(Extended Abstract)

Complexity of Buchberger's Algorithm for Gröbner Bases

T. Dubé *B. Mishra* *Chee-Keng Yap*[1]

Courant Institute of Mathematical Sciences
New York University
251 Mercer Street
New York, NY 10012.

July 6, 1995

[1]Supported in part by NSF grants #DCR-84-01898 and #DCR-84-01633.

Abstract

In computational algebraic geometry, Buchberger's algorithm for constructing a Gröbner basis has been recognized as a fundamental tool. Despite great interest in analyzing the complexity of this algorithm, relatively little progress has been made. We undertake a systematic study of his algorithm. The concepts of admissible orderings and normal form algorithm are basic in Buchberger's algorithm. We present a new constructive and elementary proof of Robbiano's characterization theorem for admissible orderings. Using this characterization, we give a bound on the complexity of the normal form algorithm for arbitrary admissible orderings. Next we obtain for the first time a complexity analysis for the overall algorithm, showing that for each fixed number of variables, the complexity is in the finite Wainer hierarchy, and hence primitive recursive. Again, these results hold for arbitrary admissible ordering.

1 Introduction

Many application areas such as robotics and solid modeling have a strong computational geometry component. Computational geometry in higher dimensions eventually turns into problems of algebraic geometry (root finding, intersection of algebraic curves and surfaces, cell decompositions). Gröbner basis has become increasingly recognized as an important algorithmic tool in computational algebraic geometry [Buchberger 1985]. There is now a considerable literature on the subject and it is rapidly growing. Much of the pioneering work is due to Buchberger. In particular, Buchberger gave an algorithm for constructing a Gröbner basis and this algorithm is the focus of the current paper. It turns out that the concept of Gröbner basis was independently proposed by Hironaka in his famous dissertation on resolution of singularities [Hironaka 1964]. Unfortunately, he only indicated existence of such bases; it was Buchberger who actually provided such an algorithm in his dissertation under Gröbner. It seems that some form of the algorithm was also discovered by David Spear [Spear 1977]. See [Mishra and Yap 1986] for a self-contained introduction to the subject.

Before going into the algorithm itself, we make some connections to three algorithms familiar to complexity theorists. This also indicates the centrality of this algorithm.

- If we specialize the algorithm to the one variable case, we have essentially Euclid's algorithm for computing the GCD of several univariate polynomials. This in itself is arguably the most intensely studied problem in symbolic computation (see [Buchberger, Collins and Loos 1982]).

- If we restrict each variable to be linear, then we have essentially the Gaussian elimination algorithm. See [Bayer 1982],[Lazard 1983].

- If we generalize the algorithm to general term rewriting system, then the algorithm is seen to be a version of the Knuth-Bendix completion algorithm [Buchberger, Collins and Loos 1982].

However, one must not conclude that Buchberger's algorithm is thereby understood. The reason is that the domain of polynomials has considerably more special structure than the general Knuth-Bendix setting; yet it is also a considerably more general setting than the linear case of Gaussian elimination or the univariate case of Euclidean algorithm.

In Gröbner basis theory, we are interested in ideals I in polynomial ring $R = K[x_1, \ldots, x_n]$ for some field K. Hilbert's basis theorem says we may assume that ideals are represented by some finite set of polynomials that generate it. The power of Gröbner basis lies in the widespread applications: basically, one may say that most computational questions involving polynomial ideals can be reduced to the computation of the Gröbner basis of ideals. For instance, the 'fundamental problem in ideal theory' (Van der Waerden) asks: "given a polynomial $f \in R$ and an ideal $I \subseteq R$, is f in I?" It turns out that the problem is easily solved once we have a Gröbner basis of I. On the other hand, Mayr and Meyer [Mayr and Meyer 1982] proved this problem is exponential-space hard. It follows that the construction of Gröbner basis has this lower bound on its complexity. Another important application of Gröbner basis is in the solution of a system of algebraic equations. It can be used in factorizing multivariate polynomials and in performing Hensel lifting [Gianni and Trager 1985]. The subject is also intimately connected with differential operators [Galligo 1985]. We refer to [Buchberger 1985] for a long and growing list of applications.

One of the central open problems in this subject is the complexity of the algorithm of Buchberger. Although the 'basic' algorithm of Buchberger has an extremely simple structure, its analyses has proven to be remarkably hard. Several variations of this basic algorithm has appeared (e.g. [Spear 1977],[Buchberger 1979],[Gebauer and Möller 1986], [Pohst and Yun 1981]). Although they appear to improve the worst-case complexity, no conclusive analyses have been done. The basic complexity behavior of the algorithm and its variations have only been analyzed for special cases. We now review what is known:

- It is known that the normal form algorithm and Buchberger's algorithm terminates. The proof, among other things, uses a result known as Dixon's lemma ([Dixon 1913][Mishra and Yap 1986]).

- For two variables, Buchberger has shown a polynomial time bound on the algorithm ([Buchberger 1979],[Buchberger 1983]). [Giusti 1985] has further improved Buchberger's bounds. For three variables, Winkler shows a single exponential upper bound [Winkler 1984]. Both these results depend crucially on an improved version of the basic algorithm.

- Assuming a uniform probability distribution, Lazard has shown that with probability 1, the degree of the polynomials in a Gröbner basis is linear in the input degrees [Lazard 1983]. This implies a polynomial time complexity. However, the use of probability distributions seems to be unrealistic, since ideals of interest tend to be highly structured (cf. the analogous situation with the Satisfiability problem which also has a small complexity assuming some simple probability distribution.)

- Bayer's thesis [Bayer 1982] has a chapter discussing complexity. He indicated some polynomial bounds for special ideals. These restrictions are technical and they are unsatisfactory from a complexity theoretic viewpoint: one cannot translate these restrictions into something easily verified. Möller and Mora [Möller and Mora 1984] have extended the results of Bayer but again they are unable to remove ideal-theoretic technical restrictions (e.g. the ideal has a known H-basis).

Buchberger's algorithm is actually a family of algorithms, depending on the choice of a suitable total ordering on its power products. These orderings, known as 'admissible orderings', are defined below. It suffices to say here that the two obvious orderings, 'lexicographical' and 'total degree' orderings are admissible. It is known that different orderings profoundly affect the complexity behavior of the algorithms (e.g. [Buchberger 1985] p.228). Furthermore, for some problems (e.g. solving systems of algebraic equations) there are advantages to one ordering over another. With this in mind, we note another serious gap in the cited results: most of them assume a total-degree orderings.

Overview:
The rest of the paper is organized as follows: After establishing some terms in section 2, we present three sets of results. Section 3 gives a new constructive proof of Robbiano's characterization of admissible orderings. Using this new characterization, section 4 gives for the first time bounds on normal form algorithm for any admissible ordering. We prove a lower bound that essentially matches the upper bound. Next, we show that a variant of the normal form algorithm is exponentially more efficient. Section 5 presents the first known upper bound on the complexity of Buchberger's algorithm. The bound lies in the Wainer's hierarchy for each fixed number of variables and hence, it is primitive recursive.

2 Preliminary

Since this area is relatively new to computer science, we briefly define the terms we use. Through out the paper, n refers to the number of variables and *is assumed to be fixed.* Also $R = K[x_1, x_2, \ldots, x_n]$, for some field K.

The fundamental concept here is the 'reduction' operation on polynomials. In order to introduce this, we first let $\mathrm{PP} = \mathrm{PP}(x_1, \ldots, x_n)$ be the set of all *power products* (or *terms*)

$$\prod_{i=1}^{n} x_i^{e_i}$$

where $e_i \geq 0$ are natural numbers. The *exponent of a power product* $M = \prod_{i=1}^{n} x_i^{e_i}$ is $(e_1, e_2, \ldots, e_n) \in \mathbf{N}^n$, and is denoted by $\exp(M)$. A total ordering $\underset{\mathrm{A}}{>}$ on the set PP is said to be *admissible* if the following two axioms are satisfied.

1. $p \underset{\mathrm{A}}{>} 1$ for all $p (\neq 1) \in \mathrm{PP}$
2. $p \underset{\mathrm{A}}{>} q \implies rp \underset{\mathrm{A}}{>} rq$ for all $p, q, r \in \mathrm{PP}$

Relative to such an ordering $\underset{\mathrm{A}}{>}$, we may define the *head monomial*, $\mathrm{Hmono}(f)$, of any polynomial $f \in R$ to be that monomial in f whose power product is the greatest under $\underset{\mathrm{A}}{>}$.

Given a polynomial $f \in R$ and a set of polynomials $G \subseteq R$, we say f is *reducible* by G, if there exists a polynomial $g \in G$ such that $\mathrm{Hmono}(g)$ divides some monomial m of f. Let $m = c \cdot \mathrm{Hmono}(g)$ for some monomial c. We say the polynomial $h = f - c \cdot g$ is the *reduct* of f by the set G and denote the relationship by

$$f \xrightarrow{G} h.$$

We denote the reflexive transitive closure of $\xrightarrow{G}$ by $\xrightarrow[*]{G}$. We say h is a G-*normal form* of f, if $f \xrightarrow[*]{G} h$ but h is not reducible by G. We write $\mathrm{NF}_G(f)$ for the set of all G-normal forms of f.

If the set G is finite then the G-normal form of any polynomial $f \in R$ can be derived by the following (trivial) non-deterministic algorithm: repeatedly apply the reduction step $\xrightarrow{G}$ to f and its reducts until a normal form of f is reached. This algorithm is referred to as the *normal form algorithm*. We show below that the process halts regardless of the choice of reduction—hence $|\mathrm{NF}_G(f)| \geq 1$ for any f.

In general, a G-normal form of f is not unique, and the central idea in Gröbner basis is to enlarge G so that it becomes unique: A finite set $G \subseteq R$ is said to be a *Gröbner basis* of the ideal generated by G, if the G-normal form of every polynomial $f \in R$ is unique, i.e., $|\mathrm{NF}_G(f)| = 1$. Given a finite set of generators $F \subseteq R$ for the ideal (F), Buchberger presented an algorithm to compute the Gröbner basis G of (F); the normal form algorithm is a basic step in Buchberger's algorithm.

3 Results I: Characterization of Admissible Orderings

Our first goal is to obtain bounds for the normal form algorithm for any choice of admissible ordering. In order to do this, we need to understand all admissible orderings. We will develop an elementary and constructive proof of a characterization theorem for all admissible orderings. The characterization was first given in [Robbiano 1985] and [Robbiano 1986] but his proof is highly non-constructive.

Theorem 1 [Robbiano] *Every admissible ordering* $\underset{A}{<}$ *on* PP *can be characterized by* s $(1 \leq s \leq n)$ *orthogonal real vectors* $W_1, W_2, \ldots, W_s \in \mathbf{R}^n$ *such that the map* $\alpha\colon (\mathrm{PP}, \underset{A}{<}) \to (\mathbf{R}^s, \underset{LEX}{<})$*, defined by*

$$\alpha(M) \;=\; (W_1 \cdot \exp(M), W_2 \cdot exp(M), \ldots, W_s \cdot \exp(M)),$$

is an injective ordered homomorphism.

Here $\underset{LEX}{<}$ is the 'usual' lexicographical ordering, and $W_i \cdot \exp(M)$ is the scalar dot product. By an abuse of notation, we will write $W_i(M)$ to mean $W_i \cdot \exp(M)$, and refer to W_i as the i^{th} *weight function*. Robbiano's theorem says that for every admissible ordering $\underset{A}{\leq}$, there is a set of weight functions W_1, W_2, …, W_s that satisfy the following condition: for all $M, M' \in PP$,

$$M \underset{A}{<} M' \Longleftrightarrow (W_1(M), \ldots, W_s(M)) \underset{LEX}{<} (W_1(M'), \ldots, W_s(M'))$$

In general, these weights can be negative although one bonus of our explicit construction is to show that the numbers can be chosen to be non-negative. Such nonnegative weights play an important role in our derivation of the complexity of the normal form algorithm.

We construct two sets of weight functions W_i and U_i, in n stages; k^{th} stage defines W_k and U_k explicitly. The weights U_i's are constructed to have only nonnegative elements.

For any admissible ordering, there is a unique extension to the set PP(x_1, …, x_n; $\mathbf{Q}$) in which the exponents are allowed to be rationals.

Stage 1:

Define z_1 to be x_j where for all x_i, $x_j \underset{A}{\geq} x_i$.

Choose $W_{1,i}$ to be $\sup\left\{s \in \mathbf{Q} \mid \ x_i \underset{A}{\geq} z_1^s\right\}$.

Set $\begin{cases} \mathbf{X_1} &= \{x_1, \ldots, x_n\} - \{z_1\} \\ \mathbf{Z_1} &= \{z_1\} \end{cases}$

Choose $U_1 = W_1$.

Stage k ($k = 2$ to n):

Assume that W_j and U_j have been constructed for $j = 1, \ldots, k-1$.

Case 1:

There exists $x_i \in \mathbf{X_{k-1}}$ such that for all $\alpha, \beta_1, \ldots, \beta_{k-1}$

$$\alpha \neq 0 \implies (\exists_{j<k}) W_j\left(z_1^{\beta_1} z_2^{\beta_2} \ldots z_{k-1}^{\beta_{k-1}} x_i^{\alpha}\right) \neq 0$$

Choose z_k to be any such x_i (for example the least).

For every j, set $W_{k,j} = 0$.

Case 2:

For every $x_i \in \mathbf{X_{k-1}}$, there exists $\beta_1, \ldots, \beta_{k-1}$ and a non-zero α such that for all $j < k$,

$$W_j\left(z_1^{\beta_1} z_2^{\beta_2} \ldots z_{k-1}^{\beta_{k-1}} x_i^{\alpha}\right) = 0$$

Define $\overline{M}_k(M)$, $\gamma_{k,i}$, and $\hat{x}_i$ as follows:

$\overline{M}_k(M) = z_1^{\beta_1} z_2^{\beta_2} \ldots z_{k-1}^{\beta_{k-1}} M$ where $(\forall_{j<k}) W_j\left(\overline{M}_k(M)\right) = 0$

$$\gamma_{k,i} = \begin{cases} 1 & \text{if } \overline{M}_k(x_i) \underset{A}{\geq} 1 \\ -1 & \text{otherwise} \end{cases}$$

$\hat{x}_i \ = x_i^{\gamma_{k,i}}$
Choose z_k to be the $x_j \in \mathbf{X_{k-1}}$ such that for all x_i, $\overline{M}_k\,(\hat{x}_j) \underset{A}{\geq} \overline{M}_k\,(\hat{x}_i)$
and let $\hat{z}_k = \hat{x}_j$ where $z_k = x_j$

Then choose, $W_{k,i} = \begin{cases} 0 & if\ \ x_i \notin \mathbf{X_{k-1}} \\ \gamma_{k,i} \sup \left\{ s \in \mathbf{Q} \ : \ \overline{M}_k\,(\hat{x}_i) \underset{A}{\geq} \overline{M}_k\,(\hat{z}_k^s) \right\} \end{cases}$

(Either case:)

Set $\begin{cases} \mathbf{X_k} &= \mathbf{X_{k-1}} - \{z_k\} \\ \mathbf{Z_k} &= \mathbf{Z_{k-1}} \cup \{z_k\} \end{cases}$

Choose $U_k = W_k + U_{k-1} * \max\left\{ \frac{-W_{k,i}}{U_{k-1,i}} \middle| W_{k,i} < 0 \right\}$.

Theorem 2 *Every admissible ordering $\underset{A}{<}$ on* PP *can be characterized by n vectors U_1, U_2, …, $U_n \in \mathbf{R}^n$ such that*

1. *For all i, $j \in \{1, \ldots, n\}$, $U_i(j) \geq 0$, and*
2. *For all $M, M' \in PP$,*

$$M \underset{A}{<} M' \Longleftrightarrow (U_1(M), \ldots, U_n(M)) \underset{LEX}{<} (U_1(M'), \ldots, U_n(M')).$$

4 Results II: Bounds on normal form algorithms

Our next set of results concern bounds for the normal form algorithm. Buchberger has shown that the normal form algorithm terminates if the admissible ordering of choice is the 'total degree' ordering. We now show that the normal form algorithm terminates for arbitrary admissible orderings.

Lemma 3

1. *Every admissible ordering $\underset{A}{\leq}$ on* PP *is a well-ordering.*
2. *We define $\underset{A}{\leq}'$ to be the following total ordering on the polynomials induced by $\underset{A}{\leq}$: Let $f = f_1 + f_2 + \cdots + f_s$ and $g = g_1 + g_2 + \cdots + g_t$ be two polynomials, each presented with its monomials in decreasing order (with respect to $\underset{A}{\leq}$). Then $f \underset{A}{\leq}' g$, if*

(a) *either for some* $i < \min(s,t)$, $\mathrm{Hterm}(f_1) = \mathrm{Hterm}(g_1)$, ..., $\mathrm{Hterm}(f_i) = \mathrm{Hterm}(g_i)$ *and* $\mathrm{Hterm}(f_{i+1}) \underset{A}{<} \mathrm{Hterm}(g_{i+1})$,

(b) *or else the sequence* $(\mathrm{Hterm}(f_1), \ldots, \mathrm{Hterm}(f_s))$ *is a prefix of the sequence* $(\mathrm{Hterm}(g_1), \ldots, \mathrm{Hterm})(g_t))$ *(thus* $s \le t$*).*

The induced ordering $\underset{A}{\le'}$ *is a well-ordering.*

The first part of the above lemma follows from Dixon's Lemma (see [Dixon 1913]). Using the above lemma it can be shown that

Theorem 4 *Let* $f \in R$ *be a polynomial and* $G \subseteq R$*, a finite set of polynomials. If* $f \xrightarrow{G} h$ *then* $f \underset{A}{>'} h$*. Hence, the normal form algorithm terminates.*

Although the above theorem shows termination of the normal form algorithm for any admissible ordering, it provides no bound for the complexity of the algorithm. Previously, the only bounds known are for the simple case where $\underset{A}{>}$ is the total degree ordering. Using the constructive version of the characterization theorem for the admissible ordering (see previous section), we are able to show the following:

Theorem 5 *Let* $\underset{A}{>}$ *be any admissible ordering,* $f \in R$ *a polynomial and* $G \subseteq R$ *a finite set of polynomials. Then the length of any sequence of reductions* $f \xrightarrow[*]{G} h$ *is at most*

$$\bar{W}_G(f) \le \begin{cases} L & \ell = 1 \\ L(1 + rU) & \ell = 2 \\ L\, 2^{rU} & \ell \ge 3 \end{cases}$$

where

L *is the length of* f *(i.e. the number of monomials in* f*)*

ℓ *is the maximum length of a polynomial in* G

r *is a constant depending only on the admissible ordering* $\underset{A}{>}$ *and* G

U *is the maximum of the weights* $U_k(f_i)$ *where* f_i *is a monomial of* f*.*

Corollary *Assuming that the admissible ordering $\underset{A}{>}$ and the set of polynomials G are fixed, the number of reduction steps of the normal form algorithm is bounded from above by*

$$L\ O(1)^{\deg(f)},$$

where $\deg(f)$ *is the total degree of f, L is the length of f.*

We also show that the upper bound of the above theorem is essentially tight:

Theorem 6 *Let $n \geq 1$, $\ell \geq 2$, $d \geq \ell - 2$, $L \geq 1$ and $D > L$ be a set of arbitrary numbers. Then it is possible to construct a set of $n + \ell - 1$ polynomials G and a polynomial f such that*

ℓ is the maximum length of any polynomial in G

d is the maximum degree of any polynomial of G in any variable x_i

L is the length of f

$D + L$ is the total degree of f

and the number of reduction steps required for f with respect to G under 'lexicographic' ordering is at least

$$L\ 2^{\left(\frac{l-2}{l-1}\right)(d+1)^{n-1}D}.$$

The above bounds, however, can be improved for $\ell \geq 3$ under the assumption that the normal form algorithm always chooses to eliminate the $\underset{A}{>}$-largest monomial that could be eliminated. We call such a reduction sequence, an *ordered reduction*.

Theorem 7 *For any admissible ordering $\underset{A}{>}$, the length of any sequence of ordered reductions beginning from an input polynomial f is at most $O(U^n)$, where U is the maximum of the weights*

$$\{U_k(f_i)\ :\ f_i \textit{ is a monomial of } f\}.$$

5 Results III: Bounds on Buchberger's Algorithm

In this section, we provide a simple though crude upper bound on the complexity of Buchberger's algorithm. For simplicity, we restrict ourselves to the 'total degree' ordering; the same proof techniques generalize to any admissible orderings.

Our bound is primitive recursive for each fixed number of variables, and is somewhat unexpected, since an initial impression is that this complexity is at least Ackerman or general recursive. Furthermore, our bound may very well be tight without additional assumptions about the algorithm. In fact if n is the number of variables, then the bound is majorized by the $n+1^{\text{st}}$ function in the Wainer hierarchy [Wainer 1970].

Let $X = \{x_1, \ldots, x_n\}$ and $Y \subseteq X$. For any $M \in \mathrm{PP}$, let $\Pi_Y(M) \in \mathrm{PP}(Y)$ be the projection of M onto $\mathrm{PP}(Y)$. In the following, we implicitly assume $Y \neq \emptyset$ whenever we use the notation $\Pi_Y(M)$.

Fix a finite subset $\Sigma \in \mathrm{PP}$. Let $\Sigma_* = \{MM' : M \in \mathrm{PP}, M' \in \Sigma\}$, the set of multiples of terms in Σ. We now define

$$\begin{aligned} T_Y = T_Y(\Sigma) \;\; = \;\; & \{M \in \mathrm{PP}(Y) : (\forall M' \in \mathrm{PP}(X-Y))[MM' \notin \Sigma_*] \\ & \wedge (\forall Z \subset Y)[\Pi_Z(M) \notin T_Z]\} \end{aligned}$$

Note that in this definition, if $Y = X$ then $\mathrm{PP}(X-Y)$ is defined to be the set $\{1\}$. For $i = 1, \ldots, n$, we define

$$E_i = E_i(\Sigma) = \sum\{|T_Y| : Y \subseteq X, |Y| = i\},$$

and

$$D = D(\Sigma) = max\{deg_x(LCM(\Sigma)) : x \in X\},$$

the maximum degree of any variable in Σ.

Lemma 8 *E_i is bounded by $C_i(D)$, where $C_i(D)$ is defined as*

$$C_i(D) = \binom{n}{i} (D-1)^i$$

We now derive a primitive recursive bound on the number of iterations of the main while-loop in Buchberger's basic algorithm. At any stage of the algorithm, let the set of head terms of polynomials in G be Σ. Then there is the associated set of complexity parameters

$$D = D(\Sigma), E_1 = E_1(\Sigma), \ldots, E_n = E_n(\Sigma).$$

Suppose that at the next stage, Σ becomes $\Sigma' = \Sigma \cup \{M\}$, with the addition of the head term M of the most recently normalized S-polynomial. Let the corresponding set of complexity parameters be

$$D' = D(\Sigma'), E_1' = E_1(\Sigma'), \ldots, E_n' = E_n(\Sigma').$$

We note that $D' \leq 2D$ since M is obtained as the product of two terms in the polynomials in G.

Lemma 9 *Let the function $f_n(D, E_1, \ldots, E_n)$ be as follows:*

1. $f_n(D, E_1, \ldots, E_n) = 0$ *if* $E_1 = \cdots = E_n = 0$.
2. *Otherwise,*

$$\begin{aligned} &f_n(D, E_1, \ldots, E_n) = 1+ \\ &\quad \max\{f_n(2D, E_1, \ldots, E_{i-1}, E_i - 1, C_{i+1}(2D), \ldots, C_n(2D)) \ : \\ &\qquad\qquad E_i > 0 \wedge i = 1, \ldots, n\}. \end{aligned}$$

Then $f_n(D(\Sigma), E_1(\Sigma), \ldots, E_n(\Sigma))$ is a bound on the number of iterations of Buchberger's basic algorithm where Σ is the set of head terms of the input polynomial set.

We now define a slight variant of the Wainer hierarchy of functions mapping from $\mathbf{N}$ to $\mathbf{N}$:

$$\begin{aligned} W_0(N) &= N + 1; \\ W_n(N) &= W_{n-1}^{(N+1)}(N), \quad \text{if } n \text{ is a successor ordinal;} \\ &\qquad (W^{(m)} \text{ is the } m\text{-fold application of } W.) \\ W_\lambda(N) &= \max_{j \leq N} \left(W_{\{\lambda\}_j}(N) \right), \quad \text{if } \lambda \text{ is a limit ordinal.} \\ &\qquad (\text{Here } \{\lambda\}_j \text{ is the } j^{\text{th}} \text{ element in a canonical fundamental} \\ &\qquad\qquad \text{sequence converging to } \lambda.) \end{aligned}$$

Lemma 10 *The function* $f_n(D, E_1, \ldots, E_n)$ *is primitive recursive for each* n*. The function* f_n *is monotonic, increasing in each variable and is majorized by the* $n+1^{\text{st}}$ *function in the Wainer hierarchy, i.e.,*

$$f_n(N, N, \ldots, N) \leq W_{n+1}(n(N+4)), \quad (\textit{almost everywhere.})$$

We define an *elementary operation* to be one of the following.

1. A field operation over the underlying field K
2. A test for zero of a field element
3. The sum, difference or product of two monomials
4. A test for divisibility of one monomial by another

The *cost* of the (1) and (2) is constant and the *cost* of (3) and (4) is $O(n \log d)$ where n is the number of variables involved and d is the total degree of the product of the two monomials. For example, the cost of forming the product of two polynomials is determined by reduction to these elementary operations (1)-(4). Our main result is:

Theorem 11 *For any finite set* F *of polynomials in* n *variables, let* $\text{size}(F)$ *be the sum of the total degree of polynomials in* F*, plus the sum of the lengths of polynomials in* F*. There is a primitive recursive function* ϕ_n *such that for any input* F *to Buchberger's basic algorithm, the total cost of elementary operations required by the algorithm is at most* $\phi_n(\text{size}(F))$*. Furthermore* $\phi_n(\text{size}(F))$ *is majorized by the function* $W^{(2)}_{n+1}(\text{size}(F)+1)$*.*

6 Conclusion

Despite much interest in the complexity of Buchberger's algorithm, all previous bounds are unsatisfactory in some ways: they are very specialized (two or three variables), apply only to total degree ordering, have technical restrictions, or are not bounded as a function of the input size. We have tried to overcome these shortcomings. Some of our bounds are tight and although our general bound is somewhat crude, it may well be tight. However we were unable to construct suitable lower bounds.

References

[Bachmair and Buchberger 1980] L. Bachmair and Bruno Buchberger, *A simplified proof of the characterization theorem for Gröbner-bases*, **SIGSAM 14**, 29-34.

[Bayer 1982] David Allen Bayer, *The division algorithm and the Hilbert scheme*, **PhD thesis**, Department of Mathematics, Harvard.

[Buchberger 1979] Bruno Buchberger, *A criterion for detecting unnecessary reductions in the construction of Gröbner bases*, **SYMSAC 79**, Lecture Notes in C.S. 72, Springer-Verlag, 3-21.

[Buchberger, Collins and Loos 1982] B. Buchberger, G. E. Collins and R. Loos (editors), *Computer Algebra*, second edition, Springer-Verlag.

[Buchberger 1983] Bruno Buchberger, *A note on the complexity of constructing Gröbner bases*, **Proc. EUROCAL 83**, Lecture Notes in C.S. 162, Springer-Verlag, 137-145.

[Buchberger 1985] Bruno Buchberger, *Gröbner basis: An algorithmic method in polynomial ideal theory*, in chapter 6 of **Multidimensional Systems Theory**, (editor, N. K. Bose), D.Reidel Publishing Company, 184-229.

[Dixon 1913] Leonard E. Dixon, *Finiteness of the odd perfect and primitive abundant numbers with n distinct prime factors*, **Amer. J. of Math. 35**, 413-426.

[Dubé, Mishra and Yap 1986] T. Dubé, B. Mishra and C. Yap, *Admissible orderings and bounds for Gröbner bases normal form algorithm*, **NYU-Courant Robotics Lab. Report 88**, November, 1986.

[Galligo 1985] André Galligo, *Some algorithmic questions on ideals of differential operators*, **EUROCAL 85**, Linz, B. F. Caviness (ed.), Lecture Notes in C.S. 204, Springer-Verlag, 413-421.

[Gebauer and Möller 1986] R. Gebauer and H. M. Möller, *Buchberger's algorithm and staggered linear bases SYMSAC 86*, Waterloo.

[Gianni and Trager 1985] P. Gianni and B. Trager, *GCD's and factoring multivariate polynomials using Gröbner bases*, **EUROCAL 85**, Linz, B. F. Caviness (ed.), Lecture Notes in C.S. 204, Springer-Verlag, 409-410.

[Giusti 1985] Marc Giusti, *A note on the complexity of constructing standard bases*, **EUROCAL 85**, Linz, B. F. Caviness (ed.), Lecture Notes in C.S. 204, Springer-Verlag, 411-412.

[Hironaka 1964] H. Hironaka, *Resolution of singularities of an algebraic variety over a field of characteristic zero: I & II*, **Annals of Math. 79**, 109-326.

[Lazard 1983] Daniel Lazard, *Gröbner bases, Gaussian elimination, and resolution of systems of algebraic equations*, **EUROCAL 83**, London, H. van Hulzen (ed.), Lecture Notes in C.S. 162, Springer-Verlag, 146-156.

[Mayr and Meyer 1982] E. W. Mayr and A. R. Meyer, *The complexity of the word problems for commutative semigroups and polynomial ideals* **Advances in Mathematics 46**, 305-329.

[Mishra and Yap 1986] B. Mishra and C.Yap, *Notes on Gröbner Bases*, **NYU-Courant Robotics Lab. Report 87**, November 1986.

[Möller and Mora 1984] H. M. Möller and F. Mora, *Upper and lower bounds for the degree of Groebner bases*, **EUROSAM 84**, Lecture Notes in C.S. 174, Springer-Verlag, 172-183.

[Pohst and Yun 1981] M. E. Pohst and D. Y. Y. Yun, *On solving systems of algebraic equations via ideal bases and elimination theory*, **IBM Research Report RC 8891 (#38991)**, 10 pages.

[Seidenberg 1974] A. Seidenberg, *Constructions in algebra*, **Trans. AMS 197**, 273-313.

[Robbiano 1985] L. Robbiano, *Term Orderings on the Polynomial Ring*, 513-517.

[Robbiano 1986] L. Robbiano, *On the Theory of Graded Structures*, **J. Symbolic Computation 2**, 139-170.

[Spear 1977] David Spear, *A constructive approach to commutative ring theory*, **Proc. MACSYMA User's conf., Berkeley**, Richard Fateman (editor), NASA Publication CP-2012, Washington, D.C, 369-376.

[Wainer 1970] S. S. Wainer, *A classification of the ordinal recursive functions*, **Arch. math. Logik 13**, 136-153.

[Winkler 1984] Franz Winkler, *On the complexity of the Gröbner-bases algorithm over K[x,y,z]*, **EUROSAM 84**, Lecture Notes in C.S. 174, Springer-Verlag, 184-194.