Computation of free non-commutative Gröbner Bases over Z with Singular:Letterplace
Viktor Levandovskyy, Tobias Metzlaff, Karim Zeid

To cite this version:
Viktor Levandovskyy, Tobias Metzlaff, Karim Zeid. Computation of free non-commutative Gröbner Bases over Z with Singular:Letterplace. 2020. hal-02496535

HAL Id: hal-02496535
https://hal.archives-ouvertes.fr/hal-02496535
Preprint submitted on 3 Mar 2020
Computation of free non-commutative Gröbner Bases over $\mathbb{Z}$ with SINGULAR:LETTERPLACE

Viktor Levandovskyy
Lehrstuhl D für Mathematik, RWTH Aachen University
Aachen, Germany
Viktor.Levandovskyy@math.rwth-aachen.de

Tobias Metzlaff
AROMATH, INRIA Méditerranée
Université Côte d’Azur
Sophia Antipolis, France
tobias.metzlaff@inria.fr

Karim Abou Zeid
Lehrstuhl D für Mathematik, RWTH Aachen University
Aachen, Germany
karim.abou.zeid@rwth-aachen.de

ABSTRACT
The extension of Gröbner bases concept from polynomial algebras over fields to polynomial rings over rings allows to tackle numerous applications, both of theoretical and of practical importance. Gröbner and Gröbner-Shirshov bases can be defined for various non-commutative and even non-associative algebraic structures. We study the case of associative rings and aim at free algebras over principal ideal rings. We concentrate ourselves on the case of commutative coefficient ring without zero divisors (i. e. a domain). Even working over $\mathbb{Z}$ allows one to do computations, which can be treated as universal for fields of arbitrary characteristic. By using the systematic approach, we revisit the theory and present the algorithms in the implementable form. We show drastic differences in the behavior of Gröbner bases between free algebras and algebras, close to commutative. Even the formation of critical pairs has to be reengineered, together with the criteria for their quick discarding. We present an implementation of algorithms in the SINGULAR subsystem called LETTERPLACE, which internally uses Letterplace techniques (and Letterplace Gröbner bases), due to La Scala and Levandovskyy. Interesting examples accompany our presentation.

CCS CONCEPTS
• Computing methodologies → Algebraic algorithms; Special-purpose algebraic systems.

KEYWORDS
Non-commutative algebra; Gröbner bases; Coefficients in rings; Algorithms

INTRODUCTION
In the recent years a somewhat strange attitude has established itself around Gröbner bases: non-commutative generalizations of various concepts, related to algorithms and, in particular, Gröbner bases, are often met with expressions like “as expected”, “straightforward”, “more or less clear” and so on. This is not true in general for generalizations to various flavours of non-commutativity require deep analysis of procedures (algorithms) based on very good knowledge of properties of rings and modules over them. Characteristically, in this paper we demonstrate in e. g. Example 2.4 and 2.5, how intrinsically different Gröbner bases over $\mathbb{Z}(X)$ are even when compared with Gröbner bases over $\mathbb{Q}(X)$, not taking the commutative case into account. An example can illustrate this better than a thousand words: the same set $\{2x, 3y\}$ delivers a finite strong Gröbner basis $\{3x, 3y, xy, xy\}$ over $\mathbb{Z}(x, y)$ and an infinite Gröbner basis over $\mathbb{Z}(x, y, z_1, \ldots, z_m)$ for any $m \geq 1$, containing e. g. $x^k y^k z_i x$ for any natural $k$.

In his recent articles and in the book [18] Teo Mora has presented “a manual for creating your own Gröbner bases theory” over effective associative rings. This development is hard to underestimate, for it presents a unifying theoretical framework for handling very general rings. On the other hand, procedures and even algorithms related to Gröbner bases in such frameworks are still very complicated. Therefore, when aiming at implementation, one faces the classical dilemma: generality versus performance. Perhaps the most general implementation which exists is the JAS system by Heinz Kredel [8]. In our attempts we balance the generality with the performance; based on SINGULAR, we utilize its’ long and successful experience with data structures and algorithms in commutative algebra. Notably, the recent years have seen the in-depth development of Gröbner bases in commutative algebras with coefficients in principal ideal rings (O. Wienand, G. Pfister, A. Frühbis-Krüger, A. Popescu, C. Eder, T. Hofmann and others), see e. g. [5–7, 16]. This required massive changes in the structure of algorithms; ideally, one has one code for several instances of Gröbner bases with specialization to particular cases. In particular, the very generation of critical pairs and the criteria for discarding them without much effort were intensively studied. These developments were additional motivation for us in the task of attacking Gröbner bases in free algebras over commutative principal ideal rings, with $\mathbb{Z}$ at the first place. There are plenty of other motivations for doing these: currently, to the best of our knowledge, no computer algebra system is able to do such computations. Also, a number of highly interesting applications wait to be solved: in studying representation theory of a finitely presented algebra (i. e. the one, given by generators and relations), computations over $\mathbb{Z}$...
remain valid after specification to any characteristic and thus encode a universal information. In the system FeLiX by Apel et al. [2, 3], such computations were experimentally available, though not documented. In his paper [1], Apel demonstrates Gröbner bases of several nontrivial examples over \( \mathbb{Z}(X) \), the correctness of which we can easily confirm now.

Our secret weapon is the Letterplace technology [9–11, 14], which allows the usage of commutative data structures at the lowest level of algorithms. We speak, however, in theory, the language of free algebras over rings, since this is mutually bijective with the language of Letterplace.

This paper is organized as follows: In the first chapter we fix the notations which are necessary when dealing with polynomial rings. Subsequently, in the second chapter we generalize the notion of Gröbner bases for our setup, present a theoretical version of Buchberger’s algorithm and give examples to visualize significant differences compared to the field case or the commutative case. Implementation of Buchberger’s algorithm depends on and benefits from the choice of pairs, which we will discuss in the third chapter. This is followed up by computational examples and implementational aspects to confirm important examples from the cited literature in the fourth and fifth chapter.

1 GRÖBNER BASICS

All rings are assumed to be associative and unital, but not necessarily commutative.

We want to discuss non-commutative Gröbner bases over the integers \( \mathbb{Z} \). Equivalently one can take any commutative Euclidean domain or principal ideal domain \( \mathbb{R} \).

We work towards an implementation and therefore we are interested in algorithms, that is in procedures, which terminate after a finite number of steps. Since \( \mathbb{Z}(X) \) is not Noetherian, exist finite generating sets whose Gröbner bases are infinite with respect to any monomial well-ordering. Therefore, our typical computation is executed subject to the length bound (where length is meant literally, applied to words from the free monoid \( \langle X \rangle \)), specified in the input, and therefore terminates per assumption. Thus, we talk about algorithms in this sense.

Our main goal is to obtain an algorithm to construct a Gröbner basis over such a ring, finding or adjusting criteria for critical pairs and giving an effective method to implement Buchberger’s algorithm in the computer algebra system SINGULAR. The problem of applying the statements of commutative Gröbner basis over Euclidean domains and principal ideal rings, such as in [6, 7, 16, 17], are divisibility conditions of type \( \text{LM}(f) \mid \text{LM}(g) \). We start with the construction of \( S \)-polynomials.

Let \( X = \{x_1, \ldots, x_n\} \) denote the finite alphabet with \( n \) letters. We set \( \mathcal{P} = \mathcal{R}(X) \), the free \( \mathcal{R} \)-algebra of \( X \), where all words on \( X \) form a basis \( \mathcal{B} \) of \( \mathcal{P} \) as of the free \( \mathcal{R} \)-module (from now on we say shortly “\( \mathcal{B} \) is an \( \mathcal{R} \)-basis”). Moreover, let \( \mathcal{P}^e = \mathcal{P} \otimes_\mathcal{R} \mathcal{P}^\mathcal{B} \) be the free enveloping \( \mathcal{R} \)-algebra with basis \( \mathcal{B}^e = \{x \otimes y \mid x, y \in \mathcal{B}\} \). The natural action \( \mathcal{P}^e \times \mathcal{P} \to \mathcal{P} \), \( (x \otimes y, r) \mapsto x \cdot r \cdot y \) makes a bimodule \( \mathcal{P} \) into a left \( \mathcal{P}^e \)-module. We call the elements of \( \mathcal{B} \) monomials.

Let \( \preceq \) be a monomial well-ordering on \( \mathcal{B} \). An element \( f \in \mathcal{P} \) is a polynomial which is either zero or has a degree \( \deg(f) \in \mathbb{N}_0 \) and, w.r.t. \( \preceq \), a leading coefficient \( \text{LC}(f) \in \mathcal{R} \setminus \{0\} \), a leading monomial \( \text{LM}(f) \in \mathcal{B} \) and a leading term \( \text{LT}(f) = \text{LC}(f) \cdot \text{LM}(f) \neq 0 \). We denote by \(|w|\) the length of the word \( w \in \mathcal{X} \). An ordering \( \preceq \) is called length-compatible, if \( u \preceq w \) implies \(|u| \leq |w|\).

Every subset \( \mathcal{G} \subseteq \mathcal{P} \) yields a two-sided ideal, the ideal of leading terms \( \mathcal{L}(\mathcal{G}) = \{\mathcal{L}(f) \mid f \in \mathcal{G} \setminus \{0\}\} \).

Naturally, the definitions of leading coefficient, monomial and term carry over to an element \( h \in \mathcal{P}^e \) by considering \( h \cdot 1 \in \mathcal{P} \).

Definition 1.1. Let \( x, y \in \mathcal{P} \). We say that \( x \) and \( y \) have an overlap, if exist monomials \( a_1, a_2 \in \mathcal{B} \), such that at least one of the following four cases

1. \( xa_1 = a_2y \)
2. \( a_1x = ya_2 \)
3. \( a_1xa_2 = y \)
4. \( x = a_1ya_2 \)

holds. Additionally we say that \( x \) and \( y \) have a non-trivial overlap, if in the first two cases \(|a_1| < |y|\) and \(|a_2| < |x|\). In the third, respectively fourth case, we say that \( x \) divides \( y \), respectively \( y \) divides \( x \). The set of all elements which are divisible by both \( x \) and \( y \) is denoted by \( \text{CM}(x, y) \) (’common multiple’). The set of all minimal, non-trivial elements which are divisible by both \( x \) and \( y \) is denoted by \( \text{LCM}(x, y) \) (’least...’), i.e., \( t \in \text{LCM}(x, y) \), if and only if there exist \( \tau_x, \tau_y \in \mathcal{B}^e \), such that \( t = \tau_x x = \tau_y y \), representing non-trivial overlaps of \( x \) and \( y \), and if \( t, \tau \in \text{LCM}(x, y) \) with \( \tau \) for \( \tau \in \mathcal{B}^e \), then \( t = \tau \) and \( \tau = 1 \cdot 1 \). If there are only trivial overlaps, then \( \text{LCM}(x, y) = \emptyset \).

If \( \text{LM}(g) \) divides \( \text{LM}(f) \) for \( f, g \in \mathcal{P} \), then \( \text{LM}(g) \leq \text{LM}(g) \), because \( \leq \) is a monomial well-ordering with 1 (representing the empty word) as the smallest element.

2 NON-COMMUTATIVE GRÖBNER BASSES

A Gröbner basis \( \mathcal{G} \subseteq \mathcal{P} \setminus \{0\} \) is a generating set for a two-sided ideal \( \mathcal{I} \subseteq \mathcal{P} \) with the property \( L(\mathcal{I}) \subseteq L(\mathcal{G}) \). In the field case, this guarantees the existence of a so-called Gröbner representation, which we will redefine subsequently, and for any \( f \in \mathcal{I} \setminus \{0\} \) the existence of an element \( g \in \mathcal{G} \), such that \( \text{LT}(g) \) divides \( \text{LT}(f) \).

Definition 2.1. Let \( f, g \in \mathcal{P} \setminus \{0\}, \mathcal{G} \subseteq \mathcal{P} \setminus \{0\} \) be a countable set and \( I \subseteq \mathcal{P} \) be an ideal. Assume, that we fix a monomial well-ordering \( \preceq \).

We say that \( g \) \( \text{LM}-\)reduces \( f \), if \( \text{LM}(g) \) divides \( \text{LM}(f) \) with \( \text{LM}(f) = \tau \cdot \text{LM}(g) \) for some \( \tau \in \mathcal{B}^e \) and there are \( a, b \in \mathcal{R}, a \neq 0 \) and \( b <_E \text{LC}(f) \) (in the Euclidean norm), such that \( \text{LC}(f) = a \cdot \text{LC}(g) + b \). Then the \( \text{LM}-\text{reduction} \) of \( f \) by \( g \) is given by \( f - \text{arg} \).

We say that \( f \) has a strong Gröbner representation w.r.t. \( \mathcal{G} \), if \( f = \sum_{i=0}^m h_i g_i \) with \( m \in \mathbb{N}, g_i \in \mathcal{G}, h_i \in \mathcal{P}^e \) and there exists a unique \( 1 \leq j \leq m \), such that \( \text{LM}(f) = \text{LM}(h_j g_j) \) and \( \text{LM}(f) > \text{LM}(h_i g_i) \) for all \( i \neq j \) where \( h_j \neq 0 \).

\( \mathcal{G} \) is called a strong Gröbner basis for \( I \), if \( \mathcal{G} \) is a Gröbner basis for \( I \) and for all \( f' \in \mathcal{I} \setminus \{0\} \) there exists \( g' \in \mathcal{G} \), such that \( \text{LT}(g') \) divides \( \text{LT}(f') \).

\( \text{LM}-\text{reductions} \) are the key to obtain a remainder after division through a set \( \mathcal{G} \) (usually a generating set) and used in Buchberger’s algorithm to construct a Gröbner basis from \( \mathcal{G} \). In this sense, the idea of a Gröbner basis is to deliver a unique remainder when dividing through it. Since we operate in a polynomial ring of multiple
variables, the expression "reduction" is more justified than "division" to describe a chain of LM-reductions. The outcome of such a reduction, i.e. the remainder of the division, is then known as a normal form.

The following strong normal form algorithm uses LM-reductions and can be compared to the normal form algorithms in algebras over fields (cf. [12]).

NormalForm

input: $f \in \mathcal{P} \setminus \{0\}$, $\mathcal{G} \subseteq \mathcal{G}$ finite and partially ordered
output: normal form of $f$ w.r.t. $\mathcal{G}$
01: $h = f$
02: while $h \neq 0$ and $\mathcal{G}_h = \{g \in \mathcal{G} \mid g \text{LM-reduces } h\} \neq \emptyset$ do
03: choose $g \in \mathcal{G}_h$
04: choose $a, b \in \mathcal{R}$ with:
05: $a \neq 0$, $\text{LC}(h) = a \text{LC}(g) + b$ and $||b|| < E(||\text{LC}(h)||)$
06: $h = h - \text{arg}, \text{the LM-reduction of } h\text{ by } g$
07: end while
08: return $h$

A normal form of the zero-polynomial is always unique and zero. Termination and correctness are analogous to the classical proofs.

The output of the algorithm is in general not unique, but depends on the choice of elements $g \in \mathcal{G}_h$ which are used for reduction.

We confirm, that the proof of the following theorem carries over verbatim from the commutative case.

**Theorem 2.2.** Let $\mathcal{G} \subseteq \mathcal{P} \setminus \{0\}$ and $\{0\} \neq I \subseteq \mathcal{P}$. Then the following statements with respect to $\mathcal{G}$ and $\leq$, are equivalent.

1. $\mathcal{G}$ is a strong Gröbner basis for $I$.
2. Every $f \in I \setminus \{0\}$ has a strong Gröbner representation.
3. Every $f \in \mathcal{P} \setminus \{0\}$ has a unique normal form after reduction.

The proof is analogous to the commutative case in [16]. An early "weak" non-commutative version was proven by Pritchard in [20].

Such a strong Gröbner basis can be computed with Buchberger's algorithm using syzygy relations between leading monomials of generating polynomials. In the field case, this is done with S-polynomials. However, it does not suffice, when leading coefficients are non-invertible.

**Definition 2.3.** Let $f, g \in \mathcal{P} \setminus \{0\}$. There exist $\tau_f, \tau_g \in \mathcal{B}_f$, such that $\tau_f \text{LM}(f) = \tau_g \text{LM}(g) \in \text{cm}(\text{LM}(f), \text{LM}(g))$. Furthermore, let $a = \text{lcm}(\text{LC}(f), \text{LC}(g))$ and $a_f, a_g \in \mathcal{R}$, such that $a = a_f \text{LC}(f) = a_g \text{LC}(g)$. In a Euclidean domain, the least common multiple is uniquely determined up to a sign and so are $a_f, a_g$. Then a $G$-polynomial of $f$ and $g$ is defined as

$$\text{spoly}(f, g) := a_f \tau_f f - a_g \tau_g g.$$ 

It is well known from the commutative case over rings that it does not suffice to take such S-polynomials to obtain a strong Gröbner basis. Let $I = \{f = 3x, g = 2y\}$. Then every S-polynomial of $f$ and $g$ is zero, but clearly $xy = f - xg \in I$ has a leading term which is neither divisible by $\text{LT}(f)$ nor $\text{LT}(g)$. Thus, $\{f, g\}$ is not a strong Gröbner basis for $I$. The problematic polynomial $xy$ could be constructed by looking at the greatest common divisor of the leading coefficients of $f$ and $g$.

Let $b = \text{gcd}(\text{LC}(f), \text{LC}(g))$ and $b_f, b_g \in \mathcal{R}$, such that $b = b_f \text{LC}(f) + b_g \text{LC}(g)$ (the Bézout identity for the leading coefficients). As above, $b$ is unique in a Euclidean domain as a greatest common divisor, although the Bézout coefficients $b_f, b_g$ may not be, but depend on the implementation of a Euclidean algorithm. A $G$-polynomial of $f$ and $g$ is defined as

$$\text{gpoly}(f, g) := b_f \tau_f f + b_g \tau_g g.$$ 

So far everything seems to work out as in the commutative case. We consider some examples to see, that this assumption is wrong.

**Example 2.4.** Let $f = 2xy + x$, $g = 3yz + z$. Usually we would compute an S-polynomial (which is zero) and a G-polynomial

$$\text{gpoly}(f, g) := (-1) \cdot 2xy \cdot z + 1 \cdot x \cdot 3yz = xyz$$

and add them to $(f, g)$ to obtain a strong Gröbner basis for $I = \langle f, g \rangle \subseteq \mathcal{P}$. But clearly

$$\text{gpoly}(f, g) := (-1) \cdot 2xy \cdot w \cdot yz + 1 \cdot x \cdot y \cdot w \cdot 3yz = xywyz$$

is also a G-polynomial of $f, g$ for every $w \in \mathcal{B}$ and must be added to the basis. In other words there is no finite Gröbner basis for $I$ and we have to be satisfied with computing up to a fixed maximal leading monomial or length. Note that in the case of gpoly we computed a G-polynomial in the canonical way by looking for a non-trivial overlap of $xy$ and $yz$. In the case of gpoly' we ignored this overlaps. In the commutative case this is irrelevant, because then gpoly$(f, g)$ is a G-polynomial. Furthermore, in the field case this is also irrelevant, because then we do not need G-polynomials.

**Example 2.5.** A similar problem occurs with S-polynomials. Let $f = 2xy + x$, $g = 3yz + z$. Then spoly$(f, g) = 3fz - 2xg = 3xz - 2xz = xz$ is an S-polynomial of $f$ and $g$.

However, so are all polynomials

$$\text{spoly}(f, g) := 3fyz - 2xywz = 3xwyz - 2xwyz$$

for any monomial $w \in \mathcal{B}$. Now we can reduce spoly$(f, g)$ with $f$ and $g$ to

$$\text{spoly}(f, g) - xwz) + fyz = -2xywz + fyz = xwz$$

which does not reduce any further. Therefore, we have to add spoly$(f, g)$ to the basis. And even this is not enough. For $f = 2xy + x$ we see that

$$\text{spoly}(f, f) := fwyx - xwyf = xwxy - xwyx \neq 0$$

is an S-polynomial of $f$ with itself which does not reduce any further, because the leading coefficient of $f$ is not a unit and we need $\text{LMI}(f)w \text{LM}(f) \in \text{cm}(\text{LM}(f), \text{LM}(f))$, although it is clearly not contained in $\text{LCM}(\text{LM}(f), \text{LM}(f))$. So even principal ideals do not have finite strong Gröbner bases in general. This case of S-polynomials does not occur over fields and is completely new for non-commutative polynomials over $\mathcal{R}$.

Also, note that we do not consider any further extensions of the leading monomials, meaning that the $S$- and $G$-polynomial corresponding to $t \in \text{LCM}(f)$, $\text{LM}(g)$ or $\text{LM}(f)w \text{LM}(g)$ make any further (trivial) overlap relations $\tau f$ or $\tau (\text{LC}(f)w \text{LM}(g))$ for $\tau \in \mathcal{B}_f$ redundant. Therefore, in the definition of $\text{LCM}(x, y)$ we attached importance to the minimality.
The previous example shows that we have to consider all possible S- and G-polynomials, but those are infinitely many. Moreover, the set \( \text{cm}(LM(f), LM(g)) \) contains too many elements that are redundant whereas the set \( \text{LCM}(LM(f), LM(g)) \) is too small. The following definition is made to classify two types of S- and G-polynomials, namely those corresponding to non-trivial overlap relations and those corresponding to trivial ones.

**Definition 2.6.** Let \( f, g \in P \setminus \{0\} \) and \( a_f, a_g, b_f, b_g \in R \) as in 2.3. We distinguish the following two cases.

If \( LM(f) \) and \( LM(g) \) have a non-trivial overlap, then there exist \( t \in \text{LCM}(LM(f), LM(g)) \) and \( \tau_f, \tau_g \in B^*, \) such that \( t = \tau_f LM(f) = \tau_g LM(g). \) Furthermore, we assume that \( \tau_f = 1 \otimes \tau_f, \tau_g = t_g \otimes t_g' \) for \( \tau_f, \tau_g, t_g' \in B \) with \( |t_f| < |LM(g)|, |a_f|, |t_g'| < |LM(f)|. \) We define a first type S-polynomial of \( f \) and \( g \) w.r.t. \( t \) as

\[
\text{spoly}_1^e(f, g) := a_f \tau_f f - a_g \tau_g g
\]

and a first type G-polynomial of \( f \) and \( g \) w.r.t. \( t \) as

\[
\text{gpoly}_1^e(f, g) := b_f \tau_f f + b_g \tau_g g.
\]

If such \( \tau_f, \tau_g \) do not exist then we set the first type S- and G-polynomials both to zero. Since two monomials may have several non-trivial overlaps, these \( \tau_f, \tau_g \) are not unique. More precisely, this results from \( P \) not being a unique (but merely a finite) factorization domain.

For any \( w \in B \) we define the second type S-polynomial of \( f \) and \( g \) w.r.t. \( w \) by

\[
\text{spoly}_2^w(f, g) := a_f f wLM(g) - a_g LM(f) w g
\]

and the second type G-polynomial of \( f \) and \( g \) w.r.t. \( w \) as

\[
\text{gpoly}_2^w(f, g) := b_f f wLM(g) + b_g LM(f) w g.
\]

**Remark 2.7.** Clearly, it only makes sense to consider first type S- and G-polynomials if there is a non-trivial overlap of the leading monomials. However, as Example 2.4 shows, we always need to consider second type S- and G-polynomials. For any \( w \in B \) we have \( LM(f) \text{w}LM(g) \in \text{cm}(LM(f), LM(g)) \) and \( LM(g) \text{w}LM(f) \in \text{cm}(LM(f), LM(g)), \) which are distinct in general. Therefore, we need to consider both \( \text{spoly}_2^w(f, g) \) and \( \text{spoly}_2^w(g, f) \) and the same holds for second type G-polynomials. Also, note that the set of first type S- and G-polynomials is finite, because our monomial ordering is a well-ordering, whereas the set of second type S- and G-polynomials is infinite. Therefore, we need to fix an upper bound for the length of monomials which may be involved.

It is important to point out, that the elements \( \tau_f, \tau_g \) are not uniquely determined. Take for example \( f = 2xy + y, g = 3x + 1. \) Then \( t := xyz = LM(f) = xyLM(g) \in \text{LCM}(LM(f), LM(g)), \) but also \( t = LM(g) xyz \) and thus \( \text{spoly}_1^e(f, g) = -3f + 2gy = 2xy - 3y \) and \( \text{spoly}_1^e(f, g) = -3f + 2gy = 2xy - 3y \) are both first type S-polynomials with different leading monomials.

A finite set \( G \subseteq P \) is called length-bounded strong Gröbner basis for an ideal \( I, \) if there is a Gröbner basis \( G' \) for \( I, \) such that \( G \subseteq G' \) contains precisely the elements of \( G' \) of length smaller or equal to \( d \) for some \( d \in N. \)

The following algorithm uses Buchberger’s criterion 2.8 as a characterization for strong Gröbner bases, which we will prove subsequently. It computes S- and G-polynomials up to a fixed degree and reduces them with the algorithm NormalForm in order to obtain a length-bounded strong Gröbner basis for an input ideal.

**BuchbergerAlgorithm**

**Input:** \( I = \langle f_1, \ldots, f_k \rangle \subseteq R(X), d \in N, \text{NormalForm} \)

**Output:** length-bounded strong Gröbner basis \( G \) for \( I \)

1. \( G = \{ f_1, \ldots, f_k \} \)
2. \( L = \{ \text{spoly}_1^e(f_i, f_j), \text{gpoly}_1^e(f_i, f_j) \forall t^*, i, j \} \)
3. \( L = L \cup \{ \text{spoly}_1^w(f_i, f_j), \text{gpoly}_1^w(f_i, f_j) \forall w^*, i, j \} \)
4. **while** \( L \neq \emptyset \) **do**
5. **end do**
6. **if** \( h \neq 0 \) **then**
7. **end if**
8. **end while**
9. **return** \( G \)

\* \( t \in \text{LCM}, \) such that \( |t| < d \)
\*
\* \( w \in B, \) such that \( |LM(f_i)| + |w| + |LM(f_j)| < d \)
\** \( w \in B, \) such that \( |LM(h)| + |w| + |LM(g)| < d \)

For the algorithm to terminate we need the set \( L \) to eventually become empty. This happens, if and only if after finitely many steps every S- and G-polynomial based on any combination of leading terms has normal form zero w.r.t. \( G, \) i.e. there exists a chain of LM-reductions, such that the current S- or G-polynomial reduces to zero. However, LM-reductions only use polynomials of smaller or equal length and all of these are being computed. Therefore, the algorithm terminates.

For the correctness of the algorithm we still need a version of Buchberger’s criterion. More precisely, we want \( G \) to be a Gröbner basis for \( I, \) if and only if for every pair \( f, g \in G \) all their S- and G-polynomials reduce to zero. Moreover, we only want to consider first and second type S- and G-polynomials, i.e. only use \( t \in \text{cm}(LM(f), LM(g)), \) such that one of the following four cases holds for \( t, t_f', t_g', t_g'' \in B. \) This excludes all cases where \( t \) is not minimal, i.e. \( t = t \tau' \) for \( \tau \in B^* \) and \( t' \) satisfying one of the above four cases. Pritchard has proven in [20], that for a generating set of the left syzygy module (which is not finitely generated in general) we may use only minimal syzygies.

**Lemma 2.8.** Let \( G \subseteq P \setminus \{0\}, \) then \( G \) is a strong Gröbner basis for \( I := \langle G \rangle, \) if and only if for every pair \( f, g \in G \) their first and second type S- and G-polynomials reduce to zero w.r.t. \( G. \)
Theorem 2.2. Since the $S$- and the $G$-polynomials are of first or second type, they are first or second type $S$- and $G$-polynomials and \( LM(\text{spoly}_1(\cdot)) \) or \( \text{spoly}_2(\cdot) \) identity for the leading coefficients). Now, if such an $f$ induces a well-ordering, we can choose a representation where \( \{ \text{spoly}_1(\cdot), \text{spoly}_2(\cdot) \} \subseteq B \), such that \( t = \tau T \) and \( LM(f) \) and we have a strong standard representation of $f$ w.r.t. $G$. Suppose otherwise that $\text{card}(M) > 1$.

Proof. The proof is similar to the commutative case and the idea goes back to [16]. The "only if" part follows immediately from Theorem 2.2.

For "if" let $f \in \mathcal{I} \setminus \{ 0 \}$ with $f = \sum_{i \in \mathbb{P}^e} h_i g_i$ for some $h_i \in \mathbb{P}^e$. We set $t := \max(\text{LM}(h_i g_i))$ and $M := \{ i \in \mathbb{N} | \text{LM}(h_i g_i) = t \}$. Clearly \( LM(f) \leq t \) and we may assume that there is no other representation of $f$ where $t$ is smaller. Without loss of generality let $M = \{ 1, \ldots, m \}$. Moreover, since the Euclidean norm induces a well-ordering, we can choose a representation where $\sum_{i \in \mathcal{I}} |\text{LC}(h_i g_i)|$ is minimal w.r.t. $t$. If $M$ contains exactly one element, then $t = \text{LM}(f)$ and we have a strong standard representation of $f$ w.r.t. $G$.

This is similar to a statement over fields which can be found in [21]. The point is that these overlap relations or "obstructions" $t_j \text{LM}(f) h_j \prec t_j \text{LM}(g)$ correspond to $S$- and $G$-polynomials up to coefficients. But, since the coefficients are uniquely determined by $f$ and $g$ and we compute $S$- and $G$-polynomials for all pairs, we do not lose any information. Now let $t_f = t_f \otimes t_f \otimes t_f = t_f \otimes t_f' \in B^e$, $t = \tau T \text{LM}(f)$, $\text{LM}(g)$ with $t = \tau T \text{LM}(f) = \tau T \text{LM}(g)$. Then there exists a $t' \in \text{cml}(\text{LM}(f), \text{LM}(g))$ that satisfies one of the above four cases \((1) - (4)\) and $t, t_f, t_f' \in B^e$, such that $t = t_f t_f' \text{LM}(f) = t_f' \text{LM}(g)$ and $t_f \otimes t_f' = t_f' \otimes t_f$. Let

\begin{equation}
\text{spoly}(f, g) = a_f \tau t_f - a_g \tau g_f \quad \text{gpoly}(f, g) = b_f \tau t_f + b_g \tau g_f
\end{equation}

be the corresponding $S$- and $G$-polynomials. Clearly \( \text{spoly}(f, g) \) and \( \text{gpoly}(f, g) \) are first or second type $S$- and $G$-polynomials and we have \( \text{spoly}(f, g) = \tau \text{spoly}(f, g) \) and \( \text{gpoly}(f, g) = \tau \text{gpoly}(f, g) \). Therefore, if \( \text{spoly}(f, g) \) and \( \text{gpoly}(f, g) \) reduce to zero w.r.t. $G$, then do \( \text{spoly}(f, g) \) and \( \text{gpoly}(f, g) \).

It is possible to define monic or reduced Gröbner basis in our setup. For monic Gröbner bases, this was done by Li in [15]. Such a set is a Gröbner basis where every element is a monic polynomial, i.e., has leading coefficient 1. A similar notion is reduced Gröbner basis which satisfies three properties. A proposal for this definition was also made in [19]. Let $G \subseteq P \setminus \{ 0 \}$. Then $G$ is called a reduced Gröbner basis, if

\begin{enumerate}
\item every $g \in G$ has leading coefficient with signum 1,
\item $L(G \setminus \{ g \}) \subseteq L(G)$ for every $g \in G$ and
\item LT($\text{tail}(g)) \neq L(G)$ for every $g \in G$.
\end{enumerate}

The first condition states that, in the case on $R = \mathbb{Z}$, every element of a reduced Gröbner basis has leading coefficient in $\mathbb{Z}^+$. The second condition is sometimes referred to as “simplicity” and means that the leading ideal becomes strictly smaller when removing an element, thus no element is useless. The third condition, “tail-reduced”, is required in the classical field case with commutative polynomials to ensure that a reduced Gröbner basis is unique. However, this does not suffice in our setup: for instance, Pritchard gave a counterexample in [20].

Let $f = 2y^2, g = 3x^2 + y^2$ and $I = \langle f, g \rangle$. Then \( \{ f, g \} \) is a Gröbner basis for $I$ with respect to any ordering $x > y$ and satisfies the above three conditions. On the other hand, this is also true for \( \{ f, g' \} \) where \( g' = g - f = 3x^2 - y^2 \), so we have two different reduced Gröbner bases for $I$. In the field case the polynomial $g$ is not tail-reduced. This example can be used in both the commutative and non-commutative case.
When implementing a version of Buchberger’s algorithm, one should always aim to have a reduced Gröbner basis as an output. In fact this is more practical, because removing elements which are not simplified or tail reduced speeds up the computation, since we do not need to consider them in critical pairs.

**Lemma 2.9.** Suppose that \( \mathcal{G} \subset R(\mathcal{X}) \) is a result of a Gröbner basis computation up to a length bound \( d \in \mathbb{N} \), and thus finite. \( \mathcal{G} \) is a strong Gröbner basis of the ideal it generates, if and only if a Gröbner basis computation up to a length bound \( 2d - 1 \) does not change \( L(G) \).

**Proof.** It suffices to prove the “if” part. Assume that \( \mathcal{G} \) is a result of a computation up to degree \( 2d - 1 \) and \( L(G) = L(G') \). This means that all overlap relations of length \( 2d - 1 \), which are precisely the non-trivial overlap relations for polynomials of degree up to \( d \), do not enlarge the leading ideal. In other words, all first kind S- and G-polynomials reduce to zero. Because \( \mathcal{G} \) is finite and since for a Gröbner basis over fields or respectively for a “weak” (not strong) Gröbner basis over rings, we only need non-trivial overlap relations, this is the characterizing property of a Gröbner basis.

If we additionally assume that a Gröbner basis computation up to degree \( 2d \) does not change \( L(G) \), then this means that the trivial overlap relations \( LM(f) \cdot LM(g) \), which are of length \( \leq 2d \), do not add new polynomials to the basis. It remains to prove that this suffices for all trivial overlap relations \( LM(f) \cdot wLM(g) \) with \( w \in B \) to be irrelevant. Moreover, we need to take the divisibility condition \( LT(g) | LT(f) \) into account. As a consequence we could replace “Gröbner basis” with “strong Gröbner basis” in Lemma 2.9.

### 3 Critical Pairs

To improve Buchberger’s algorithm, we need criteria to determine which pairs of polynomials of the input set yield S- and G-polynomials which reduce to zero. In the following we will recall the criteria for discarding critical pairs known from the commutative case and analyze, which of them can be applied in the case \( R(\mathcal{X}) \). We also consider special situations and give counterexamples when no criterion can be derived from them.

**Remark 3.1.** First we consider the case where \( t = \text{lcm}(f) \) is divisible by (or even equals to) \( \text{lcm}(g) \). Then \( \text{lcm}(LM(f), LM(g)) \) contains exactly one element, namely \( t \), because it is the only minimal element that is divisible by both leading monomials. Therefore, \( \text{spoly}^{(1)}_{w}(f, g) \) and \( \text{gpoly}^{(1)}_{w}(f, g) \) are the only first type S- and G-polynomials. However, these are not uniquely determined, we might have more overlap relations of \( LM(f), LM(g) \), as we have seen in the previous example of Remark 2.7, and we still need second type S-polynomials.

The following Lemma has the obvious consequence that G-polynomials are redundant over fields.

**Lemma 3.2.** (cf. [7, 16]) Let \( f, g \in \mathcal{P} \setminus \{0\} \). If \( \text{lcm}(f) \mid \text{lcm}(g) \) in \( \mathcal{R} \), then every G-polynomial of \( f \) and \( g \) is redundant.

**Proof.** By the hypothesis we have \( b = \text{lcm}(\text{LC}(f), \text{LC}(g)) = \text{LC}(f) \). Let \( r \in \mathcal{R} \), such that \( r \text{LC}(f) = \text{LC}(g) \). Then \( \text{LC}(f) = (n + 1)\text{LC}(g) \) yields any possible Bézout identity for \( b \), where \( n \in \mathbb{Z} \). Thus, with \( t = \tau \text{LM}(f) = \tau g \text{LM}(g) \), every G-polynomial of \( f \) and \( g \) has shape \( \text{gpoly}(f, g) = (n + 1)\tau f - n\tau g = \text{LC}(f)t + n(\tau f - \tau g) \). Subtracting \( \tau f \), we can reduce this to \( n(\tau f - \tau g) \). Note that \( \tau f - \tau g \) is an S-polynomial of \( f \) and \( g \). Hence, every G-polynomial of \( f \) and \( g \) reduces to zero, after we compute their S-polynomials.

For \( f \in \mathcal{P} \setminus \{0\} \) we define recursively \( \text{tail}^0(f) := f \) and \( \text{tail}^i(f) := \text{tail}(\text{tail}^{i-1}(f)) \) for \( i \geq 1 \) when \( \text{tail}^{i-1}(f) \neq 0 \).

**Lemma 3.3.** (Buchberger’s product criterion, cf. [7, 16]) Let \( f, g \in \mathcal{P} \setminus \{0\} \) and \( w \in B \), such that

1. \( \text{LC}(f) \) and \( \text{LC}(g) \) are coprime over \( \mathcal{R} \),
2. \( \text{LM}(f) \) and \( \text{LM}(g) \) only have trivial overlaps and
3. for all \( i, j \geq 1 \), \( w \) does not satisfy:
   \[ \text{LM}(\text{tail}^i(f)w\text{LM}(g)) = \text{LM}(f)w\text{LM}(\text{tail}^i(g)) \].

Then \( s := \text{spoly}^{(w)}_{w}(f, g) \) reduces to zero w.r.t. \( \{f, g\} \).

**Proof.** Under the assumptions (1) and (2) we have \( s = f w\text{LT}(g) - \text{LT}(f)w g = f w - \text{tail}(g) - \text{tau}(f)w g - \text{tau}(f)w \). Note that \( \text{tail}(f)w g \) reduces to zero w.r.t. \( g \) and \( f w \text{tail}(g) \) reduces to zero w.r.t. \( f \).

By (3) we can assume without loss of generality that \( \text{LT}(s) = \text{LT}(\text{tail}(f)w)\text{LT}(g) \). Then \( s \) reduces to \( s' := s - \text{LT}(\text{tail}(f)w) \text{g} - \text{tail}(g)w ef \) and \( \text{LM}(s') \times \text{LM}(s) \). Again by (3) there is no cancellation of leading terms and, since \( < \) is a well ordering, we iteratively see that \( s \) reduces to zero.

**Remark 3.4.** The commutative version of Buchberger’s product criterion (cf. [7, 16]) criterion states, that the S-polynomial reduces to zero, if the leading terms are coprime over \( K[X] \).

Condition (3), or rather its negation, describes a very specific relation between the monomials of \( f \) and \( g \), which can occur infinitely often in theory and yield irreducible S-polynomials. The reader is reminded here that there is only a finite amount of such \( w \in B \), that satisfy this relation and are considered in Buchberger’s algorithm, because we only compute with monomials up to a certain length.

The version over fields for this criterion is much simpler, because then we only consider \( w \) to be the empty word which clearly satisfies (3). Moreover, (1) is redundant and Buchberger’s product criterion states that an S-polynomial reduces to zero when the leading monomials have only trivial overlap relations.

We consider further situation where we might find applications for criteria.

**Example 3.5.** If \( \text{LM}(f) \) and \( \text{LM}(g) \) do not overlap and the leading coefficients are not coprime, i.e., \( \text{lcm}(\text{LC}(f), \text{LC}(g)) \neq 1 \), then we can make no a priori statement about reduction. This only applies to second type S- and G-polynomials. Take for example \( f = 4xy + x, g = 6yz + z \in \mathbb{Z}(X) = \mathbb{Z}(x, y, z) \) in the degree left lexicographical ordering with \( x > y > z \). Then \( \text{spoly}^{(3)}_{w}(f, g) = 3fyz - 2xyg = 3xyz - 2xyz \) and \( \text{gpoly}^{(3)}_{w}(f, g) = -(1)fy + 1xyg = 2xyz + xy \). Since both do not reduce any further and thus must be added to the Gröbner basis just as any other second type S- and G-polynomial.

Also, for first type S- and G-polynomials no statement can be made when the leading coefficients are not coprime. For example in the case of \( f = 4xy + y, g = 6yz + y \) we have \( \text{spoly}^{(4)}_{w}(f, g) = 3fz - 2xy = 3yz - 2xy \) and \( \text{gpoly}^{(4)}_{w}(f, g) = -(1)fz + 1xyg = 2xyz - yz + xy \) which do not reduce any further.
Remark 3.6. Recall that the pair \((f, g)\) can be replaced in the commutative case (cf. [7]) by \(\{\text{spoly}(f, g), \text{gpoly}(f, g)\}\), if \(t = \text{LM}(f) = \text{LM}(g)\) (cf. [7]). Now, if \(\text{LM}(f) = \text{LM}(g)\) then in the definition of first type \(S\) and \(G\)-polynomials we have \(\tau_f = \tau_g = 1\) and therefore \(\text{spoly}_1(f, g) = af - agf\) and \(\text{gpoly}_1(f, g) = bf + bgf\). This yields a linear equation
\[
\begin{pmatrix}
\text{spoly}_1(f, g) \\
\text{gpoly}_1(f, g)
\end{pmatrix}
= 
\begin{pmatrix}
a_f & -a_g \\
b_f & b_g
\end{pmatrix}
\begin{pmatrix}
f \\
g
\end{pmatrix},
\]
where the defining matrix has determinant \(a_f b_g + a_g b_f = 1\), and thus is invertible over \(R\). Hence, we can obtain \(f\) and \(g\) from their \(S\) and \(G\)-polynomial and replace them. The importance of this statement was discussed for the commutative case in [7] and translates equivalently to the non-commutative one.

The following two lemmata are chain criteria, which are based on the idea to have two critical piles and derive a third one from them under certain conditions. The commutative versions for both criteria were proven in [7].

**Lemma 3.7. (Buchberger’s S-chain criterion, cf. [7, 16])** Let \(\mathcal{G} \subseteq \mathcal{P} \setminus \{0\}\) and \(f, g, h \in \mathcal{G}\). For all \(a, b \in \{f, g, h\}\) let \(\text{LCM}(\text{LM}(a), \text{LM}(b)) \neq \emptyset\) and fix \(T_{ab} \in \text{LCM}(\text{LM}(a), \text{LM}(b))\) and choose \(\tau_{ab} \in \mathcal{B}^e\) with \(\tau_{ab}\text{LM}(a) = T_{ab}\). There exist \(\tau_a \in \mathcal{B}^e\), such that \(\tau_a\text{LM}(a) = T_{ba}\). Assume that \(T_{ab} = T_{ba}\). Furthermore, let
\[
\begin{align*}
\text{spoly}_1^{T_{ab}}(f, g) & = \frac{c_{gh}}{c_{hf}} \text{delta}_{gf}\text{spoly}_1^{T_f}(f, g) - \frac{c_{gh}}{c_{gf}} \delta_{hf}\text{spoly}_1^{T_g}(f, g) \\
\text{gpoly}_1^{T_{ab}}(f, g) & = \frac{c_{gh}}{c_{hf}} \delta_{gf}(c_{hf}\text{tau}_h f - c_{hf}\text{tau}_g h) - \frac{c_{gh}}{c_{gf}} \delta_{hf}(c_{fg}\tau_f g - c_{fg}\tau_g f) \\
= & c_{gh}\delta_{hf}\tau_f g - c_{gh}\delta_{gf}\tau_h f + \left(\frac{c_{gh} c_{hf}}{c_{fy}} \delta_{gf}\text{tau}_f h - \frac{c_{gh} c_{hf}}{c_{fy}} \delta_{hf}\text{tau}_g f\right) f
\end{align*}
\]
and with \(\tau_{bh}\text{LM}(h) = T_{hg} = \delta_{gf}\text{tau}_h f = \delta_{hf}\text{tau}_g f\), \(\tau_{gh}\text{LM}(h) = \delta_{gf}\text{tau}_h f = \delta_{hf}\text{tau}_g f\), we have \(\delta_{gf}\text{tau}_h f = \tau_{gh}\in \mathcal{B}^e\). Analogously \(\delta_{hf}\text{tau}_g f = \tau_{gh}\) and thus the first term equals \(\text{spoly}_1^{T_{gh}}(g, h)\). Moreover, one can observe that \(c_{gh} c_{fy} = c_{fg} c_{gy} c_{fy}\). Finally, \(\text{spoly}_1^{T_{gh}} = \delta_{gf}\text{tau}_h f = T_{hg}\) and \(\text{spoly}_1^{T_{gh}} = \delta_{hf}\text{tau}_g f = T_{gh}\) implies \(\delta_{gf}\text{tau}_h f = \delta_{hf}\text{tau}_g f\) in \(\mathcal{P}^e\), hence
\[
\begin{align*}
\frac{c_{gh}}{c_{hf}} \delta_{gf}\text{spoly}_1^{T_{gh}}(f, g) & = \frac{c_{gh}}{c_{gf}} \delta_{hf}\text{spoly}_1^{T_{gh}}(f, g) = \text{spoly}_1^{T_{gh}}(g, h),
\end{align*}
\]
which shows that \(\text{spoly}_1^{T_{gh}}(g, h)\) has a strong Gröbner representation w.r.t. \(\mathcal{G}\). Clearly this also works for second type \(S\)-polynomials \(\text{spoly}_2^w(g, h)\) or \(\text{spoly}_2^w(h, g)\) if we choose \(w\) or \(w\), such that \(\text{LM}(g)\text{wLM}(h) = T_{gh}\) or \(\text{LM}(h)\text{wLM}(g) = T_{gh}\).

**Lemma 3.8. (Buchberger’s G-chain criterion, cf. [7, 16])** Let \(\mathcal{G} \subseteq \mathcal{P} \setminus \{0\}\) and \(f, g, h \in \mathcal{G}\). We use the notation \(T_{ab}\) and \(\tau_{ab}\) from Buchberger’s \(S\)-chain criterion. Let
\[
T_{gh} = T_{hg} = \text{tau}_h f, \quad T_{gf} = T_{fg} = \text{tau}_g f
\]
with \(\delta_{gf}\text{tau}_h f = \delta_{hf}\text{tau}_g f\) for some \(\delta_{gf}, \delta_{hf} \in \mathcal{B}^e\) and
\[
(2) \quad \text{LC}(f) | \gcd(\text{LC}(g), \text{LC}(g)) \quad \text{with} \quad d = \frac{\gcd(\text{LC}(g), \text{LC}(g))}{\text{LC}(f)}.
\]
Then \(\text{gpoly}_1^{T_{gh}}(g, h)\) has a strong Gröbner representation w.r.t. \(\mathcal{G}\).

**Proof.** First of all note that
\[
\text{gpoly}_1^{T_{gh}} = \gcd(\text{LC}(g), \text{LC}(h))\text{tau}_g h + b_h\text{tau}_g h\text{tail}(g) + b_h\text{tau}_g h\text{tail}(h)
\]
and
\[
\text{spoly}_1^{T_{gh}} = \frac{\text{LC}(g)}{\text{LC}(f)}\text{tau}_g f - \tau_{gf} g = \frac{\text{LC}(g)}{\text{LC}(f)}\text{tau}_g f - \tau_{gf} g
\]
(essentially a G-polynomial type S- and G-polynomials. Computations can show how hard these
computations can also be applied in the non-commutative case can also be applied in the non-commutative
then we obtain
\[
\begin{align*}
\text{spoly}_1^{T_{gh}} & = \text{dwtail}(f) = \text{dwtail}(f) + b_h\text{tau}_g h\text{tail}(g) + b_h\text{tau}_g h\text{tail}(f) \\
& = \frac{\text{LC}(h)}{\text{LC}(f)}\text{tau}_h f - \tau_{hf} h = \frac{\text{LC}(h)}{\text{LC}(f)}\text{tau}_h f - \tau_{hf} h
\end{align*}
\]
(leading out arguments of \(S\)- and \(G\)-polynomials. Since \(T_{gh}\) divides \(T_{gh}\), there exists \(w\in \mathcal{B}^e\) with \(w\text{LC}(f) = T_{gh}\). Then
\[
\text{wLM}(f) = T_{gh} = \delta_{gf}\text{tau}_f h = \delta_{hf}\text{tau}_f h\text{LM}(f).
\]
Hence, \(w = \delta_{gf}\text{tau}_f h\) and analogously \(w = \delta_{hf}\text{tau}_h f\). Moreover, \(d\text{wLC}(f)\text{LM}(f) = \gcd(\text{LC}(g), \text{LC}(h))\text{tau}_g h\) and finally we obtain
\[
\begin{align*}
\text{gpoly}_1^{T_{gh}} & = \text{dwtail}(f) + b_g\delta_{hf}\text{spoly}_1^{T_{gh}} + b_h\delta_{gf}\text{spoly}_1^{T_{gh}} \\
& = \frac{\text{LC}(g)}{\text{LC}(f)}\text{tau}_g f - \tau_{gf} g = \frac{\text{LC}(g)}{\text{LC}(f)}\text{tau}_g f - \tau_{gf} g
\end{align*}
\]
with \(\delta_{gf}\text{tau}_h f = \delta_{hf}\text{tau}_g f\) and \(\text{spoly}_1^{T_{gh}}(g, h)\) has a strong Gröbner representation w.r.t. \(\mathcal{G}\).

Thus, \(\text{gpoly}_1^{T_{gh}} = \text{dwtail}(f) + b_g\delta_{hf}\text{spoly}_1^{T_{gh}} - b_h\delta_{gf}\text{spoly}_1^{T_{gh}}\) is a strong Gröbner representation of \(\text{gpoly}_1^{T_{gh}}(g, h)\). \(\square\)

We conclude that the well-known criteria for \(S\)- and \(G\)-polynomials from the commutative case can also be applied in the non-commutative case with modifications, if we distinguish between first and second type \(S\)- and \(G\)-polynomials. Computations can show how hard these requirements are to be fulfilled compared to the commutative case by specifically counting the number of applications of product and chain criteria.

**4 Examples**

We give examples for Gröbner bases that have been computed up to a certain length bound over the integers. These examples also show that although computing over \(\mathbb{Z}\) delivers infinite results much
more often than when computing over fields, non-commutative Gröbner bases over \( \mathbb{Z} \) can be finite as well.

For the examples which will follow, let \( \mathcal{P} = \mathbb{Z}(x, y, z) \) with the degree left lexicographical ordering and \( x > y > z \).

**Example 4.1.** (cf. [1]) We consider the ideal \( I = \langle f_1 = yx - 3xy - 3z, f_2 = 2z - 2x + y, f_3 = yz - yz - x \rangle \subset \mathcal{P} \).

At first, we analyze this ideal over the field \( \mathbb{Q} \):

```plaintext
LIB "freegb.lib"; // initialization of free algebras
ring r = 0,(z,y,x),Dp; // degree left lexicographical ordering and
ideal I = twostd(I); // compute a two-sided GB of I
option(intStrategy); // avoid divisions by coefficients
ideal J = twostd(I); // compute a two-sided GB of I
// prints generators of J
```

As we see, original generators have decomposed. In order to compute the output has plenty of elements in each degree (which is the same
infinite Gröbner basis (what we confirm below) and the elements,
length because of the degree ordering), what hints at potentially
they were decomposed.

**Example 4.2.** Let \( I = \langle f_1 \rangle \) be the ideal of the free bimodule, which commutes only with constants.

```plaintext
\begin{align*}
\langle & 2z - 2x + y, 2x - 2y, 4y^2 - 2x^2, 2z^2 - 2xy \rangle.
\end{align*}
```

As we can see, the leading coefficients of the Gröbner basis above might vanish, if we pass to the field of characteristic 2. Therefore
the bimodule \( M = \mathbb{Z}(x, y, z)/I \) might have nontrivial 2-torsion, i. e.
there is a nonzero submodule \( \mathbb{T}_2(M) := \{ p \in M : 2n \in \mathbb{N}_0 \ 2^n \cdot p \in I \} \). By adopting the classical method of Caboara and Traverso for computing colon (or quotient) ideals to our situation, where we use the fact that the ground ring is central (i.e. commutes with all variables), we do the following:

```plaintext
LIB "freegb.lib";
ring r = integer,(x,y,z),dp; // degree over-term order
ring R = freeAlgebra(r,7,2); // 2==number of components
ideal I = y*x - 3*x*y - z, z*x - x*z + y, z*y+y*z-x;
ring S = option(redSB); option(redTail); // for minimal reduced GB
ideal J = twostd(I); // compute a two-sided GB of I
// prints generators of J
```

Above, \( \text{gen}(1) \) stands for the \( i \)-th canonical basis vector (commuting with everything) and \( \text{ncgen}(1) \) - for the \( i \)-th canonical generator of the free bimodule, which commutes only with constants.

The output, which is a list of vectors, looks as follows:

```plaintext
\begin{align*}
\{ & yz - yz - x, 2xz - x^2, 4yx + x, 2xz - y^2, 2x^2 - y^2, 2z^2 - xy - yz \}.
\end{align*}
```

Another step of the colon computation terminates, therefore we have computed the saturation ideal of \( I, I \subset \mathbb{Z}(x, y, z) \). It is the presentation for the 2-torsion
submodule \( \mathbb{T}_2(M) = \mathbb{Z}(x, y, z)/I \) and, moreover, \( 2 \cdot L \subset I \subset L \) holds.

**Example 4.3.** Another ideal that has a finite Gröbner basis is
\( I = \langle f_1 = xy - 3xy - 3z, f_2 = 2x - xz + y, f_3 = yz - yz - x \rangle \subset \mathcal{P} \). Then \( I \) has a finite strong Gröbner basis, namely
\( \langle f_1, f_2, f_3, 8xy + 2z, 4xz - 2y, 4yz + 2x, 2x^2 - 2y^2, 4y^2 - 2x^2, 2z^2 - 2xy \rangle. \)

We can show that for every \( 2 \leq i \leq 10 \) there are the only polynomials which have to be added to
the above set in order to obtain a Gröbner basis for \( I \). Therefore
this Gröbner basis is infinite, but can be presented in finite terms.

Also, we note that the original generators have been preserved in
any Gröbner basis, while over \( \mathbb{Q} \) they were decomposed.

**Example 4.4.** In this example we have to run a Gröbner basis of
\( \langle f_1 = yz - yz - 3x, f_2 = xz + y^2, f_3 = yx - 3xy \rangle \) up to length bound
11, in order to prove with the Lemma 2.9 that we have computed a
finite Gröbner basis. We use degree right lexicographical ordering,
while its version and elimination orderings do not result in
finite sets.
We have created a powerful implementation called Letterplace. All these can be executed with respect to the orderings like degree cations of our algorithms and their implementation, in particular speed. Following Mora’s “manual for creating own Gröbner basis theory” 6 CONCLUSION AND FUTURE WORK

As we can see from the leading terms, the corresponding module might have 2- and 7-torsion submodules.

There have been 17068 critical pairs created, and internal total degree of intermediate elements was 11. The product criterion has been used 196 times, while the chain criterion was invoked 36711 times. Totally, up to 2.9 GB of memory was allocated.

In the contrast, the Gröbner basis computation of the same input over Q over Z has been used 36711−196=36515. The product criterion has considered only 14 critical pairs, went up to total degree 6, used no product criterion and 9 times the chain criterion with less than 1 MB of memory. The result is

\[
\{f_1, f_2, f_3, 2y^3 + y^2z - 2yz^2 + 2z^3, \\
y^2z^2 - 4yz + 6z^4, y^4 + 27xy^2z - 54xy^2 + 54xz^3, \\
54xy^2z - y^3z - 108xyz^2 + 108x^3z + 62yz^3 - 124z^4, 14z^5, \\
14yz^3 - 28z^4, 2y^4 - 6z^5, 2xy^3 - 4xz^4, xy^2z, 2z^6, 2xz^5\}.
\]

As we can see from the leading terms, the corresponding module might have 2- and 7-torsion submodules.

7 ACKNOWLEDGEMENTS

The authors are grateful to Gerhard Pfister (Kaiserslautern), Anne Frühbis-Krüger (Oldenburg), Leonard Schmitz, Eva Zerz (RWTH Aachen) and Evelyn Hubert (INRIA) for fruitful discussions.

The first and third authors (V. Levandovskyy and K. Abou Zeid) have been supported by Project II.6 of SFB-TRR 195 “Symbolic Tools in Mathematics and their Applications” of the German Research Foundation (DFG).

The work of the second author (T. Metzlaff) has been supported by European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Actions, grant agreement 813211 (POEMA).

REFERENCES


