, then $q = r$.

Proof: Although the proof is rather involved, we present the details in order to further acquaint the reader with the subtleties of reduction. We proceed in three stages.

(i) $\Longrightarrow$ (ii): This is clear, since Spoly$(p, q) \in <G>$ implies that

$$\text{Spoly}(p, q) \longmapsto^{+}_{G} 0 = \text{Reduce}(0, G) .$$

(ii) $\Longrightarrow$ (iii): We proceed by induction on the headterm of p. First, consider the case hterm$(p) = 1$. Clearly the assertion is true, since either p is irreducible (i.e. it is already reduced) or reduces in one step to 0. Suppose, then, that (iii) holds for all p such that hterm$(p) <_T t$ for some fixed $t \in \mathrm{T}_{\mathbf{x}}$ (the "main" induction hypothesis); consider p such that hterm$(p) = t$. If t is irreducible (modulo G), $p \longmapsto^{\cdot}_{G} q$ and $p \longmapsto^{\cdot}_{G} r$, the result is fairly clear. This is because the reductions may involve only the lower order terms; i.e., if

$$p = \mathrm{M}(p) + p - \mathrm{M}(p) \longmapsto^{\cdot}_{G} \mathrm{M}(p) + p_1 = q$$

and hence

$$p \longmapsto^{\cdot}_{G} \mathrm{M}(p) + p_2 = r ,$$

the induction hypothesis (applied to p−M(p)) implies $p_1 = p_2$ and hence $q = r$. We therefore assume that t is reducible, and write

$$R_{p,G} = \{g_1, \ldots, g_m\} ,$$

where the order is fixed but arbitrary. Take p_1, p_2, q such that

$$\mathrm{M}(p) \longmapsto_{g_1} p_1 , \; p - \mathrm{M}(p) \longmapsto^{\cdot}_{G} p_2 , \; p_1 + p_2 \longmapsto^{\cdot}_{G} q , \tag{10.3a}$$

and hence also

$$p \longmapsto^{\cdot}_{G} q . \tag{10.3b}$$

(This is always possible by reducing the lower order terms first, since hterm(p_1) $<_T t$.) Now suppose that there is also an r such that $p \longmapsto^{\cdot}_{G} r$. We consider two cases.

(a) For the time being, assume that the latter reduction is of the form

$$\mathrm{M}(p) \longmapsto_{g_1} p_1 , \; p - \mathrm{M}(p) \longmapsto^{+}_{G} p_3 , \; p_1 + p_3 \longmapsto^{\cdot}_{G} r . \tag{10.4}$$

We claim that, under the present conditions, $p_1 + p_2$ and $p_1 + p_3$ have a common successor. This is established by induction on k, the number of steps in the reduction $p_3 \longmapsto^{*}_{G} p_2$ (which is always possible in view of the induction hypothesis). If $k = 0$, this is trivial. Let us assume that, say, $p_1 + f$ and $p_1 + p_3$ have a common successor if $p_3 \longmapsto^{+}_{G} f$ in l steps. Now let $\tilde{f}$ be such that $p_3 \longmapsto^{+}_{G} \tilde{f}$ in l steps, and $\tilde{f} \longmapsto_{G} p_2$. By Theorem 10.4, $\exists\, g$ such that

$$p_1 + \tilde{f} \overset{+}{\mapsto}_G g \ , \ p_1 + p_2 \overset{+}{\mapsto}_G g \ .$$

Since, by hypothesis, $\exists\, h$ such that

$$p_1 + p_3 \overset{+}{\mapsto}_G h \ , \ p_1 + \tilde{f} \overset{+}{\mapsto}_G h \ ,$$

it follows that for some $\tilde{g}$, $\tilde{h}$,

$$p_1 + \tilde{f} \overset{\bullet}{\mapsto}_G \tilde{g} \ , \ p_1 + p_2 \overset{\bullet}{\mapsto}_G \tilde{g} \ ,$$

$$p_1 + p_3 \overset{\bullet}{\mapsto}_G \tilde{h} \ , \ p_1 + \tilde{f} \overset{\bullet}{\mapsto}_G \tilde{h} \ .$$

Since the headterms of all these polyomials are smaller than t, the main induction hypothesis implies $\tilde{g} = \tilde{h}$; i.e., p_1+p_2 and p_1+p_3 have a common successor. Together with (10.3a), (10.4) and the main induction hypothesis, this implies that $r = q$.

(b) Assume that we have $\tilde{p}_1$, p_3 such that

$$\mathrm{M}(p) \mapsto_{g_n} \tilde{p}_1 \ , \ p - \mathrm{M}(p) \overset{+}{\mapsto}_G p_3 \ , \ \tilde{p}_1 + p_3 \overset{\bullet}{\mapsto}_G r \ , \tag{10.5a}$$

where $2 \le n \le m$, and

$$p \overset{\bullet}{\mapsto}_G r \ . \tag{10.5b}$$

Consider also the reductions

$$\mathrm{M}(p) \mapsto_{g_1} p_1 \ , \ p - \mathrm{M}(p) \overset{+}{\mapsto}_G p_3 \ , \ p_1 + p_3 \overset{\bullet}{\mapsto}_G \tilde{r} \ ; \tag{10.6a}$$

i.e.

$$p \overset{\bullet}{\mapsto}_G \tilde{r} = q \tag{10.6b}$$

(noting the result of case (a)). Now, we find that

$$\begin{aligned}(\tilde{p}_1 + p_3) - (p_1 + p_3) &= \tilde{p}_1 - p_1 \\ &= \mathrm{M}(p)\,[\,\frac{g_1}{\mathrm{M}(g_1)} - \frac{g_n}{\mathrm{M}(g_n)}\,] \ .\end{aligned} \tag{10.7}$$

Since g_1, $g_n \in R_{p,G}$, the above quantity is the product of Spoly(g_1, g_n) times a monomial. Applying (ii) and Theorem 10.3, $\exists\, f$ such that

$$\tilde{p}_1 + p_3 \overset{+}{\mapsto}_G f \ , \ p_1 + p_3 \overset{+}{\mapsto}_G f \ .$$

Therefore, by (10.5b), (10.6b), and the main hypothesis, $r = q$.

(iii) $\Rightarrow$ (i): If $p \in \langle G \rangle$, then $\exists\, h_i \in \mathrm{F}[\mathrm{x}]$ such that

$$p = \sum_{i=1}^{l} h_i\, g_i \ . \tag{10.8}$$

We proceed by induction on the maximal term t among the headterms of h_1g_1, h_2g_2, $\ldots$, h_lg_l. First, if $t = 1$ the result is trivial. (Either $p = 0$, or $p \in \mathrm{F}$ and $p \longmapsto_G 0$.) Now assume that for some t, we have $p \overset{+}{\longmapsto}_G 0$ whenever (10.8) holds with $\mathrm{hterm}(h_ig_i) <_T t$ for $1 \leq i \leq l$; then consider a polynomial p with $\mathrm{hterm}(h_ig_i) \leq_T t$ for some $1 \leq i \leq l$. We suppose (without loss of generality) that $\{h_1g_1, \ldots, h_mg_m\}$ are the (nonzero) polynomials in (10.8) which have headterm t. We will show that $p \overset{+}{\longmapsto}_G 0$ by induction on m. If $m = 1$, then

$$p = h_1g_1 + \sum_{i=2}^{l} h_ig_i \longmapsto_{g_1} \tilde{p} = (h_1 - \mathrm{M}(h_1))g_1 + \sum_{i=2}^{l} h_ig_i \ ,$$

and by the main hypothesis $\tilde{p} \overset{+}{\longmapsto}_G 0$. Now assume that $p \overset{+}{\longmapsto}_G 0$ when $m \leq k$ and consider $m = k+1$. (That is, the representation (10.8) of p has $k+1$ components with headterm t.) Now, for $\alpha \in \mathrm{F}$ write

$$p = h_1g_1 + h_2g_2 + \sum_{i=3}^{l} h_ig_i = \bar{p} + p' \ ,$$

where

$$\bar{p} = \mathrm{M}(h_1)g_1 + [\,\mathrm{M}(h_2) + \alpha{\cdot}\mathrm{hterm}(h_2)\,]{\cdot}g_2 \ ,$$

$$p' = (h_1 - \mathrm{M}(h_1))g_1 + [\,h_2 - \mathrm{M}(h_2) - \alpha{\cdot}\mathrm{hterm}(h_2)\,]{\cdot}g_2 + \sum_{i=3}^{l} h_ig_i \ , \quad (10.9)$$

and choose α such that

$$\bar{p} = \beta{\cdot}u{\cdot}\mathrm{Spoly}(g_1, g_2)$$

for some $\beta \in \mathrm{F}$, $u \in \mathrm{T}_x$ (Exercise 10.4). On one hand, the representation (10.9) has at most k components of headterm t; hence by hypothesis $p' \overset{+}{\longmapsto}_G 0$. On the other hand, we can show that $\bar{p} \overset{+}{\longmapsto}_G 0$, as follows. We note that

$$\frac{\mathrm{LCM}(\mathrm{M}(g_1), \mathrm{M}(g_2))}{\mathrm{M}(g_1)}{\cdot}g_1 \longmapsto_{g_2} \mathrm{Spoly}(g_1, g_2) \overset{*}{\longmapsto}_G q$$

for some q. But also

$$\frac{\mathrm{LCM}(\mathrm{M}(g_1), \mathrm{M}(g_2))}{\mathrm{M}(g_1)}{\cdot}g_1 \longmapsto_{g_1} 0 \ ,$$

which in view of (iii) implies $q = 0$. It follows that $\bar{p} = p - p' \overset{+}{\longmapsto}_G 0$ as well. Therefore, by Theorem 10.3 and (iii), $\exists\, r$ such that $p \overset{*}{\longmapsto}_G r$ and $p' \overset{*}{\longmapsto}_G r$. But since $p' \overset{*}{\longmapsto}_G 0 = \mathrm{Reduce}(0, G)$, we conclude that $r = 0$.

■

Corollary 10.6. G is a Groebner basis if and only if $\forall\, f, g \in G$ either

(1) $\mathrm{Spoly}(f, g) \overset{+}{\longmapsto}_G 0$, or

(2) $\exists\, h \in G,\ f \neq h \neq g$, such that

$$\mathrm{M}(h) \mid \mathrm{LCM}(\mathrm{M}(f), \mathrm{M}(g)) \,, \tag{10.10}$$

$$\mathrm{Spoly}(f, h) \overset{+}{\longmapsto}_G 0 \,,\ \mathrm{Spoly}(h, g) \overset{+}{\longmapsto}_G 0 \,. \tag{10.11}$$

Proof: If we replace (ii) in Theorem 10.5 with the above condition, we need only extend part (b) of the proof that (ii) $\Longrightarrow$ (iii) when (2) holds; we therefore resume the proof up to (10.6). We first note that by (10.10), $h \in R_{p,G}$. As before, we let p_1', s be such that

$$\mathrm{M}(p) \longmapsto_h p_1' \,,\ p - \mathrm{M}(p) \overset{+}{\longmapsto}_G p_3 \,,\ p_1' + p_3 \overset{\cdot}{\longmapsto}_G s \,, \tag{10.12a}$$

$$p \overset{+}{\longmapsto}_G \mathrm{M}(p) + p_3 \longmapsto_G p_1' + p_3 \overset{\cdot}{\longmapsto}_G s \,. \tag{10.12b}$$

Also (as before!),

$$(p_1' + p_3) - (p_1 + p_3) = \mathrm{M}(p)\,[\,\frac{f}{\mathrm{M}(f)} - \frac{h}{\mathrm{M}(h)}\,] \overset{+}{\longmapsto}_G 0 \,,$$

$$(\tilde{p}_1 + p_3) - (p_1' + p_3) = \mathrm{M}(p)\,[\,\frac{h}{\mathrm{M}(h)} - \frac{g}{\mathrm{M}(g)}\,] \overset{+}{\longmapsto}_G 0 \,,$$

using (10.11). Thus, by Theorem 10.3 and the induction hypothesis, we conclude that $r = s = q$.

■

Corollary 10.7. If G is a Groebner basis, then

$$\mathrm{Reduce}(p, G) = \mathrm{Reduce}(q, G) \iff p - q \in \langle G \rangle \,.$$

Proof:

$\Longrightarrow$: Suppose $r = \mathrm{Reduce}(p, G) = \mathrm{Reduce}(q, G)$. Then $p - r \in \langle G \rangle$ and $q - r \in \langle G \rangle$. Therefore,

$$(p - r) - (q - r) = p - q \in \langle G \rangle \,.$$

$\Longleftarrow$: Apply Theorem 10.3 (noting $p - q \in \langle G \rangle$), and then part (iii) of Theorem 10.5.

■

The need for Corollary 10.6 will become apparent in the next section. Corollary 10.7 shows that if G is a Groebner basis, then its reduction algorithm is not only a normal simplifier, but also a canonical simplifier (cf. Chapter 3). Decision procedures follow for a host of related problems in polynomial ideal theory, including ideal inclusion (Exercise 10.5) and computing in the quotient ring $F[x]/\langle G \rangle$. We postpone discussion of these, and other applications, until later sections.

Instead, we will now fulfill our promise to present an algorithm for the computation of Groebner bases.

Buchberger's Algorithm

Characterization (ii) of Theorem 10.5 suggests how we may transform an arbitrary ideal basis into a Groebner basis. Given a finite set $P \subset \mathrm{F}[\mathbf{x}]$, we may immediately test P by checking whether

$$\mathrm{Spoly}(p, q) \mapsto^{+}_{P} 0 \quad \forall\ p, q \in P,\ p \neq q\ .$$

If we find a pair (p, q) such that

$$\mathrm{Spoly}(p, q) \stackrel{\cdot}{\mapsto}_{P}\ r \neq 0\ ,$$

then $<P> \ = \ <P,\ r>$ and $\mathrm{Spoly}(p,\ q) \stackrel{\cdot}{\mapsto}_{P \cup \{r\}} 0$. That is, we may add the nonzero result to the basis, and begin testing of the *augmented* set. To see that such a process will terminate, let H_i be the set of headterms of the basis after the i-th new polynomial is added. Since new headterms are not multiples of old ones, the inclusions

$$<H_1> \ \subset \ <H_2> \ \subset \ \cdots$$

are proper. Given that F[x] is a Noetherian integral domain, the chain must terminate by Hilbert's "divisor chain condition" (see van der Waerden [33] for example.) The resulting algorithm appears below as Algorithm 10.2. As in Algorithm 10.1, we have used a procedure selectpair to denote that some selection is made from the nonempty set "B". Since the particular selection is of no *theoretical* importance, the reader may assume for now that the first pair is chosen. (The reason "G" appears as an argument to selectpoly will be explained later.)

Example 10.10. Consider the set $P \subset \mathrm{Q}[x,y,z]$ defined by

$$P \ = \ \{x^2 + yz - 2,\ y^2 + xz - 3,\ xy + z^2 - 5\}\ ,$$

using the degree ordering $<_D$. (As usual, the terms in each polynomial are written in descending order.) We first set $G = P$, $k = 3$, and $B = \{[1,2],\ [1,3],\ [2,3]\}$. Then

$$\begin{aligned} \mathrm{Spoly}(G_1,\ G_2) &= y^2\cdot(x^2 + yz - 2) - x^2\cdot(y^2 + xz - 3) \\ &= -x^3z + y^3z + 3x^2 - 2y^2 \\ &\mapsto_{G_1}\ y^3z + xyz^2 + 3x^2 - 2y^2 - 2xz \\ &\mapsto_{G_2}\ 3x^2 - 2y^2 - 2xz + 3yz \\ &\mapsto_{G_1}\ -2y^2 - 2xz + 6 \end{aligned}$$

Algorithm 10.2 Buchberger's Algorithm for Groebner Bases

```
procedure Gbasis($P$)
    # Given a set of polynomials $P$, compute $G$ such
    # that $<G> = <P>$ and $G$ is a Groebner basis.

    $G \leftarrow P$ ; $k \leftarrow$ length($G$)

    # We denote the $i$-th element of the ordered set
    # $G$ by $G_i$.

    $B \leftarrow \{ [i, j] \mid 1 \le i < j \le k \}$
    while $B \ne \emptyset$ do {
        $[i, j] \leftarrow$ selectpair($B$, $G$) ; $B \leftarrow B - \{[i, j]\}$
        $h \leftarrow$ Reduce(Spoly($G_i, G_j$), $G$)
        if $h \ne 0$ then {
            $G \leftarrow G \cup \{h\}$ ; $k \leftarrow k + 1$
            $B \leftarrow B \cup \{ [i, k] \mid 1 \le i < k \}$    } }
    return( $G$ )
end
```

$$\mapsto_{G_2} 0 \ ,$$

whereupon $B = \{[1, 3], [2, 3]\}$. Then

$$\begin{aligned} \text{Spoly}(G_1, G_3) &= y^2z - xz^2 + 5x - 2y \\ &\mapsto_{G_2} -2xz^2 + 5x - 2y + 3z \ , \end{aligned}$$

which is irreducible. We therefore set $G_4 = -2xz^2 + 5x - 2y + 3z$, ($k = 4$,) and $B = \{[2, 3], [1, 4], [2, 4], [3, 4]\}$. Continuing in this manner:

$$\text{Spoly}(G_2, G_3) \mapsto_{G_1} G_5 = -2yz^2 - 3x + 5y + 2z \ ,$$

$$B = \{[1, 4], [2, 4], [3, 4], [1, 5], [2, 5], [3, 5], [4, 5]\} \ ;$$

$$\text{Spoly}(G_1, G_4) \overset{*}{\mapsto}_G 0 \ ,$$

$$B = \{[2, 4], [3, 4], [1, 5], [2, 5], [3, 5], [4, 5]\} \ ;$$

$$\text{Spoly}(G_2, G_4) \overset{*}{\mapsto}_G 0 \ ,$$

$$B = \{[3, 4], [1, 5], [2, 5], [3, 5], [4, 5]\} \ ;$$

$$\mathrm{Spoly}(G_3, G_4) \ \mapsto^{+}_{G} \ G_6 \ = \ -2z^4 - 2xz - 3yz + 15z^2 - 19 \ ,$$

$$B \ = \ \{[1, 5], [2, 5], [3, 5], [4, 5], [1, 6], [2, 6], [3, 6], [4, 6], [5, 6]\} \, ,$$

after which all further S-polynomial reductions lead to 0.

■

When applied to linear polynomials, Algorithm 10.2 specializes to a Gaussian elimination algorithm. When applied to univariate polynomials, it specializes to Euclid's algorithm for several polynomials. The relationship with polynomial division processes is, in the bivariate case, fully specified by Lazard [29]. It has also been shown that Algorithm 10.2 and the Knuth-Bendix [26] algorithm for rewrite rules are both instances of a more general "critical pair/completion" algorithm. (See Buchberger [12] or LeChenadec [15], for example.) This connection has been exploited by Bachmair and Buchberger [2] to shorten the proof of Theorem 10.5, and by Winkler [34] to carry over improvements to Algorithm 10.2 to the Knuth-Bendix procedure. We mention also that the algorithm has been generalized to various Euclidean domains (e.g. Z); see Buchberger [13] or Kandri-Rody and Kapur [24] , for example.

10.4. IMPROVING BUCHBERGER'S ALGORITHM

It must be noted that if, in Example 10.10, we had used a different permutation of variables, or another term ordering, we would have obtained a completely different basis. It should also be pointed out that, these issues aside, Groebner bases are by no means unique.

Example 10.11. Consider the set P and corresponding Groebner basis G of Example 10.8. It may be shown (Exercise 10.7) that $G - \{p_1\}$ is also a Groebner basis for $<P>$.

■

Reduced Groebner Bases

Fortunately, the problem of non-uniqueness is very easily remedied, as we now show.

Definition 10.8. A set $G \subset \mathrm{F}[\mathrm{x}]$ is *reduced* if $\forall \ g \in G, \ g = \mathrm{Reduce}(g, G-\{g\})$; it is *monic* if $\forall \ g \in G, \ \mathrm{hcoeff}(g) = 1$.

■

Theorem 10.8. (Buchberger [7]) If G, H are reduced, monic Groebner bases such that $<G> \ = \ <H>$, then $G = H$.

■

We see that if the polynomials are scaled in *any* consistent manner, a Groebner basis may be made unique by ensuring that each element is reduced modulo the others. A possible algorithm to perform such a transformation appears as

Algorithm 10.3.

Algorithm 10.3 Construction of a Reduced Ideal Basis

```
procedure ReduceSet($E$)
    # Given a set $E$ (not necessarily a Groebner basis),
    # compute $\tilde{E}$ such that $<E> = <\tilde{E}>$ and $\tilde{E}$ is reduced.

    # First, remove any redundant elemements.

    $R \leftarrow$ E ; P $\leftarrow \emptyset$
    while $R \neq \emptyset$ do {
        $h \leftarrow$ selectpoly($R$) ; $R \leftarrow R - \{h\}$
        $h \leftarrow$ Reduce($h$, $P$)
        if $h \neq 0$ then {
            $Q \leftarrow \{ q \in P$ such that hterm($h$) | hterm($q$) $\}$
            $R \leftarrow R \cup Q$ ; $P \leftarrow P - Q \cup \{h\}$ } }

    # Ensure each element is reduced modulo the others.

    $\tilde{E} \leftarrow \emptyset$ ; $S \leftarrow P$
    foreach $h \in P$ do {
        $h \leftarrow$ Reduce($h$, $S-\{h\}$)
        $\tilde{E} \leftarrow \tilde{E} \cup \{h\}$ }
    return($\tilde{E}$)
end
```

A proof that Algorithm 10.3 terminates (similar to, but more involved than that of Theorem 10.1) is given by Buchberger [5]. When applied at the end of Algorithm 10.2, it is easy to see that only a subset of G must be reduced. Namely, for any G_i, G_j in Algorithm 10.2 such that hterm(G_j)|hterm(G_i), Spoly(G_i, G_j) is equal (up to a rescaling) to the reduced form of G_i modulo G_j; hence, G_i may be discarded at the end of the algorithm. And, although the result will not be unique if the input set "E" is not a Groebner basis, Algorithm 10.3 may also be applied before Algorithm 10.2. In fact, a reformulation of Algorithm 10.2 is possible in which the partial basis G is reduced after each new polynomial is added. Still, it is not clear how much pre- or inter-reduction is best in practice. (See Czapor [17] , for example.)

The Problem of Unnecessary Reductions

It should be apparent from Example 10.10 that Algorithm 10.2 is capable of producing extremely complex calculations from (apparently) modest input polynomials. Note, for example, that as "k" (the number of polynomials) grows, the number of S-polynomials in "B" grows rapidly. It turns out that when applied to polynomials of the form $s-t$, where $s, t \in T_X$, Algorithm 10.2 specializes to one for the "uniform word problem" for commutative semigroups (see Ballantyne and Lankford [3]). This relationship is used by Mayr and Meyer [30] to demonstrate that the congruence problem for polynomial ideals is exponentially space complete. Hence, the problem of constructing Groebner bases is intrinsically hard. This does not mean that Algorithm 10.2 is of no practical use; however, it is well worth considering some refinements which will improve its performance.

It is also clear that most of the computational cost of the algorithm is in the polynomial arithmetic of the reduction step. Now, it is easy to see that *full* reduction of each S-polynomial is not actually necessary; a partially reduced form $\tilde{h} \neq 0$ will suffice as long as $M(\tilde{h})$ is irreducible (i.e. it is not possible that $\tilde{h} \mapsto^{+}_{G} 0$). However, it may happen that the fully reduced form leads to simpler polynomials later in the algorithm; so, the actual benefits of this approach are difficult to assess. Quite typically, though, only a relatively small proportion of the S-polynomials which are reduced will yield new (nonzero) results. Therefore, a great deal of computation is wasted. Fortunately, Buchberger has shown that many of these 0-reductions may be detected *a priori*, without a significant amount of computation. This is accomplished, in part, using the following result:

Theorem 10.9. If $\mathrm{LCM}(\mathrm{hterm}(p), \mathrm{hterm}(q)) = \mathrm{hterm}(p)\cdot\mathrm{hterm}(q)$, then

$$\mathrm{Spoly}(p, q) \mapsto^{+}_{\{p,q\}} 0 .$$

Proof: We obtain

$$\begin{aligned}\mathrm{Spoly}(p, q) &= \alpha(M(q)\cdot p - M(p)\cdot q) \\ &= \alpha[M(q)\cdot(p - M(p)) - M(p)\cdot(q - M(q))] ,\end{aligned}$$

where $\alpha = \mathrm{GCD}(\mathrm{hcoeff}(p), \mathrm{hcoeff}(q))^{-1}$. No terms cancel in the subtraction above, since the terms of the two polynomials $p - M(p)$ and $q - M(p)$ are distinct. (This is an easy consequence of the fact that $M(p)$, $M(q)$ must contain distinct sets of variables.) Then note that

$$M(p) \mapsto_{p} p - M(p) , \quad M(q) \mapsto_{q} q - M(q) .$$

∎

The above result provides a condition under which certain S-polynomials (i.e. pairs $[i, j]$) may be skipped. Namely, $[i, j]$ may be safely ignored if it does not satisfy the function

$$\text{criterion1}([i, j], G) \iff \text{LCM}(\text{hterm}(G_i), \text{hterm}(G_j)) \neq \text{hterm}(G_i)\,\text{hterm}(G_j)\,.$$

In addition, Corollary 10.6 implies that we may skip Spoly(i, j) if $[i, j]$ does not satisfy

$$\begin{aligned} &\text{criterion2}([i, j], B, G) \iff \\ &\quad \neg\, \exists\, k,\ 1 \le k \le \text{length}(G),\ \text{such that} \\ &\qquad \{ i \neq k \neq j\,, \\ &\qquad\ \text{hterm}(G_k) \mid \text{LCM}(\text{hterm}(G_i), \text{hterm}(G_j)), \\ &\qquad\ [i, k] \notin B,\ [k, j] \notin B \}\,. \end{aligned}$$

The reader is referred to Buchberger [9] and Buchberger and Winkler [10], which together supply greater insight into the derivation of criterion2. In practical terms, the effect of using these criteria is dramatic. According to Buchberger, for example, criterion2 results (roughly speaking) in a reduction of the number of S-polynomial reductions from $O(K^2)$ to $O(K)$, where K is the final length of the basis. The improved form of Buchberger's algorithm appears below as Algorithm 10.4.

Buchberger and Winkler [10] also present an important argument regarding the procedure selectpair. It can be shown that if we always select $[i, j]$ such that

$$\text{LCM}(\text{hterm}(G_i), \text{hterm}(G_j)) = \min_{<_T} \{ \text{LCM}(\text{hterm}(G_u), \text{hterm}(G_v)) \mid [u, v] \in B \} \tag{10.13}$$

(the "normal" selection strategy), then criterion1, criterion2 are "good" in the sense that *all* possible reductions (i.e. not just one particular reduction) of Spoly(G_i, G_j) will yield 0. Moreover, the likelihood that criterion2 can even be applied is increased. Finally, if the degree ordering is used, this strategy would seem to lead to simpler polynomials than other choices. (Although, it has recently become apparent that the same cannot be said when the lexicographic ordering is used; see Czapor [17].)

Computational Complexity

We conclude this section with some brief remarks on the complexity of Buchberger's algorithm. It is useful to determine bounds on the maximum degree of any polynomial produced by the algorithm; this, in turn, may bound the number of polynomials and the (maximum) number of reduction steps required for each. Althought this is difficult in general, Buchberger [9] has shown the following: in the bivariate case, when a criterion similar to criterion2 is used (in conjunction with the normal selection strategy) the polynomials produced by the algorithm with the degree ordering are bounded by $4D(P)$, where

Algorithm 10.4 [Improved] Construction of Reduced Groebner Basis

```
procedure Gbasis($P$)

    # Given polynomials $P$, find the corresponding reduced
    # Groebner basis $G$.

    # First, pre-reduce the raw input set;
    # optionally, just set $G \leftarrow P$.

    $G \leftarrow$ ReduceSet($P$) ; $k \leftarrow$ length($G$)
    $B \leftarrow \{ [i, j] \mid 1 \leq i < j \leq k \}$
    while $B \neq \emptyset$ do {
        $[i, j] \leftarrow$ selectpair($B$, $G$) ; $B \leftarrow B - \{[i, j]\}$
        if criterion1($[i, j]$, $G$) and criterion2($[i, j]$, $B$, $G$) then {
            $h \leftarrow$ Reduce(Spoly($G_i$, $G_j$), $G$)
            if $h \neq 0$ then {
                $G \leftarrow G \cup \{h\}$ ; $k \leftarrow k + 1$
                $B \leftarrow B \cup \{ [i, k] \mid 1 \leq i < k \}$   } } }
    $R \leftarrow \{g \in G$ such that $\exists\, h \in G$ with $h \neq g$, hterm$(h) \mid$hterm$(g)\}$
    return( ReduceSet( $G - R$ ) )
end
```

$$\mathrm{D}(P) = \max \{\deg(P_i) \mid 1 \leq i \leq \mathrm{length}(P)\} ;$$

the number of computational steps is then bounded by

$$2(\mathrm{length}(P) + 16\mathrm{D}(P)^2)^4 .$$

Of course, the actual computational cost depends on the coefficent field as well. Recent results (see Winkler [35], for example) show progress with regard to development of a version of Algorithm 10.2 which uses a homomorphism/lifting approach similar to the EZ-GCD scheme (cf. Chapter 7). Since algorithms involving polynomial division (e.g. PRS algorithms) are plagued by the problem of coefficient growth, this is an imprtant area of study. The role of the term ordering used is illustrated by the following result of Buchberger [11] : for every natural number n, $\exists\, P \subset F[x,y]$ with $n = \mathrm{D}(P)$ such that

(a) for all Groebner bases for P with respect to $<_D$, $\mathrm{D}(G) \geq 2n - 1$;

(b) for all Groebner bases for P with respect to $<_L$, $\mathrm{D}(G) \geq n^2 - n + 1$.

Apparently, the complexity of the algorithm is lower when using $<_D$ than when using $<_L$. (See also Exercise 10.14.) Lazard [28] shows that for $<_D$ (or similar orderings), the maximum degree is usually below $\Sigma\deg(P_i)-n+1$, where n is the number of variables. Other important results are obtained by Moeller and Mora [31] and Giusti [20].

10.5. APPLICATIONS OF GROEBNER BASES

We have seen that Buchberger's algorithm completely solves the simplification problem for polynomials modulo side relations. That is, when $G \subset F[x]$ is a Groebner basis the corresponding reduction algorithm Reduce($\cdot$, G) is a canonical function for [F[x]; ~], where ~ is the "equivalence modulo G" relation used in Section 1. In view of the central role of polynomial domains in symbolic computation, this alone establishes the importance of Groebner bases. However, a survey of some of the applications of this powerful technique suggests that it is indeed one of the fundamental algorithms of symbolic computation. We will not attempt to list all such applications here; this is an active area of research, and any such list would soon be incomplete. Moreover, a discussion of the recent use of Groebner bases in such fields as bifurcation theory (Armbruster [1]) and spline theory (Billera and Rose [4]) is beyond the scope of this book. We instead restrict our attention to a few simple, but important, examples.

Computing in Quotient Rings

The close connection between the simplification problem and arithmetic in the quotient ring $F[x]/\langle G\rangle$ is illustrated by the following:

Theorem 10.10. Suppose G is a Groebner basis, and define

$$U = \{[u], \text{ where } u \in T_x \text{ is such that } \neg\exists\ g \in G \text{ with } \mathrm{hterm}(g)|u\}, \qquad (10.14)$$

where $[u]$ is the congruence class of u modulo G. Then U is a linearly independent [vector space] basis for $F[x]/\langle G\rangle$.

Proof: Suppose we have a dependence

$$c_1[u_1] + \cdots + c_m[u_m] = 0\ ,$$

where $c_i \in F$, $u_i \in U$ for $1 \le i \le m$. Since we now know that for $p \in F[x]$,

$$[p] = 0 \quad \Longleftrightarrow \quad p \in \langle G\rangle\ ,$$

and that reduction modulo G is a canonical simplifier, there must be a polynomial $g = c_1u_1 + \cdots + c_mu_m \in \langle G\rangle$. But it is only possible that $\mathrm{Reduce}(g, G) = 0$ if we have $c_i = 0$, $1 \le i \le m$.

■

The reader should compare the above result to the well known fact that an extension field of F of the form

$$F[x]/\langle p\rangle \;=\; \{p_0 + p_1 x + \cdots + p_{n-1}x^{n-1} \mid p_i \in F\}$$

where $p \in F[x]$ is an irreducible polynomial of degree n, is a vector space of dimension n with basis $[1]$, $[x]$, ..., $[x^{n-1}]$.

Theorem 10.10 allows us to easily decide if the quotient ring is finite dimensional (when considered as a vector space), since this is so if and only if the set U has finitely many elements. This observation will prove useful later on in this section. However, its immediate importance is that it guarantees we can perform arithmetic in the quotient ring.

Example 10.12. We recall that the set

$$G = \{x^2 + yz - 2,\ y^2 + xz - 3,\ xy + z^2 - 5,\ -2xz^2 + 5x - 2y + 3z,$$
$$-2yz^2 - 3x + 5y + 2z,\ -2z^4 - 2xz - 3yz + 15z^2 - 19\}$$

computed in Example 10.10 is a Groebner basis in $Q[x,y,z]$ with respect to $<_D$. In fact, it is also a reduced Groebner basis. Then

$$U = \{[1], [x], [y], [z], [xz], [yz], [z^2], [z^3]\}$$

is a basis for $Q[x,y,z]/\langle G\rangle$. To compute $[xz]\cdot[yz]$, for example, we merely find

$$\mathrm{Reduce}(xz\cdot yz,\ G) \;=\; xz + \frac{3}{2}yz - \frac{5}{2}z^2 + \frac{19}{2}\ ;$$

then

$$[xz]\cdot[yz] \;=\; 1[xz] + \frac{3}{2}[yz] - \frac{5}{2}[z^2] + \frac{19}{2}[1]\ .$$

■

In addition to the basic arithmetic operations, Theorem 10.10 allows us to compute inverses, when they exist, in $F[x]/\langle G\rangle$.

Example 10.13. Consider again the sets G, U of Example 10.12. Since U has finitely many entries, it may be possible to compute ring inverses for some of those entries. For example, if $[x]$ has an inverse, it must be of the form

$$[x]\cdot(a_0[1] + a_1[x] + a_2[y] + a_3[z]$$
$$+ a_4[xz] + a_5[yz] + a_6[z^2] + a_7[z^3]) = 1\ .$$

Then Theorem 10.10 implies that the reduced form of the polynomial

$$p = x(a_0 + a_1 x + a_2 y + a_3 z + a_4 xz + a_5 yz + a_6 z^2 + a_7 z^3) - 1$$

vanishes. Since we find that

$$\mathrm{Reduce}(p,\ G) = (-1 + 2a_1 + 5a_2) + (a_0 + \frac{3}{2}a_4 + \frac{5}{2}a_6)x$$

$$+ (-\frac{5}{2}a_4 - a_6)y + (a_4 + 5a_5 + \frac{3}{2}a_6)z + (-a_1 - a_7)yz$$
$$+ (a_3 + \frac{5}{2}a_7)xz + (-a_2 + \frac{3}{2}a_7)z^2 - a_5z^3 \ ,$$

we obtain the system of linear equations

$$2a_1 + 5a_2 = 1 \ , \quad a_0 + \frac{3}{2}a_4 + \frac{5}{2}a_6 = 0 \ ,$$
$$-\frac{5}{2}a_4 - a_6 = 0 \ , \ a_4 + 5a_5 + \frac{3}{2}a_6 = 0 \ ,$$
$$-a_1 - a_7 = 0 \ , \quad a_3 + \frac{5}{2}a_7 = 0 \ ,$$
$$-a_2 + \frac{3}{2}a_7 = 0 \ , \ -a_5 = 0 \ .$$

If we solve this system (e.g., by the one of the methods of Chapter 9), we find the solution

$$a_0 = a_4 = a_5 = a_6 = 0, \ a_1 = -\frac{2}{11}, \ a_2 = \frac{3}{11}, \ a_3 = -\frac{5}{11}, \ a_7 = \frac{2}{11} \ ;$$

hence

$$[x]^{-1} \ = \ \frac{2}{11}[z^3] - \frac{2}{11}[x] + \frac{3}{11}[y] - \frac{5}{11}[z] \ .$$

■

This type of construction turns out to be very useful in the next subsection.

Solution of Systems of Polynomial Equations

We now turn our attention to the more common problem of solving systems of polynomial equations. To this end, we will take a somewhat more modern approach than that of Section 9.5. Namely, we will view a set of equations over a field F

$$p_i(x_1, x_2, \ldots, x_n) \ = \ 0, \quad 1 \le i \le k \ ,$$

in terms of the ideal $<p_1, p_2, \ldots, p_k>$. It is easily established that if $<P> \ = \ <G>$, then the sets of common zeros of the sets P, $G \subset$ F[x] are identical. (Exercise 10.10.) If G is a Groebner basis for $<P>$, then one expects (by now!) to be able to obtain more information about these zeros from G than from P. This is indeed the case, as the following results of Buchberger [6] show.

Theorem 10.11. Let G be a monic Groebner basis for $<P> \ = \ <p_1, \ldots, p_k> \ \subseteq$ F[x]. Then P, viewed as a system of algebraic equations, is solvable if and only if $1 \notin G$.

Proof: It is well known from (modern) algebra (see for example Hilbert's "Nullstellensatz", in van der Waerden [33]) that P is unsolvable if and only if there exists a combination of the p_i (over F[x]) which equals a nonzero constant, say 1. Since $<P> = <G>$, this is equivalent to $1 \in <G>$. Since G is a Groebner basis, this implies that Reduce(1, G) = 0; this, in turn, means that $1 \in G$.

■

We note that a system P is unsolvable if and only if a Groebner basis for $<P>$ contains an element of headterm 1. In such a case, the reduced monic Groebner basis will simply be $\{1\}$.

Example 10.14. The reduced, monic Groebner basis (over $\mathbf{Q}[x,y]$) for the ideal $<p_1, p_2, p_3>$ where

$$p_1 = x^2y + 4y^2 - 17\ ,\ \ p_2 = 2xy - 3y^3 + 8\ ,\ \ p_3 = xy^2 - 5xy + 1\ ,$$

is $\{1\}$, irrespective of the term ordering used. Therefore, the corresponding system of algebraic equations

$$p_1 = 0\ ,\quad p_2 = 0\ ,\quad p_3 = 0$$

has no solutions.

■

Theorem 10.12. Let G be a Groebner basis for $<P> \subseteq \mathrm{F}[\mathrm{x}]$, and let H be the set

$$H = \{\mathrm{hterm}(g) \mid g \in G\}\ .$$

Then the system of equations corresponding to P has finitely many solutions if and only if for all $1 \le i \le n$, there is an $m \in \mathbf{N}$ such that $(x_i)^m \in H$.

Proof: The headterms of G have the required "separation property" iff the set U defined in Theorem 10.10 has finitely many entries; i.e., $\mathrm{F}[\mathrm{x}]/<G>$ is finite dimensional as a vector space. This, however, is true if and only if the set G (or P) has finitely many solutions. (This is plausible in view of our earlier remark on algebraic extension fields of F. However, the reader is referred to Groebner [21], or van der Waerden [33] for more details.)

■

It must be noted that these powerful results do not depend on the term ordering chosen to construct the Groebner basis. Neither do they require that the solutions themselves be produced. The latter fact may be important in practice, since the construction of solutions may (for some reason) be impractical when Algorithm 10.4 is not.

Example 10.15. We recall from Example 10.12 that the reduced Groebner basis for

$$<P> = <x^2 + yz - 2,\ y^2 + xz - 3,\ xy + z^2 - 5> \subset \mathbf{Q}[x,y,z]$$

with respect to $<_D$ has 6 polynomials with headterms

$$H \;=\; \{x^2,\, y^2,\, xy,\, xz^2,\, yz^2,\, z^4\}\,.$$

Since $1 \notin H$, the system corresponding to P is solvable; also, by Theorem 10.12, there are finitely many solutions.

■

Example 10.16. Consider the set of polynomials (and system of equations corresponding to)

$$<P> = <zx + yx - x + z^2 - 2,\; xy^2 + 2zx - 3x + z + y - 1,$$
$$2z^2 + zy^2 - 3z + 2zy + y^3 - 3y>.$$

If we order $T_{\{x,y,z\}}$ with the lexicographic order $<_L$, we may obtain a reduced Groebner basis for $<P>$,

$$\{\, xz^2 - 2x - z^4 + 4z^2 - 4,$$
$$y + z^4 + 2z^3 - 5z^2 - 3z + 5,$$
$$z^6 + 2z^5 - 7z^4 - 8z^3 + 15z^2 + 8z - 10 \,\}.$$

Again, we see that the system corresponding to P is solvable; however, in this case there are infinitely many solutions.

■

The above examples illustrate an important distinction between "total degree" and "lexicographic" Groebner bases. The degree basis shown in Example 10.12 offers no direct insight into the solutions of the system; however, a quick inspection of the lexicographic basis of Example 10.16 suggests a more powerful result. Apparently, it will be more difficult to obtain solutions from some types of Groebner bases than from others. Since the choice of term ordering affects the complexity (and practical behaviour) of Algorithm 10.4, it is well worth developing solution methods for both $<_D$ and $<_L$.

We consider first the use of the degree ordering. In Example 10.13, we exploited the fact that if a polynomial (with indeterminate coefficients) $p = \Sigma a_j t_j$ is in $<G>$, the requirement that Reduce(p, G) = 0 yields conditions on the indeterminates a_i. The difficulty lies in determining which $t_i \in T_X$ to include in the representation of p. But, if G has finitely many solutions, certain types of polynomials are guaranteed to exist. Namely, for each x_i, $1 \le i \le n$, there must exist a univariate polynomial $p_i = \Sigma a_{ij}(x_i)^j$ whose roots contain all possible values of x_i which may appear in solutions of G. For a set $P \subset F[x]$ and $\tilde{x} \in x$, the polynomial of least degree in $<P> \cap F[\tilde{x}]$ may be constructed by Algorithm 10.5 below.

Algorithm 10.5 Solution of System P in Variable $\bar{x}$

```
    procedure Solve1($P$, $\bar{x}$)
        # Given a system $P$ with finitely many solutions, find
        # the smallest polynomial containing the solutions in $\bar{x}$.
        $G \leftarrow$ Gbasis($P$)
        # Assume a polynomial of form $\Sigma a_k \bar{x}^k$;
        # then require that
        # Reduce($\Sigma a_k \bar{x}^k$, $G$) = $\Sigma a_k$Reduce($\bar{x}^k$, $G$) = 0.
        $k \leftarrow 0$
        # If $G$ does not satisfy Theorem 10.12, the following loop
        # may be infinite!
        do {
            $p_k \leftarrow$ Reduce($\bar{x}^k$, $G$)
            if $\exists\ (a_0, \ldots, a_k) \neq (0, \ldots, 0)$ such that $\sum_{j=0}^{k} a_j p_j = 0$ then
                return( $a_k^{-1} \cdot \sum_{j=0}^{k} a_j \bar{x}^j$ )
            else $k \leftarrow k + 1$  }
    end
```

The above algorithm is clearly valid for *any* admissible term ordering, although we will soon see that for $<_L$ it is unnecessary.

Example 10.17. Consider the set $P \subset Q[x,y,z]$ defined in Example 10.10, along with the corresponding total degree basis G. In order to find the polynomial $p \in\ <P> \cap\ Q[z]$ of least degree, we note that 1, z, z^2, z^3 are irreducible modulo G. Therefore, we first let $p = a_0 + a_1 z + a_2 z^2 + a_3 z^3 + a_4 z^4$, and set

$$\begin{aligned}\text{Reduce}(p,\ G) &= (a_0 - \frac{19}{2}a_4) + a_1 z + \ (a_2 + \frac{15}{2}a_4)z^2 \\ &\qquad - \frac{3}{2}a_4 yz - a_4 xz + a_3 z^3 \\ &= 0\ .\end{aligned}$$

This implies that $a_0 = \ \cdots\ = a_4 = 0$. In like manner, the polynomials of degrees 5, 6, and 7 all vanish identically. When we try $p = \sum_{k=0}^{8} a_k z^k$,

$$\begin{aligned}\text{Reduce}(p,\,G) = {} & (a_0 - \tfrac{19}{2}a_4 - \tfrac{3325}{8}a_8 - \tfrac{285}{4}a_6) + (a_1 - \tfrac{25}{2}a_5 - \tfrac{775}{8}a_7)z \\ & + (-\tfrac{109}{4}a_7 - \tfrac{11}{4}a_5)y + (\tfrac{13}{8}a_7 - \tfrac{1}{4}a_5)x \\ & + (a_2 + \tfrac{175}{4}a_6 + \tfrac{925}{4}a_8 + \tfrac{15}{2}a_4)z^2 + (-\tfrac{743}{8}a_8 - 14a_6 - \tfrac{3}{2}a_4)yz \\ & + (-\tfrac{337}{8}a_8 - \tfrac{31}{4}a_6 - a_4)xz + (a_3 + \tfrac{175}{4}a_7 + \tfrac{15}{2}a_5)z^3 \,.\end{aligned}$$

The resulting system of (linear, homogeneous) equations has the nontrivial solution

$$a_0 = \frac{361}{8}a_8,\;\; a_1 = 0,\;\; a_2 = -95a_8,\;\; a_3 = 0,$$
$$a_4 = \frac{219}{4}a_8,\;\; a_5 = 0,\;\; a_6 = -\frac{25}{2}a_8,\;\; a_7 = 0\,.$$

Without loss of generality, we choose $a_8 = 1$ to obtain

$$p = z^8 - \frac{25}{2}z^6 + \frac{219}{4}z^4 - 95z^2 + \frac{361}{8}\,.$$

∎

A complete set of n univariate polynomials obtained in the above manner constitutes a finite inclusion of the roots of the original system. As noted in Section 9.5, though, not all n-tuples so defined are roots. One can do better if one of the univariate polynomials splits into factors over F. For example, if $p_1 \in$ $<P> \cap \mathrm{F}[x_1]$ admits a factorization $p_1 = q_1^{e_1} q_2^{e_2} \cdots q_m^{e_m}$, then Gbasis($P$) may be refined to Gbasis($P \cup \{q_i\}$) with respect to each irreducible, distinct factor q_i. Thereafter, each component basis will yield different (smaller) univariate polynomials in $x_2, \ldots, x_n$. Carried even further, this approach suggests a scheme (specified by Algorithm 10.6) to explicitly determine the roots of P.

Of course, it will not always be possible to solve all univariate polynomials exactly. Moreover, the successive refinement of each Groebner basis may be impractical if complicated extensions of F are involved. (Note that F may be a rational function field!) Still, Algorithm 10.6 provides a complete solution in theory when P has finitely many solutions.

Lexicographic Bases and Elimination

We now recall from Example 10.16 that Groebner bases with respect to $<_L$ seem to provide more information, in a way, than total degree bases. So, in spite of the increased difficulty of computing such bases, their use may offer a valuable alternative to Algorithm 10.6. The basis for such a method is the following:

Theorem 10.13. Let $<_T$ be an admissible ordering on T_X which is such that $s <_T t$ whevever $s \in \mathrm{T}_{\{x_i, \ldots, x_n\}}$ and $t \in \mathrm{T}_{\{x_1, \ldots, x_{i-1}\}}$; let G be a Groebner basis over F with

```
Algorithm 10.6 Complete Solution of System $P$

  procedure GroebnerSolve($P$)

    # Given system $P \subset$ F[x] with finitely many solutions,
    # find these solutions over an "appropriate" extension of F.

    # We store partially refined bases and partial roots in $Q$.

    $Q \leftarrow \{ [P, ()] \}$
    for $k$ from $n$ by $-1$ to 1 do {
      $S \leftarrow \emptyset$

      # Refine/extend each element of $Q$ one more level.

      for $[G, (\alpha_{k+1}, \ldots, \alpha_n)] \in Q$ do {
        $\tilde{G} \leftarrow \{ g(x_1, \ldots, x_k, \alpha_{k+1}, \ldots, \alpha_n) \mid g \in G \}$
        $\tilde{G} \leftarrow$ Gbasis($\tilde{G}$)
        $p \leftarrow$ Solve1($\tilde{G}$, $x_k$)

        # The roots of $p$ in $x_k$ yield several new partial roots.

        if $p \neq 1$ then
          $S \leftarrow S \cup \{ [\tilde{G}, (\alpha, \alpha_{k+1}, \ldots, \alpha_n)] \mid p(\alpha) = 0 \}$   }
      $Q \leftarrow S$   }
    $roots \leftarrow \emptyset$
    for $[G, (\alpha_1, \ldots, \alpha_n)] \in Q$ do {
      $roots \leftarrow roots \cup \{ (\alpha_1, \ldots, \alpha_n) \}$   }
    return($roots$)
  end
```

respect to $<_T$. Then

$$<G> \cap \mathrm{F}[x_k, \ldots, x_n] \;=\; <G \cap \mathrm{F}[x_k, \ldots, x_n]> ,$$

where the ideal on the right hand side is formed in $\mathrm{F}[x_k, \ldots, x_n]$.

Proof: For convenience, we define $G^{(k)} = G \cap \mathrm{F}[x_k, \ldots, x_n]$. First, suppose that $p \in <G> \cap \mathrm{F}[x_k, \ldots, x_n]$. Since G is a Groebner basis, $p \longmapsto^{*}_{G} 0$. But since p contains only the variables $x_k, \ldots, x_n$ this means that there exist $p_i \in \mathrm{F}[x_k, \ldots, x_n]$, $g_i \in G^{(k)}$ such that $p = \sum_{i=1}^{m} p_i g_i$; this implies that $p \in <G^{(k)}>$.

We remark that if G is a Groebner basis with respect to $<_T$, then $G^{(k)}$ must also be a Groebner basis, since Theorem 10.5 (ii) requires that

$$\text{Spoly}(p, q) \ \overset{*}{\longmapsto}_{G^{(k)}} \ 0$$

for all $p, q \in G^{(k)} \subseteq G$. It follows that if $p \in <G^{(k)}>$, we also have $p \in <G> \cap$ $\mathrm{F}[x_k, \ldots, x_n]$.

■

Now, consider specifically the ordering $<_L$, which satisfies the requirements of the above result. Then Theorem 10.13 says that the polynomials in G which only depend on the last $n-k+1$ variables are a Groebner basis for the "k-th elimination ideal" of G (i.e., the subset of $<G>$ which depends only on these variables). Suppose that G is the lexicographic Groebner basis of a set $P \subset \mathrm{F}[\mathbf{x}]$ which has finitely many solutions. Then by Theorem 10.12, G must contain a single univariate polynomial in x_n; namely, the polynomial in $<P> \cap \mathrm{F}[x_n]$ of least degree. In addition, G must contain at least one polynomial in each elimination ideal in which the "highest" variable is separated.

Example 10.18. The reduced, monic Groebner basis with respect to $<_L$ for

$$<P> = <x^2 + yz - 2,\ y^2 + xz - 3,\ xy + z^2 - 5> \ \subset \ \mathrm{Q}[x, y, z]$$

is

$$\begin{aligned}
&\{ x - \frac{88}{361}z^7 + \frac{872}{361}z^5 - \frac{2690}{361}z^3 + \frac{125}{19}z\ , \\
&\ \ y + \frac{8}{361}z^7 + \frac{52}{361}z^5 - \frac{740}{361}z^3 + \frac{75}{19}z\ , \\
&\ \ z^8 - \frac{25}{2}z^6 + \frac{219}{4}z^4 - 95z^2 + \frac{361}{8} \ \} .
\end{aligned}$$

Hence, in order to solve the nonlinear system associated with P we may solve instead the reduced equations

$$\begin{aligned}
x \qquad\quad & - \frac{88}{361}z^7 + \frac{872}{361}z^5 - \frac{2690}{361}z^3 + \frac{125}{19}z = 0\ , \\
y \quad & + \frac{8}{361}z^7 + \frac{52}{361}z^5 - \frac{740}{361}z^3 + \frac{75}{19}z = 0\ , \\
& z^8 - \frac{25}{2}z^6 + \frac{219}{4}z^4 - 95z^2 + \frac{361}{8} = 0\ .
\end{aligned}$$

Note how closely this resembles a triangular linear system.

■

A lexicographic basis may, of course, contain other polynomials whose head-terms are not separated. But since each subset $G \cap \mathrm{F}[x_k, \ldots, x_n]$ is also a Groebner basis, no simpler (i.e. "more separated") basis may exist for the given permutation of variables. A (simpler) counterpart to Algorithm 10.6 appears below as Algorithm 10.7. We note that the basis refinements (i.e. the additional

Algorithm 10.7 Solution of P using Lexicographic Groebner Basis

```
procedure LexSolve($P$)
    # Assume that $P \subset F[x]$ has finitely many solutions,
    # and impose the term ordering $<_L$.

    $G \leftarrow$ Gbasis($P$) ; $roots \leftarrow \varnothing$

    # First, solve the univariate polynomial in $x_n$.

    $p \leftarrow$ selectpoly($G \cap F[x_n]$)
    $roots \leftarrow roots \cup \{ (\alpha) \mid p(\alpha) = 0 \}$

    # Now, back-solve (cf. Section 9.5).

    for $k$ from $n-1$ by $-1$ to 1 do {
      $S \leftarrow \varnothing$
      for $(\alpha_{k+1}, \ldots, \alpha_n) \in roots$ do {
        $G_k \leftarrow G \cap F[x_k, \ldots, x_n] - F[x_{k+1}, \ldots, x_n]$
        $\tilde{G} \leftarrow \{ g(x_k, \alpha_{k+1}, \ldots, \alpha_n) \mid g \in G_k \}$
        $\tilde{G} \leftarrow$ Gbasis($\tilde{G}$)
        $p \leftarrow$ selectpoly($\tilde{G} \cap F[x_k]$)
        if $p \neq 1$ then
          $S \leftarrow S \cup \{ (\alpha, \alpha_{k+1}, \ldots, \alpha_n) \mid p(\alpha) = 0 \}$   }
      $roots \leftarrow S$   }
    return($roots$)
  end
```

Groebner basis computations) in Algorithm 10.7 are univariate sub-problems, and therefore amount to GCD computations. Thus, the above process is indeed simpler than Algorithm 10.6. Still, it may not be possible to carry out in practice. We mention that, as before, the Groebner basis may be decomposed into irreducible components if any of its elements factor. (In fact, these components can be computed much more efficiently by factoring *during* Algorithm 10.4; see Czapor [17].)

It should also be pointed out that even when a given system has *infinitely* many solutions, the lexicographic Groebner basis will be as "triangular" as possible. Therefore, it is often still possible to obtain the solutions directly from the basis.

Example 10.19. In Example 10.16, we computed the lexicographic basis

$$\begin{aligned} G = \{\, & xz^2 - 2x - z^4 + 4z^2 - 4\,, \\ & y + z^4 + 2z^3 - 3z - 5z^2 + 5\,, \\ & z^6 + 2z^5 - 7z^4 - 8z^3 + 15z^2 + 8z - 10\,\}, \end{aligned}$$

over $Q[x,y,z]$. If we factor the final, univariate polynomial we obtain

$$(z^2 - 2)(z^4 + 2z^3 + 5z^2 - 4z + 5)\,.$$

The roots of the larger factor,

$$p_1(z) \;=\; z^4 + 2z^3 + 5z^2 - 4z + 5 \;=\; 0$$

may be extended using Algorithm 10.7 to four complete roots for x,y,z; however, the roots of $p_2(z) = z^2 - 2 = 0$ yield only the solutions $\{y = 1 \mp \sqrt{2},\ z = \pm\sqrt{2}\}$, in which x may take any value. Alternatively, we may refine the basis with respect to each of the univariate polynomials p_1, p_2 to obtain

$$\text{Gbasis}(G \cup \{p_1\}) = \{x - z^2 + 2,\ y + z\,,\ z^4 + 2z^3 + 5z^2 - 4z + 5\,\}\,,$$

$$\text{Gbasis}(G \cup \{p_2\}) = \{y + z - 1,\ z^2 - 2\,\}\,.$$

Note that, in the irreducible components of the lexicographic basis, it becomes clear when specific variables must be viewed as parameters in the corresponding solutions.

■

Obviously, there is a strong connection between lexicographic Groebner bases and the resultant techniques of the previous chapter. However, the Groebner basis is clearly a more powerful and elegant result than a reduced system (cf. Definition 9.1). For example, the final univariate polynomial is of minimal degree, and therefore contains no extraneous roots. Also, the degree of the polynomial which must be solved at each phase of back-solving will be no larger than the number of roots.

On the other hand, for some types of input polynomials the computation of a reduced system via resultants may be much faster. Hence, Pohst and Yun [32] proposed the combined use of resultants, pseudo-division and S-polynomial reduction. Also, the speed of either scheme (for a given problem) depends very strongly on the permutation of variables $x_1, \ldots, x_n$ used for the elimination. It is much easier to choose a good permutation for the resultant method (one variable at a time) than to choose a good permutation (*a priori*) for Algorithm 10.4, when $<_L$ is used. (It is important to note that, when a degree ordering is used, the algorithm is not nearly as sensitive to this choice; see Exercise 10.14.) Although the problem of determining the *optimal* permutation is difficult, Boege et al. [14] have proposed a simple heuristic for choosing a "reasonable" permutation. Recently, it seems that an even better solution has been proposed (in the case when the system has only finitely many solutions): namely, it is possible to first compute the total degree basis and then obtain the lexicographic one by a

✓ Roiden-Theorem
[illegible] x total-degree GB to lex. GB

"change of basis" transformation. (See Gianni et al. [18].)

10.6. ADDITIONAL APPLICATIONS

Along with the applications discussed above, Buchberger's Algorithm provides constructive solutions for a great many problems in polynomial ideal theory such as computation of Hilbert functions for polynomial ideals, free resolution of polynomial ideals and syzigies, and determination of algebra membership. However, most such topics require more algebraic background than we wish to discuss here. We consider instead two very basic problems which, perhaps surprizingly, may be solved in terms of Groebner bases.

Geometry Theorem Proving

In the past few years, the automated proving of elementary geometry theorems has become a topic of great interest in symbolic computation. This is primarily due (it seems) to the recent work of Wu [36]. We will not dwell here on the foundations of the subject; rather, we will attempt to present some of the basic ideas with emphasis on the possible role of Groebner bases. The main idea is that often a theorem (i.e., a set of hypotheses implying a conclusion), for which the geometric relationships may be expressed as polynomials, can be proven algebraically. In Wu's method, one attempts to show that the set of common zeros (in an algebraically closed field) of the hypothesis polynomials is contained in the set of zeros of the conclusion polynomial. Unfortunately, in elementary geometry one is concerned with real (rather than complex) zeros; so, the method is not *complete* in the sense that not all valid theorems may be proven. Still, Wu and also Chou [16] have succeeded in proving a large number of such theorems.

It is not surprizing that, in the above problem, Groebner basis techniques have been successfully applied. We will sketch one such approach due to Kapur Kapur Geometry Proving Nullstellensatz (A different approach presented by Kutzler and Stifter [27] appears to be faster, but less powerful.) Details of the equivalence between Wu's formulation of the problem and Kapur's (and a comparison of the various methods) are given by Kapur [25].

Let F be a field of characteristic zero, and let $\bar{\mathrm{F}}$ be an algebraically closed field containing F. Suppose we can represent hypotheses as polynomials $h_i \in \mathrm{F}[\mathrm{x}]$, the conclusion as $c \in \mathrm{F}[\mathrm{x}]$, and any subsidiary hypotheses (to be explained later) as $s_i \in \mathrm{F}[\mathrm{x}]$. Then we will consider statements of the form

$$\forall\ x_1, \ldots, x_n \in \tilde{\mathrm{F}},\ \{h_1 = 0,\ h_2 = 0,\ \ldots,\ h_k = 0,\ s_1 \neq 0,\ \ldots,\ s_l \neq 0\}$$

$$\Longrightarrow \quad c = 0\ . \tag{10.15}$$

The above statement is a theorem if the zeros in $\bar{\mathrm{F}}$ of c include the admissible common zeros of the h_i. This form is actually quite general because any (quantifier-free) formula involving boolean connectives may also be expressed as a (finite) set of polynomial equations. Namely, Kapur shows in [25] that:

$$
\begin{array}{lll}
\text{(a) } p_1 = 0 \textbf{ and } p_2 = 0 & \Longleftrightarrow & \{p_1 = 0,\ p_2 = 0\}; \\
\text{(b) } p_1 = 0 \textbf{ or } p_2 = 0 & \Longleftrightarrow & \{p_1p_2 = 0\}; \\
\text{(c) } p_1 \neq 0 & \Longleftrightarrow & \{p_1z - 1 = 0\},
\end{array}
$$

where z in the above is a new indeterminate. He then proposes the following, which (as in Theorem 10.11) is based on Hilbert's Nullstellensatz:

Theorem 10.14. ([25]) The validity of a (geometry) statement of the form (10.15) is equivalent to the validity of

$$<h_1,\ \ldots,\ h_k,\ s_1z_1 - 1,\ \ldots,\ s_lz_l - 1,\ cz - 1> \;=\; <1>\,,$$

where z, $\{z_i\}$ are additional indeterminates.

■

That is, the problem reduces to that of showing that a related system (which includes the contradiction of the conclusion) is not solvable over $\tilde{\mathrm{F}}$.

Example 10.20. Consider the problem of proving the following simple proposition: if the right bisector of the hypotenuse of a right triangle intersects the right vertex, then the triangle is isosceles.

Without loss of generality, we set up a plane coordinate system in which the right vertex is at the origin, and the triangle sits in the first quadrant. Suppose the other two vertices are at $(y_1,\ 0)$ and $(0,\ y_2)$, and the midpoint of the hypotenuse is $(y_3,\ y_4)$. Then $y_4 = y_2/2$ and $y_3 = y_1/2$ (since we have a midpoint), and $y_4/y_3 = -(-y_2/y_1)^{-1}$ (since the bisector is perpendicular to the hypotenuse). Furthermore, the triangle will be isosceles if and only if $|y_1| = |y_2|$. Since the reduced, monic Groebner basis of

$$<y_1 - 2y_3,\ \ y_2 - 2y_4,\ \ y_1y_3 - y_2y_4,\ \ (y_1^2 - y_2^2)z - 1>$$

(over $\mathrm{Q}[y_1, y_2, y_3, y_4, z]$) is $\{1\}$, the theorem is valid.

■

We must, finally, mention the role of the subsidiary hypotheses $\{s_i\}$ in the above. It may happen that some theorems may only be established in the above manner when certain degenerate cases are ruled out. For example, in Example 10.20 it might have been necessary to specify that $y_1 \neq 0 \neq y_2$; thus we would have added the polynomials

$$s_1 \;=\; y_1z_1 - 1\ ,\quad s_2 \;=\; y_2z_2 - 1$$

to the set above. An example of a case in which such extra conditions are necessary is provided in Exercise 10.20. (Methods for detecting such cases are discussed by Kapur [25] and Wu [36].)

Polynomial GCD Computation

Recently, Gianni and Trager [19] outlined how Groebner basis calculations may be used to compute multivariate GCD's. While not especially practical, such methods do serve to illustrate the significance of Groebner bases. They propose the following method:

Theorem 10.15. Let $f_1, \ldots, f_m, g \in \mathrm{F}[y, \mathrm{x}]$ be primitive with respect to y and I be a maximal ideal in F[x] (i.e., I is contained in no other ideal). Suppose that

$$<f_1, \ldots, f_m, \mathrm{I}> \;=\; <1>\,,$$

$$<\mathrm{lcoeff}_y(f_i \cdot g), \mathrm{I}> \;=\; <1> \quad \text{for some } 1 \le i \le m\ ,$$

and let $G_{(k)}$ be a reduced Groebner basis for the ideal

$$<f_1 \cdot g\,, \;\ldots, \;f_m \cdot g\,, \;\mathrm{I}^k>$$

with respect to $<_D$. Then for $k > [\deg(g)]^2$, the unique polynomial $\tilde{g}$ in $G_{(k)}$ of least total degree is an associate of g.

■

The idea in the above is to produce an ideal in which the GCD is the element of least degree. This depends, in part, on the observation that

$$<f_1, \ldots, f_m, \mathrm{I}> = <1> \;\Rightarrow\; <f_1 \cdot g, \ldots, f_m \cdot g, \mathrm{I}^k> = <g, \mathrm{I}^k>$$

for $k > 0$. (The proof of this is left as an exercise for the reader.) In practice, the ideal I is chosen to be of the form

$$\mathrm{I} \;=\; <y_1 - \alpha_1,\; y_2 - \alpha_2,\; \ldots,\; y_m - \alpha_m>$$

for $\alpha_i \in \mathrm{F}$. The reader should compare the above requirements for the $\{f_i\}$, g and I with those imposed on homomorphisms in Section 7.4.

Example 10.21. Consider the problem of finding the GCD of

$$p_1 = 2yxz - 2y^3 + 4y - 7x^2z + 7xy^2 - 14x + xz^2 - zy^2 + 2z\ ,$$

$$p_2 = 3x^2z - 3xy^2 + 6x - xz^2 + zy^2 - 2z - xz + y^2 - 2\ .$$

We first note that both of these polynomials are primitive in x. If we choose $\mathrm{I} = \;<y, z-1>$, for example, then a basis for I^5 is

$$Q = \{y^5, y^4(z-1), y^3(z-1)^2, y^2(z-1)^3, y(z-1)^4, (z-1)^5\}\ .$$

It is then easy to verify that the remaining conditions of Theorem 10.15 are met. By means of Algorithm 10.4, we may compute a reduced basis (with respect to $<_D$) for $<p_1, p_2, Q>$, namely

$$\begin{aligned}\{&yz^4 - 4yz^3 + 6yz^2 - 4yz + y,\ z^5 - 5z^4 + 10z^3 - 10z^2 + 5z - 1,\\ &xyz^2 - 2yz^3 - 2xyz + 8yz^2 + xy + 4y - 10yz,\\ &x^3 - 12xz^2 + 16z^3 + 6x^2 + 24xz + 96z - 72z^2 - 32,\\ &xz^3 - 2z^4 - 3xz^2 + 10z^3 + 3xz - x + 14z - 18z^2 - 4,\\ &x^2y - 4xyz + 4yz^2 + 8xy + 16y - 16yz,\\ &x^2z - 4xz^2 + 4z^3 - x^2 + 12xz - 8x + 32z - 20z^2 - 16,\\ &y^2 - xz - 2\}\ .\end{aligned}$$

The polynomial of least degree in this basis is

$$g = y^2 - xz - 2\ .$$

Since we have

$$p_1 = (2y - 7x + z)(xz - y^2 + 2),\ \ p_2 = (3x - z - 1)(xz - y^2 + 2)\,,$$

g is indeed the required GCD.

■

We mention that, in addition, a method is given by Gianni and Trager [19] for performing multivariate factorization.

Exercises

(Those exercises marked with an * may require a significant amount of computer time; time limits should be set at "appropriate" values.)

1. Consider arbitrary $p,\ q,\ r \in \mathrm{F}[\mathrm{x}]$ and $P \subset \mathrm{F}[\mathrm{x}]$ such that $p \longmapsto_P q$. Is it true that $p + r \longmapsto_P q + r$? Is it true that $p + r \overset{+}{\longmapsto}_P q + r$?

2. Show that if $p,\ q,\ r \in \mathrm{F}[\mathrm{x}]$ are such that $p = q \cdot r$, then $p \overset{+}{\longmapsto}_{\{q\}} 0$ for any term ordering $<_T$.

3. Formulate a strategy for the procedure selectpoly, which selects the "best" of several reducers in Algorithm 10.1 when the degree ordering is used. (Hint: show that the number of distinct n-variate terms of total degree k is
$$\binom{k+n-1}{n-1}\ ;$$
hence the number of distinct terms of degree less than or equal to d is
$$\sum_{k=0}^{d} \binom{k+n-1}{n-1} = \binom{d+n}{n}\ .)$$

4. For arbitrary $g_1, g_2, h_1, h_2 \in \mathrm{F}[\mathbf{x}]$, find $\alpha, \beta \in \mathrm{F}$ and $u \in \mathrm{T}_{\mathbf{x}}$ such that

$$\mathrm{M}(h_1)g_1 + [\,\mathrm{M}(h_2) + \alpha{\cdot}\mathrm{hterm}(h_2)\,]{\cdot}g_2 \;=\; \beta{\cdot}u{\cdot}\mathrm{Spoly}(g_1, g_2)\ .$$

5. Devise an algorithm to decide, given $P_1, P_2 \subset \mathrm{F}[\mathbf{x}]$, if $<P_1> \subseteq <P_2>$.

6. Using your favorite computer algebra system, implement Algorithm 10.2 using both term orderings $<_D$ and $<_L$ over $\mathbf{Q}[\mathbf{x}]$. (Hint: you need only construct different leading monomial functions M_D and M_L.) You should use the "first available" pair selection strategy, and make all polynomials monic as they are added to the partial basis. Test your code for $<_D$ on Examples 10.8 and 10.10, and for $<_L$ on Examples 10.16 and 10.18.

7. Implement Algorithm 10.3, and hence modify the code from Exercise 6 to yield a reduced, monic Groebner basis. Compare the results of the new code to that of the old on Examples 10.16, 10.18 using $<_L$; repeat this comparison for Example 10.8 using $<_D$. Assuming the implementation of Exercise 6 is correct, can you devise a procedure which verifies the correctness of the new code?

8. Improve the implementation of Exercise 7 by adding criterion1 and criterion2 as in Algorithm 10.4. (Hint: the efficiency of criterion2 depends on a fast means of testing if $[u, v] \in B$.) Compare this code, and that of Exercise 7, on:

   (a) the polynomials of Example 10.10, for $<_D$ and $<_L$;

   (b)* the set

$$\{x^2z + xz^2 + yz^2 - z^2 - 2z,\ zy^2 + 2xz - 3z + x + y - 1,$$
$$2x^2 + xy^2 - 3x + 2xy + y^3 - 3y\},$$

   using the same orderings on $T_{[x,y,z]}$.

9. Further (and finally!) improve the implementation of Exercise 8 by modifying the procedure selectpair to use the "normal" selection strategy (10.13). (Hint: a careful choice of data structure for the set B will help.) Carefully compare this and the code of Exercise 8 on the following:

   (a) the polynomials of Exercise 8(b), using $<_D$;

   (b) the same set of polynomials, using $<_L$;

   (c) the polynomials of Example 10.10 using $<_L$.

10. Prove that if sets $P, Q \subset F[\mathbf{x}]$ are such that $<P> = <Q>$, then the roots of P, Q are identical.

11. Implement Algorithm 10.5 in your favorite computer algebra system, i.e. whichever one was used in Exercises 6-9. (Note: in order to save some trouble, you may use whatever system routines are available for the solution of systems of linear equations.) Test your implementation by computing the polynomial $p(z)$ found in Example 10.17. Then, for the set P defined in Example 10.10, use your implementation to compute the counterparts $q(y)$, $r(x)$ to p.

12. Show that the set $\{p, q, r\}$ computed in Exercise 11 is a reduced, monic Groebner basis. (Note that this does not require computation of this set.) Why does this not contradict the uniqueness of the basis computed in Example 10.10?

13. Is it possible for a reduced ideal basis, *not* composed entirely of univariate polynomials, to be a Groebner basis with respect to more than one term ordering? Give an example (if possible) of such a set which is a Groebner basis with respect to an arbitrary admissible ordering.

14. Compute monic, reduced Groebner bases with respect to $<_L$ for the set

$$\{y_3y_4 + 19y_1y_2 + 5y_1^3 + 45,\ y_4 - 7y_3 + 9y_2 - y_1 + 44,$$

$$53y_3y_4 + 2y_1y_2 + 11y_2y_3 + 454,\ y_1y_3y_4 + 3y_1^2 - 6y_3 + 30\},$$

using the following permutations of variables:

(a) $\mathbf{x} = \{y_4, y_3, y_2, y_1\}$;

(b) $\mathbf{x} = \{y_2, y_4, y_1, y_3\}$;

(c)* $\mathbf{x} = \{y_1, y_2, y_3, y_4\}$.

Compare the times required for the above computations with those required using $<_D$. Suggest a procedure for choosing a permutation for which the lexicographic computation is relatively simple.

15. Compute the lexicographic basis of Exercise 14(c), by first computing the degree basis for the same permutation of variables. (Hint: guess the likely form for the lexicographic basis, and then use the fact that the reduced form of each polynomial must vanish.)

16. Solve the following systems of equations (or the systems corresponding to given sets of polynomials) as explicitly as possible:

(a) the set of Exercise 8(b) for x, y, z;

(b) the system of Exercise 9.X (noting that your implementation of Buchberger's algorithm may have to be modified for coefficients in $\mathbf{Q}(c)$);

(c)* the set

$$\{x_3x_4 + 19x_1x_2 + 5x_1^2 + 45,\ 33x_3x_4 + 2x_2^2 + 11x_2x_3 + 454,$$
$$x_2x_3^2 - 2x_3x_4 - 2x_1x_4 - 14,\ x_4 - 7x_3 + 9x_2 - x_1 + 44,\ x_1x_3x_4 - 6x_3 + 30\},$$

for x_1, x_2, x_3, x_4.

17. Prove that an ordering $<_M$ on $\mathrm{T_X}$ defined for $1 \le m \le n$ by

$$s = x_1^{i_1} \cdots x_n^{i_n} <_M x_1^{j_1} \cdots x_n^{j_n} = t \quad \Leftrightarrow$$
$$\{\exists\, l,\ 1 \le l \le m \text{ such that } i_l < j_l \text{ and } i_k = j_k,\ 1 \le k < l\}, \text{ or}$$
$$\{i_k = j_k,\ 1 \le k \le m\ , \text{ and } x_{m+1}^{i_{m+1}} \cdots x_n^{i_n} <_D x_{m+1}^{j_{m+1}} \cdots x_n^{j_n}\}$$

is admissible according to Definition 10.1. Suggest two possible uses for such an ordering.

18. Implement (i.e. modify one of your previous implementations of) Buchberger's algorithm for Groebner bases over $\mathbf{Z}_p$, where p is prime. Compute Groebner bases over $\mathbf{Z}_{17}[x,y,z]$ with respect to $<_D$ for the following:

(a) $\{x^4y + 2x^3y + 5x^3,\ x^2y^2 - 3xy,\ xy^4 + xy^2\}$;

(b) $\{2xy^2 + 7y^2 + 9x + 2,\ xy + 4x - 5y + 11\}$.

Compare both results above to the corresponding bases over $\mathbf{Q}[x,y,z]$. What conclusions may be drawn?

19. Devise a modification of Buchberger's algorithm which, given $p \in \langle f_1, \ldots, f_m\rangle \subseteq \mathrm{F}[\mathrm{x}]$, finds $a_i \in \mathrm{F}[\mathrm{x}]$ such that $p = \sum_{i=1}^{m} a_i f_i$. Implement your scheme, and use it to find a_1, a_2, a_3 such that

$$a_1\cdot(x^3yz - xz^2) + a_2\cdot(xy^2z - xyz) + a_3\cdot(x^2y^2 - z^2) \;=\; xz^4 - xyz^3\ .$$

20. Use Groebner bases to prove that a parallelogram is a square iff its diagonals are perpendicular and equal in length. (Hint: does the problem make sense if your "arbitrary points" do not really define a parallelogram?)

References

1. D. Armbruster, "Bifurcation Theory and Computer Algebra: An Initial Approach," *Proc. EUROCAL 85, Vol. 2, Lecture Notes in Computer Science,* **204** pp. 126-137 Springer-Verlag, (1985).
2. L. Bachmair and B. Buchberger, "A Simplified Proof of the Characterization Theorem for Gröbner Bases," *ACM SIGSAM Bull.*, **14**(4) pp. 29-34 (1980).
3. A.M. Ballantyne and D.S. Lankford, "New Decision Algorithms for Finitely Presented Commutative Semigroups," *Comp. Math. Appl.*, **7** pp. 159-165 (1981).
4. L.J. Billera and L.L. Rose, "Gröbner Basis Methods for Multivariate Splines," RRR # 1-89, Rutgers Univ. Department of Mathematics and Center for Operations Research (1989).
5. B. Buchberger, "An Algorithm for Finding a Basis for the Residue Class Ring of a Zero-Dimensional Polynomial Ideal," Ph.D. Thesis, Univ. of Innsbruck, Math. Inst. (1965).
6. B. Buchberger, "An Algorithmical Criterion for the Solvability of Algebraic Systems of Equations," *Aequationes math.*, **4**(3) pp. 374-383 (1970).
7. B. Buchberger, "Some Properties of Gröbner-Bases for Polynomial Ideals," *ACM SIGSAM Bull.*, **10**(4) pp. 19-24 (1976).
8. B. Buchberger, "A Theoretical Basis for the Reduction of Polynomials to Canonical Forms," *ACM SIGSAM Bull.*, **10**(3) pp. 19-29 (1976).
9. B. Buchberger, "A Criterion for Detecting Unnecessary Reductions in the Construction of Gröbner Bases," *Proc. EUROSAM 79, Lecture Notes in Computer Science*, **72** pp. 3-21 Springer-Verlag, (1979).
10. B. Buchberger and F. Winkler, "Miscellaneous Results on the Construction of Gröbner Bases for Polynomial Ideals," Tech. Rep. 137, Univ. of Linz, Math. Inst. (1979).
11. B. Buchberger, "A Note on the Complexity of Constructing Gröbner Bases," *Proc. EUROCAL 83, Lecture Notes in Computer Science*, **162** pp. 137-145 Springer-Verlag, (1983).
12. B. Buchberger and R. Loos, "Algebraic Simplification," pp. 11-43 in *Computer Algebra - Symbolic and Algebraic Computation, 2nd ed.*, ed. B. Buchberger, G. Collins, R. Loos, Springer-Verlag, Wein - New York (1983).
13. B. Buchberger, "Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory," pp. 184-232 in *Progress, directions and open problems in multidimensional systems theory*, ed. N.K. Bose, D. Reidel Publishing Co. (1985).

14. W. Böge, R. Gebauer, and H. Kredel, "Some Examples for Solving Systems of Algebraic Equations by Calculating Gröbner Bases," *J. Symbolic Comp.*, 2(1) pp. 83-98 (1986).

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

16. S.C. Chou, "Proving Elementary Geometry Theorems Using Wu's Algorithm," *Contemporary Math.*, **29** pp. 243-286 (1984).

17. S.R. Czapor, "Gröbner Basis Methods for Solving Algebraic Equations," Ph.D. Thesis, University of Waterloo, Dept. of Applied Math. (1988).

18. P. Gianni, B. Trager, and D. Lazard, *UNKNOWN.*

19. P. Gianni and B. Trager, "GCD's and Factoring Multivariate Polynomials Using Gröbner Bases," *Proc. EUROCAL 85, Vol. 2, Springer Lecture Notes in Computer Science*, **204** pp. 409-410 (1985).

20. M. Giusti, "A Note on the Complexity of Constructing Standard Bases," *Proc. EUROCAL 85, Vol. 2, Springer Lecture Notes in Computer Science*, **204** pp. 411-412 (1985).

21. W. Gröbner, *Modern Algebraic Geometry (German)*, Springer-Verlag, Wien-Innsbruck (1949).

22. G. Hermann, "The Question of Finitely Many Steps in Polynomial Ideal Theory (German)," *Math. Ann.*, 95 pp. 736-788 (1926).

23. H. Hironaka, "Resolution of Singularities of an Algebraic Variety over a Field of Characteristic Zero I, II," *Ann. Math.*, 79 pp. 109-326 (1964).

24. A. Kandri-Rody and D. Kapur, "Algorithms for Computing Gröbner Bases of Polynomial Ideals over Various Euclidean Rings," *Proc. EUROSAM 84, Springer Lecture Notes in Computer Science*, **174** pp. 195-206 (1984).

25. D. Kapur, "Geometry Theorem Proving Using Hilbert's Nullstellensatz," *Proc. SYMSAC '86*, pp. 202-208 (1986).

26. D.E. Knuth and P.B. Bendix, "Simple Word Problems in Universal Algebras," pp. 263-298 in *Proc. of the Conf. on Computational Problems in Abstract Algebra (OXFORD '67)*, ed. J. Leech, Pergamon Press, Oxford (1970).

27. B. Kutzler and S. Stifter, "Automated Geometry Theorem Proving Using Buchberger's Algorithm," *Proc. SYMSAC 86*, pp. 209-214 (1986).

28. D. Lazard, "Gröbner Bases, Gaussian Elimination, and Resolution of Systems of Algebraic Equations," *Proc. EUROCAL 83, Springer Lecture Notes in Computer Science*, **162** pp. 146-156 (1983).

29. D. Lazard, "Ideal Bases and Primary Decomposition: Case of Two Variables," *J. Symbolic Comp.*, 1(3) pp. 261-270 (1985).

30. E. Mayr and A. Meyer, "The Complexity of the Word Problems for Commutative Semigroups and Polynomial Ideals," Report LCS/TM-199, M.I.T. Lab. of Computer Science (1981).

31. H.M. Möller and F. Mora, "Upper and Lower Bounds for the Degree of Gröbner Bases," *Proc. EUROSAM 84, Springer Lecture Notes in Computer Science*, **174** pp. 172-183 (1984).

32. M.E. Pohst and D.Y.Y. Yun, "On Solving Systems of Algebraic Equations via Ideal Bases and Elimination Theory," *Proc. SYMSAC 81*, pp. 206-211 (1981).

33. B.L. van der Waerden, *Modern Algebra (Vols. I and II)*, Ungar (1970).

34. F. Winkler, "Reducing the Complexity of the Knuth-Bendix Completion Algorithm: A Unification of Different Approaches," *Proc. EUROCAL 85, Vol. 2, Springer Lecture Notes in Computer Science*, **204** pp. 378-389 (1985).

35. F. Winkler, "A p-adic Approach to the Computation of Gröbner Bases," *J. Symbolic Comp.*, **6** pp. 287-304 (1988).

36. W. Wu, "Basic Principles of Mechanical Theorem Proving in Elementary Geometries," *J. Syst. Sci. and Math. Sci.*, **4**(3) pp. 207-235 (1984).