A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5)

Jean-Charles Faugère
Ver 1.2
SPACES/LIP6/CNRS/Université Paris VI/INRIA
case 168, 4 pl. Jussieu, F-75252 Paris Cedex 05
E-mail: jcf@calfor.lip6.fr

ABSTRACT

This paper introduces a new efficient algorithm for computing Gröbner bases. We replace the Buchberger criteria by an optimal criteria. We give a proof that the resulting algorithm (called F_5) generates *no useless critical pairs if the input is a regular sequence*. This a new result by itself but a first implementation of the algorithm F_5 shows that it is also very efficient in practice: for instance previously untractable problems can be solved (cyclic 10). In practice for most examples there is no reduction to zero. We illustrate this algorithm by one detailed example.

1. INTRODUCTION

Solving polynomial systems is an important part of Computer Algebra since a lot of practical problems (cryptography, robotics, celestial mechanics, error correcting codes, signal theory, ...) can be solved with these algorithms. Among all available methods for solving polynomial systems, computation of Gröbner bases remains one of the more powerful. Historically, the Buchberger algorithm was the first algorithm for computing such Gröbner bases.

It may eventually be possible to suggest two improvements for the Buchberger algorithm [3, 4, 5]. The first improvement is concerned with strategies: during a Gröbner computation, several choices can be made (select a critical pair, choose a reductor) this aspect of the problem is not directly studied in this paper, but is implemented in other algorithms (F_4 [6] for instance). The other open issue was to remove useless computations: since 90% of the time is spent in computing zero it is a very challenging question to have a more powerful criterion to remove useless critical pairs. This is precisely the goal of this paper to give a theoretical and practical answer.

In [9] the link between the computation of a Gröbner basis of $F = [f_1, \ldots, f_m]$ and linear algebra is done: the Buchberger algorithm can be considered as a triangularisation of a submatrix of the sylvester matrix. The reduction of a polynomial to zero can be interpreted as a linear dependence of the rows of this matrix. Since each row of the matrix is a product $t \times f$ where t is a term and $f \in F$, a linear dependence is $\sum \lambda t f = 0$ or by grouping terms: $\sum_{i=1}^{m} g_i f_i = 0$. In other words, $(g_1, \ldots, g_m)$ is a syzygy.

Several papers investigate those issues: Buchberger [4] proposes two criteria to remove a lot of useless critical pairs; staggered linear bases are used in [7]; the idea of [10] is to compute simultaneously a Gröbner basis and a basis of the module of syzygies: a critical pair is not considered if the corresponding syzygy is a linear combination of some elements of the current basis of the module of syzygies. They have in all in common to use implicitly or explicitly the trivial sysygies $f_i f_j = f_j f_i$. Another common point is that all the algorithms are nearly Buchberger's algorithm except that some reductions are avoided. The efficiency of those algorithms is not yet satisfactory in theory and practice because a lot of useless critical pairs are not removed. For instance we quote from [10] that "many useless pairs are discovered, but it involves a lot of extra computation, so the execution time is increased". Another approch is involutive bases [11] which is based on the concept of involutive monomial division: some reductions are forbidden and so some computations are not considered.

The strategy in this paper is to take into account only the trivial syzygies $f_i f_j - f_j f_i = 0$ but not to compute the module of syzygies. This imply (see section 2 and 4) two major differences with the standard Buchberger algorithm or the F_4 algorithm: first we need to compute *all* the Gröbner basis of the following ideals (f_m), (f_{m-1}, f_m), ..., $(f_1, \ldots, f_m)$. The second difference is that some reductions are not allowed; as a result the reduction of *one* polynomial by a list of polynomials may be *several* polynomials. A consequence of the restriction to trivial syzygies is that, in worst cases, the algorithm does not avoid all the useless pairs: for instance if we have two times the same polynomial in the original equations there is a reduction to zero. However we give the proof (see corollary 3) that if the input system is a regular sequences then there is no reduction to zero. Moreover, in practice, for most systems there is no reduction to zero (experimental evidences are given in 9.1). Another important point is that the new algorithm does not improve the theoretical worst case complexity for computing Gröbner bases but experimentally (see section 9.2 for some some CPU timings and comparison with other algorithms), the F_5 is faster than all the previously implemented algorithms. The limited length of the paper impose us to make some choices: we give a full description of the algorithm and a detailed example but the proofs of correctness and termination are only sketched. For the same reason the experimental section 9 is minimal. A full paper describing the algorithm in the most general case is in preparation.

The plan of the paper is as follows. The section 5 is devoted to presenting the new criterion, and a theorem giving an equivalent condition for a set of polynomials to be a Gröbner basis. The resulting algorithm is described in section 7. This section includes also the proof of the correctness of the algorithm. In section 2 we give the idea of the algorithm. The necessary mathematical notations (we make the choice to use the same notations as in the book [2]) are reviewed in section 3. In section 8 we compute the example from [10] in full. The name of this algorithm is simply algorithm number 5. In the rest of this paper F_5 stands for this algorithm.

2. THE IDEA

We consider the following systems of degree 2 in 3 variables x,y,z depending on the parameter $b \in \{0,1\}$:

$$\mathscr{S}_b \begin{cases} f_3 = x^2+18xy+19y^2+8xz+5yz+7z^2 \\ f_2 = 3x^2+(7+b)xy+22xz+11yz+22z^2+8y^2 \\ f_1 = 6x^2+12xy+4y^2+14xz+9yz+7z^2 \end{cases}$$

We want to compute a Gröbner basis of f_1, f_2, f_3 modulo 23 for a total degree ordering with $x > y > z$. This can be done with the Buchberger algorithm (including the Buchberger criteria): there is 5 useless pairs and 5 useful ones. First we suppose that $b = 0$. To compute the Gröbner basis, we proceed degree by degree. For the degree 2 there is no choice to construct the matrix:

$$A_2 = \begin{array}{c} \\ f_3 \\ f_2 \\ f_1 \end{array} \begin{array}{c} \begin{array}{cccccc} x^2 & xy & y^2 & xz & yz & z^2 \end{array} \\ \begin{pmatrix} 1 & 18 & 19 & 8 & 5 & 7 \\ 3 & 7 & 8 & 22 & 11 & 22 \\ 6 & 12 & 4 & 14 & 9 & 7 \end{pmatrix} \end{array}$$

and after triangulation of the matrix A_2:

$$B_2 = \begin{array}{c} \\ f_3 \\ f_2 \\ f_1 \end{array} \begin{array}{c} \begin{array}{cccccc} x^2 & xy & y^2 & xz & yz & z^2 \end{array} \\ \begin{pmatrix} 1 & 18 & 19 & 8 & 5 & 7 \\ 0 & 1 & 3 & 2 & 4 & -1 \\ 0 & 0 & 1 & -11 & -3 & -5 \end{pmatrix} \end{array}$$

and we have constructed two "new" polynomials in the ideal $f_4 = xy+4yz+2xz+3y^2-z^2$ and $f_5 = y^2-11xz-3yz-5z^2$.
In degree 3 the first idea is to construct the matrix:

$$\begin{array}{c} \\ zf_3 \\ yf_3 \\ xf_3 \\ zf_2 \\ yf_2 \\ xf_2 \\ zf_1 \\ yf_1 \\ xf_1 \end{array} \begin{array}{c} \begin{array}{cccccc} x^3 & x^2y & xy^2 & y^3 & x^2z & \dots \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & \dots \\ 0 & 1 & 18 & 19 & 0 & \dots \\ 1 & 18 & 19 & 0 & 8 & \dots \\ 0 & 0 & 0 & 0 & 3 & \dots \\ 0 & 3 & 7 & 8 & 0 & \dots \\ 3 & 7 & 8 & 0 & 22 & \dots \\ 0 & 0 & 0 & 0 & 6 & \dots \\ 0 & 6 & 12 & 4 & 0 & \dots \\ 6 & 12 & 4 & 0 & 14 & \dots \end{pmatrix} \end{array}$$

To triangulate the matrix the first operation might be to simplify rows xf_2 and xf_1 with the row xf_3. But this this is a waste of time since this as already be done in the previous step: for instance $f_4 = -f_2+3f_3$, so that $xf_4 = -xf_2+3xf_3$. This is an important idea of the Buchberger algorithm: try to reuse as much as possible the previous computations. It is also clear that we should not put into the matrix f_1 and f_4 since they are linearly depends. So we construct a matrix with f_4 (resp. f_5) instead of f_2 (resp. f_1):

$$\begin{array}{c} \\ zf_3 \\ yf_3 \\ xf_3 \\ zf_4 \\ yf_4 \\ xf_4 \\ zf_5 \\ yf_5 \\ xf_5 \end{array} \begin{array}{c} \begin{array}{cccccccccc} x^3 & x^2y & xy^2 & y^3 & x^2z & xyz & y^2z & xz^2 & yz^2 & z^3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 18 & 19 & 8 & 5 & 7 \\ 0 & 1 & 18 & 19 & 0 & 8 & 5 & 0 & 7 & 0 \\ 1 & 18 & 19 & 0 & 8 & 5 & 0 & 7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3 & 2 & 4 & 22 \\ 0 & 0 & 1 & 3 & 0 & 2 & 4 & 0 & 22 & 0 \\ 0 & 1 & 3 & 0 & 2 & 4 & 0 & 22 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 12 & 20 & 18 \\ 0 & 0 & 0 & 1 & 0 & 12 & 20 & 0 & 18 & 0 \\ 0 & 0 & 1 & 0 & 12 & 20 & 0 & 18 & 0 & 0 \end{pmatrix} \end{array}$$

After triangulation

$$\begin{array}{c} \\ xf_3 \\ yf_3 \\ yf_2 \\ \mathbf{xf_2} \\ zf_3 \\ zf_2 \\ zf_1 \\ \mathbf{yf_1} \\ \mathbf{xf_1} \end{array} \begin{array}{c} \begin{array}{cccccccccc} x^3 & x^2y & xy^2 & y^3 & x^2z & xyz & y^2z & xz^2 & yz^2 & z^3 \end{array} \\ \begin{pmatrix} 1 & 18 & 19 & 0 & 8 & 5 & 0 & 7 & 0 & 0 \\ & 1 & 18 & 19 & 0 & 8 & 5 & 0 & 7 & 0 \\ & & 1 & 3 & 0 & 2 & 4 & 0 & 22 & 0 \\ & & & \mathbf{1} & 0 & 0 & 8 & 1 & 18 & 15 \\ & & & & 1 & 18 & 19 & 8 & 5 & 7 \\ & & & & & 1 & 3 & 2 & 4 & 22 \\ & & & & & & 1 & 12 & 20 & 18 \\ & & & & & & & \mathbf{1} & 11 & 13 \\ & & & & & & & & \mathbf{1} & 18 \end{pmatrix} \end{array}$$

So we have constructed 3 new polynomials (black bold font). For instance $f_6 = y^3+8y^2z+xz^2+18yz^2+15z^3$ and we recall that this polynomial comes from the row $x f_4$ or equivalently $x f_2$.
In degree 4 there is a new interesting point: the matrix A_4 whose rows are

$$x^2 f_i, xy f_i, y^2 f_i, xz f_i, yz f_i, z^2 f_i,\ i = 1,2,3$$

is not full rank ! (this correspond to 3 useless pairs in the Buchberger algorithm). The reason is that $f_2 f_3 - f_3 f_2 = 0$ or written differently:

$$\begin{array}{l} 3x^2 f_3 + (7+b)xy f_3 + 8y^2 f_3 + 22xz f_3 + 11yz f_3 + 22z^2 f_3 \\ -x^2 f_2 - 18xy f_2 - 19y^2 f_2 - 8xz f_2 - 5yz f_2 - 7z^2 f_2 = 0 \end{array}$$

Hence we can remove the row $x^2 f_2$ from A_4. By using $f_1 f_3 - f_3 f_1 = 0$ we can remove in the same way $x^2 f_1$ from A_4. Since there is another relation $f_1 f_2 = f_2 f_1$ we know that there is another useless row in the matrix A_4. Suppose that we return to the original problem $\mathscr{S}_b$ with $b \in \{0,1\}$; we have

$$\begin{array}{rcl} 0 & = & (f_2 f_1 - f_1 f_2) - 3(f_3 f_1 - f_1 f_3) \\ 0 & = & (f_2 - 3f_3) f_1 - f_1 f_2 + 3 f_1 f_3 \\ 0 & = & f_4 f_1 - f_1 f_2 + 3 f_1 f_3 \\ 0 & = & \left((1-b)xy + 4yz + 2xz + 3y^2 - z^2\right) f_1 \\ & & -(6x^2+\cdots) f_2 + 3(6x^2+\cdots) f_3 \end{array}$$

We deduce from this equality that we can remove $xy f_1$ from A_4 if $b \neq 0$ and $yz f_1$ if $b = 0$. In other words it is impossible to know without computation which row is useless (since it depends on the value of b). On the other hand a combination of the trivial relations $f_i f_j = f_j f_i$ can always be written:

$$u(f_2 f_1 - f_1 f_2) + v(f_3 f_1 - f_1 f_3) + w(f_2 f_3 - f_3 f_2)$$

where u, v, w are arbitrary polynomials. This can be rewritten

$$(u f_2 + v f_3) f_1 - u f_1 f_2 - v f_1 f_3 + w f_2 f_3 - w f_3 f_2$$

Hence all the (trivial) relation $h f_1$ are such that h is in the ideal generated by f_2 and f_3. So it is easy to remove lines if we have already computed a Gröbner basis of (f_2, f_3). More precisely we can always remove the rows $m f_1$ where m is a monomial divisible by the leading term of an element of $Id(f_2, \dots, f_m)$. If G_{prev} is an already computed Gröbner basis $(f_2, \dots, f_m)$ and we want to compute a Gröbner basis of $(f_1) + G_{\text{prev}}$ then we will construct matrices whose rows are $m f_1$ such that m is a monomial not divisible by the leading term of an element of G_{prev}.

To finish this example ($b = 0$) and in order to reuse the previous computations we have to apply the following simplification rule (in that order):

$$\begin{array}{cc} xf_2 \to f_6 & f_2 \to f_4 \\ xf_1 \to f_8 & yf_1 \to f_7 \\ f_1 \to f_5 & \end{array}$$

Now the rows of the matrix A_4 are

$$\begin{array}{c} yf_7, zf_8, zf_7, z^2 f_5, yf_6, y^2 f_4, zf_6, yzf_4, \\ z^2 f_4, x^2 f_3, xyf_3, y^2 f_3, xzf_3, yzf_3, z^2 f_3, \end{array}$$

Hence A_4 is almost a triangular matrix except a 5×5 block:

$$\begin{array}{c} \\ z^2 f_4 \\ z^2 f_5 \\ z f_7 \\ z f_8 \\ y f_7 \end{array} \begin{array}{c} \begin{array}{ccccc} xyz^2 & y^2z^2 & xz^3 & yz^3 & z^4 \end{array} \\ \begin{pmatrix} 1 & 3 & 2 & 4 & 22 \\ 0 & 1 & 12 & 20 & 18 \\ 0 & 0 & 1 & 11 & 13 \\ 0 & 0 & 0 & 1 & 18 \\ 1 & 11 & 0 & 13 & 0 \end{pmatrix} \end{array}$$

The reduction of the matrix give us a new polynomial $f_9 = z^4$. Remark that none useless pair (a line in the matrix reducing to zero) has remained.

The conclusion of this example is that in order to reuse the previous computations in lower degrees: first we need to give a unique "name" or "signature" (see section 4) to each row of the matrix (for instance the true name of the rows xf_4, f_6 is xf_2 in the previous example).The second thing is that we have to implement the simplification rules (see section 6).

3. STANDARD NOTATIONS

In the rest of the paper we suppose that all the polynomials are homogeneous and that the coefficients of the polynomials are in a field.

We use the notations of [2] for basic definitions: $\mathscr{K}$ is the ground field, $\mathscr{P} = \mathscr{K}[x_1, \ldots, x_n]$ is the polynomial ring. $\mathbf{N}$ is the set of non negative integers. We denote by $T(x_1, \ldots, x_n)$, or simply by T, the set of all terms in these variables. We choose $<$ an admissible ordering on T. If $t = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \in T$, then the *total degree* of t is defined as $\deg(t) = \sum_{i=1}^{n} \alpha_i$. Now let $0 \neq f \in \mathscr{P}$, so that $f = \sum c(\alpha_1, \ldots, \alpha_n) x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ (where $c(\alpha_1, \ldots, \alpha_n)$ are elements of $\mathscr{K}$). The *total degree* of f is defined as

$$\deg(f) = max\left\{\alpha_1 + \cdots + \alpha_n \mid c(\alpha_1, \ldots) \neq 0\right\}.$$

We use the notation $\mathrm{HM}(f)$ (resp. $\mathrm{HT}(f)$, $\mathrm{HC}(f)$) for the *head monomial* (resp. *head term*, *head coefficient*) of f.

Let $f, g, p \in \mathscr{P}$ with $p \neq 0$, and let F be a finite subset of $\mathscr{P}$. Then we say that f *is reducible modulo* P if there exists $g \in \mathscr{P}$ such that $f \xrightarrow[P]{} g$. $f \xrightarrow[P]{*} g$ is the reflexive-transitive closure of $\xrightarrow[P]{}$. If G is a Gröbner basis then $\mathrm{NF}(f, G) = g$ where $f \xrightarrow[G]{*} g$ is the *normal form* of f w.r.t. G. The *S-polynomial* of f and g is defined as

$$Spol(f,g) = \mathrm{HC}(g)\frac{\tau}{\mathrm{HT}(f)} f - \mathrm{HC}(f)\frac{\tau}{\mathrm{HT}(g)} g$$

where $\tau = lcm(\mathrm{HT}(f), \mathrm{HT}(g))$.

4. SIGNATURE OF A POLYNOMIAL

Let $(f_1, \ldots, f_m)$ be a polynomial m-tuple (an element of the free module $\mathscr{P}^m$) and I the ideal generated by $(f_1, \ldots, f_m)$. The goal of this section is to associate a unique and canonical "signature" for all the elements of $T(I)$ that is to say all the leading terms of all the polynomials in the ideal.

In the following $\mathbf{F}_i$ is the canonical i-th unit vector in $\mathscr{P}^m$. We consider the evaluation function:

$$v\left(\begin{array}{ccc} \mathscr{P}^m & \longrightarrow & \mathscr{P} \\ \mathbf{g} = (g_1, \ldots, g_m) & \mapsto & \sum_{i=1}^{m} f_i g_i \end{array}\right)$$

We have $v(\mathbf{F}_i) = f_i$ and $\mathbf{g} = \sum_{i=1}^{m} g_i \mathbf{F}_i$. An m-tuple $\mathbf{g} = (g_1, \ldots, g_m)$ is called a *syzygy* if $v(\mathbf{g}) = 0$. The so called *principal* syzygies $\mathbf{s}_{i,j} = f_j \mathbf{F}_i - f_i \mathbf{F}_j$ are syzygies. The set of all syzygies is a module and abbreviated by Syz (for more information on syzygies we refer to [2] or to [1]). Let PSyz be the module generated by the principal syzygies. For a generic (random) polynomial system $(f_1, \ldots, f_m)$, $\mathrm{Syz} = \mathrm{PSyz}$.

We can extend the admissible ordering $<$ to $\mathscr{P}^m$ with the following definition:

$$\sum_{k=i}^{m} g_k \mathbf{F}_k \prec \sum_{k=j}^{m} h_k \mathbf{F}_k \text{ iff } \begin{cases} i > j \text{ and } h_j \neq 0 \\ \text{or} \\ i = j \text{ and } \mathrm{HT}(g_i) < \mathrm{HT}(h_i) \end{cases}$$

In particular we have $\mathbf{F}_1 \succ \mathbf{F}_2 \succ \cdots \succ \mathbf{F}_m$. For all $\mathbf{g} \in \mathscr{P}^m$ there is an index i such that $\mathbf{g} = \sum_{k=i}^{m} g_k \mathbf{F}_k$ with $g_i \neq 0$. This i will be denoted as the *index* of $\mathbf{g}$, $\mathrm{index}(\mathbf{g})$. For the new ordering $\prec$ we have

$$\mathrm{HT}(\mathbf{g}) = \mathrm{HT}(g_i)\mathbf{F}_i$$

We define the degree of $\mathbf{g} = \sum_{i=1}^{m} g_i \mathbf{F}_i$, $\deg(\mathbf{g})$ by

$$max\left\{\deg(g_i) + \deg(f_i) \text{ for } i \in \{1, \ldots, m\}\right\}$$

Let $\mathbf{T_i}$ be $\left\{t\mathbf{F}_i \mid t \in T\right\}$ so that $\mathrm{HT}(\mathbf{g}) \in \mathbf{T_i}$. $\mathbf{T} = \cup_{i=1}^{m} \mathbf{T_i}$ will be the set of the signatures of all the polynomials in the ideal I. Of course if $t \in T$, $W(t) = \{\mathbf{g} \in \mathscr{P}^m \mid \mathrm{HT}(v(\mathbf{g})) = t\}$ can contain more than one element so we have to choose one of them:

PROPOSITION 1.

$$\textit{Let } w \textit{ be } \left(\begin{array}{ccc} T & \longrightarrow & \mathscr{P}^m \\ t & \mapsto & min_{\prec} W(t) \end{array}\right)$$

If $(t_1, t_2) \in T(I)^2$, *then* $HT(w(t_1)) \neq HT(w(t_2))$ *if* $t_1 \neq t_2$.

COROLLARY 1. *For all the polynomials p in the ideal I we define $v_1(p)$ to be $HT(w(HT(p)))$. If p_1 and p_2 are two polynomials of I with distinct head terms ($HT(p_1) \neq HT(p_2)$) we have $v_1(p_1) \neq v_1(p_2)$.*

In the following algorithm F_5, $v_1(p)$ will be the "signature" of the polynomial p: it is unique and does not depend on the order of the computations. We need to store these data in the internal representation of a polynomial. Mathematically the representation of polynomials will be $R = \mathbf{T} \times \mathscr{P}$. If $r = (tF_i, f) \in R$ we define:

$$\begin{array}{c} \mathrm{poly}(r) = f \in \mathscr{P} \\ \mathscr{S}(r) = t\mathbf{F}_i \in \mathbf{T} \\ \mathrm{index}(r) = i \in N \end{array}$$

We will see that during the execution of the algorithm the property $\mathscr{S}(r) = v_1(\mathrm{poly}(r))$ is conserved. We say that $r \in R$ is admissible if there exists $\mathbf{g} \in v^{-1}(\mathrm{poly}(r))$ such that $\mathrm{HT}(\mathbf{g}) = \mathscr{S}(r)$. Let $0 \neq \lambda \in \mathscr{K}$, $v \in T$, $\mathbf{t} = w\mathbf{F}_k \in \mathbf{T}$ and $r = (u\mathbf{F}_i, p) \in R$ we define $\lambda r = (u\mathbf{F}_i, \lambda p)$, $v\mathbf{t} = (vw)\mathbf{F}_k$ and $vr = (uv\mathbf{F}_i, vp)$. We do not define an addition. We also extend the definition of usual operators to R:

for $r \in R$ $\mathrm{HT}(r) = \mathrm{HT}(\mathrm{poly}(r))$.
for $r \in R$ $\mathrm{HC}(r) = \mathrm{HC}(\mathrm{poly}(r))$.
for $r \in R$ and $G \subset \mathscr{P}$, $\mathrm{NF}(r, G) = (\mathscr{S}(r), NF(\mathrm{poly}(r), G))$.

5. NEW CRITERION

DEFINITION 1. *Let P be a finite subset of R, and $r \in R$, and $t \in R$. If*

$$poly(r) = \textstyle\sum_{p \in P} m_p p \;\; m_p \in \mathscr{P}$$

we say that it is a t-representation *of r wrt P if $HT(t) \geq HT(m_p p)$ and $\mathscr{S}(r) \geq \mathscr{S}(m_p p)$ for all $p \in P$. This property will be denoted as $r = \mathscr{O}_P(t)$. We use the notation $r = o_P(t)$ if there exists $t' \in R$ such that $\mathscr{S}(t') \leq \mathscr{S}(t)$ and $HT(t') < HT(t)$ such that $r = \mathscr{O}_P(t')$.*

DEFINITION 2. *We say that $r \in R$ is* normalized *if $\mathscr{S}(r) = e\mathbf{F}_k$ and e is not top reducible by $Id(f_{k+1}, \ldots, f_m)$.*
We say that $(u, r) \in T \times R$ is normalized if ur is normalized. We say that a pair $(r_i, r_j) \in R^2$ is normalized *if $u_j \mathscr{S}(r_j) \prec u_i \mathscr{S}(r_i)$,*

(u_i, r_i) and (u_j, r_j) are normalized where
$\tau_{i,j} = lcm(HT(r_i), HT(r_j))$, $u_i = \frac{\tau_{i,j}}{HT(r_i)}$, $u_j = \frac{\tau_{i,j}}{HT(r_j)}$.

THEOREM 1. *Let $F = [f_1, \ldots, f_m]$ be a list of monic polynomials. Let $G = [r_1, \ldots, r_{n_G}] \in R^{n_G}$ such that*

(i) $F \subset poly(G)$. Let $g_i = poly(r_i)$ and $G_1 = [g_1, \ldots, g_{n_G}]$.

(ii) all the r_i are admissible and monic ($i = 1, \ldots, n_G$).

(iii) for all $(i, j) \in \{1, \ldots, n_G\}$, such that the pair (r_i, r_j) is normalized then $spol(g_i, g_j) = o_G(u_i r_i)$ (or 0) where

$$u_i = \frac{lcm(HT(g_i), HT(g_j))}{HT(r_i)}.$$

Then G_1 is a Gröbner basis of I.

PROOF. Let f be an element of $I = Id(G_1)$. We define $\mathscr{V} = \{(\mathbf{s}, \sigma) \in \mathscr{P}^{n_G} \times \mathscr{S}_n \,|\, \sum_{i=1}^{n_G} s_i g_{\sigma(i)} = f \text{ and } \mathscr{S}(s_1 r_{\sigma(1)}) \geq \mathscr{S}(s_2 r_{\sigma(2)}) \geq \cdots\}$. We define a new ordering $(\mathbf{s}, \sigma) <_1 (\mathbf{s}', \sigma')$. We use the notation

$$\overline{\nu} = (\mathscr{S}(s_1 r_{\sigma(1)}), \mathscr{S}(s_2 r_{\sigma(2)}), \cdots)$$

and

$$\overline{\nu'} = (\mathscr{S}(s'_1 r_{\sigma'(1)}), \mathscr{S}(s'_2 r_{\sigma'(2)}), \cdots).$$

We define $(\mathbf{s}, \sigma) <_1 (\mathbf{s}', \sigma')$ if one of the following conditions is true

(i) $\overline{\nu} \prec_{lex} \overline{\nu'}$

(ii) $\overline{\nu} = \overline{\nu'}$ and $max_i \mathrm{HT}(s_i g_{\sigma(i)}) < max_i \mathrm{HT}(s'_i g_{\sigma'(i)})$

(iii) $\overline{\nu} = \overline{\nu'}$ and $t = max_i \mathrm{HT}(s_i g_{\sigma(i)}) = max_i \mathrm{HT}(s'_i g_{\sigma'(i)})$ and $\#\{i \,|\, \mathrm{HT}(s_i g_{\sigma(i)}) = t\} < \#\{i \,|\, \mathrm{HT}(s'_i g_{\sigma'(i)}) = t\}$

We take $\mathbf{s} = min_{<_1} \mathscr{V}$. Wlog we may assume that σ is the identity (by renumbering G) Let $t = max_i \mathrm{HT}(s_i g_i)$ and $\mathscr{I} = \{i \,|\, \mathrm{HT}(s_i g_i) = t\}$, $r = \#\mathscr{I}$. Suppose for a contradiction that $t > \mathrm{HT}(f)$. Necessarily $r \geq 2$. Suppose that there exists i such that (s_i, r_i) is not normalized. That is to say $\mathscr{S}(r_i) = uF_k$ and $\mathrm{HT}(s_i)u \in \mathrm{HT}(Id(f_{k+1}, \ldots, f_m))$. Since r_i is admissible, one can write $g_i = \sum_{j=k}^{m} w_j f_j$ such that $\mathrm{HT}(w_k) = u$.

$$s_i w_k = r + \sum_{r \in G \,\mathscr{S}(r) \prec \mathbf{F}_k}^{m} \lambda_j \mathrm{poly}(g)$$

with $\mathrm{HT}(r) < \mathrm{HT}(s_i w_k)$ and $\mathrm{HT}(\lambda_j \mathrm{poly}(g)) \leq \mathrm{HT}(u_k u)$. Then

$$\begin{aligned} f &= \sum_{j \neq i} s_j g_j + s_i w_k g_k + \sum_{j=k+1}^{m} s_i w_j g_j \\ &= \sum_{j \neq i} s_j g_j + r g_k + \sum_{r \in G \;\mathscr{S}(r) \prec \mathbf{F}_k}^{m} g_k \lambda_j \mathrm{poly}(g) + \sum_{j=k+1}^{m} s_i w_j g_j \end{aligned}$$

This expression is $<_1 \mathbf{s}$ and there is a contradiction. Therefore all the (s_i, r_i) are normalized.
Let $w = max\{\mathscr{S}(s_i r_i) \,|, i \in \mathscr{I}\}$ and $\mathscr{J} = \{i \in \mathscr{I} \,|\, \mathscr{S}(s_i r_i) = w\}$. If $\#\mathscr{J} > 1$, since the r_i are admissible then for all $i \in \mathscr{J}$, $g_i = \sum_{j=j_0}^{m} w_{i,j} f_j$ with $\mathrm{HT}(s_i w_{i,j_0}) \mathbf{F}_{j_0} = w$. We can write f as follow:

$$\begin{aligned} f = \quad & \textstyle\sum_{i < min \mathscr{J}} s_i g_i + (\sum_{i \in \mathscr{J}} s_i w_{i,j_0}) g_{j_0} \\ & + (\textstyle\sum_{i \in \mathscr{J}} \sum_{j=j_0+1}^{m} w_{i,j} g_j + \sum_{i > max \mathscr{J}} s_i g_i) \end{aligned}$$

so we find another expression of f with is $<_1$ than $\mathbf{s}$. Consequently $\#\mathscr{J} = 1$ and let $k \in \mathscr{J}$ and $l \in \mathscr{I} \setminus \{k\}$. By construction we have $\mathscr{S}(s_l r_l) \prec \mathscr{S}(s_k r_k)$. We write f as follow:

$$f = s_k g_k - \frac{\mathrm{HC}(s_k)}{\mathrm{HC}(s_l)} s_l g_l + \left[1 + \frac{\mathrm{HC}(s_k)}{\mathrm{HC}(s_l)}\right] s_l g_l + \textstyle\sum_{i \neq k,l} s_i g_i$$

Let $m_k = \mathrm{HM}(s_k)$ and $m_l = \frac{\mathrm{HC}(s_k)}{\mathrm{HC}(s_l)} \mathrm{HM}(s_l)$ and $s'_i = s_i - \mathrm{HM}(s_i)$. Hence $t = \mathrm{HT}(m_k g_k) = \mathrm{HT}(m_l g_l)$, and consequently $\tau_{k,l} = lcm(\mathrm{HT}(g_k), \mathrm{HT}(g_l))$ divides t, that is to say:

$$m_k g_k - m_l g_l \quad = \frac{HC(s_k) t}{\tau_{k,l}} spol(g_k, g_l)$$

Since (s_k, g_k) and (s_l, g_l) are normalized we deduce that (g_k, g_l) is normalized, so that

$$\begin{aligned} m_k g_k - m_l g_l \quad &= \tfrac{t}{\tau_{k,l}} o_G(u_k r_k) \quad \text{where } u_k = \tfrac{\tau_{k,l}}{\mathrm{HT}(r_k)} \\ &= o_G(s_k r_k) \end{aligned}$$

Hence

$$f = o_G(s_k r_k) + s'_k g_k - \frac{\mathrm{HC}(s_k)}{\mathrm{HC}(s_l)} s'_l g_l + \alpha s_l g_l + \textstyle\sum_{i \neq k,l} s_i g_i$$

where $s'_i = s_i - \mathrm{HM}(s_i)$ $(\mathrm{HT}(s'_i) < \mathrm{HT}(s_i))$ and $\alpha = 1 + \frac{\mathrm{HC}(s_k)}{\mathrm{HC}(s_l)} \in \mathscr{K}$. This is a new expression of f which is $<_1 \mathbf{s}$. This is a contradiction and $t \leq \mathrm{HT}(f)$. So we can reduce f by an element of G_1. $f \xrightarrow[G_1]{*} 0$. □

REMARK 1. *In the theorem if we restrict (iii) to the critical pair of degree less than d we make the proof that G is Gröbner basis up to degree d.*

6. SIMPLIFICATION RULES

We describe now how to implement the simplification rules (for instance $x\mathbf{F}_2 \to f_6$ and $\mathbf{F}_2 \to f_4$ in the previous example).
We use an array Rule to store the rules. Each element of Rule is a list of elements of $T \times \mathbf{N}$. At the beginning there is no rules:

Reset simplification rules
  Input: m the number of polynomials
  for $i := 1, 2, \ldots, m$ **do**
    $\mathrm{Rule}[i] := \emptyset$

Add Rule $(r_k = (t\mathbf{F}_i, p) \in R)$
  $\mathrm{Rule}[i] := concat([[t, k]], \mathrm{Rule}[i])$

The following procedure try to simplify a product $u \times r_k$:

Rewritten $(u \in T$ a term, $r_k = (t\mathbf{F}_i, p) \in R)$
  $L := \mathrm{Rule}[i] = [[t_1, k_1], \ldots, [t_r, k_r]]$
  for $i = 1, \ldots, r$ **do**
    if ut divisible by t_i **then**
      return $(\frac{ut}{t_i}, r_{k_i})$
  return (u, r_k)

The following function return *true* if the $u \times r_k$ can be rewritten differently.

Rewritten? $(u \in T$ a term, $r_k = (t\mathbf{F}_i, p) \in R)$
  $(v, r_l) := \mathrm{Rewritten}(u, r_k)$
  return $l \neq k$

Example: If $r_4 = (\mathbf{F}_2, f_4)$ and $r_6 = (x\mathbf{F}_2, f_6)$ as in the previous example then $AddRule(r_4)$ and $AddRule(r_6)$ add two new rules $xF_2 \to f_6$ and $\mathbf{F}_2 \to f_4$. Now $\mathrm{Rewritten}(xy, r_4)$ returns (y, r_6) and $\mathrm{Rewritten?}(y^2, r_4)$ returns *true*.

7. DESCRIPTION OF THE ALGORITHM

7.1 The main algorithm

Since the algorithm is *incremental* the main loop of the algorithm iterates on the number of polynomials:

Algorithm incremental F_5

Input: $\begin{cases} F = (f_1, \ldots, f_m) \text{ a list of homogeneous} \\ \text{polynomials and } < \text{an admissible ordering} \end{cases}$

$N := m$ (the number of polynomials $r_1, \ldots, r_N$ occurring in the algorithm)

Reset simplification rules(m).

$r_m := (\mathbf{F}_m, f_m) \in R$, $G_m := [r_m]$

for $i := (m-1), \ldots, 1$ (in that order) **do**

$G_i := AlgorithmF_5(i, f_i, G_{i+1})$

return $\text{poly}(G) = \left[poly(r) \mid r \in G_1\right]$

In this algorithm the critical pairs are oriented:

DEFINITION 3. *The critical pair of* $(r_1, r_2) \in R^2$ *is*

$$CritPair(r_1, r_2) = (lcm_{r_1,r_2}, u_1, r_1, u_2, r_2)$$

(this is an element of $T^2 \times R \times T \times R$*) such that:*

$$\begin{aligned} lcm(CritPair(r_1, r_2)) &= lcm_{r_1,r_2} \\ &= u_1 HT(r_1) = u_2 HT(r_2) \\ &= lcm(HT(r_1), HT(r_2)) \end{aligned}$$

and

$$\mathscr{S}(u_1 r_1) \succ \mathscr{S}(u_2 r_2)$$

We say that the degree of such a critical pair is $deg(lcm_{r_1,r_2})$.

The basic version of our algorithm is now described. To simplify the presentation, we make the choice to describe the algorithm similarly to the description of the Buchberger algorithm, that is to say using polynomials and not linear algebra. However, from the efficiency point of view, it is recommended to translate the algorithm in a F_4 [6] fashion. The only structural difference with a standard Buchberger algorithm is that the reduction of one polynomial wrt a list of polynomials may return several polynomials. The algorithm uses 3 auxiliary functions: the definitions of "CritPair" (construction of critical pair if the new criterion cannot apply), "Spol" (construction of the Spolynomial), and "Reduction" (reduction of polynomials wrt the current list) are postponed until the end of this section:

Algorithm F_5

Input: $\begin{cases} i \text{ an integer and } f_i \text{ a polynomial} \\ G_{i+1} \text{ a finite subset of R,} \\ \quad \text{such that poly}(G_{i+1}) \text{ is a Gröbner basis} \\ \quad\quad \text{of } Id(f_{i+1}, \ldots, f_m) \end{cases}$

$r_i := (\mathbf{F}_i, f_i) \in R$

$\varphi_{i+1} = \text{NF}(., poly(G_{i+1}))$

$G_i := G_{i+1} \cup \{r_i\}$

$P :=$ Sort $\left[\text{CritPair}(r_i, r, i, \varphi_{i+1}) \mid r \in G_{i+1}\right])$ by degree

while $P \neq \emptyset$ **do**

$d := deg(first(P))$

$P_d := \{p \in P \mid deg(p) = d\}$

$P := P \backslash P_d$

$F := Spol(P_d)$

$R_d := Reduction(F, G_i, i, \varphi_{i+1})$

for $r \in R_d$ **do**

$P := P \cup \left\{\text{CritPair}(r, p, i, \varphi_{i+1}) \mid p \in G_i\right]\right)$

$G_i := G_i \cup \{r\}$

$P :=$ Sort P for the degree

return G_i

7.2 New criterion: implementation

We can now define the construction of a critical pair which implements the new criterion:

Algorithm CritPair (r_1, r_2, k, φ)

Input: $\begin{cases} k \text{ an integer} \\ r_1, r_2 \text{ polynomials in } R \\ \varphi \text{ a normal Form} \end{cases}$

$p_i := \text{poly}(r_i)$ for $i = 1, 2$

$t := lcm(\text{HT}(p_1), \text{HT}(p_2))$

$u_i := \dfrac{t}{\text{HT}(p_i)}$ for $i = 1, 2$

if $u_1 \mathscr{S}(r_1) \prec u_2 \mathscr{S}(r_2)$ **then**

Swap r_1 and r_2

$t_i \mathbf{F}_{k_i} := \mathscr{S}(r_i)$ for $i = 1, 2$

if $k_1 > k$ **then return** $\emptyset$

if $\varphi(u_1 t_1) \neq u_1 t_1$ **then return** $\emptyset$

if $k_2 = k$ and $\varphi(u_2 t_2) \neq u_2 t_2$ **then return** $\emptyset$

return $\left[t, u_1, r_1, u_2, r_2\right]$

Algorithm Spol

Input: $P = [p_1, \ldots, p_h]$ a list of critical pairs

Let $p_l = \left[t_l, u_l, r_{i_l}, v_l, r_{j_l}\right]$ for $l = 1, \ldots, h$

$F := \emptyset$

for l **from** 1 **to** h **do**

if and $\begin{matrix} (\text{not Rewritten?}(u_l, r_{i_l})) \\ (\text{not Rewritten?}(v_l, r_{j_l})) \end{matrix}$ **then**

$N := N + 1$

$r_N := (u_l \mathscr{S}(r_{i_l}), u_l \,\text{poly}(r_{i_l}) - v_l \,\text{poly}(r_{j_l}))$

Add Rule (r_N)

$F := F \cup \left\{r_N\right\}$

$F :=$ Sort F by increasing $\mathscr{S}$

return F

7.3 Reductions of polynomials

A major difference with Buchberger algorithm is that the reduction of a polynomial wrt a list of polynomials may return several polynomials so we have to modify the standard Reduction function: we use an auxiliary function TopReduction to perform an elementary reduction step. The result of TopReduction is a pair (r, F') where $r \in R$ and F' a list of polynomials. $F' = \emptyset$ means that r is irreducible (or zero). If $F' \neq \emptyset$ (then $r = \emptyset$) and it means that we have to rerun TopReduction on all the elements of F'.

Algorithm Reduction

Input: $\begin{cases} ToDo \text{ a finite list of polynomials} \\ G \text{ a list of polynomials of } R \\ k \text{ an integer} \\ \varphi \text{ a normal Form} \end{cases}$

$Done := \emptyset$

while $ToDo \neq \emptyset$ **do**

$h :=$ the minimal of $ToDo$ for $\mathscr{S}$

$ToDo := ToDo \backslash \{h\}$

$(h_1, ToDo_1) := TopReduction(\varphi(h), G \cup Done, k, \varphi)$

$Done := Done \cup h_1$

$ToDo := ToDo \cup ToDo_1$

return *Done*

To implement TopReduction we need a function to test the divisibility of the leading term of polynomial wrt a list of polynomials. The result is a reductor or $\emptyset$ if it is (top) irreducible.

Algorithm IsReducible

Input: $\begin{cases} r_{i_0} \text{ a polynomial of } R \\ G = [r_{i_1}, \ldots, g_{i_s}] \text{ where } g_i \in R \\ k \text{ an integer} \\ \varphi \text{ a normal Form} \end{cases}$

$t_j \mathbf{F}_{k_j} := \mathscr{S}(r_{i_j})\ j = 0, 1, \ldots, s$

for j **from** 1 **to** s **do**

if all the following conditions are true

(a) $u = \dfrac{\mathrm{HT}(r_{i_0})}{\mathrm{HT}(r_{i_j})}$ is a term (i.e. $u \in T$)

(b) $\varphi(ut_j) = ut_j$

(c) not Rewritten?(u, r_{i_j})

(d) $ut_j \mathbf{F}_{k_j} \neq t_0 \mathbf{F}_{k_0}$

then return r_{i_j}

return $\emptyset$

It is easy to give an interpretation of the four conditions:

(a) the usual divisibility test.

(b) test the new criterion: (u, r_{i_j}) is normalized.

(c) test if we can use a previous computation to avoid a waste of time (see the example in section 2).

(d) remove identical rows in the matrix.

Algorithm TopReduction

Input: $\begin{cases} r_{k_0} \text{ a polynomial of } R \\ G \text{ a list of polynomials of } R \\ k \text{ an integer} \\ \varphi \text{ a normal Form} \end{cases}$

if $\mathrm{poly}(r_{k_0}) = 0$ **then**

Warning "the system is not a regular sequence"

return $(\emptyset, \emptyset)$

$r' = IsReducible(r_{k_0}, G, k, \varphi)$

if $r' = \emptyset$ **then**

return $(\frac{1}{\mathrm{HC}(r_{k_0})} r_{k_0}, \emptyset)$

else

$r_{k_1} = r'$

$u = \dfrac{\mathrm{HT}(r_{k_0})}{\mathrm{HT}(r_{k_1})} \in T$

if $u\mathscr{S}(r_{k_1}) \prec \mathscr{S}(r_{k_0})$ **then**

$\mathrm{poly}(r_{k_0}) = \mathrm{poly}(r_{k_0}) - u\,\mathrm{poly}(r_{k_1})$

return $(\emptyset, \{r_{k_0}\})$

else

$N := N + 1$

$r_N = (u\mathscr{S}(r_{k_1}), u\,\mathrm{poly}(r_{k_1}) - \mathrm{poly}(r_{k_0})) \in R$

Add Rule (r_N)

return $(\emptyset, \{r_N, r_{k_0}\})$

7.4 Proof of the algorithm

Let $\tilde{R}$ be the set of all the polynomials occurring in the execution of the algorithm. In the following we give a proof of the termination in a restricted case (when there is no reduction to zero) but it is possible to modify slightly the F_5 algorithm so that we can always ensure the termination of the algorithm in all the cases.

PROPOSITION 2. *For all $r \in \tilde{R}$, r is admissible.*

PROOF. By induction on m. Then $r_1 = (\mathbf{F}_1, f_1)$ is obviously admissible. The operation to construct a new $r \in \tilde{R}$ is $r' = (\mathscr{S}(r_k), \mathrm{poly}(r_k) - u\,\mathrm{poly}(r_l))$ where $\mathscr{S}(r_k) > \mathscr{S}(ur_l)$ and r_k, r_l admissible. Hence we can write $r_i = \sum_{j=1}^{m} s_{i,j} f_j$ $(i = k, l)$ such that $\mathrm{HT}(s_{k,1})\mathbf{F}_1 = \mathscr{S}(r_k)$. Hence $\mathrm{poly}(r') = \sum_{j=1}^{m} (s_{k,j} - u s_{l,j}) f_j$ and $\mathrm{HT}(s_{k,1} - u s_{l,1}) = \mathrm{HT}(s_{k,1})$ since $\mathrm{HT}(us_{l,1})$ is zero or is less than $\mathrm{HT}(s_{k,1})$. r' is admissible. □

PROPOSITION 3. *If G_{i+1} is a Gröbner basis of $Id(f_{i+1}, \ldots, f_m)$, then all the polynomials occurring in Algorithm $F_5(i, f_i, G_{i+1})$ are normalized.*

THEOREM 2. *For all d, the result of the algorithm F_5 is a (non reduced) Gröbner basis up to degree d.*

PROOF. The proof is by induction on m the number of polynomials. We suppose that G_2 is a Gröbner basis up to degree d and we want to proof that G_1 is a Gröbner basis up to degree d. For all (r, r') such as in theorem 1, let r'' be the result of the reduction of $\mathrm{spol}(r, r')$ by G_1. Let τ be $lcm(\mathrm{HT}(r), \mathrm{HT}(r'))$ and u be $\frac{\tau}{\mathrm{HT}(r)}$. We have

$$\begin{array}{l} \mathscr{S}(r'') = u\mathscr{S}(r) \\ \text{and } \mathrm{HT}(\mathrm{poly}(r'')) < lcm(\mathrm{HT}(r), \mathrm{HT}(r')) = u\mathrm{HT}(r) \end{array}$$

so that

$$\mathrm{spol}(r, r') = r'' + o_{G_1}(ur) = o_{G \cup r''}(ur)$$

From proposition 2 and proposition 3 we can apply theorem 1 and we deduce that G_1 is a Gröbner basis of the ideal generated by $(f_1, \ldots, f_m)$ (up to degree d). □

THEOREM 3. *We suppose that all the f_i are homogeneous and that there is no reduction to zero. For all d, the result of Reduction in the algorithm F_5 is R_d. Then $Id(HT(G_i)) \neq Id(HT(G_i \cup R_d))$.*

COROLLARY 2. *This makes the proof of the termination of the algorithm F_5.*

PROOF. Without loss of generality we can suppose than $i = 1$ in the algorithm F_5 and G_2 the result of the algorithm on $[f_2, \ldots, f_m]$ which is a Gröbner basis by previous theorem. Let $u\mathbf{F}_1$ be the maximum of $\{\mathscr{S}(r) \mid r \in R_d\}$, so there exists $r \in R_d$ such that $\mathscr{S}(r) = u\mathbf{F}_1$. Suppose for a contradiction that there is $r' \in G_1 \cup R_d \backslash \{r\}$ such that $u = \frac{\mathrm{HT}(r)}{\mathrm{HT}(r')} \in T$. If $u\mathscr{S}(r')$ is not top reducible by G_2 then

a) if $u\mathscr{S}(r') > \mathscr{S}(r)$ then the critical pair $(r', r) = (u, r', 1, r)$ was introduced in the list and since there is no reduction to zero $ur' \in R_d$. This is a contradiction since r was the maximum.

b) if $u\mathscr{S}(r') < \mathscr{S}(r)$ then r can be reduced by r'. Contradiction.

We have to study now the case $u\mathscr{S}(r')$ top reducible by G_2. Since r' is admissible $r' = \sum_{i=1}^{m} s'_i f_i$ with $\mathrm{HT}(s'_1)\mathbf{F}_1 = \mathscr{S}(r')$ and $us'_1 = v + \sum_{i=2}^{m} \lambda_i f_i$ where v is fully reduced by G_2 so that $\mathrm{HT}(v) < \mathrm{HT}(us'_1)$

$$\begin{aligned} u\mathrm{poly}(r') &= us'_1 f_1 + \textstyle\sum_{i=2}^{m} us'_i f_i \\ &= v f_1 + \textstyle\sum_{i=2}^{m} (\lambda_i f_1 + us'_i) f_i \end{aligned}$$

Let $\mathscr{T} = \{\mathrm{HT}(t f_1) > \mathrm{HT}(r) \,|\, t \in T(v)\}$. If $\mathscr{T}$ is non empty then for all $t \in \mathscr{T}$, (t, f_1) is normalized so that it should have been put in the list of critical pair. In the reduction process we find a polynomial r'' such that $\mathscr{S}(r'') = \mathrm{HT}(v)\mathbf{F}_1$ and $\mathrm{HT}(r'') = \mathrm{HT}(r)$. Contradiction. If $\mathscr{T} = \emptyset$, then $\sum_{i=2}^{m} (\lambda_i f_1 + us'_i) f_i = \sum_{g \in G_2} \mu_g g$ where $\mathrm{HT}(\mu_g g) \leq \mathrm{HT}(r)$. Hence $\mathrm{HT}(r) = \mathrm{HT}(\mu_g g)$ for some $g \in G_2$ or $\mathrm{HT}(r) = \mathrm{HT}(v f_1)$. So we can reduced r by $G_2 \cup \{f_1\}$. □

THEOREM 4. *If the algorithm finds a reduction to zero, $r_{i_k} \to 0$ then there exists $\mathbf{s} \in Syz \backslash PSyz$ with $HT(\mathbf{s}) = \mathscr{S}(r_{i_k})$.*

PROOF. We may suppose wlog that $\mathscr{S}(r_{i_k}) = t\mathbf{F}_1$ for some $t \in T$. Now for all $\mathbf{s} \in \mathrm{PSyz}$ with $\mathrm{index}(\mathbf{s}) = 1$ we have

$$\begin{aligned} \mathbf{s} &= \textstyle\sum_{i=1}^{m} \sum_{j=i+1}^{m} \lambda_{i,j} \mathbf{s}_{i,j} \\ &= \textstyle\sum_{i=1}^{m} \sum_{j=i+1}^{m} \lambda_{i,j} f_j \mathbf{F}_i - \sum_{i=1}^{m} \sum_{j=i+1}^{m} \lambda_{i,j} f_i \mathbf{F}_j \\ &= \textstyle\sum_{j=2}^{m} \lambda_{1,j} f_j \mathbf{F}_1 + \sum_{i=2}^{m} (\cdots) \mathbf{F}_i \end{aligned}$$

Consequently $\mathrm{HT}(\mathbf{s}) = \mathrm{HT}(\sum_{j=2}^{m} \lambda_{1,j} f_j)\mathbf{F}_1$, that is to say $\mathrm{HT}(\mathbf{s}) \in \mathrm{HT}(Id(f_2, \ldots, f_m))$. Hence if $r_{i_k} = 0$, then $\mathscr{S}(r_{i_k}) = \mathrm{HT}(\mathbf{s})$ for some $s \in Syz$. Since r_{i_k} is normalized $\mathrm{HT}(s) \notin \mathrm{HT}(Id(f_2, \ldots, f_m))$, hence $\mathbf{s} \notin PSyz$. □

COROLLARY 3. *If the input system is a regular sequence there is no reduction to zero.*

8. EXAMPLE

We compute one example from [10] in full. We are using the Degree Reverse Lexico ordering $x > y > z > t$ and the coefficients are rational numbers.

$$\begin{aligned} f_3 &= x^2 y - z^2 t \\ f_2 &= xz^2 - y^2 t \\ f_1 &= yz^3 - x^2 t^2 \end{aligned}$$

The algorithm computes successively Gröbner bases of (f_3), (f_3, f_2) and (f_3, f_2, f_1). Since the last computation is the most difficult we may skip these first steps. The corresponding Gröbner bases are

$G_3 = [r_3]$ and $G_2 = [r_3, r_2, r_4, r_5]$ where $r_3 = (\mathbf{F}_3, f_3)$, $r_2 = (\mathbf{F}_2, f_2)$, $r_4 = (xy\mathbf{F}_2, xy^3 t - z^4 t)$, $r_5 = (xyz^2\mathbf{F}_2, z^6 t - y^5 t^2)$.

$\varphi_2 = \mathrm{NormalForm}(., [r_3, r_2, r_4, r_5])$
$r_1 = (\mathbf{F}_1, f_1)$
$G_1 = G_2 \cup \{r_1\} = [r_3, r_2, r_4, r_5, r_1]$
There are four critical pairs: $p_7 = (xyz^3, x, r_1, yz, r_2)$, $p_8 = (x^2 yz^3, x^2, r_1, z^3, r_3)$, $p_9 = (yz^6 t, z^3 t, r_1, y, r_5)$, $p_{10} = (xy^3 z^3 t, xy^2 t, r_1, z^3, r_4)$. $\mathscr{S}(p_7), \ldots, \mathscr{S}(p_{10})$ are resp. $x\mathbf{F}_1, x^2\mathbf{F}_1, z^3\mathbf{F}_1, xy^2\mathbf{F}_1$ are all invariants by φ_2.
$P = [p_7, p_8, p_9, p_{10}]$
$\boxed{d = 5}$, enter Spol(P_5) with $P_5 = [p_7]$ and $P = [p_8, p_9, p_{10}]$
$r_6 = (x\mathbf{F}_1, y^3 zt - x^3 t^2)$ and $F := [r_6]$
We add a new rule $x\mathbf{F}_1 \to r_6$
There is obviously no reduction of r_6 by G_1 so the returned result is $R_5 = [r_6]$
$G_1 = [r_3, r_2, r_4, r_5, r_1, r_6]$
We update the list of critical pairs: $p_{11} = (y^3 z^3 t, z^2, r_6, y^2 t, r_1)$, $p_{12} = (y^3 z^6 t, z^5, r_6, y^3, r_5)$, $p_{13} = (xy^3 zt, x, r_6, z, r_4)$, $p_{14} = (x^2 y^3 zt, x^2, r_6, y^2 zt, r_3)$, $p_{15} = (xy^3 z^2 t, xz, r_6, y^3 t, r_2)$. We check that $\mathscr{S}(z^2 r_6) = xz^2\mathbf{F}_1$ and $\mathscr{S}(z^5 r_6) = xz^5\mathbf{F}_1$ are reducible by φ_2 so that the pairs p_{11} and p_{12} are rejected.
Hence $P = [p_8, p_9, p_{10}, p_{13}, p_{14}, p_{15}]$.
$\boxed{d = 6}$, enter Spol(P_6) with $P_6 = [p_8, p_{13}]$ and $P = [p_9, p_{10}, p_{14}, p_{15}]$
We check that Rewritten$(x^2, r_1) = (x, r_6)$ so we do not keep p_8
For the other pair p_{13}: Rewritten?$(x, r_6) = false$ and Rewritten?$(z, r_4) = false$ so that $r_7 = (x^2\mathbf{F}_1, z^5 t - x^4 t^2)$
We add a new rule $x^2\mathbf{F}_1 \to r_7$
There is obviously no reduction of r_7 by G_1 so the returned result is $R_6 = [r_7]$
$G_1 = [r_3, r_2, r_4, r_5, r_1, r_6, r_7]$
Among all the critical pairs we check as usual that (r_7, r_1), (r_7, r_6), (r_7, r_3) and (r_7, r_4) are not valid.
The new critical pairs are $p_{16} = (z^6 t, z, r_7, 1, r_5)$ and $p_{17} = (xz^5 t, x, r_7, z^3 t, r_2)$.
$\boxed{d = 7}$, enter Spol(P_7) with $P_7 = [p_{15}, p_{16}, p_{17}, p_{14}]$ and $P = [p_9, p_{10}]$
We check that Rewritten$(xz, r_6) = (z, r_7)$ so we do not keep p_{15}
p_{16} is valid and $r_8 = (x^2 z\mathbf{F}_1, y^5 t^2 - x^4 zt^2)$ is computed
We add a new rule $x^2 z\mathbf{F}_1 \to r_8$
p_{17} is valid and $r_9 = (x^3\mathbf{F}_1, -x^5 t^2 + y^2 z^3 t^2)$ is computed
We add a new rule $x^3\mathbf{F}_1 \to r_9$
We check that Rewritten$(x^2, r_6) = (1, r_9)$ so we do not keep p_{14}
There are two Spolys to reduce $F = [r_8, r_9]$
The elements of F are not top reducible by G_1 as described in the algorithm but it is possible to *fully* reduce r_9 by $yt^2 \times r_1$: $r_9 = (x^3\mathbf{F}_1, -x^5 t^2 + x^2 yt^4)$ and the final result is $r_9 = -\varphi_2(r_9) = (x^3\mathbf{F}_1, x^5 t^2 - z^2 t^5)$
The result of Reduction is $R_7 = [r_9, r_8]$
$G_1 = [r_3, r_2, r_4, r_5, r_1, r_6, r_7, r_8, r_9]$
The critical pairs (r_9, r_1), (r_9, r_6), (r_9, r_7), (r_9, r_2), (r_9, r_3), (r_9, r_4), (r_9, r_5), (r_8, r_1), (r_8, r_6), (r_8, r_7), (r_8, r_9), (r_8, r_2) and (r_8, r_5) are not valid.
The new critical pairs are $p_{18} = (xy^5 t^2, x, r_8, y^2 t, r_4)$ and $p_{19} = (x^2 y^5 t^2, x^2, r_8, y^4 t^2, r_3)$.
$\boxed{d = 8}$, enter Spol(P_8) with $P_8 = [p_9, p_{10}, p_{18}]$ and $P = [p_{19}]$
p_9 is valid and $r_{10} = (z^3 t\mathbf{F}_1, y^6 t^2 - x^2 z^3 t^3)$ is computed
We add a new rule $z^3 t\mathbf{F}_1 \to r_{10}$
We check that Rewritten$(xy^2 t, r_1) = (y^2 t, r_6)$ so we do not keep p_{10}
We check that Rewritten$(x, r_8) = (z, r_9)$ so we do not keep p_{18}
Now $r_{10} = \varphi_2(r_{10}) = (z^3 t\mathbf{F}_1, y^6 t^2 - xy^2 zt^4)$ is fully reduced, the result is $R_8 = [r_{10}]$.
$G_1 = [r_3, r_2, r_4, r_5, r_1, r_6, r_7, r_8, r_9, r_{10}]$
All the new possible critical pairs (r_{10}, r_i) $(i = 1, \ldots, 8)$ are rejected
$\boxed{d = 9}$, enter Spol(P_9) with $P_9 = [p_{19}]$ and $P = \emptyset$
We check that Rewritten$(x^2, r_8) = (xz, r_9)$ so we do not keep p_{19}
$F = \emptyset$ and $R_9 = \emptyset$
The algorithm stops and returns G_1.
Remark that no useless pair has remained. With the Buchberger algorithm (resp. the algorithm [10]) there was 7 (resp. 1) useless pairs and 5 (resp. 5) useful ones.

Example	[5]	[7]	[10]	F_5	remark
Raksanyi	1	?	0	**0**	
Hairer1	10	?	4	**0**	
Rose	22	19	?	**0**	
Trinks6	17	8	6	**0**	
Trinks7	12	11	6	**4**	over constrained
Katsura3	1	?	1	**0**	
Katsura4	18	10	7	**0**	
Katsura5	50	28	?	**0**	
Katsura10	3936	?	?	**0**	
Binary10	2147	?	?	**0**	
Noon8	7886	?	?	**0**	
Eco 6	61	?	?	**7**	see text
Eco 6 fact	63	?	?	**0**	
Eco 8 fact	315	?	?	**0**	

Fig 9.1: Number of useless critical pairs

Cyclic	7	8	9	10
F_4	1.26	36.0	4949.1	
F_4 inc	1.4	171.3		
F_5	1.0	27.9		
F_5'	0.4	7.2	1002.3	57600
F_5''	0.8	3.95	676.2	

Fig 9.2: Comparison of F_4 and F_5 for the Cyclic n problem modulo p (Inspiron PIII 1Ghz): CPU Time in seconds.

9. EXPERIMENTAL RESULTS

9.1 Number of useless pairs

This is interesting to compare the number of useless critical pairs in practice for the various algorithms because this number does not depend on the implementation (at least for the F_5 algorithm). The first line of the following tabular (figure 9.1) contains all the examples of [7] and [10] the other are well known. Note that reductions to zero are unavoidable for `Trinks7` (7 equations, 6 variables). The table brought the `Eco n` to our attention since the number of useless pairs is not zero: we found that the system can be straightforwardly rewritten by factorizing the original equations. By reformulating these problem we obtain an equivalent system `Eco n fact` with no reduction to zero ! The conclusion is that for a lot of practical examples there is no reduction to zero.

9.2 First implementation

A first implementation of the F_5 has been made in the Maple computer algebra system and then translated in Gb (C++) and FGb (C). From a traditional implementation of the Buchberger algorithm it is very easy to implement the new algorithm: the only data type to modify is to add to the property list of each polynomial r an integer (the index k of r) and a power product t ($\mathscr{S}(r) = r\mathbf{F}_k$). Hence the extra memory cost is very small. The behavior of the algorithm is very good: it is at least one order of magnitude faster than the fastest known algorithm/implementation (F_4) and two order of magnitude faster than one of the fastest programs (Singular 2.0 [8]). In tabular 9.2 we give the timings for the well known cyclic n problem: a Gröbner basis of Cyclic 10 was computed for the first time. In table 9.2 "F_4 inc" is the F_4 algorithm applied incrementally. "F_5'" and "F_5''" are different version of the F_5 algorithm that will be described in a future paper. We report now detailed CPU timings for the Katsura n problem modulo a small prime p (there is no useless pairs for this example).

The algorithm F_5 is not always faster than F_4: for cyclic n the basic version of the F_5 algorithm is just a little faster than F_4; the maximal efficiency of the F_5 algorithm is expected when the number of equation is less or equal than the number of variables. On the contrary bad performance is expected when the system is overconstrained: for instance compute a Gröbner basis for a total degree and then rerun the F_5 algorithm on the result.

n	Singular	Gb	F_4	F_5	Singular
	2-0-0				1-2-3
7	1.6	2.2	0.4	**0.15**	3.1
8	13.6	22.25	2.8	**0.8**	36.4
9	135.3	252.5	23.1	**4.1**	411.2
10	1140.2	2907.1	220.2	**25.5**	4311.8
11	11671	34903	2097	**174.2**	58174.6
12			25161	**1460.7**	
13			240667	**10748**	

Fig 9.2: Katsura n PII 400 Mhz (CPU time in seconds)

n	F_4/F_5	Sing/Gb	Gb/F_4	O Sing/Gb	**Sing**/F_5
7	2.7	0.7	5.2	1.4	**10.6**
8	3.3	0.6	8.0	1.6	**16.4**
9	5.6	0.5	10.9	1.6	**33.1**
10	8.6	0.4	13.2	1.5	**44.8**
11	12.0	0.3	16.6	1.7	**67.0**
12	17.2				
13	22.4				

Fig 9.2: Katsura n PII 400 Mhz (Speedup)

In the following tables 9.2 we compute the speedup: for instance `O Sing/Gb` is the CPU time for the old version of Singular (1-2-3) divided by the CPU of Gb on the same example.

Acknowledgements

We would like to thank the FRE 2341 Medicis.

10. REFERENCES

[1] ADAMS, W., AND LOUSTAUNAU, P. *An introduction to Gröbner Bases*, vol. 3 of *Graduate Studies in Mathematics*. American Mathematical Society, 1994.

[2] BECKER T. AND WEISPFENNING V. *Groebner Bases, a Computationnal Approach to Commutative Algebra*. Graduate Texts in Mathematics. Springer-Verlag, 1993.

[3] BUCHBERGER B. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.* PhD thesis, Innsbruck, 1965.

[4] BUCHBERGER B. A Criterion for Detecting Unnecessary Reductions in the Construction of Gröbner Basis. In *Proc. EUROSAM 79* (1979), vol. 72 of *Lect. Notes in Comp. Sci.*, Springer Verlag, pp. 3–21.

[5] BUCHBERGER B. Gröbner Bases : an Algorithmic Method in Polynomial Ideal Theory. In *Recent trends in multidimensional system theory*, Reidel, Ed. Bose, 1985.

[6] FAUGÈRE J.C. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra 139*, 1–3 (June 1999), 61–88.

[7] GEBAUER, R., AND MOLLER, H. M. Buchberger's algorithm and staggered linear bases. In *Proceedings of the 1986 Symposium on Symbolic and Algebraic Computation* (July 1986), pp. 218–221.

[8] GREUEL G.-M. AND PFISTER G. AND SCHOENEMANN H. *SINGULAR 2.0*, July 2002. `http://www.singular.uni-kl.de/`.

[9] LAZARD D. Gaussian Elimination and Resolution of Systems of Algebraic Equations. In *Proc. EUROCAL 83* (1983), vol. 162 of *Lect. Notes in Comp. Sci*, pp. 146–157.

[10] MORA, T. AND MÖLLER, H.M. AND TRAVERSO, C. Gröbner Bases Computation Using Syzygies. In *ISSAC 92* (July 1992), P. S. Wang, Ed., ACM Press, pp. 320–328.

[11] V.P. GERDT AND YU.A.BLINKOV. Involutive bases of polynomial ideals. *athematics and Computers in Simulation 45* (1998), 519–542.