Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

72

Symbolic and Algebraic Computation
EUROSAM '79, An International Symposium on Symbolic and Algebraic Manipulation, Marseille, France, June 1979

Edited by Edward W. Ng

Springer-Verlag
Berlin Heidelberg New York 1979
A CRITERION FOR DETECTING UNNECESSARY REDUCTIONS IN THE CONSTRUCTION OF GRÖBNER-BASES *) **)

B. Buchberger

Institut für Mathematik
Johannes-Kepler-Universität
A-4045 Linz, Austria

*) Dedicated to the 80th birthday of Prof. W. Gröbner
**) Invited paper, EUROSAM 79, Marseille (Springer LN in Comp. Scie., vol. 72)

Abstract

We present a new criterion that may be applied in an algorithm for constructing Gröbner-bases of polynomial ideals. The application of the criterion may drastically reduce the number of reductions of polynomials in the course of the algorithm. Incidentally, the new criterion allows to derive a realistic upper bound for the degrees of the polynomials in the Gröbner-bases computed by the algorithm in the case of polynomials in two variables.

1. Introduction

We present two results concerning the algorithmic construction of Gröbner-bases for polynomial ideals:

1. We develop a method for speeding up the algorithm given in /1/.
2. We derive a realistic upper bound for the complexity of the new version of the algorithm in the case of polynomials in two variables.

The usefulness of Gröbner-bases /2,3/ stems from the fact that a number of important problems in the theory of polynomial ideals may be solved...
easily as soon as a Gröbner-basis for the given ideal is available. Such problems are: the question whether a given polynomial belongs to a given ideal, the construction of a vector space basis for the residue class ring modulo a given ideal, the question whether a given set of algebraic equations is solvable, the question whether a given ideal has dimension zero, the question whether a given ideal is a principal ideal, the reduction of polynomials to canonical forms in the presence of side relations. For detailed motivation see /1,4,5,6,7/.

Gröbner-bases may be characterized in a number of different ways /7/.

Informally,
a finite set \( F \) of polynomials is a Gröbner-basis (for the ideal generated by \( F \)) iff
a certain reduction process for polynomials, which is defined relative to the "side relations" in \( F \), always leads to a unique result.

The algorithm described in /1/ solves the following problem:
given a finite set \( F \) of polynomials (in \( n \) variables)
find a finite set \( G \) of polynomials such that
\( F \) and \( G \) generate the same ideal and
\( G \) is a Gröbner-basis.

A rough description of the algorithm would read as follows:

For all pairs of polynomials in \( F \):
form a certain type of "resolvent" of the pair and reduce the resolvent relative to \( F \).
If the reduction does not lead to zero,
add the resulting polynomial to the basis $F$.

Note that in the case when a new polynomial is added to the basis, simultaneously the set of pairs of polynomials to be treated by the algorithm is expanded. Thus, the termination of the algorithm is a non-trivial problem. A termination proof is given in [1]. However, so far, no general upper bound for the complexity of the algorithm is known (in the author's dissertation, which is an early version of [1], a rough upper bound for the case of polynomials in two variables is given.)

The algorithm has been implemented several times [2,8,9,6,10] and, in fact, experience shows that it produces complex computations.

In this paper our main concern will be to develop a criterion by which the reduction of some of the resolvents may be avoided under certain circumstances. In practical examples, this results in a drastic saving of computation time. Incidentally, for the new version of the algorithm we will be able to derive an a priori upper complexity bound in the case of polynomials in two variables. This upper bound will be "realistic" in the sense that one can give examples where the bound is nearly reached.

2. Definition and Basic Facts

In order to make the paper easy to read we give the definitions of the basic notions by examples. For the formal definitions the reader is referred to [2].

Throughout the paper $f,g,h$ will denote polynomials in $n$ variables over a field $K$ and $F,G$ will denote finite sequences of polynomials. $F_i$ is the $i$-th element of the sequence $F$ and $L_F$ is the length of the sequence $F$. The terms are ordered "lexicographically" in the following sense
(example n=3):

\[ 1 <_T x <_T y <_T z <_T \]
\[ <_T x^2 <_T xy <_T xz <_T y^2 <_T yz <_T z^2 <_T \]
\[ <_T x^3 <_T x^2 y <_T x^2 z <_T xy^2 <_T \ldots \]

The "headterm of f" is the greatest term in f with respect to the ordering \(<_T\). For example:

\[ f := -x^2 z + 5xy^2 + 1 \]

headterm of f.

"f reduces to g with respect to F" (abbreviated \( f \triangleright g \))

if g is obtained from f by a sequence of steps of the following kind:

if a term t in f is a multiple of the headterm of an \( F_i \), add a suitable multiple of \( F_i \) to f in order that the coefficient of t disappears. For example

\[ F_1 := x^2 yz - xy, F_2 := xyz^2 - 2z^2, \]
\[ f := x^3 yz + 5xy^2 z^2 - 3xyz \triangleright F \]
\[ F \quad x^2 y + 5xy^2 z^2 - 3xyz \quad F \quad x^2 y + 10yz^2 - 3xyz. \]

We underline a polynomial by \( \underline{\ldots} \) if it is in normal form with respect to F, i.e. if it cannot be reduced further.

It is clear, /2/, that \( f \triangleright 0 \quad F \quad f \in \text{Ideal}(F) \)

(where \( \text{Ideal}(F) \) is the ideal generated by F).

However, in general,
\[ f \in \text{Ideal}(F) \not\rightarrow f = 0. \]

\[ \begin{align*}
& f \in \text{Ideal}(F) \\
& f = g \\
& f = h \\
& \not\rightarrow g = h.
\end{align*} \]

**Definition:** \( F \) is a **Gröbner-basis** \( \iff \)

\[ (G6) \quad \bigwedge_f \left( f \in \text{Ideal}(F) \implies f \neq 0 \right) \]

(It turns out, /2/, that we could equivalently define

\[ F \text{ is a Gröbner-basis} \iff \]

\[ (G3) \quad \bigwedge_{F, g, h} \left( f = g, f = h \iff g = h \right). \]

The following theorem is the basis for the algorithm given in /1/:

**Theorem 1** /2/ : \( F \) is a Gröbner-basis \( \iff \)

\[ (G2) \quad \bigwedge_{1 \leq i < j \leq L_F} \text{S-polynomial of } F_i \text{ and } F_j \neq 0. \]

The "S-polynomial of" two polynomials \( f \) and \( g \) (abbreviated \( \text{SP}(f,g) \)) is formed by the following process: determine the least common multiple \( t \) of the headterms of \( f \) and \( g \), then multiply \( f \) and \( g \) by appropriate factors \( f' \) and \( g' \), such that both the headterm of \( f'.f \) and the headterm of \( g'.g \) is \( t \) and both headterms have the same coefficient. Then form \( f'.f - g'.g \). This is the S-polynomial of \( f \) and \( g \). Note that \( t \) does not occur any more in the S-polynomial. The S-polynomial is the type of "resolvent" alluded to in the introduction.

We give an example: \( f := 5xy - 3x, \quad g := 7y^2 + 2x, \)

least common multiple \( \{xy, y^2\} = xy^2, \)

S-polynomial of \( f \) and \( g = (f).f - (5x).g = -21xy - 10x^2. \)
3. Basic Form of the Algorithm

Problem: given $F$, find $G$ such that

$$\text{Ideal}(F) = \text{Ideal}(G) \text{ and } G \text{ is a Gr"ubner-basis}$$

**Algorithm GBQ (1):**

1. $G := F$; $B := \{(i,j) / 1 \leq i,j \leq |F|\}$

A

2. while exist $(i,j) \in B$ do
   3. $h := \text{SP}(G_i, G_j)$; $h := \text{NF}(h, G)$;
   4. if $h \neq 0$ then
      5. $G := (G, h)$; $B := B \cup \{(i,|G|) / 1 \leq i < |G|\}$;
      6. $B := B - \{(i,j)\}$

$NF$ is a procedure that reduces a given polynomial to a polynomial that
is in normal form with respect to a given sequence of polynomials. $(G,h)$
denotes the sequence that is obtained from $G$ by adding $h$ at the end.

The while exist - statement seems to be self-explanatory. Instead of
begin/end we use indentation.

It is easy to see that the algorithm is correct by using the following
inductive assertion at cut point A:

$$\text{Ideal}(G) = \text{Ideal}(F) \text{ and } \forall_{1 \leq i,j \leq |G|, \{i,j\} \notin B} \text{SP}(G_i, G_j) = 0.$$

On termination ($B = \emptyset$) we then have

$$\text{Ideal}(G) = \text{Ideal}(F) \text{ and } \forall_{1 \leq i,j \leq |G|} \text{SP}(G_i, G_j) = 0.$$
By Theorem 1 this implies that $G$ is a Gröbner-basis.

The termination proof of this and all the other algorithms in this paper is based on the Theorem on p.380 in /1/.

**Example 1:**

$G_1 := F_1 := x^3yz - xz^2$, $G_2 := F_2 := xy^2z = xzy$, $G_3 := F_3 := x^2y^2 - z^2$.

Note that we have a lot of freedom in choosing $(I,J) \in B$. Every choice is correct. However, it will turn out that the sequence of choices has an essential influence on the complexity of the algorithm. Anticipating an argument of Section 4, we proceed by choosing $(I,J)$ such that the least common multiple of the head terms of $G_I$ and $G_J$ is minimal with respect to $\preceq_r$.

We write $I,J \rightarrow h$, if the reduction of the S-polynomial of $G_I$ and $G_J$ leads to the normal form $h$. With this notation the trace of a possible computation could be as follows:

- $2,3 \rightarrow G_4 := x^2yz - z^3$
- $1,4 \rightarrow G_5 := xz^3 - xz^2$
- $2,4 \rightarrow G_6 := yz^3 - z^3$
- $3,4 \rightarrow 0$
- $5,6 \rightarrow G_7 := x^2y^2 - xz^2$
- $4,7 \rightarrow G_8 := z^4 - x^2z^2$
- $2,7 \rightarrow 0$
- $5,7 \rightarrow 0$
- $6,7 \rightarrow 0$
- $5,8 \rightarrow G_9 := x^3z^2 - xz^2$
- $6,8 \rightarrow 0$
- $1,2 \rightarrow 0$
- $3,8 \rightarrow 0$
Altogether \( \binom{35}{2} \) reductions are necessary. From 6,7 one of them yield zero. In this example, we used very simple coefficients in order to make the reductions easy for hand computation. Nevertheless, from this example, we get a good impression of how a typical computation proceeds and we get an idea where we could save computation time: we should try to guess in advance which reductions will yield zero without actually carrying them out.

In /1/ we already gave two criteria of this kind, in this paper we shall develop a more powerful one.

4. Improved Form of the Algorithm: First Version

The improvement is based on the following theorem which has been obtained by a careful analysis of the proof of Theorem 1.

**Theorem 2** /1/ : \( F \) is a Gröbner-basis if

\[
1 \leq i < j \leq F, \quad 1 \leq k \leq F, \quad 1 \leq u_1, \ldots, u_k \leq F
\]

\[
\left[ i = u_j, \quad u_k = j, \quad H_F(u_1, \ldots, u_k) \leq_H H_F(i, j), \quad \text{SP}(F_{u_1} F_{u_2}) > 0 \right]
\]

Here we use the following notations: \( s \leq_H t : \quad t \text{ is a multiple of term } s \) (for instance, \( xyz^2 \leq_H x^3y^2z^3 \)), and

\[
H_F(u_1, \ldots, u_k) = \text{least common multiple of the head terms of } F_{u_1} \ldots F_{u_k}
\]

(for instance, in Example 1, \( H_G(4,5,6) = x^2yz^3, \quad H_G(4) = x^2yz \).)
Algorithm GB1:

\[ \text{G} := \text{F}; \quad \text{B} := \{(i,j) / 1 \leq i < j \leq L_F\}; \]

\[ \text{A} \]

\[ \text{while exist } \{i,j\} \in \text{B} \text{ do} \]

\[ \text{if} \ 	ext{Criterion 1} \]

\[ \text{then} \]

\[ h := \text{SP}(G_i, G_j); \quad h := \text{NF}(h, G); \]

\[ \text{if} h \neq 0 \text{ then} \]

\[ G := (G, h); \quad B := B \cup \{(i, L_G) / 1 \leq i \leq L_G\}; \]

\[ B := B \setminus \{(i,j)\}; \]

Criterion 1 is the following expression:

\[ \forall 1 \leq k \leq L_G \quad 1 \leq u_1, \ldots, u_k \leq L_G \]

\[ \{i = u_1, u_k = j, \quad H_G(u_1, \ldots, u_k) \subseteq H_G(i, j), \quad \text{not } \{(u_1, u_1+1) \in B\} \}

Algorithm GB1 can be proved correct by using the following inductive assertion at cut point (A):

\[ \text{Ideal}(G) = \text{Ideal}(F), \]

\[ \forall 1 \leq i, j \leq L_G \quad \text{not } \{(i,j) \in B\} \]

\[ \{i = u_1, u_k = j, \quad H_G(u_1, \ldots, u_k) \subseteq H_G(i, j), \quad \text{not } \{u_1, u_{1+1} \in G\} \}

Upon termination (B=\emptyset) this assertion implies (GB) for \( G \), i.e. \( G \) is a Gröbner-basis.

Note that in \( B \) we store those combinations \( \{i,j\} \) of indices, for which \( \text{SP}(G_i, G_j) \) is still to be reduced. The additional if - statement in
Algorithm GB1 allows to skip the reduction process for certain combinations \((i,j)\). In fact, in general, many combinations \((i,j)\) are of this kind. For instance, in Example 1, only 11 instead of 36 reductions have to be carried out when using Algorithm GB1. Computational experience shows that the number of reductions tends to be linear in the length of \(G\) in Algorithm GB1 whereas it is quadratic in GB0. A theoretical analysis of this phenomenon has not yet been achieved.

Algorithm GB1, still, is not satisfactory, because it is difficult to handle Criterion 1 algorithmically. In essence, it involves the computation of the transitive closure of the relations \(R_{i,j}\):

\[(u,v) \in R_{i,j} : \iff H_{G}(u,v) \leq_{T} H_{G}(i,j), \quad (u,v) \notin B\]

for all \(i,j\).

We now develop a criterion which is "nearly as powerful" as Criterion 1, however, it is much easier to handle.

Before we do this we observe that Algorithm GB1 suggests a strategy for the choice of \((i,j)\) in \(B\): choose \((i,j) \in B\) such that \(H_{G}(i,j)\) is minimal among all the \(H_{G}(i,j), \quad (i,j) \in B\), with respect to \(\leq_{T}\). If we chose an \((i,j) \in B\) such that \(H_{G}(i,j) \geq \min(H_{G}(i,j) +(i,j) \in B)\) it may happen that Criterion 1 is not yet fulfilled, i.e. we would have to reduce \(SP(G_{i},G_{j})\), whereas it might be fulfilled in a later stage of the algorithm, when some of the \((i,j)\) which are in \(B\) so far will be \(\notin B\). Thus we would have missed a chance to save a reduction.

The deviation from this strategy may even reverse the positive effect of the application of a criterion: In general, if a criterion is applicable to a pair \((i,j)\) at a certain stage of the algorithm, this does not mean, that all reductions of the S-polynomial of \(G_{i}\) and \(G_{j}\) would already lead
to zero. Thus we could possibly obtain a new polynomial in the basis, if we did not apply the criterion. If we apply the criterion, a new polynomial, possibly of higher degree, will necessarily appear at a later stage of the algorithm. This may lead to longer computations than that obtained without application of the criterion. Thus, for a "good" criterion it must be shown that all reductions of \( SP(G_1, G_2) \) would lead to zero in case the criterion is applicable to \((I, J)\). Criterion 1 and our final Criterion 3 are "good" ones in this sense, if applied in connection with the above strategy.

5. Improved Form of the Algorithm: Second Version

Algorithm GB2:

Exactly as Algorithm GB1 with the Criterion 1 replaced by the following Criterion 2

\[
\begin{align*}
&\exists \ u \in G \\
&I \neq u \neq J, \ H_g(u) \not\subseteq H_g(I, J), \\
&(I, u) \not\in B \lor D_g(I, u) < D_g(I, J), \quad (u, J) \not\in B \lor D_g(u, J) < D_g(I, J).
\end{align*}
\]

The correctness proof for Algorithm GB2 is more complicated. We use the following inductive assertion at cut point A:

\[
\text{Ideal}(G) = \text{Ideal}(F) \quad \text{and} \quad \forall i, j, k \leq G, \forall u_1, \ldots, u_k \leq G,
\begin{align*}
&i = u_1, \ u_k = j, \ H_G(u_1, \ldots, u_k) \not\subseteq H_G(i, j), \\
&(SP(G_{u_1}, G_{u_{k+1}}) = 0 \lor ((u_1, u_{k+1}) \in B \land D_G(u_1, u_{k+1}) < D_G(i, j))).
\end{align*}
\]
On termination (\(B=\emptyset\)) we then, again, get (G8) for \(G\), i.e. \(G\) is a Gröbner-basis. The proof that the inductive assertion is invariant is tedious and will be omitted here.

We note that the condition \((I,u)\notin B \lor D_G(I,u) < D_G(I,J)\), because of \(H_G(u) \subseteq H_G(I,J)\), is equivalent to \((I,u)\notin B \lor ((I,u)\in B \land D_G(I,u) \neq D_G(I,J))\). This is "nearly the same" as \((I,u)\notin B \lor (I,u)\in B\). Thus, one might conjecture that Criterion 2 can simply be replaced by

\[
\neg \bigwedge_{1 \leq u \leq l_G} \left[ I \neq u \neq J, \quad H_G(u) \subseteq H_G(I,J) \right].
\]

However, the resulting algorithm is incorrect.

On the other hand, the following Criterion 3

\[
\neg \bigwedge_{1 \leq u \leq l_G} \left[ I \neq u \neq J, \quad H_G(u) \subseteq H_G(I,J), \quad (I,u) \notin B, \quad (u,J) \notin B \right]
\]

though correct, in general, is not as efficient as Criterion 2. If, however, we use the selection strategy for \((I,J)\in B\) described in Section 4, Criterion 3 is as powerful as Criterion 2, because, in this case, \(D_G(I,u) < D_G(I,J) \implies (I,u) \notin B\). Also, Criterion 3 is a "good" one. This can be proven by adding the assertion

\[
\left( \text{Headterm of } f \leftarrow \hat{\epsilon}, \quad f \cdot g, \quad f \cdot h \right) \implies g = h,
\]

where \(\hat{\epsilon} := \min_{I,J} \{H_G(i,j)/(i,j)\in B\}\), to the inductive assertion at cut point A (extensive use of the proof techniques developed in /2/ has to be made).

Criterion 3 is easy to handle. A suitable data structure for \(B\) is as follows: simultaneously store \(B\) as a sequence of lists, the \(l\)-th list containing all the \((i,j)\) with \(D_G(i,j) = l\) (thus, the implementation of
the above selection strategy for \((i,j) \in B\) is easy, and as a boolean array \(C (C_{i,j} = \text{true} \iff (i,j) \in B)\) such that the decisions \((i,u) \notin B, (u,j) \notin B\) can be made easily.

Applying Algorithm GB3 (which is the result of replacing Criterion 1 in Algorithm GB1 by Criterion 3) to Example 1 (with the above selection strategy) yields a computation that actually reduces \(SP(S_i, S_j)\) only for the following 11 combinations of indices \((i,j)\): \((2,3), (1,4), (2,4), (5,6), (4,7), (2,7), (5,7), (5,8), (6,8), (1,9), (5,9)\). These are the same as would be treated by Algorithm GB1. In fact, one can show that using the above selection strategy for \((i,j) \in B\), Criterion 3 is as powerful as Criterion 1, except for those cases where at a certain stage of the algorithm

\[
\sqrt{\left( (i',j') \neq (i,j), \quad H_g(i',j') = H_g(i,j) \right)}.
\]

In this case Criterion 1 may judge \((i,j)\) to be "superflous" whereas Criterion 3 possibly does not detect this fact.

5. The Complexity of Algorithm GB3 for \(n=2\)

In this section we consider only sequences of polynomials in two variables. We give an upper bound \(B_P\) for the degrees of the polynomials that appear in the basis \(G\) constructed from a given \(F\) by Algorithm GB2.

If we know \(B_P\) we also have an upper bound \(N_F\) for the number of polynomials in the Gröbner-basis \(G\):

\[
N_F = L_F + S(B_P), \quad \text{where}
\]

\[
S(d) := \text{number of terms of degree} \leq d = \frac{(d+2)(d+1)}{2},
\]

because the head term of a polynomial \(G_{F+j+1}^k\) \((j = 0,1,...)\) constructed
by Algorithm G82 is distinct from the headterms of all the $G_1, \ldots, G_{L_F+1}$.

Thus, an upper bound $R_F$ for the number of possible $SP(G_i, G_j)$ that have to be reduced is

$$R_F = \frac{N_F \cdot (N_F - 1)}{2}.$$

A reduction of a polynomial $f$, whose headterm has the number $t$ in the lexicographical ordering of terms, needs at most $t \cdot N_F$ comparisons for determining the polynomials from the basis that have to be inserted into $f$ and at most $\frac{t(t-1)}{2}$ comparisons, multiplications and additions for the insertions. Since also the degree of all the $SP(G_i, G_j)$ is bounded by $B_F$ (see below) we need at most

$$\frac{N_F \cdot (N_F - 1)}{2} \cdot \left( S(B_F) \cdot N_F + \frac{S(B_F) \cdot (S(B_F) - 1)}{2} \right)$$

steps. Roughly, $B_F$ may be bounded by $4 \cdot P_F$, where $P_F := \max(D_F(i) / 1 \leq i \leq L_F)$ (where $D_F(i) := \text{degree of } H_F(i)$). Summarizing, the number of steps may be bounded by

$$T_F := 2 \cdot (L_F + 16 \cdot P_F^2)^4.$$

(We need the factor 2, if we take into account that, before a reduction, in any case Criterion 2 (or 3) has to be checked).

We now concentrate on determining $B_F$. An idea for getting $B_F$ may be obtained from the following graphical representation of the $H_G(i)$ and $H_G(i, j)$ of a given $G$:
In the drawing we represent a term $x^k y^l$ by the point $(k, l)$ in the cartesian plane. We mark the $H_G(i)$ (in our case: $i = 1, 2, 3, 4$) by drawing a circle $\bigcirc$ and the $H_G(i, j)$, for which

$$\bigvee_{1 \leq u \leq L_G} \left[ i \neq u \neq j, \quad H_G(u) < H_G(i, j), \quad D_G(i, u) < D_G(i, j), \quad D_G(u, j) < D_G(i, j) \right].$$

by a square $\square$. Let $n_G$ be the maximal degree of these $H_G(i, j)$ and let

$W_G$ be the "width" of $G$, i.e. the length of the thick line in the drawing, formally:

$$W_G := \left( \min_{1 \leq i \leq L_G} E^1_G(i) \right) + \left( \min_{1 \leq i \leq L_G} E^2_G(i) \right),$$

where $E^1_G(i) (E^2_G(i))$ is the exponent of $x$ ($y$) in $H_G(i)$.

(The shaded region is the region of terms $x^k y^l$ that are multiples of some $H_G(i)$.)
Now, a reduction according to Algorithm GB2 can only occur for a "squared" $M_G(i,j)$. The new polynomial, for instance $G_5$, then, satisfies $D_G(5) \leq M_G$ and the new squared $M_G(5,1)$ will satisfy $D_G(5,1) \leq M_G + W_G$.

On the other hand, incidentally, $W_G$ is reduced. This suggests that $M_G + W_G$ does not increase during the algorithm and therefore

\[
\bigcup_{1 \leq i \leq L_G} D_G(i) \leq M_G + W_G \leq M_F + W_F
\]

upon termination of the algorithm. Thus,

\[
B_F := M_F + W_F
\]

is a suitable upper bound for the degrees of the polynomials in the Gröbner-Basis computed.

Before we give more details about the above sketch of the derivation, let us consider the following example:

\[
G_1 := F_1 := xy^9 - x^{10}, \quad G_2 := F_2 := y^{10}
\]

\[
M_G = M_F = D_F(1,2) = 11, \quad W_G = W_F = 9.
\]

Reducing the $SP(G_1,G_2)$ we obtain $G_3 := x^{10}y$.

The new values of $M_G$ and $W_G$ are $M_G = D_G(1,3) = 19, \quad W_G = 1$.

Reducing $SP(G_1,G_3)$ yields $G_4 := x^{19}$.

Thus, $B_F = M_F + W_F = 20$ is nearly reached. From this example, we see that $B_F$ is a "realistic" bound. The other bounds $M_F, R_F, P_F$ and $T_F$, however, seem to be too coarse.

For a formal proof of $\bigcup_{1 \leq i \leq L_G} D_G(i) \leq B_F$ (where $G$ is the Gröbner-basis obtained from $F$ by Algorithm GB2) one has to show:

(1) $\bigcup_{1 \leq i \leq L_G} D_G(i) \leq M_G$ (for arbitrary $G$)
and

(2) the inductive assertion \( M_G + W_G \leq M_F + W_F \), added at cut point

A in Algorithm GB2, is invariant (it certainly is true at the beginning!).

A detailed proof of both (1) and (2) needs a relatively complicated case analysis. The intuitions, however, may easily be obtained from the drawing. We only point out the major steps in the proof of (2): The interesting case is when a new \( h \) is added to \( G \). In this case we show

(3) Degree of \( h \leq M_G \).

(4) \( M(G,h) + W(G,h) \leq M_G + W_G \).

For (3) observe that if Criterion 2 is fulfilled then also

\[
\forall u \in L_G \left[ \exists i \neq u \neq j, \ H_G(u) \leq M H_G(I,J), \ D_G(I,u) < D_G(I,J) \right],
\]

\[ D_G(u,J) < D_G(I,J) \]

must hold.

In order to prove (4) we distinguish the cases

a) \( h \) is in the "interior" of \( G \) (i.e. \( A \succ^H B \) head term of \( h \)).

b) \( h \) is "below" \( G \) (i.e. head term of \( h \leq^H X Y \)).

c) \( h \) is in the "exterior" of \( G \) (i.e. neither a) nor b)

(here, \( A := \min E^1(i) \), \( B := \min E^2_G(i) \)).

In case a) one can show that: \( M(G,h) \leq M_G \), \( W(G,h) = W_G \).

In case b) one can show that: \( M(G,h) \leq M_G \), \( W(G,h) < W_G \).

In case c) (which is the most interesting one)

\( M(G,h) \leq M_G + d \), \( W(G,h) = W_G + d \).
holds, where

\[
\begin{cases}
\min_{1 \leq i \leq L_G} \epsilon_0^{2}(i) - \text{exponent of } y \text{ in head term of } h, \\
\min_{1 \leq i \leq L_G} \epsilon_0^{1}(i) - \text{exponent of } x \text{ in head term of } h,
\end{cases}
\]

if this difference is \(> 0\).

\[d := \begin{cases}
\min_{1 \leq i \leq L_G} \epsilon_0^{2}(i) - \text{exponent of } y \text{ in head term of } h, \\
\min_{1 \leq i \leq L_G} \epsilon_0^{1}(i) - \text{exponent of } x \text{ in head term of } h,
\end{cases}
\]

if this difference is \(> 0\).

For generalizing the method given here for the case of \(n \geq 3\) a suitable definition of the "width" of \(G\) would be necessary. An attempt has been made in /12/. However, the problem seems to be difficult.

Acknowledgement: I gratefully acknowledge my indebtedness to F. Winkler for fruitful discussions on selection strategies.

References


/2/ B. Buchberger, A Theoretical Basis for the Reduction of Polynomials to Canonical Forms, ACM SIGSAM Bulletin 8, 19-29, August 1976


/10/ F. Winkler, Implementierung eines Algorithmus zur Konstruktion von Gröbner-Basen, Diplomarbeit, Univ. Linz (Austria), Math. Inst., 1978
