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 /2/. 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 /1,8,9,6,10/ 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 refered 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):

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

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

$$f := -x^2z + 5xy^2 + 1$$

headterm of f.

"f reduces to g with respect to F" (abbreviated $f \underset{F}{\triangleright} g$)

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

$$F_1 := x^2yz - xy, \quad F_2 := xyz^2 - 2z^2,$$

$$f := x^3yz + 5xy^2z^2 - 3xyz \underset{F}{\triangleright} x^2y + 5xy^2z^2 - 3xyz \underset{F}{\triangleright} \underline{x^2y + 10yz^2 - 3xyz}.$$

We underline a polynomial by ____ if it is in normalform with respect to F, i.e. if it cannot be reduced further.

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

(where Ideal(F) is the ideal generated by F).

However, in general,

$$f \in \mathrm{Ideal}(F) \not\Longrightarrow f \underset{F}{\triangleright} 0,$$

$$\left.\begin{array}{l} f \underset{F}{\triangleright} \underline{g} \\ f \underset{F}{\triangleright} \underline{h} \end{array}\right\} \not\Longrightarrow g = h.$$

Definition: F is a Gröbner-basis : $\Longleftrightarrow$

(G6) $\bigwedge_{f} (f \in \mathrm{Ideal}(F) \longrightarrow f \underset{F}{\triangleright} 0)$

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

F is a Gröbner-basis : $\longleftrightarrow$

(G3) $\bigwedge_{f,g,h} (f \underset{F}{\triangleright} \underline{g}, f \underset{F}{\triangleright} \underline{h} \longrightarrow g = h).$)

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

Theorem 1 /2/ : F is a Gröbner-basis $\longleftrightarrow$

(G2) $\bigwedge_{1 \le i < j \le L_F}$ S-polynomial of F_i and $F_j \underset{F}{\triangleright} 0$.

The "S-polynomial of" two polynomials f and g (abbreviated 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: $f := 5xy - 3x$, $g := 7y^2 + 2x$,
least common multiple $(xy, y^2) = xy^2$,
S-polynomial of f and g = $(7y).f - (5x).g = -21xy - 10x^2$.

3. Basic Form of the Algorithm

Problem: given F, find G such that

Ideal(F) = Ideal(G) and G is a Gröbner-basis

Algorithm GBO /1/ :

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

Ⓐ

while exist (I,J) ∈ B do

h := $SP(G_I, G_J)$; h := NF(h,G);

if h ≠ 0 then

G := (G,h); B := B ∪ {(i,L_G) / $1 \leq i < L_G$};

B := B - {(I,J)};

NF is a procedure that reduces a given polynomial to a polynomial that is in normalform 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 Ⓐ:

$$\text{Ideal}(G) = \text{Ideal}(F) \text{ and } \bigwedge_{\substack{1 \leq i < j \leq L_G \\ (i,j) \notin B}} SP(G_i, G_j) \underset{G}{\to} 0.$$

On termination (B = ∅!) we then have

$$\text{Ideal}(G) = \text{Ideal}(F) \text{ and } \bigwedge_{1 \leq i < j \leq L_G} SP(G_i, G_j) \underset{G}{\to} 0.$$

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:

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

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 headterms of G_I and G_J is minimal with respect to $<_T$.

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

$2,3 \longrightarrow G_4 := x^2yz - z^3$
$1,4 \longrightarrow G_5 := xz^3 - xz^2$
$2,4 \longrightarrow G_6 := yz^3 - z^3$
$3,4 \longrightarrow 0$
$5,6 \longrightarrow G_7 := xyz^2 - xz^2$
$4,7 \longrightarrow G_8 := z^4 - x^2z^2$
$2,7 \longrightarrow 0$
$5,7 \longrightarrow 0$
$6,7 \longrightarrow 0$
$5,8 \longrightarrow G_9 := x^3z^2 - xz^2$
$6,8 \longrightarrow 0$
$1,2 \longrightarrow 0$
$\vdots$
$3,8 \longrightarrow 0$

Altogether $\binom{8}{2} = 36$ reductions are necessary. From 6,7 on all of them yield zero. In this example, we used very simple coefficients in order to make the reductions easy for handcomputation. 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 /11/ : F is a Gröbner-basis $\Longleftrightarrow$

(G8) $\bigwedge_{1\le i<j\le L_F} \bigvee_{1\le k\le L_F} \bigwedge_{1\le u_1,..,u_k\le L_F}$

$$\left[i = u_1,\ u_k = j,\quad H_F(u_1,..,u_k) \le_M H_F(i,j),\quad \bigvee_{1\le l<k} SP(F_{u_l},F_{u_{l+1}})_F \ne 0\right]$$

Here we use the following notations: $s \le_M t$: $\Longleftrightarrow$
term t is a multiple of term s (for instance, $xyz^2 \le_M x^3yz^3$), and
$H_F(u_1,..,u_k)$:= least common multiple of the headterms of $F_{u_1},..,F_{u_k}$
(for instance, in Example 1, $H_G(4,5,6) = x^2yz^3$, $H_G(4) = x^2yz$.).

Algorithm GB1 :

```
G := F;  B := {{i,j} / 1≤i<j≤L_F};
Ⓐ
    while exist {I,J} ∈ B do
        if  Criterion 1
            then
                h := SP(G_I,G_J);  h := NF(h,G);
                if h ≠ 0 then
                    G := (G,h);  B := B ∪ {{i,L_G} / 1≤i<L_G};
        B := B - {{I,J}};
```

Criterion 1 is the following expression

$$\neg \bigvee_{1\le k\le L_G} \bigwedge_{1\le u_1,\ldots,u_k\le L_G} \Big[I = u_1,\ u_k = J,\quad H_G(u_1,\ldots,u_k) \le_M H_G(I,J),\quad \bigvee_{1\le l<k} (u_l,u_{l+1}) \notin B \Big]$$

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

$$\text{Ideal}(G) = \text{Ideal}(F),$$

$$\bigvee_{\substack{1\le i<j\le L_G \\ (i,j)\notin B}} \bigvee_{1\le k\le L_G} \bigwedge_{1\le u_1,\ldots,u_k\le L_G} \Big[i = u_1,\ u_k = j,\quad H_G(u_1,\ldots,u_k) \le_M H_G(i,j),\quad \bigvee_{1\le l<k} SP(G_{u_l},G_{u_{l+1}}) \underset{G}{\to} 0 \Big]$$

Upon termination (B=∅!) this assertion implies (G8) for G, i.e. G is a Gröbner-basis.

Note that in B we store those combinations (i,j) of indices, for which $SP(G_i,G_j)$ 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 GBO. 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 $R_{I,J}$

$$(u,v) \in R_{I,J} :\Longleftrightarrow H_G(u,v) \leq_M H_G(I,J),\ \{u,v\} \notin B$$

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 $H_G(i,j)$, {i,j} $\in$ B, with respect to $<_T$. If we chose an {I,J} $\in$ B such that $H_G(I,J) >_T \min(H_G(i,j)/\{i,j\} \in B)$ 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 <u>all</u> 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 $SP(G_I, G_J)$ 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

$$\neg \bigvee_{1 \le u \le L_G} \Big[I \neq u \neq J, \quad H_G(u) \le_M H_G(I,J), \quad \{I,u\} \notin B \ \vee \ D_G(I,u) < D_G(I,J), \quad \{u,J\} \notin B \ \vee \ D_G(u,J) < D_G(I,J) \Big].$$

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

$$\text{Ideal}(G) = \text{Ideal}(F) \quad \text{and}$$

$$\bigwedge_{\substack{1 \le i < j \le L_G \\ \{i,j\} \notin B}} \ \bigvee_{1 \le k < L_G} \ \bigwedge_{1 \le u_1, \ldots, u_k \le L_G} \Big[i = u_1, \ u_k = j, \quad H_G(u_1, \ldots, u_k) \le_M H_G(i,j), \quad \bigvee_{1 \le l < k} (SP(G_{u_l}, G_{u_{l+1}}) \underset{G}{>} 0 \quad \vee \quad \{u_l, u_{l+1}\} \in B \wedge D_G(u_l, u_{l+1}) < D_G(i,j)) \Big]$$

On termination (B=∅!) 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 $\{I,u\} \notin B \quad \vee \quad D_G(I,u) < D_G(I,J)$, because of $H_G(u) \leq_M H_G(I,J)$, is equivalent to $\{I,u\} \notin B \quad \vee \quad (\{I,u\} \in B \wedge D_G(I,u) \neq D_G(I,J))$. This is "nearly the same" as $\{I,u\} \notin B \quad \vee \quad \{I,u\} \in B$. Thus, one might conjecture that Criterion 2 can simply be replaced by

$$\neg \bigvee_{I \leq u \leq L_G} \left[I \neq u \neq J, \quad H_G(u) \leq_M H_G(I,J) \right].$$

However, the resulting algorithm is incorrect.

On the other hand, the following Criterion 3

$$\neg \bigvee_{I \leq u \leq L_G} \left[I \neq u \neq J, \quad H_G(u) \leq_M H_G(I,J), \quad \{I,u\} \notin B, \quad \{u,J\} \notin B \right]$$

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, $D_G(I,u) < D_G(I,J) \Longrightarrow \{I,u\} \notin B$. Also, Criterion 3 is a "good" one. This can be proven by adding the assertion

$$\bigwedge_{f,g,h} ((\text{Headterm of } f <_T \hat{t}\ ,\ f \triangleright \underline{g},\ f \triangleright \underline{h}) \Longrightarrow g = h)\ ,\ \text{where}$$

$\hat{t} := \min_{<_T} \{H_G(i,j) / \{i,j\} \in B\}$, to the inductive assertion at cut point Ⓐ (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} is easy), and as a boolean array C ($C_{i,j}$ = true :⟺ {i,j} ∈ B) such that the decisions {I,u} ∉ B, {u,J}∉ 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 $SP(G_I,G_J)$ 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} ∈ B, Criterion 3 is as powerful as Criterion 1, except for those cases where at a certain stage of the algorithm $\bigvee_{I',J'} [\{I',J'\} \neq \{I,J\}, \quad H_G(I',J') = H_G(I,J)]$.

In this case Criterion 1 may judge {I,J} to be "superflous" whereas Criterion 3 possibly does not detect this fact.

6. The Complexity of Algorithm GB2 for n=2

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

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

$N_F = L_F + S(B_F)$, where

S(d) := number of terms of degree ≤ d

$= \frac{(d+2)(d+1)}{2}$,

because the headterm of a polynomial G_{L_F+j+1} (j = 0,1,...) constructed

by Algorithm GB2 is distinct from the headterms of all the $G_1,\ldots,G_{L_F+j}$.

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

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

A reduction of a polynomial f, whose headterm has the number t in the lexicographical ordering of terms, needs at most $t.N_F$ 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

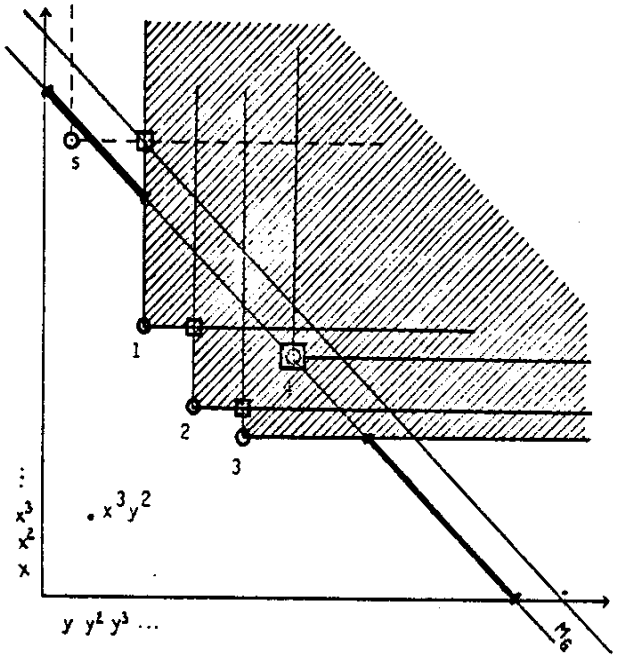
$$\frac{N_F.(N_F-1)}{2} . \left(S(B_F).N_F + \frac{S(B_F).(S(B_F)-1))}{2} \right)$$

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

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

(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 ○ and the $H_G(i,j)$, for which

$$\neg \bigvee_{1 \le u \le L_G} \big[i \neq u \neq j, \quad H_G(u) \le_M H_G(i,j), \quad D_G(I,u) < D_G(i,j), \quad D_G(u,J) < D_G(i,j) \big],$$

by a square □. Let M_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:

$$W_G := \Big(\min_{1 \le i \le L_G} E^1_G(i) \Big) + \Big(\min_{1 \le i \le L_G} E^2_G(i) \Big),$$

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" $H_G(I,J)$. The new polynomial, for instance G_5, then, satisfies $D_G(5) \leq M_G$ and the new squared $H_G(5,1)$ 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

$$\bigwedge_{1\leq i\leq L_G} D_G(i) \leq M_G + W_G \leq M_F + W_F$$

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:

$G_1 := F_1 := xy^9 - x^{10}$, $G_2 := F_2 := y^{10}$.
$M_G = M_F = D_F(1,2) = 11$, $W_G = W_F = 9$.
Reducing the $SP(G_1,G_2)$ we obtain $G_3 := x^{10}y$.
The new values of M_G and W_G are $M_G = D_G(1,3) = 19$, $W_G = 1$.
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 N_F, R_F, P_F and T_F, however seem to be too coarse.

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

(1) $\bigwedge_{1\leq i\leq L_G} D_G(i) \leq M_G$ (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

$$\neg \bigvee_{1 \leq u \leq L_G} \left[I \neq u \neq J,\ H_G(u) \leq_M H_G(I,J),\ D_G(I,u) < D_G(I,J),\right.$$
$$\left. D_G(u,J) < D_G(I,J) \right]$$

must hold.

In order to prove (4) we distinguish the cases

a) h is in the "Interior" of G (i.e. $x^A y^B \leq_M$ headterm of h)

b) h is "below" G (i.e. headterm of $h <_M x^A y^B$)

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

(here, $A := \min_{1 \leq i \leq L_G} E_G^1(i)$, $B := \min_{1 \leq i \leq L_G} E_G^2(i)$)

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)

$$M_{(G,h)} \leq M_G + d, \quad W_{(G,h)} + d = W_G$$

holds, where

$$d := \begin{cases} \min\limits_{1 \le i \le L_G} \varepsilon_G^2(i) & - \text{ exponent of } y \text{ in headterm of } h, \\ & \quad \text{if this difference is } > 0, \\ \min\limits_{1 \le i \le L_G} \varepsilon_G^1(i) & - \text{ exponent of } x \text{ in headterm of } h, \\ & \quad \text{if this difference is } > 0. \end{cases}$$

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 indebtness to F.Winkler for fruitful discussions on selection strategies.

References

/1/ B.Buchberger, Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems, Aequationes mathematicea, Vol.4/3, 374-383, 1970

/2/ B.Buchberger, A Theoretical Basis for the Reduction of Polynomials to Canonical Forms, ACM SIGSAM Bulletin 39, 19-29, August 1976

/3/ B.Buchberger, Some Properties of Gröbner-Bases for Polynomial Ideals, ACM SIGSAM Bulletin 40, 19-24, November 1976

/4/ R. Loos, Toward a Formal Implementation of Computer Algebra, Proceedings of EUROSAM 1974, ACM SIGSAM Bulletin 8, 9-16, August 1974

/5/ M.Lauer, Canonical Representatives for Residue Classes of a Polynomial Ideal, Proceedings of ACM SIGSAM 1976, 339-345

/6/ D.A.Spear, A Constructive Approach to Commutative Ring Theory, Proceedings of the MACSYMA User's Conference 1977, M.I.T.

/7/ W.Trinks, Über B.Buchbergers Verfahren, Systeme algebraischer Gleichungen zu lösen, J. of Number Theory, Vol.10/4, 475-488, 1978

/8/ R.Schrader, Diplomarbeit, Univ. Karlsruhe, Math. Inst., 1977

/9/ M.Lauer, Diplomarbeit, Univ. Karlsruhe, Inst. für Informatik, 1977

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

/11/ C.Kollreider, B.Buchberger, An Improved Algorithmic Construction of Gröbner-Bases for Polynomial Ideals, ACM SIGSAM Bulletin, to appear 1979

/12/ C.Kollreider, Polynomial Reduction: A Termination Proof for a Reduction Algorithm, Bericht Nr. 123, Math. Inst., Univ. Linz (Austria) 1978