J. Symbolic Computation (2000) **30**, 429–449
doi:10.1006/jsco.1999.0415
Available online at http://www.idealibrary.com on IDEAL

Computing Gröbner Bases by FGLM Techniques in a Non-commutative Setting

M. A. BORGES-TRENARD[†§], M. BORGES-QUINTANA[†] AND T. MORA[‡¶]

[†] *Department of Mathematics, Faculty of Sciences, University of Oriente, Santiago de Cuba 90500, Cuba*
[‡] *DISI, University of Genova, Italy*

A generalization of the FGLM technique is given to compute Gröbner bases for two-sided ideals of free finitely generated algebras. Specializations of this algorithm are presented for the cases in which the ideal is determined by either functionals or monoid (group) presentations. Generalizations are discussed in order to compute Gröbner bases on (twisted) semigroup rings.

© 2000 Academic Press

1. Introduction

It is well known that the complexity of Gröbner bases computation strongly depends on the term ordering, moreover, elimination orderings often yield a greater complexity. This remark led to the so-called FGLM conversion problem, i.e. **given** a Gröbner basis w.r.t. a certain term ordering,[‖] **find** a Gröbner basis of the same ideal w.r.t. another term ordering. One of the efficient approaches for solving this problem, in the zero-dimensional case, is the FGLM algorithm (see Faugère *et al.*, 1993).

The key ideas of this algorithm were successfully generalized in Marinari *et al.* (1993) with the objective of computing Gröbner bases of zero-dimensional ideals that are determined by functionals (in the sense that they are kernels of finite sets of linear morphisms from the polynomial ring to the base field). In fact Buchberger and Möller (1982) pioneered the work of FGLM and these algorithms.

The main goal of this paper is to generalize the FGLM algorithm to non-commutative polynomial rings.[**] Before giving a brief summary of the sections of this paper, let us introduce some familiar notation.

$X := \{x_1, \ldots, x_n\}$	finite alphabet
$\langle X \rangle$	free monoid on X
1	the empty word in $\langle X \rangle$
$K\langle X \rangle$	free associative K algebra on X (K a field)
I	two-sided ideal of $K\langle X \rangle$

[§] E-mail: {mborges, mijail}@csd.uo.edu.cu
[¶] E-mail: theomora@dima.unige.it

[‖] Usually, it is a total degree ordering, where computing complexity is lower.

[**] The theory presented here in the case of two-sided ideals can be generalized to left-modules of non-commutative polynomial rings (see Alonso *et al.*, 1995).

0747–7171/00/100429 + 21 $35.00/0 © 2000 Academic Press

$Ideal(F)$	two-sided ideal of $K\langle X\rangle$ generated by $F \subset K\langle X\rangle$
$K\langle X\rangle/I$	residue class algebra of $K\langle X\rangle$ modulo I
$L(s)$	length of the word $s \in \langle X\rangle$
$\mathrm{Card}(C)$	cardinal of the set C

1.1. OVERVIEW

Section 2 deals with basic Gröbner theory. The partition of $\langle X\rangle$ in different regions which is induced by a a semigroup ideal is characterized. In particular, the notion of Border bases is generalized from Marinari *et al.* (1993); these are specific Gröbner bases that allow us to compute canonical forms in polynomial time. In Section 3 we introduce our main algorithm (Algorithm 10); it is presented in such a fashion that makes essential ideas of algorithms like FGLM clear and, at the same time, allows us to specialize it on several particular settings. Section 4 generalizes the pattern, introduced in Marinari *et al.* (1993), of computing Gröbner bases for ideals that are determined by functionals. Three cases are shown that are compatible with this approach. Section 5 shows that the viewpoint of Section 4 is not general enough, that is, there are instances where Algorithm 10 can be itemized in a better way than the one given in Section 4, covering finite monoids given by concrete representation that allow word multiplication by generators and recovering the ideas introduced in Labontè (1990). All the algorithms that are designed thus far turn out to be polynomial in their input (number of variables, dimension of the corresponding residue class vector space, maximal length of the words in canonical form, etc.). Lastly, in Section 6, some considerations are given in order to design algorithms like FGLM for (twisted) semigroup rings. Almost everywhere in this paper I is considered to be zero-dimensional; consequently, $K\langle X\rangle/I$ will have finite dimension in that case.

2. Border Bases for Two-sided Ideals

The main objective of this section is to introduce non-commutative Gröbner bases techniques for algorithms like FGLM and, in particular, generalize the notion of border bases on that setting. This notion appears for the first time in Faugère *et al.* (1993) where its information is essentially contained in the so-called Matphi function; subsequently, its formal definition was given in Marinari *et al.* (1993). The new results that are included in this section are Proposition 1 and from Definition 6 on. Proposition 1 is outside Gröbner bases theory, but contributes to it because the set of the maximal terms of an ideal is a semigroup ideal. Definition 6 and the subsequent results deal with border bases. On the other hand, from Theorem 2 to Definition 5, the reader will just find well known Gröbner bases tools.

Let $\tau \subset \langle X\rangle$ be a semigroup ideal of $\langle X\rangle$, i.e. for $s, u \in \langle X\rangle$ and $t \in \tau$, $stu \in \tau$. Then, it is well known that τ has a unique set $G(\tau)$ of irredundant generators (probably infinite). We are going to introduce for τ some notation and terminology, which are similar to those introduced in Marinari *et al.* (1993). The difference, in the non-commutative case, is that the border of τ is divided into two one-sided borders, each of them is enough in order to generate τ and their intersection is $G(\tau)$.

For $s := x_{i_1} \cdots x_{i_m} \in \langle X\rangle$ we set:

$$rr(s) := \begin{cases} 1 & \text{for } m = 1 \\ x_{i_2} \cdots x_{i_m} & \text{otherwise} \end{cases} \quad (\textit{right rest of } s),$$

$$lr(s) := \begin{cases} 1 & \text{for } m = 1 \\ x_{i_1} \cdots x_{i_{m-1}} & \text{otherwise} \end{cases} \quad (\textit{left rest of } s).$$

Then, let

$$\begin{aligned} N(\tau) &:= \{s \in \langle X \rangle \mid s \notin \tau\} && (\textit{outside of } \tau), \\ rB(\tau) &:= \{t \in \tau \mid rr(t) \in N(\tau)\} && (\textit{right border of } \tau), \\ lB(\tau) &:= \{t \in \tau \mid lr(t) \in N(\tau)\} && (\textit{left border of } \tau), \\ B(\tau) &:= rB(\tau) \cup lB(\tau) && (\textit{border of } \tau), \\ I(\tau) &:= \tau \setminus B(\tau) && (\textit{interior of } \tau). \end{aligned}$$

We remark that $t \in \tau$ lies in $G(\tau)$ if all its proper divisors are in $N(\tau)$. In the following proposition, some basic results concerning τ and its regions are summarized. Although they are very easy to prove, their importance is crucial for non-commutative FGLM techniques.

Proposition 1. (Properties of the Semigroup Ideal Regions) (i) *For each* $u \in \tau$ *there exist* $s_1 \in \langle X \rangle$ *(*$s_2 \in \langle X \rangle$*) and* $t_1 \in rB(\tau)$ *(*$t_2 \in lB(\tau)$*) such that* $u = s_1 t_1$ *(*$u = t_2 s_2$*).*

(ii) *For* $x \in X$*:*

- *If* $s \in N(\tau)$*, then* $xs \in N(\tau) \cup rB(\tau)$ *and* $sx \in N(\tau) \cup lB(\tau)$*.*
- *If* $s \in rB(\tau)$ *(*$s \in lB(\tau)$*), then* $sx \in rB(\tau) \cup I(\tau)$ *(*$xs \in lB(\tau) \cup I(\tau)$*).*
- *If* $s \in I(\tau)$*, then* $xs, sx \in I(\tau)$*.*

(iii) $N(\tau), N(\tau) \cup G(\tau), N(\tau) \cup B(\tau)$ *are order ideals, i.e. if* u *belongs in one of these subsets and* s *divides* u*, then* s *also belongs to those sets.*

(iv) $G(\tau) = rB(\tau) \cap lB(\tau)$*.*

Now let $<$ be a semigroup total well ordering on $\langle X \rangle$ (such an ordering is also called admissible), then the following notations are quite familiar: For $f \in K\langle X \rangle \setminus \{0\}$, $T_<(f)$ is the maximal term of f w.r.t. $<$, $LC_<(f)$ is the leading coefficient of f w.r.t. $<$. Similarly, for $F \subset K\langle X \rangle$, $T_<\{F\}$ is the set of maximal terms of non-zero polynomials in F, $T_<(F)$ is the semigroup two-sided ideal generated by $T\{F\}$. Moreover, for the sake of simplicity in notation, $U_<(F)$ will be used instead of $U(T_<(F))$, where U lies in $\{G, N, rB, lB, B, I\}$.[†]

Theorem 2. (The Vector Space of Canonical Forms Modulo an Ideal) *Let* $\mathrm{Span}_K(N_<(I))$ *be the* K*-vector space whose basis is* $N_<(I)$*. Then the following holds:*

(i) $K\langle X \rangle = I \oplus \mathrm{Span}_K(N_<(I))$ *(this sum is considered as a direct sum of vector spaces).*

(ii) *For each* $f \in K\langle X \rangle$ *there is a unique polynomial of* $\mathrm{Span}_K(N_<(I))$*, denoted by* $\mathrm{Can}(f, I, <) = \mathrm{Can}(f, I)$*, such that* $f - \mathrm{Can}(f, I, <) \in I$*; moreover:*

- $\mathrm{Can}(f, I, <) = \mathrm{Can}(g, I, <)$ *iff* $f - g \in I$*.*

[†]Of course, given an ideal I and two different admissible orderings $<$ and $\prec$, in general we have $U(T_<(I)) \neq U(T_\prec(I))$ for all U. Notwithstanding this strong dependence on $<$, while a single admissible ordering $<$ is considered, so that no confusion can arise, we will often simply write $U(F)$ for $U_<(F)$.

- $\mathrm{Can}(f, I, <) = 0$ *iff* $f \in I$.

(iii) *There is a* K*-vector space isomorphism between* $K\langle X\rangle/I$ *and* $\mathrm{Span}_K(N_<(I))$ *(the isomorphism associates the class of* f *modulo* I *with the canonical form* $\mathrm{Can}(f, I, <)$ *of* f *modulo* I*).*

$\mathrm{Can}(f, I, <) = \mathrm{Can}(f, I)$ is called the canonical (normal) form of f modulo I (and the dependence on $<$ is omitted if no confusion arises).

The following definitions and results (Theorem 3, Proposition 4, and Definition 5) belong to the non-commutative Gröbner bases theory on free algebras (cf. Mora, 1994).

Theorem 3. (Some Characterizations of Gröbner Bases) *Let* $F \subset I \setminus \{0\}$. *Then, the following properties are equivalent:*

(i) $T_<(F) = T_<(I)$.
(ii) $N_<(F)$ *is a* K*-basis of* $\mathrm{Span}_K(N_<(I))$.
(iii) $\{\pi(s) \mid s \in N_<(F)\}$ *is a* K*-basis of* $K\langle X\rangle/I$*, where* $\pi : K\langle X\rangle \to K\langle X\rangle/I$ *is the canonical projection.*

A subset F with the above properties is called a Gröbner basis of I w.r.t. the given term ordering $<$.

Proposition 4. (Characterization of Zero-dimensional Ideals) *Let* F *be a Gröbner basis of* I *w.r.t.* $<$*. Then,* I *is a zero-dimensional ideal (i.e.* $\dim_K K\langle X\rangle/I < \infty$*) iff* $N_<(F)$ *is finite. Moreover, in such a case,* $\dim_K K\langle X\rangle/I = \mathrm{Card}(N_<(F))$.

We will set, for the zero dimensional case, $d := \dim_K K\langle X\rangle/I$.

Definition 5. (Reduced Gröbner Basis) A subset $F \in I \setminus \{0\}$ is called the reduced Gröbner basis of I w.r.t. $<$ if the following holds.

(i) $T_<\{F\} = G_<(I)$ (i.e. $T_<\{F\}$ is the set of irredundant generators of $T_<(I)$).
(ii) For $f \in F$, $f = T_<(f) - \mathrm{Can}(T(f), I, <)$.

We will denote by $rGb(F, <)$ the reduced Gröbner basis of $\mathrm{Ideal}(F)$ with respect the $<$, and by $rGb(F)$ when there is no reason to specify $<$.

The computation of $\mathrm{Can}(f, \mathrm{Ideal}(F), <)$, where F is a finite Gröbner basis, may be carried out by means of a reduction procedure; however, this method has proved to be inefficient (cf. Faugère *et al.*, 1993). In a commutative polynomial ring a more efficient approach was proposed in Faugère *et al.* (1993) and formalized via the notion of border basis (B-basis) of an ideal in Marinari *et al.* (1993, Definition 3.9).† Shortly, the B-basis of an ideal I is a certain Gröbner basis of I, which contains $rGb(I, <)$ and whose set

†In the non-commutative case, the notion of (left) Border basis essentially coincides with the one of prefix Gröbner basis introduced in Madlener and Reinert (1998) and Reinert (1995). In the commutative case, it is strictly related with the notion of Janet basis introduced by Zharkov (1996) and the papers cited there and also Apel (1998), as a generalization of Janet's (1929) theory of partial differential equations.

of maximal terms is the border of $T_<(I)$; it allows the user to compute $\mathrm{Can}(f, I, <)$ in polynomial time. Border bases can be successfully generalized to non-commutative free algebras as is shown from Definition 6 to Remark 9.

Definition 6. (Right (Left) Border Basis) *The right (left) border basis of I w.r.t. $<$ is the subset $r\mathbf{B}(I,<) \subset I$ ($l\mathbf{B}(I,<) \subset I$) defined by*:

$$r\mathbf{B}(I,<) := \{s - \mathrm{Can}(s, I, <) \mid s \in rB_<(I)\} \quad (rB\text{-basis of } I),$$
$$l\mathbf{B}(I,<) := \{s - \mathrm{Can}(s, I, <) \mid s \in lB_<(I)\} \quad (lB\text{-basis of } I).$$

The following results, Theorem 7 to Remark 9, are equally valid (of course) if one replaces $r\mathbf{B}$ by $l\mathbf{B}$.

Theorem 7. (Border Bases Properties) (i) *$r\mathbf{B}(I,<)$ is a Gröbner basis of I.*
(ii) *$rGb(I,<) = r\mathbf{B}(I,<) \cap l\mathbf{B}(I,<)$.*
(iii) *If I is a zero dimensional ideal, then $r\mathbf{B}(I,<)$ is finite and $\mathrm{Card}(r\mathbf{B}(I,<)) \preceq nd$.*
(iv) *If I is a zero dimensional ideal, then $rGb(I,<)$ is finite and its cardinal is bounded by nd.*

Proof. (i) Let us see that $r\mathbf{B}(I,<)$ satisfies the Gröbner basis characterization Theorem 3(i): On one side, by Theorem 2(ii), $r\mathbf{B}(I,<) \subset I \setminus \{0\}$. On the other side, by Proposition 1(i) $\langle X\rangle T_<\{r\mathbf{B}(I,<)\} = T_<(I)$.
(ii) See Proposition 1(iv) and the structure of the polynomials in $rGb(I,<)$ (Definition 5).
(iii) As $N_<(I)$ is finite and $d = \mathrm{Card}(N_<(I))$ (see in Proposition 4 the characterization of zero-dimensional ideals), one gets the result from the structure of the words in $T_<\{r\mathbf{B}(I,<)\}$.
(iv) It is a consequence of (ii) and (iii). □

Proposition 8. (Complexity Analysis of Computing Normal Forms by Means of Border Bases) *Let I be a zero dimensional ideal, then $r\mathbf{B}(I,<)$ can be used to compute canonical forms with the following complexity (where $d = \mathrm{Card}(N_<(I))$):*

(i) *If $u \in \langle X\rangle$ and $s \in N_<(I)$, then $\mathrm{Can}(us, I, <)$ is computed in $\mathrm{O}(L(u)d^2)$ arithmetical operations.*
(ii) *If $u \in \langle X\rangle$, then $\mathrm{Can}(u, I, <)$ is computed in $\mathrm{O}(L(u)d^2)$ arithmetical operations.*
(iii) *If $f := \sum_{i=1}^{k} c_i u_i$, where, for $i \in [1,k]$, $c_i \in K \setminus \{0\}$ and $u_i \in \langle X\rangle$, then $\mathrm{Can}(f, I, <)$ is computed in $\mathrm{O}(kmd^2)$ arithmetical operations, where $m := \max\{L(u_i) \mid i \in [1,k]\}$.*

Proof. Following Faugère *et al.* (1993), we store the information of $r\mathbf{B}(I,<)$ as a function of three arguments, denoting for each $x \in X$, and $s,t \in N_<(I)$, $\mathbf{\Phi_r}[x,s,t]$ the coefficient of t in the expression of $\mathrm{Can}(xs, I, <)$ as a linear combination of vectors in $N_<(I)$ so that $\mathrm{Can}(xs, I, <) = \sum_{t \in N_<(I)} \mathbf{\Phi_r}[x,s,t]t$.

With this representation in mind, it is easy to see that the size of $r\mathbf{B}(I,<)$ is bounded by $d\mathbf{M} + nd^2$ (denoting $n := \mathrm{Card}(X)$ and considering size 1 for elements of the field of

coefficients),[†] where $\mathbf{M}$ is the maximum of the set $\{L(u) \mid u \in N_<(I)\}$.[‡] Nevertheless, in an efficient implementation, $\mathbf{\Phi_r}$ requires to be defined only for those arguments whose images are different from zero.

(i) Let $u = u_1 x$, where $u_1 \in \langle X \rangle$ and $x \in X$; then,

$$\mathrm{Can}(us, I, <) = \mathrm{Can}(u_1 \mathrm{Can}(xs, I, <), I, <) = \mathrm{Can}\left(u_1 \sum_{i=1}^{d} \mathbf{\Phi_r}[x, s, t_i] t_i, I, <\right).$$

Now, if $u_1 \neq 1$, one can factor u_1 as $u_2 y$, where $u_2 \in \langle X \rangle$ and $y \in X$; hence,

$$\begin{aligned}\mathrm{Can}(us, I, <) &= \mathrm{Can}\left(u_2 \sum_{i=1}^{d} \mathbf{\Phi_r}[x, s, t_i] \mathrm{Can}(yt_i, I, <), I, <\right) \\ &= \sum_{j=1}^{d} \sum_{i=1}^{d} \mathbf{\Phi_r}[x, s, t_i] \mathbf{\Phi_r}[y, t_i, t_j] \mathrm{Can}(u_2 t_j, I, <).\end{aligned}$$

For each of the d summands $\sum_{i=1}^{d} \mathbf{\Phi_r}[x, s, t_i] \mathbf{\Phi_r}[y, t_i, t_j]$ requires d multiplications in K at most; consequently, expressing $\mathrm{Can}(us, I, <)$ in terms of linear combinations of $\mathrm{Can}(u_2 t_j, I, <)$ needs $\mathrm{O}(d^2)$ arithmetical operations. One can repeat this process while the remainder word (now u_2) is different from 1; and so, we are done.

(ii) In (i) above, set $s := 1$, which is a canonical form modulo I except in the trivial case $I = K\langle X \rangle$.

(iii) It follows from (ii) above and the additivity of the function Can. □

Remark 9. (i) The assumption of unit cost for the field operations has already been done by Marinari *et al.* (1993) and Faugère *et al.* (1993) and requires a not entirely realistic computational model. More realistically, Faugère *et al.* (1993) also considered the growth of the coefficients in the computation of *rGb* when the term ordering is changed and the old ordering is a degree compatible one. Briefly, they concluded in that paper that, for that case, the new basis may be computed in a time which is exponential in n, but polynomial in d; they also consider this exponential behaviour to be unavoidable and related to cases where the result is too big to be useful. One cannot hope to produce a complexity analysis like the one in Faugère *et al.* (1993) for the non-commutative case in a more realistic computational model than the one assumed in Marinari *et al.* (1993). In that model, instead, following the argument Marinari *et al.* (1993, p. 144), it is easy to deduce that the computation of canonical forms, for the non-commutative case, requires a similar number of arithmetical operations to its commutative predecessor.

(ii) On the other hand, for certain interesting classes of ideals the complexity behaviour may be lower than the one in Proposition 8; that is, for example, the case for ideals generated by binomials (binomial ideals).[§] It is not difficult to see that for a binomial ideal I and a word $u \in \langle X \rangle$, $\mathrm{Can}(u, I, <) = ct$, where $c \in K \setminus \{0\}$

[†]Remark that it is sufficient to store an indexed list $\{t_1, \ldots, t_d\}$ of the elements in $N_<(I)$ and the nd^2 coefficients $\mathbf{\Phi_r}[x, s, t]$ s.t. $\mathrm{Can}(xs, I, <) = \sum_{t \in N_<(I)} \mathbf{\Phi_r}[x, s, t] t$.

[‡]$\mathbf{M}$ may be computed directly once a Gröbner basis for I is known; a general bound is $\mathbf{M} \leq d$ (cf. Proposition 1(iii)), but often could be a too big bound.

[§]And so for term rewriting theory.

and $t \in N_<(I)$ (the reader could consult Borges and Borges (1998, 4.4(ii)) for details). Therefore, for binomial ideals, the function $\mathbf{\Phi_r}$ that is mentioned in the proof of Proposition 8(i) is no longer useful by representing $r\mathbf{B}(I,<)$; instead, it is more practical to define, for $x \in X$, $s \in N_<(I)$, $\mathbf{\Phi_r}[x,s] := (c,i)$, where $c \in K \setminus \{0\}, i \in \{1,\ldots,d\}$ and $\mathrm{Can}(xs,I,<) = ct_i$. With this representation, the size of $r\mathbf{B}(I,<)$ is bounded by $d(\mathbf{M}+n)$. Thus, in this case, the number of operations for computing $\mathrm{Can}(us,I,<) = \mathrm{Can}(u_1\mathbf{\Phi_r}[x,s],I,<)$ in Proposition 8(i) is $L(u)$ and, as a consequence, that number is $\mathrm{O}(km)$ to compute $\mathrm{Can}(f,I,<)$ in Proposition 8(iii).

As another example in the same direction as above, we have that the computation of $\mathrm{Can}(us,I,<)$ in Proposition 8(i) when the binomial ideal I is generated by binomials having the form $s-t$, in fact does not involve arithmetical operations; its complexity is rather characterized by $L(u)$ reduction steps. This kind of ideal is strongly related to monoid presentations (cf. Madlener and Reinert (1998) for a recent study regarding this relation). We also remark that in the same mood FGLM algorithm has been used in Reinert *et al.* (1998) in their interpretation of the Todd-Coxeter Algorithm in terms of Gröbner techniques.

3. FGLM Algorithm for Free Associative Algebras

In this section we present our generalization, for free associative algebras, of the FGLM algorithm. The procedure we are presenting is based on a sort of black-box pattern: in fact the description of Steps 5 and 6 is only made in terms of their input and output. More precisely, we are assuming that a term ordering $\prec$ is fixed on $\langle X\rangle$, I is a zero-dimensional ideal,[†] and that the K-vector space $\mathrm{Span}_K(N_\prec(I))$ is represented by giving

- a K-vector space E which is endowed of an *effective* function

$$\mathbf{LinearDependency}[v,\{v_1,\ldots,v_r\}]$$

which, for each finite set $\{v_1,\ldots,v_r\} \subset E$ of linearly independent vectors and for each vector $v \in E$ returns the value defined by

$$\begin{cases} \{\lambda_1,\ldots,\lambda_r\} \subset K & \text{if } v = \sum_{i=1}^r \lambda_i v_i \\ \mathbf{False} & \text{if } v \text{ is not a linear combination of } \{v_1,\ldots,v_r\} \end{cases}$$

- a linear injective morphism $\xi : \mathrm{Span}_K(N_\prec(I)) \mapsto E$.

This informal approach allows a free choice of a suitable representation of the space $\mathrm{Span}_K(N_\prec(I))$ both to us in our complexity analysis and to the user in its efficient implementation of these techniques. Moreover, as an aside effect, it enables us to present this generalization in such a way that it can be applied on several more particular patterns and helps to make key ideas behind the FGLM algorithm less obscure.

Let us start making some references to some subroutines of the algorithm.

InsertNexts$[t, List]$ inserts properly the products xt (for $x \in X$) in $List$, and sorts it by increasing ordering w.r.t. the ordering $<$ (the reader should notice that **InsertNexts**, unlike its commutative predecessor in Faugère *et al.* (1993), does not produce duplicates).

NextTerm$[List]$ removes the first element from $List$ and returns it.

[†]Without this restriction the algorithm does not terminate.

Algorithm 10. (Non-commutative FGLM Algorithm)
Input: $<$, a term ordering on $\langle X\rangle$; $\xi : \mathrm{Span}_K(N_\prec(I)) \mapsto E$, a suitable representation of $\mathrm{Span}_K(N_\prec(I))$ as specified above.
Output: $rGb(I,<)$.

1. $G := \emptyset$; $List := \{1\}$; $N := \emptyset$; $r := 0$;
2. While $List \neq \emptyset$ **do**
3. $t := \mathbf{NextTerm}[List]$;
4. **If** $t \notin T_<(G) \setminus T_<\{G\}$ **then** (*it occurs iff* $t = 1$ or $lr(t) \in N$);†
5. $v := \xi(\mathrm{Can}(t, I, \prec))$;
6. $\Lambda := \mathbf{LinearDependency}[v, \{v_1, \ldots, v_r\}]$;
7. **If False** $\neq \Lambda$ **then** $G := G \cup \{t - \sum_{i=1}^{r} \lambda_i t_i\}$ (*where* $\Lambda = (\lambda_1, \ldots, \lambda_r)$)
8. **else** $r := r + 1$;
9. $v_r := v$;
10. $t_r := t$; $N := N \cup \{t_r\}$;
11. $List := \mathbf{InsertNexts}[t_r, List]$;
12. Return$[G]$

Justification of the algorithm:

The proof of correctness follows the same idea as Marinari *et al.* (1993); however, we include it here in order to highlight its main arguments, itemize the aspects concerning the non-commutative case and clarify some obscure details in the proof given in Marinari *et al.* (1993).

LinearDependency guarantees that N is a linearly independent set modulo I; on the other hand, the words in Step 3 are taken into account in increasing order (thanks to **InsertNexts**); hence, all things considered, N is a subset of $N_<(I)$. Now, we also have to prove that $G \subset I$, but:

$$t - \sum_{i=1}^{r} \lambda_i t_i \in I \iff \mathrm{Can}(t, I, \prec) = \sum_{i=1}^{r} \lambda_i \mathrm{Can}(t_i, I, \prec) \tag{3.1}$$

and, by the construction of $v_1, \ldots, v_r$ and v, the right side of (3.1) holds iff $v = \sum_{i=1}^{r} \lambda_i v_i$.

Moreover, after termination, G is a subset of $rGb(I,<)$ (compare the property of border bases Theorem 7(ii) with Step 4 and note that, for $t \in N$, $\mathrm{Can}(t, I, <) = t$).‡ Therefore, $\langle X\rangle = N \cup T_<(G) \subset N_<(I) \cup T_<(I) = \langle X\rangle$ and, since $T_<(G) \subset T_<(I)$, one infers $T_<(G) = T_<(I)$ and $N = N_<(I)$; consequently, $G = rGb(I,<)$.

This proves the correctness of the algorithm; termination is guaranteed by the finiteness of $N_<(I)$.

Remark 11. (i) A key idea in algorithms like FGLM is to use the relationship between membership to an ideal I and linear dependency modulo I, namely $\forall c_i \in K, s_i \in K\langle X\rangle$:

$$\sum_{i=1}^{r} c_i s_i \in I \setminus \{0\} \iff \{s_1, \ldots, s_r\} \text{ is linearly dependent modulo } I.$$

†Since the function **InsertNexts** produces only elements in $N_<(I) \cup rB_<(I)$ and because of Theorem 7(ii).

‡Note that (by the way N is being built) each term t is considered after the corresponding term $lr(t)$.

This connection with linear algebra was used for the first time in Gröbner bases theory as early as Buchberger (1970).

ii. It is clear that one can compute $r\mathbf{B}(I, <)$ just by eliminating the performance of Step 4 in Algorithm 10.

iii. With the presentation of the algorithm given above, one can do only a complexity analysis of the management of *List* in **InsertNexts** and the test of Step 4. It is an easy exercise to prove, on the basis of the proof given in Faugère *et al.* (1993, Theorem 5.1), that the former complexity is $\mathrm{O}(nd^2\mathbf{M})$, where $\mathbf{M}$ was defined in the proof of Proposition 8, and the latter one is also dominated by $nd^2\mathbf{M}$ (this test is a searching in the set N, which has d elements at most, each search needs to compare with each word in N, which has a cost bounded by $\mathbf{M}$; moreover, the number of searches is nd).

4. FGLM Algorithm for Ideals that are Defined by Functionals: Non-commutative Case

An algorithm to compute Gröbner bases of ideals that are defined by linear forms, which is based on the FGLM algorithm, has proved to be successful in the commutative case and turned out to be general enough for being applicable on several interesting instances (see, Marinari *et al.*, 1993, Algorithm 1).

The main goal of this section is to generalize that algorithm for computing Gröbner bases in free associative algebras. In order to do so, we are going to state first some generalizations of concepts and results that are given in Marinari *et al.* (1993) for the specific case of commutative polynomial rings. As a matter of fact, several considerations, in Marinari *et al.* (1993, Section 1) rather depend on relations between a vector space and its dual than on properties of polynomial rings.

Thus, the structure of this section is as follows. Section 4.1 extends to any vector space the duality theory that is given in Marinari *et al.* (1993) for the specific case of commutative polynomial vector spaces. Proofs are really the same as those given in Marinari *et al.* (1993), so our contribution is to present these results in a more general context than its predecessor; besides, we summarize in Theorem 12 some results that appeared dispersed (and sometimes not connected) in Marinari *et al.* (1993). Section 4.2 characterizes when a finite set of functionals on a (twisted) semigroup ring determines a two-sided ideal of this ring and, if so, the type of ideal that is determined. Section 4.3 presents our algorithm for computing the reduced Gröbner basis of an ideal $I \subset K\langle X\rangle$, when this ideal is given by a set of functionals on $K\langle X\rangle$. We prove, in this section, that the designed algorithm is as efficient as its commutative predecessor in Marinari *et al.* (1993). Section 4.4 shows how Algorithm 15 is applied when the functionals are determined by the coefficients of $\mathrm{Can}(\cdot, I, <)$ w.r.t. a given $N_<(I)$. Section 4.5 exemplifies Algorithm 16, which computes the right border basis starting from the reduced Gröbner basis.

4.1. some relations between a vector space and its dual

Let P be a K-vector space and P^* its dual, i.e. the vector space over K of functionals (linear forms) on P.

4.1.1. BIORTHOGONAL AND TRIANGULAR SEQUENCES

We say that $\{q_1, \ldots, q_s\} \subset P$ is a biorthogonal sequence (b.o.s.) for $L_1, \ldots, L_s \in P^*$ if $L_i(q_j) = \delta_i^j$, for $i, j \in [1, s]$, where δ_i^j is the Kronecker symbol. Similarly, $\{q_1, \ldots, q_s\}$ is a triangular sequence (t.s.) for $L_1, \ldots, L_s$ if $L_i(q_j) = 0$ for $i < j$ and $L_j(q_j) = 1$. By the same argument as Marinari *et al.* (1993, Remark 1.5), it can be constructively proved that, given a t.s., it is also possible to get a b.o.s. through a sort of Gram–Schmidt orthogonalization process.

4.1.2. ORTHOGONALITY RELATIONS BETWEEN SUBSPACES OF P AND P^*

Let Q (respectively V) be a K-vector subspace of P (P^*). Let us denote, as Marinari *et al.* (1993) did, by $L(Q)$ ($Z(V)$) the subspace of P^* (P) that is defined as follows:

$$L(Q) := \{L \in P^* | \, \forall f \in Q \; L(f) = 0\} \qquad (Z(V) := \{f \in P | \, \forall L \in V \; L(f) = 0\}).$$

The reader is able to find in Marinari *et al.* (1993, Lemma 1.1, Corollary 1.7) a proof of the following facts: $Z(L(Q)) = Q$ (if V is a finite dimensional K-subspace, then $L(Z(V)) = V$). Proofs in Marinari *et al.* (1993) are given under the hypothesis of P being a vector space of commutative polynomials, but this condition is not really used in the proofs, so the results are valid for any P whatsoever.

The proof for the statement inside the last parenthesis, given in Marinari *et al.* (1993), used an equivalent form of the linear independency property for a set $\{L_1, \ldots, L_s\} \subset P^*$; this characterization is based on the existence of a b.o.s. for $L_1, \ldots, L_s$. In fact, there are interesting characterizations of the linear independency for subsets of P^*; we collect some of them in the following theorem.

THEOREM 12. (SOME CHARACTERIZATIONS OF LINEAR INDEPENDENCY IN P^*) *Let $L := \{L_1, \ldots, L_s\}$ be a subset of P^*. Then the following statements are equivalent:*

(i) *L is linear independent in P^*.*
(ii) *There exists a b.o.s. for $L_1, \ldots, L_s$.*
(iii) *There exists a t.s. for $L_1, \ldots, L_s$.*
(iv) *The linear mapping $\Psi : P \to K^s$ that transforms each $f \in P$ into $(L_1(f), \ldots, L_s(f))$ is surjective.*

PROOF. (ii) $\Rightarrow$ (iii) $\Rightarrow$ (iv) $\Rightarrow$ (i) have a straightforward verification. For (i) $\Rightarrow$ (ii), the reader is able to consult the proof in Marinari *et al.* (1993, Lemma 1.6). □

Surjectiveness of Ψ and b.o.s. are related to polynomial interpolation problems. Connection between Gröbner bases and Interpolation Theory is given in Buchberger and Möller (1982) and Marinari *et al.* (1993), and more recently in Möller (1998), Sauer (1998) and other papers quoted there. In fact, it may be possible to extend basic results of this connection to the setting of this paper, but while it would be interesting to have examples of generalized interpolation problems on non-commutative algebras, we do not have any so far.

4.2. IDEALS DEFINED BY FUNCTIONALS

Now let S be a semigroup, $K[S]$ the free K-vector space on S (i.e. the vectors of $K[S]$ are the finite formal linear combinations of elements of S); let $*$ be a binary operation on $K[S]$ that behaves properly with field multiplication[†] endows this set with a ring structure which will be denoted by $K[S,*]$ ($*$ is not necessarily the natural extension of the multiplication of S, in which particular case one would get the classical semigroup ring $K[S]$ of S with coefficients in K). When $K[S,*] \neq K[S]$, $K[S,*]$ is often called a twisted semigroup ring.

Our intention is now to compute Gröbner bases of a two-sided ideal $I \subset K\langle X\rangle$[‡] that is defined by functionals, that is to say, input is not (as usual in Gröbner bases theory) a generating set of I but a finite set V of functionals on $K\langle X\rangle$ that "characterize" I by

$$I := \{f \in K\langle X\rangle | \, \forall L \in V \; L(f) = 0\}.$$

For doing so, we begin by exhibiting some results that are valid in a more general context than $K\langle X\rangle$, i.e. $K[S,*]$.

Taking into consideration Section 4.1.2, we have that if $L_1, \ldots, L_s$ are functionals on $K[S,*]$, then $Q := Z(\mathrm{Span}_K(L_1, \ldots, L_s))$ is uniquely determined by the set $\{L_1, \ldots, L_s\}$:

$$\forall q \in K[S,*] \;\; q \in Q \Longleftrightarrow L_1(q) = \cdots = L_s(q) = 0 \Longleftrightarrow q \in \mathrm{Ker}(\Psi).^{\S}$$

Moreover, $L(Q) = \mathrm{Span}_K(L_1, \ldots, L_s)$. Thus, a natural question arises:

Given $\{L_1, \ldots, L_s\} \subset K[S,*]^*$:
Is $Q := Z(\mathrm{Span}_K(L_1, \ldots, L_s))$ a two-sided ideal of $K[S,*]$?
An answer is given in Theorem 13.[§]

THEOREM 13. (CHARACTERIZATION OF IDEALS THAT ARE DEFINED BY FUNCTIONALS) *Let $\{L_1, \ldots, L_s\} \subset K[S,*]^*$. Then $Q := Z(\mathrm{Span}_K(L_1, \ldots, L_s))$ is a two-sided ideal of $K[S,*]$ iff, for $i \in [1,s]$, t_1, $t_2 \in S$, the linear forms $L_{\{i,t_1,t_2\}}$ ($L_{\{i,t_1,t_2\}}(f) := L_i(t_1 * f * t_2)$) are linear combinations of $L_1, \ldots, L_s$.*

PROOF. $\Rightarrow$ As Q is a two-sided ideal, for each $f \in Q, t_1, t_2 \in S, t_1 * f * t_2 \in Q$ so that $L_{\{i,t_1,t_2\}} \in L(Q)$ for all $i \in [1,s]$, t_1, $t_2 \in S$. Since, by the orthogonality relations between subspaces of P and P^* (4.1.2),

$$L(Q) = L(Z(\mathrm{Span}_K(L_1, \ldots, L_s))) = \mathrm{Span}_K(L_1, \ldots, L_s),$$

every $L_{\{i,t_1,t_2\}}$ lies in $\mathrm{Span}_K(L_1, \ldots, L_s)$.

$\Leftarrow$ The point is that every $h \in K[S,*]$ is a linear combination $\sum \alpha_i t_i$ (where $\alpha_i \in K \setminus \{0\}$ and $t_i \in S$) and so, the two-sided ideal condition w.r.t. the product is equivalent to:

$$\forall \, t_1, \, t_2 \in S, \, f \in Q \;\; t_1 * f * t_2 \in Q. \square \tag{4.2}$$

REMARK 14. (i) At any rate, $\dim_K K[S,*]/Q \leq s$ (since $Q = \mathrm{Ker}(\Psi)$) and equality holds iff $\{L_1, \ldots, L_s\}$ is linearly independent (see the characterization 12(iv) of

[†] i.e. $(as) * (bt) = ab(s * t), \forall a, b \in K, s, t \in S$.
[‡] $K\langle X\rangle$ is, of course, equal to $K[S]$ for $S = \langle X\rangle$.
[§] See definition of Ψ in Theorem 12(iv).

linear dependency at P^*). In addition, by the rank theorem,

$$\dim_K K[S,*]/Q = \dim_K \mathrm{Span}_K(L_1,\ldots,L_s).$$

(ii) When $K[S,*]$ is $K\langle X\rangle$, condition (4.2) amounts to a property that can be verified in a finite number of steps, namely:

$$\text{For } x_j,\, x_k \in X, L_{\{i,x_j,x_k\}} \in \mathrm{Span}_K(L_1,\ldots,L_s).$$

It may be also possible to get, starting from (4.2), practical criteria for another specific (twisted or not) semigroup rings.

(iii) Similar conclusions to Theorem 13 and (i) and (ii) above are contained in Marinari *et al.* (1993, Proposition 1.3, Remark 1.9). We have wished here, in addition to present generalizations of these results to twisted semigroup rings, to highlight related problems, like, given a finite set of functionals

- how to verify that they determine an ideal?
- what kind of ideal they determine? and, consequently:
- when they can be taken as an input for computing Gröbner bases?

4.3. An algorithm to compute Gröbner bases of ideals that are defined by functionals

The procedure we are going to discuss in this section is the non-commutative FGLM algorithm for the case in which the input is a finite set of functionals defining an ideal.

Algorithm 15. (Non-commutative FGLM Algorithm for Ideals Defined by Functionals)

Input: $<$, a term ordering on $\langle X\rangle$; $L_1,\ldots,L_s$, functionals on $K\langle X\rangle$ such that $Q := Z(\mathrm{Span}_K(L_1,\ldots,L_s))$ is a two-sided ideal of $K\langle X\rangle$. (By Remark 14(i), Q is zero-dimensional.)

Output: $rGb(Q,<)$; $\{q_1,\ldots,q_m\}$, a t.s. for $L'_1,\ldots,L'_m$, where the L'_is are a maximal l.i. subset of $\{L_1,\ldots,L_s\}$.

1. $G := \emptyset$; $List := \{1\}$; $N := \emptyset$; $r := 0$;
2. **While** $List \neq \emptyset$ **do**
3. $t :=$ **NextTerm**$[List]$;
4. **If** $t \notin T_<(G) \setminus T_<\{G\}$ **then** (*it occurs iff* $t = 1$ or $lr(t) \in N$)
5. $v := (L_1(t),\ldots,L_s(t))$;
6. $(p,v) :=$ **Gauss-reduce**$[t,v,q_1,\ldots,q_r,v_1,\ldots,v_r]$;
7. **If** $v = 0$ **then** $G := G \cup \{p\}$
8. **else** $r := r+1$;
8.1. $j := min\{i \mid L_i(p) \neq 0\}$;
8.2. $L'_r := L_j$;
9. $v_r := L_j(p)^{-1}v$;
9.1. $q_r := L_j(p)^{-1}p$;
10. $t_r := t$; $N := N \cup \{t_r\}$;
11. $List :=$ **InsertNexts**$[t_r, List]$;
12. **Return**$[G]$

Gauss-reduce was already defined in Marinari *et al.* (1993); we rewrite it here with the

aim of self containment:

Gauss-reduce $[p, v, q_1, \ldots, q_r, v_1, \ldots, v_r]$
For $i = 1 \cdots r$ **do** $v := v - L'_i(p)v_i$; $p := p - L'_i(p)q_i$.

Justification:
The key is that the representation of $\mathrm{Span}_K(N_<(Q))$ by a linear injective morphism

$$\xi : \mathrm{Span}_K(N_<(Q)) \mapsto E$$

which we choose in Algorithm 10 is the morphism $\xi : \mathrm{Span}_K(N_<(Q)) \mapsto K^s$ defined by

$$\forall f \in \mathrm{Span}_K(N_<(Q))\ \ \xi(f) := \Psi(f) = (L_1(f), \ldots, L_s(f)).$$

This justifies Step 5; moreover, **Gauss-reduce** plays in essence the same role as **Linear Dependency** of Algorithm 10. The reader can easily see that the set of vectors v_i's that is built in Algorithm 10 is equivalent to the set of v_i's that is built in the present algorithm, but the latter is built as an echelon set (see also Step 9), in order to influence the efficiency of testing linear dependency. Moreover, **Gauss-reduce**, and Steps 8.1, 8.2, and 9.1, guarantee that the q_i's form a triangular sequence for a permutation of the L_i's, in case the functionals are l.i., otherwise, there will be functionals out of selection in Step 8.1 (see Remark 14(i)).

Finally, **Gauss-reduce** again decides, at the same time, whether the v_is are linearly dependent (in case $v = 0$) and builds, in that case, the corresponding new polynomial p of G (note that $T_<(p) = t$, and, for $i \in [1, s]$, $q_i \in \mathrm{Span}_K(N_<(Q))$, $L_i(p) = 0$).

For a better understanding of the above algorithm, it might be helpful to take into account the following, easy to verify, facts. In every step of the **Gauss-reduce** algorithm $\mathrm{Can}(p, I, <) = \xi^{-1}(v)$; accordingly, for every i, $q_i = \xi^{-1}(v_i)$ and $L_i(p)$ is the ith component of v.

4.3.1. number of arithmetical operations in algorithm 15

Step 5 of the algorithm requires us to evaluate s functionals on a set of words that has cardinality ns at most (equality is reached when the functionals are l.i.). Thus if $\mathbf{f}$ denotes the average cost of evaluating any of the functionals in any of the words that needs to be considered, the number of functional evaluations is $\mathrm{O}(\mathbf{f}ns^2)$. Moreover, it is also clear that the cost of **Gauss-reduce** is the same as the one given in Marinari *et al.* (1993), i.e. $\mathrm{O}(\frac{1}{2}s^3 + ns^3)$. Note, on the other hand, that Steps 9 and 9.1 only add $2s$ multiplications. Consequently, the number of arithmetical operations in Algorithm 15 is $\mathrm{O}(ns^3 + \mathbf{f}ns^2)$, hence, there is no difference between this algorithm and its predecessor, (Marinari *et al.*, 1993, Algorithm 1).

4.4. cost of evaluating the functionals when they are determined by Gröbner bases

There is a natural way to give a zero-dimensional ideal by a finite set of linearly independent functionals: Let $L_i(f)$ be the ith coefficient of $\mathrm{Can}(f, I, <)$ as a linear combination of elements in $N_<(I)$, i.e. $\mathrm{Can}(f, I, <) = \sum_{i=1}^{s} L_i(f)t_i$. In this section we also assume I to be zero-dimensional and we write $\mathbf{L}(t) := (L_1(t), \ldots, L_s(t)), \forall t \in \langle X \rangle$. There

are at least three cases where an ideal can be given by functionals; we are going to describe them now by setting their input and output.

- **Functionals given by border bases.**
  Input: The right border basis of I w.r.t. $<_1$.
  Output: The reduced Gröbner basis of I w.r.t. $<_2$.
  This is the starting point of Faugère *et al.* (1993), i.e. the basis conversion algorithm.
- **Functionals given by reduced Gröbner bases.**
  Input: $rGb(I, <)$.
  Output: $r\mathbf{B}(I, <)$.
  The corresponding algorithm has a similar goal as procedure Matphi of Faugère *et al.* (1993).
- **Functionals given by linear changes of coordinates.**
  Input: C, a linear change of coordinates ($C(x_j) := \sum_k c_{jk}x_k + c_j$); $r\mathbf{B}(I, <_1)$.
  Output: $rGb(C^{-1}(I), <_2)$.
  In this case, the functionals are given by $\mathrm{Can}(C(f), I) = \sum_i^s L_i(f)t_i$, where $s = \dim_K K\langle X\rangle/I$ and the t_i's are the elements of $N(I, <_1)$. This problem was tackled in Gianni and Mora (1989) and was one of the starting points of the results in Faugère *et al.* (1993).

In fact, complexity analysis are quite similar to those given in Marinari *et al.* (1993, 7.3, 7.4). However, the second case (4.4.2) is not detailed in Marinari *et al.* (1993) and has some differences with Matphi of Faugère *et al.* (1993).

4.4.1. functionals given by border bases

One can realize[†] that $\mathrm{O}(ns^3)$ is the number of arithmetical operations required for evaluations of functionals, which is the same result as Marinari *et al.* (1993, 7.3); consequently, $\mathbf{f} = \mathrm{O}(s)$.

4.4.2. Functionals given by reduced Gröbner bases. Border basis algorithm

In this case, one only needs to compute $r\mathbf{B}(I, <) \setminus rGb(I, <)$ (see the border bases property Theorem 7(ii)), so we will do some slight modifications on Algorithm 10:[‡]

Algorithm 16.
Input: $rGb(I, <)$.
Output: $r\mathbf{B}(I, <)$ and the function $\mathbf{\Phi_r}[x, s, t], x \in \langle X\rangle, s, t \in N(I)$.

1. $G := \emptyset$; $List := \{1\}$; $N := \emptyset$; $r := 0$;
1.1 $G_{\mathrm{aux}} := rGb(I, <)$;[§] $p := \mathbf{NextTerm}[G_{\mathrm{aux}}]$;
2. **While** $List \neq \emptyset$ **do**

[†]See Proposition 8 above and Marinari *et al.* (1993, 7.3).
[‡]In this setting and under these modifications Algorithm 2.1 becomes essentially a rewording of the one introduced by Labontè in Labontè (1990).
[§]We assume G_{aux} is ordered in increasing order of its maximal terms.

3. $t :=$ **NextTerm**$[List]$;
3.1 **If** $t \neq T_<(p)$
4. **then If** $t \neq 1$ **and** $lr(t) \notin N$
5. **then** $\Lambda := (L_1(t), \ldots, L_s(t))$;†
6. $G := G \cup \{t - \sum_{i=1}^{s} L_i(t) t_i\}$;
7. **For** $i = 1 \cdots r$ **do** $\mathbf{\Phi_r}[x, s, t_i] := L_i(t)$
(where $x \in X$, $s = rr(t)$, and $xs = t$);
8. **else** $r := r + 1$;
9. $t_r := t$; $N := N \cup \{t_r\}$;
10. $List :=$ **InsertNexts**$[t_r, List]$;
11 **For** $i = 1 \cdots r$ **do** $\mathbf{\Phi_r}[x, s, t_i] := \begin{cases} 1 & \text{if } i = r \\ 0 & \text{otherwise} \end{cases}$
11.1 **If** $T_<(p) \neq 0$ **then**
$\mathbf{\Phi_r}[x, s, t_r] :=$ "The coefficient of t_r in $\mathrm{Can}(T_<(p), I, <)$"
(where $x \in X$, $s = rr(T_<(p))$, and $xs = T_<(p)$);
11.2 **else** $G := G \cup \{p\}$;
11.3 $p :=$ **NextTerm**$[G_{\mathrm{aux}}]$
(If G_{aux} were empty p would be only a flag and $T_<(p)$ could be, for instance, equal to 0);
11.4 $\mathbf{\Phi_r}[x, s, t_i] :=$ "The coefficient of t_i in $\mathrm{Can}(T_<(p), I, <)$"
(where $x \in X$, $s = rr(T_<(p))$, and $xs = T_<(p)$);

12. Return$[rGb(I, <) \cup G]$

Justification:
The words of $List$ belong to the following union of disjoint sets: $T_<\{rGb(I,<)\} \cup N_<(I) \cup (T_<\{r\mathbf{B}(I,<)\} \setminus T_<\{rGb(I,<)\})$.

If the new t lies in the first set, which is verified in Step 3.1, then one can include directly the next p in G (Step 11.2) and consider the following polynomial of $rGb(I,<)$ (Step 11.3). Using **NextTerm** in Steps 1.1 and 11.3 is possible because the words of $List$ are taken in increasing order; **NextTerm** helps simplifying the algorithm.

If t is now in $N_<(I)$, which is decided (after Step 3.1) in Step 4‡ then t can be entered into the set N.§

Lastly, one can infer (after Steps 3.1 and 4) that $t \in T_<\{r\mathbf{B}(I,<)\} \setminus T_<\{rGb(I,<)\}$, in which case (and only in it) computing $\mathbf{L}(t)$ is necessary. It is clear that $t - \sum_{i=1}^{s} L_i(t) t_i \in I$; moreover, this polynomial is an element of $r\mathbf{B}(I,<)$ (see the definition of border bases in Definition 6), justifying Step 6.

On the other hand, $\mathbf{\Phi_r}$ is simultaneously built in order to use it for the computation of $\mathbf{L}(t)$ in Step 5 (see Steps 7, 11, and 11.4)¶ and so is a free bonus of the algorithm. $\mathbf{\Phi_r}$ is obviously equal to zero when it is evaluated on arguments that are not considered in those steps.□

Complexity analysis for Algorithm 16.

†Recall that $\mathrm{Can}(t, I, <) = \sum_{i=1}^{s} L_i(t) t_i$.

‡Setting $t = xlr(t)$, $lr(t) \in N_<(I)$ implies $t \in N_<(I) \cup T_<\{r\mathbf{B}(I,<)\}$; the conclusion follows since $t \in T_<(I)$ has been ruled out in Steps 3.1 or 4.

§In this case, for $t_i \in N$, $L_j(t_i) = \delta_j^i$.

¶More details can be found in the complexity analysis below.

$\mathbf{L}(1) := \underbrace{(1, 0, \ldots, 0)}_{s}$, let us then see how to compute $\mathbf{L}(t)$: to do so we start factoring t as xwy, where $x, y \in X$, and $w \in N$ ($t \in X$ in Step 5 is not possible because of Steps 3.1 and 4). This factorization can be made by the property of the semigroup ideal regions of Proposition 1(iii). Hence:

$$\mathrm{Can}(t, I, <) = \sum_{i=1}^{r} \mathbf{\Phi_r}[x, w, t_i]\mathrm{Can}(t_i y, I, <).$$

As $xw < t$ and $t_i y < t$, for every $t_i \in N$, we already have enough information, on the function $\mathbf{\Phi_r}$, for computing each factor $\mathbf{\Phi_r}[x, w, t_i]\mathrm{Can}(t_i y, I, <)$ in $\mathrm{O}(\mathbf{M}s^2)$ arithmetical operations (see Proposition 8(ii)). Consequently, the number of operations for the sum is also bounded by the same quantity as above and the total cost is $\mathrm{O}(n\mathbf{M}s^3)$; moreover, $\mathbf{f} = \mathrm{O}(\mathbf{M}s)$. As a consequence, Algorithm 16 has a complexity that is a little greater than its predecessor Matphi in Faugère *et al.* (1993), which has a complexity bounded by ns^3. Note, however, that Step 5 is only applied on words of $T_<\{r\mathbf{B}(I, <)\} \setminus T_<\{rGb(I, <)\}$ and so the total number of evaluations is really bounded by $c\mathbf{M}s^2$, where c is the quantity of polynomials in the above difference set.

4.4.3. FUNCTIONALS GIVEN BY LINEAR CHANGES OF COORDINATES

The analysis here does not differ from the one given in the commutative case (see Marinari *et al.*, 1993, 7.4), thus, in this case one has an algorithm that is $\mathrm{O}(n^2 s^3)$.

EXAMPLE 16. We are going to consider K to be a field of characteristic zero.

An application of Algorithm 16.

Let $rGb(I, <_{TD}) := \{p_1, p_2, p_3, p_4, p_5, p_6\}$, where $<_{TD}$ is the total-degree term ordering and $p_1 := x_2^2 + x_2x_1 + x_1x_2 + x_1^2 - 1$, $p_2 := x_1^3 - 1$, $p_3 := x_1^2x_2 - x_2x_1 - x_1^2 + 1$, $p_4 := x_1x_2x_1 - x_2 - x_1 + 1$, $p_5 := x_2x_1^2 - x_1x_2 - x_1^2 + 1$, $p_6 := x_2x_1x_2 + x_2x_1 + x_1x_2 + x_1^2 - 1$.

It is easy to see that $N(I, <_{TD}) = \{1, x_1, x_2, x_1^2, x_1x_2, x_2x_1\}$. The latter condition guarantees the finite dimensionality. Note that, in order to apply Algorithm 16, one does not need to know $\dim K\langle X\rangle/I$, it suffices to know in advance that this K-vector space has finite dimension.

$p := p_1$; $t_1 := 1$; $\mathbf{\Phi_r}[x_2, x_2, 1] := 1$; $List := \{x_1, x_2\}$;

$t_2 := x_1$; $\mathbf{\Phi_r}[x_1, 1, 1] := 0$; $\mathbf{\Phi_r}[x_1, 1, x_1] := 1$; $\mathbf{\Phi_r}[x_2, x_2, x_1] := 0$, $List := \{x_2, x_1^2, x_2x_1\}$;

$t_3 := x_2$; $\mathbf{\Phi_r}[x_2, 1, 1] := \mathbf{\Phi_r}[x_2, 1, x_1] := 0$; $\mathbf{\Phi_r}[x_2, 1, x_2] := 1$; $\mathbf{\Phi_r}[x_2, x_2, x_2] := 0$; $List := \{x_1^2, x_1x_2, x_2x_1, x_2^2\}$.

We can continue in a similar way until x_2^2, in which case: $G := \{p_1\}$; $p := p_2$; $\mathbf{\Phi_r}[x_1, x_1^2, 1] := 1$; $\mathbf{\Phi_r}[x_1, x_1^2, x_1] := \mathbf{\Phi_r}[x_1, x_1^2, x_2] := \mathbf{\Phi_r}[x_1, x_1^2, x_1^2] := \mathbf{\Phi_r}[x_1, x_1^2, x_1x_2] := \mathbf{\Phi_r}[x_1, x_1^2, x_2x_1] := 0$. The reader can verify thereby that the unique word in $T\{r\mathbf{B}(I)\} \setminus T\{rGb(I)\}$ is $x_2^2x_1$; when it is reached we have:

$$\mathrm{Can}(x_2^2x_1, I) = \mathrm{Can}\left(\left(\sum_{i=1}^{6} \mathbf{\Phi_r}[x_2, x_2, t_i]t_i\right)x_1, I\right) = \mathrm{Can}(x_1, I) - \mathrm{Can}(x_1^3, I) -$$

$$\mathrm{Can}(x_1x_2x_1, I) - \mathrm{Can}(x_2x_1^2, I) = -x_1x_2 - x_1^2 - x_2 + 1; \text{ accordingly:}$$

$$r\mathbf{B}(I, <_{TD}) = \{p_1, \ldots, p_6\} \cup \{x_2^2x_1 + x_1x_2 + x_1^2 + x_2 - 1\}.$$

5. FGLM Algorithm for Monoid and Group Rings

Let M be a finite monoid that is generated by $g_1, \ldots, g_n$; $\phi : \langle X\rangle \to M$, the canonical morphism that sends x_i to g_i; $\sigma \subset \langle X\rangle$ x $\langle X\rangle$, a presentation of M defined by ϕ. Then, it is known that the monoid ring $K[M]$ is isomorphic to $K\langle X\rangle/I(\sigma)$, where $I(\sigma)$ is the two-sided ideal generated by $P(\sigma) := \{s - t | (s,t) \in \sigma\}$; moreover, any Gröbner basis G of $I(\sigma)$ is also formed by binomials of the above form. In addition, it can be proved that $\{(s,t) | s - t \in G\}$ is another presentation of M (cf. Madlener and Reinert, 1998, Theorem 1).

We are going to show that in order to compute $rGb(I(\sigma), <)^\dagger$ one only needs to have M given by a concrete representation that allows the user to multiply words in its generators; for instance: M may be given by permutations, matrices over a finite field, or by a more abstract way (a complete or convergent presentation). Accordingly, we are going to do the necessary modifications on Algorithm 10 for this case. First of all, we represent $\mathrm{Span}_K(N_<(I(\sigma)))$ by the linear injective morphism $\xi : \mathrm{Span}_K(N_<(I(\sigma))) \mapsto K[M]$ which is the natural extension of ϕ. Hence, Step 5 will be

$$v := m_i\xi(s), \qquad \text{where } s = rr(t) \text{ and } x_is = t.$$

Moreover, **LinearDependency**$[v, \{v_1, \ldots, v_r\}]$ can be computed as

$$\begin{cases} \xi^{-1}(v) & \text{if } \ v \in \{v_1, \ldots, v_r\} \\ \textbf{False} & \text{otherwise.} \end{cases}$$

Finally, Step 7 changes into:

$$\textbf{If False} \neq \Lambda \textbf{ then } G := G \cup \{t - \xi^{-1}(v)\}.$$

Remark 17. (i) This example shows that the capability of the K-vector space E, w.r.t. **LinearDependency**, that is demanded in Algorithm 10 is required only on those sets of vectors $\{v_1, \ldots, v_r, v\}$ that are built in the algorithm.

(ii) It is clear that the cost of repeated applications of Step 5 is $\mathrm{O}(\mathbf{c}_1ns)$, where $\mathbf{c}_1$ is the cost of multiplying two elements of M and $s = \mathrm{Card}(M)$. Also the complexity of **LinearDependency** is bounded by $\mathbf{c}_2ns^2$, where $\mathbf{c}_2$ is the cost of comparing two elements of M.

†Note that M is finite iff $I(\sigma)$ is zero-dimensional.

Let us see an example where M is the alternating group A_4, which is generated by $g_1 := (1,2)(2,3),\ g_2 := (1,2)(3,4)$.

For group presentations, one needs to take into consideration the inverses, that is, $\sigma \subset \langle X \cup X^{-1}\rangle \times \langle X \cup X^{-1}\rangle$ (where X^{-1} satisfies the conditions $X \cap X^{-1} = \emptyset$ and $\mathrm{Card}(X) = \mathrm{Card}(X^{-1})$), and σ contains at least the relations $xx^{-1} = x^{-1}x = 1$, for every $x \in X$.

We will use as the term ordering the following one that is defined in Mora (1988): Let u, w be two words in $\langle X\rangle$ and let x_k, x_m be the maximal elements of X (w.r.t. $<_L$) that divide respectively u and w. If $x_k <_L x_m$ then $u <_L w$; if x_k were equal to x_m, then the comparison would be made by recursion: First, write u and w as: $u = t_1 x_k t_2 \dots t_{m_1} x_k t_{m_1+1},\ w = v_1 x_k v_2 \dots v_{m_2} x_k v_{m_2+1}$ (where the t_i's and v_j's belong to $\langle x_1, \dots, x_{k-1}\rangle$), then $u <_L w$ if

$$\begin{cases} m_1 < m_2 \ \text{ or } \ m_1 = m_2 \ \text{ and } \ t_j <_L v_j \\ (\text{where } j = \max\{i \in [1, m_1+1] \mid t_i \neq v_i\}). \end{cases}$$

This term ordering has elimination properties (see Borges and Borges, 1998). Thus, in particular, if one considers $X <_L X^{-1}$, then it suffices to work with X and, at the end of the calculus, add the corresponding relations for the inverses, i.e. $x^{-1} - \mathrm{Can}(x^{-1})$ (see details in Borges and Borges, 1998, Proposition 4.5).

Example 18. (An Application of Algorithm 10 on Group Rings) Calls to $List$ in the assignments ($List := \{\cdot\} \cup \underbrace{List} \cup \{\cdot\}$) assume that **NextTerm** has been applied previously to this set.

$t_1 := 1$; $List := \{x_1, x_2\}$; $t_2 := x_1$; $List := \{x_1^2, x_2, x_2x_1\}$; $t_3 := x_1^2$; $List := \{x_1^3\} \cup List \cup \{x_2x_1^2\}$; $G := \{x_1^3 - 1\}$; $t_4 := x_2$; $List := \{x_1x_2\} \cup List \cup \{x_2^2\}$; $t_5 := x_1x_2$; $List := \{x_1^2x_2\} \cup List \cup \{x_2x_1x_2\}$; $t_6 := x_1^2x_2$; $List := \{x_1^3x_2\} \cup List \cup \{x_2x_1^2x_2\}$; $(x_1^3x_2 \in T(G) \setminus T\{G\})$; $t_7 := x_2x_1$; $List := \{x_1x_2x_1\} \cup List \cup \{x_2^2x_1\}$; $t_8 := x_1x_2x_1$; $List := \{x_1^2x_2x_1\} \cup List \cup \{x_2x_1x_2x_1\}$; $t_9 := x_1^2x_2x_1$; $List := \{x_1^3x_2x_1\} \cup List \cup \{x_2x_1^2x_2x_1\}$; $(x_1^3x_2x_1 \in T(G) \setminus T\{G\})$; $t_{10} := x_2x_1^2$; $List := \{x_1x_2x_1^2\} \cup List \cup \{x_2^2x_1^2\}$; $t_{11} := x_1x_2x_1^2$; $List := \{x_1^2x_2x_1^2\} \cup List \cup \{x_2x_1x_2x_1^2\}$; $t_{12} := x_1^2x_2x_1^2$; $List := \{x_1^3x_2x_1^2\} \cup List \cup \{x_2x_1^2x_2x_1^2\}$.

At this point, there are no more elements for N and the set G is completed as follows:

$$G := G \cup \{x_2^2 - 1,\ x_2x_1x_2 - x_1^2x_2x_1^2,\ x_2x_1^2x_2 - x_1x_2x_1\}.$$

Now, in order to have $rGb(I(\sigma), <_L) \subset K\langle x_1, x_2, x_1^{-1}, x_2^{-1}\rangle$, one only needs to add $x_1^{-1} - x_1^2$ and $x_2^{-1} - x_2$ to the set G. On the other hand, it is not difficult to verify that

$$\begin{aligned} r\mathbf{B}(I(\sigma), <_L) = G \cup \{&x_1^3x_2 - x_2,\ x_1^3x_2x_1 - x_2x_1,\ x_2^2x_1 - x_1, \\ &x_2x_1x_2x_1 - x_1^2x_2,\ x_2x_1^2x_2x_1 - x_1x_2x_1^2,\ x_2^2x_1^2 - x_1^2, \\ &x_2x_1x_2x_1^2 - x_1^2x_2x_1,\ x_2x_1^2x_2x_1^2 - x_1x_2\}. \end{aligned}$$

Summarizing we can say that, with the procedure that is explained in this section, the user can solve the following problem: **Given** a finite monoid M by means of a generating set and an effective way to multiply words in these generators, **find** the reduced Gröbner

basis (or the right border basis) for the two-sided ideal of $K\langle X\rangle$ that determines $K[M]$; consequently, a complete presentation for M.

If the generators were, for example, permutations of degree k and $k > n$, then $\mathbf{c}_1, \mathbf{c}_2 \leq k$ and the time computing of the corresponding algorithm is $ks^2(k + \mathbf{M})$ (also take into account Remark 11(iii)).

In the example of the alternating group, see above, the corresponding border basis leads to an algorithm that allows us to multiply words in canonical form in 5 steps at most (cf. Remark 9(ii)).

6. Further Generalization

6.1. FGLM algorithm for semigroup rings

In fact, when a semigroup S is given, it can be possible to build Gröbner basis tools for $K[S]$ (ref. Mora, 1994, Section 7). Therefore, we are going to discuss here key ideas in order to design an algorithm like FGLM for two-sided ideals of $K[S]$:

There are some essential points in algorithms like FGLM, namely:

- Explore a certain finite subset T of S where all the heads of $rGb(I)$ have to be contained. Characterizations like Proposition 1 (properties of the semigroup ideal regions) are crucial in order to choose right subsets of S.
- Decide, on the basis of characterizations of $T_<\{rGb(I,<)\}$ such as Theorem 7(ii) and the linear dependency of $N_<(I)$, what elements of T give place to polynomials of $rGb(I,<)$.

There are some main problems for getting the above goal:

- How to build T.
- How to decide linear dependency (here a subproblem is: How to compute $\mathrm{Can}(t,\cdot,<)$, for every $t \in T$).

The solution for the first problem has been, up to now and from Faugère *et al.* (1993) on, to move from the smallest words, beginning from 1, and forming, for each $t \in N_<(I)$ that has been found, the new possible words of $T_<\{rGb(I,<)\}$ that are multiple of t. Nevertheless, it may not be the best way for every case.

The solution for the second problem strongly depends on the particular algebraic structure where one wanted to apply the algorithm.

6.2. FGLM algorithm for twisted semigroup rings

Gröbner bases theory has also been generalized to twisted semigroup rings $K[S,*]$ assuming the following conditions hold:

(i) $S \cup \{0\}$ is endowed with a well ordering such that:

$$\forall l, r, t_1, t_2 \in S \;\; t_1 < t_2,\; lt_1 r \neq 0, lt_2 r \neq 0 \Longrightarrow lt_1 r < lt_2 r.$$

(ii) $\forall l, r \in S,\ \forall f \in K[S]$ either $lT(f)r = 0$ or $T(l * f * r) = lT(f)r$.

On twisted semigroup rings, see Mora (1994, Section 9.1), $G \subset I \setminus \{0\}$ is a Gröbner basis of I if $T_<(I) \cup \{0\}$ is generated by $T_<\{G\} \cup \{0\}$(where the multiplication is that of the semigroup S). It has the advantage that if we had already obtained a result such as Proposition 1 for S, then that theorem would also be valid in $K[S, *]$; consequently, the corresponding FGLM algorithm does not differ so much from the one designed for $K[S]$. The essential difference is that the valuable properties that relate Can with the product continue being valid but for *, that is to say:

$$\forall f, g \in K[S] \;\; \mathrm{Can}(f * g, I, <) = \mathrm{Can}(\mathrm{Can}(f, I, <) * g, I, <) =$$

$$\mathrm{Can}(f * \mathrm{Can}(g, I, <), I, <) =$$

$$\mathrm{Can}(\mathrm{Can}(f, I, <) * \mathrm{Can}(g, I, <), I) = \mathrm{Can}(\mathrm{Can}(f * g, I, <), I, <).$$

As a consequence, if one needed to compute $\mathrm{Can}(fg, I, <)$ (as is usual in algorithms like FGLM, for example, when computing $\mathrm{Can}(xs, I, <)$ is required), then one could try to express $\mathrm{Can}(fg, I, <)$ algebraically in terms of $\mathrm{Can}(f * g, I, <)$. The corresponding expressions that relate to the canonical forms of both products will depend on the specific twisted semigroup ring and give place to specializations of the general FGLM algorithm. Studying particular cases is out of the bounds of this paper. Having Gröbner bases techniques and algorithms like FGLM for twisted semigroup rings is worth taking into consideration because this sort of rings includes important instances, e.g., the so-called, solvable polynomial rings (cf. Kandri-Rody and Weispfenning, 1987).

References

Alonso, M. E. *et al.* (1995). The big mother of all the dualities. *Draft.*

Apel, J. (1998). The computation of Gröbner bases using an alternative algorithm. *Prog. Comput. Sci. Appl. Logic*, **15**, 35–45.

Borges, M. A., Borges, M. (1998). Gröbner bases property on elimination ideal in the non-commutative case. In Buchberger, Winkler eds, pp. 323–337.

Buchberger, B. (1970). An algorithmic criterion for the solvability of algebraic systems of equations. (German). *Aequat. Math.*, **4**, 374–383.

Buchberger, B., Möller, H. M. (1982). The construction of multivariate polynomials with preassigned zeros. In *EUROCAM'82*, LNCS **144**, pp. 24–31. Berlin Heidelberg, New York, Marseille, Springer Verlag.

Buchberger, B., Winkler, F. (1998). In Buchberger, B., Winkler, F. eds, *Gröbner Bases and Applications (Proceedings of the Conference 33 Years of Gröbner Bases)*, London Mathematical Society Lecture Notes Series **251**, Cambridge University Press, ISBN 0 521 63298 6.

Faugère, J., Gianni, P., Lazard, D., Mora, T. (1993). Efficient computation of zero-dimensional Gröbner bases by change of ordering. *J. Symb. Comput.*, **16**, 329–344.

Gianni, P., Mora, T. (1989). Algebraic solution of systems of polynomial equations using Gröbner bases. In *AAECC-5,* LNCS **356**, pp. 247–257. Berlin Heidelberg, New York, Minorca, Springer Verlag.

Janet, M. (1929). *Lecons sur les systèmes d'equations aux dérivées partielles*, Paris, Gauthier-Villars.

Kandri-Rody, A., Weispfenning, V. (1987). Non-commutative Gröbner bases in algebras of solvable type. *J. Symb. Comput.*, **9**, 1–26.

Labontè, G. (1990). An algorithm for the construction of matrix representations for finite presented non-commutative algebras. *J. Symb. Comput.*, **9**, 27–38.

Madlener, K., Reinert, B. (1998). Relating rewriting techniques on monoids and rings: congruences on monoids and ideals in monoid rings. *Theor. Comput. Sci.*, **208**, 3–31.

Marinari, M. G., Möller, H. M., Mora, T. (1993). Gröbner bases of ideals defined by functionals with an application to ideals of projective points. In *AAECC 4*, pp. 103–145.

Mora, T. (1988). Gröbner bases for non-commutative algebras. In *Proceedings of the Joint Conference International Symposium on Symbolic and Algebraic Computation'88 and AAECC 6*, LNCS **358**, pp. 150–161.

Mora, T. (1994). An introduction to commutative and non-commutative Gröbner bases. *Theor. Comput. Sci.*, **134**, 131–173.

Möller, H. M. (1998). Gröbner bases and numerical analysis. In Buchberger, Winkler eds, pp. 159–178.
Reinert, B. (1995). On Gröbner bases in monoid and group rings. Ph.D. Thesis. Univ. Kaiserslautern.
Reinert, B., Madlener, K., Mora, T. (1998). A note on Nielsen reduction and coset enumeration. In *Proceeding of the International Symposium on Symbolic Algebraic Computation 98*, pp. 171–178.
Sauer, T. (1998). Polynomial interpolation of minimal degree and Gröbner bases. In Buchberger, Winkler eds, pp. 483–494.
Zharkov, A. Yu. (1996). Solving zero-dimensional involutive systems. *Prog. Math.*, **143**, 389–399.

Originally Received 27 December 1998
Accepted 28 September 1999