Computation of free non-commutative Gröbner Bases over Z with Singular:Letterplace

Viktor Levandovskyy, Tobias Metzlaff, Karim Abou Zeid

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

ACM Reference Format:
Viktor Levandovskyy, Tobias Metzlaff, and Karim Abou Zeid. 2020. Computation of free non-commutative Gröbner Bases over $\mathbb{Z}$ with Singular:Letterplace. In *Proceedings of 2020 ACM International Symposium on Symbolic and Algebraic Computation (ISSAC '20).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

ISSAC '20, July 20–23, 2020, Kalamata, Messinia, Greece

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

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 $\mathbb{Z}\langle X\rangle$ are even when compared with Gröbner bases over $\mathbb{Q}\langle X\rangle$, 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 $\{3x, 3y, yx, xy\}$ over $\mathbb{Z}\langle x, y\rangle$ and an infinite Gröbner basis over $\mathbb{Z}\langle x, y, z_1, \ldots, z_m\rangle$ for any $m \geq 1$, containing e. g. $xz_i^k y, yz_i^k x$ 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 $\mathbb{Z}\langle X\rangle$, 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[1] $\mathcal{R}$.

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 $\mathbb{Z}\langle X\rangle$ is not Noetherian, there 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 $\mathrm{LM}(f) \mid \mathrm{LM}(g)$. We start with the construction of S-polynomials.

Let $X = \{x_1, \ldots, x_n\}$ denote the finite alphabet with n letters. We set $\mathcal{P} = \mathcal{R}\langle X\rangle$, 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 $\mathcal{P}^e = \mathcal{P} \otimes_{\mathcal{R}} \mathcal{P}^{\mathrm{opp}}$ 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*.

[1] This concept can be extended to principal ideal rings. It was done in the commutative case by [5] with so-called annihilator polynomials.

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* $\mathrm{LC}(f) \in \mathcal{R} \setminus \{0\}$, a *leading monomial* $\mathrm{LM}(f) \in \mathcal{B}$ and a *leading term* $\mathrm{LT}(f) = \mathrm{LC}(f)\mathrm{LM}(f) \neq 0$. We denote by $|w|$ the length of the word $w \in \langle X\rangle$. An ordering $\succ$ is called *length-compatible*, if $u \prec w$ implies $|u| \leq |w|$. Every subset $\mathcal{G} \subseteq \mathcal{P}$ yields a two-sided ideal, the *ideal of leading terms* $\mathrm{L}(\mathcal{G}) = \langle \mathrm{LT}(f) \mid f \in \mathcal{G} \setminus \{0\}\rangle$.

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 $x, y \in \mathcal{B}$. We say that x and y have an *overlap*, if there 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 $\mathrm{cm}(x, y)$ ("common multiple"). The set of all minimal, non-trivial elements which are divisible by both x and y is denoted by $\mathrm{lcm}(x, y)$ ("least..."), i. e. $t \in \mathrm{lcm}(x, y)$, 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 $t, \tilde{t} \in \mathrm{lcm}(x, y)$ with $\tilde{t} = \tau t$ for some $\tau \in \mathcal{B}^e$, then $t = \tilde{t}$ and $\tau = 1 \otimes 1$. If there are only trivial overlaps, then $\mathrm{lcm}(x, y) = \emptyset$.

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

2 NON-COMMUTATIVE GRÖBNER BASES

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 $\mathrm{L}(\mathcal{I}) \subseteq \mathrm{L}(\mathcal{G})$. 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 $\mathrm{LT}(g)$ divides $\mathrm{LT}(f)$.

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

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

We say that f has a *strong Gröbner representation* w.r.t. $\mathcal{G}$, if $f = \sum_{i=1}^{m} h_i g_i$ 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 $\mathrm{LM}(f) = \mathrm{LM}(h_j g_j)$ and $\mathrm{LM}(f) \succ \mathrm{LM}(h_i g_i)$ for all $i \neq j$ where $h_i \neq 0$.

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

LM-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:
$a \neq 0$, $\mathrm{LC}(h) = a\mathrm{LC}(g) + b$ and $||b|| <_{\mathrm{E}} ||\mathrm{LC}(h)||$
05: choose $\tau \in \mathcal{B}^e$ with $\mathrm{LM}(h) = \tau\mathrm{LM}(g)$
06: $h = h - a\tau g$, the LM-reduction of h 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 $\{0\} \neq \mathcal{I} \subseteq \mathcal{P}$. Then the following statements with respect to $\mathcal{G}$ and $\preceq$, are equivalent.*

(1) $\mathcal{G}$ is a strong Gröbner basis for $\mathcal{I}$.
(2) Every $f \in \mathcal{I} \setminus \{0\}$ 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 $\tau_f, \tau_g \in \mathcal{B}^e$, such that $\tau_f\mathrm{LM}(f) = \tau_g\mathrm{LM}(g) \in \mathrm{cm}(\mathrm{LM}(f), \mathrm{LM}(g))$. Furthermore, let $a = \mathrm{lcm}(\mathrm{LC}(f), \mathrm{LC}(f))$ and $a_f, a_g \in \mathcal{R}$, such that $a = a_f\mathrm{LC}(f) = a_g\mathrm{LC}(g)$. In a Euclidean domain, the least common multiple is uniquely determined up to a sign and so are a_f, a_g. Then a *S-polynomial* of f and g is defined as

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

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 $\mathcal{I} = \langle f = 3x, g = 2y\rangle$. Then every S-polynomial of f and g is zero, but clearly $xy = fy - xg \in \mathcal{I}$ has a leading term which is neither divisible by $\mathrm{LT}(f)$ nor $\mathrm{LT}(g)$. Thus, $\{f, g\}$ is not a strong Gröbner basis for $\mathcal{I}$. The problematic polynomial xy could be constructed by looking at the greatest common divisor of the leading coefficients of f and g.

Let $b = \gcd(\mathrm{LC}(f), \mathrm{LC}(f))$ and $b_f, b_f \in \mathcal{R}$, such that $b = b_f\mathrm{LC}(f) + b_g\mathrm{LC}(g)$ (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

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

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 $f = 2xy$, $g = 3yz \in \mathbb{Z}\langle x, y, z\rangle$. Usually we would compute an S-polynomial (which is zero) and a G-polynomial

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

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

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

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 $\mathcal{I}$ 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 overlap. In the commutative case this is irrelevant, because then $\mathrm{gpoly}(f, g) \mid \mathrm{gpoly}'(f, g)$. 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 $\mathrm{spoly}(f, g) = 3fz - 2xg = 3xz - 2xz = xz$ is an S-polynomial of f and g. However, so are all polynomials

$$\mathrm{spoly}'(f, g) := 3fwyz - 2xywg = 3xwyz - 2xywz$$

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

$$(\mathrm{spoly}'(f, g) - xwg) + fwz = -2xywz + fwz = xwz$$

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

$$\mathrm{spoly}'(f, f) := fwxy - xywf = xwxy - xywx \neq 0$$

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 $\mathrm{LM}(f)w\mathrm{LM}(f) \in \mathrm{cm}(\mathrm{LM}(f), \mathrm{LM}(f))$, although it is clearly not contained in $\mathrm{lcm}(\mathrm{LM}(f), \mathrm{LM}(f))$. 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 $t \in \mathrm{lcm}(\mathrm{LM}(f), \mathrm{LM}(g)$ or $\mathrm{LM}(f)w\mathrm{LM}(g)$ make any further (trivial) overlap relations τt or $\tau(\mathrm{LM}(f)w\mathrm{LM}(g))$ for $\tau \in \mathcal{B}^e$ redundant. Therefore, in the definition of $\mathrm{lcm}(x, y)$ 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 $\text{CM}(\text{LM}(f), \text{LM}(g))$ contains too many elements that are redundant whereas the set $\text{LCM}(\text{LM}(f), \text{LM}(g))$ 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 $f, g \in \mathcal{P} \setminus \{0\}$ and $a_f, a_g, b_f, b_g \in \mathcal{R}$ as in 2.3. We distinguish the following two cases.

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

$$\text{spoly}_1^t(f, g) := a_f \tau_f f - a_g \tau_g g$$

and a *first type G-polynomial* of f and g w.r.t. t as

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

If such τ_f, τ_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 τ_f, τ_g are not unique. More precisely, this results from $\mathcal{P}$ not being a unique (but merely a finite) factorization domain.

For any $w \in \mathcal{B}$ we define the *second type S-polynomial* of f and g w.r.t. w by

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

and the *second type G-polynomial* of f and g w.r.t. w as

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

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 $w \in \mathcal{B}$ we have $\text{LM}(f) w \text{LM}(g) \in \text{CM}(\text{LM}(f), \text{LM}(g))$ and $\text{LM}(g) w \text{LM}(f) \in \text{CM}(\text{LM}(f), \text{LM}(g))$, 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 τ_f, τ_g are not uniquely determined. Take for example $f = 2xyx + y$, $g = 3x + 1$. Then $t := xyx = \text{LM}(f) = xy\text{LM}(g) \in \text{LCM}(\text{LM}(f), \text{LM}(f))$, but also $t = \text{LM}(g)yx$ and thus $\text{spoly}_1^t(f, g) = -3f + 2gyx = 2yx - 3y$ and $(\text{spoly}_1^t)'(f, g) = -3f + 2xyg = 2xy - 3y$ are both first type S-polynomials with different leading monomials.

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

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: $\mathcal{I} = \langle f_1, \ldots, f_k \rangle \subseteq \mathcal{R}\langle X \rangle$, $d \in \mathbb{N}$, NormalForm
output: length-bounded strong Gröbner basis $\mathcal{G}$ for $\mathcal{I}$

$\mathcal{G} = \{f_1, \ldots, f_k\}$
$\mathcal{L} = \{\text{spoly}_1^t(f_i, f_j), \text{gpoly}_1^t(f_i, f_j) \mid \forall\, t\ ^*,\ i, j\}$
$\mathcal{L} = \mathcal{L} \cup \{\text{spoly}_2^w(f_i, f_j), \text{gpoly}_2^w(f_i, f_j) \mid \forall\, w\ ^{**},\ i, j\}$
while $\mathcal{L} \neq \emptyset$ **do**
  choose $h \in \mathcal{L}$
  $\mathcal{L} = \mathcal{L} \setminus \{h\}$
  $h = \text{NormalForm}(h, \mathcal{G})$
  if $h \neq 0$ **then**
    $\mathcal{G} = \mathcal{G} \cup \{h\}$
    for $g \in \mathcal{G}$ **do**
      $\mathcal{L} = \mathcal{L} \cup \{\text{spoly}_1^t(g, h), \text{gpoly}_1^t(g, h) \mid \forall\, t\ ^*\}$
      $\mathcal{L} = \mathcal{L} \cup \{\text{spoly}_1^t(h, g), \text{gpoly}_1^t(h, g) \mid \forall\, t\ ^*\}$
      $\mathcal{L} = \mathcal{L} \cup \{\text{spoly}_2^w(g, h), \text{gpoly}_2^w(g, h) \mid \forall\, w\ ^{***}\}$
      $\mathcal{L} = \mathcal{L} \cup \{\text{spoly}_2^w(h, g), \text{gpoly}_2^w(h, g) \mid \forall\, w\ ^{***}\}$
    end do
  end if
end while
return $\mathcal{G}$

---

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

For the algorithm to terminate we need the set $\mathcal{L}$ 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 $\mathcal{G}$, 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 $\mathcal{G}$ to be a Gröbner basis for $\mathcal{I}$, if and only if for every pair $f, g \in \mathcal{G}$ 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 $t \in \text{CM}(\text{LM}(f), \text{LM}(g))$, such that one of the following four cases

$$(1)\ t = \text{LM}(f)t'_f = t_g \text{LM}(g) \qquad (2)\ t = \text{LM}(f) = t_g \text{LM}(g) t'_g$$
$$(3)\ t = t_f \text{LM}(f) = \text{LM}(g) t'_g \qquad (4)\ t = t_f \text{LM}(f) t'_f = \text{LM}(g)$$

holds for $t_f, t'_f, t_g, t'_g \in \mathcal{B}$. This excludes all cases where t is not minimal, i. e. $t = \tau t'$ for $\tau \in \mathcal{B}^e$ 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 $\mathcal{G} \subseteq \mathcal{P} \setminus \{0\}$. Then $\mathcal{G}$ is a strong Gröbner basis for $\mathcal{I} := \langle \mathcal{G} \rangle$, if and only if for every pair $f, g \in \mathcal{G}$ their first and second type S- and G-polynomials reduce to zero w.r.t. $\mathcal{G}$.*

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 $f = \sum_i h_i g_i$ for some $h_i \in \mathcal{P}^e$. We set $t := \max(\mathrm{LM}(h_i g_i))$ and $M := \{i \in \mathbb{N} \mid \mathrm{LM}(h_i g_i) = t\}$. Clearly $\mathrm{LM}(f) \preceq t$ and we may assume that there is no other representation of f where t is smaller. Withour loss of generality let $M = \{1, \ldots, m\}$. Moreover, since the Euclidean norm induces a well-ordering, we can choose a representation where $\sum_{i=1}^{m} |\mathrm{LC}(h_i)\mathrm{LC}(g_i)|$ is minimal w.r.t. t. If M contains exactly one element, then $t = \mathrm{LM}(f)$ and we have a strong standard representation of f w.r.t. $\mathcal{G}$. Suppose otherwise that $\mathrm{card}(M) > 1$. Then $t \succeq \mathrm{LM}(f)$. Note that $t = \mathrm{LM}(h_i g_i) = \mathrm{LM}(h_i)\mathrm{LM}(g_i)$ for $i \leq m$. Then there exist monomials $t_1, t_1', t_2, t_2' \in \mathcal{B}$, such that $t = t_1\mathrm{LM}(g_1)t_1' = t_2\mathrm{LM}(g_2)t_2'$. This induces an overlap relation of the leading monomials, because then there exist $s_1, s_1', s_2, s_2' \in \mathcal{B}$, such that one of the following four cases

(1) $\mathrm{LM}(g_1)s_1' = s_2\mathrm{LM}(g_2) =: T$ (2) $\mathrm{LM}(g_1) = s_2\mathrm{LM}(g_2)s_2' =: T$

(3) $s_1\mathrm{LM}(g_1) = \mathrm{LM}(g_2)s_2' =: T$ (4) $s_1\mathrm{LM}(g_1)s_1' = \mathrm{LM}(g_2) =: T$

holds and $t = \tau T$ for some $\tau \in \mathcal{B}^e$. Moreover, let τ_1, τ_2 result from s_1, s_1', s_2, s_2', such that $T = \tau_1\mathrm{LM}(g_1) = \tau_2\mathrm{LM}(g_2)$. Furthermore, let $a_1, a_2 \in \mathcal{R}$ with $a_1\mathrm{LC}(g_1) = a_2\mathrm{LC}(g_2) = \mathrm{lcm}(\mathrm{LC}(g_1), \mathrm{LC}(g_2))$ and $d := \gcd(\mathrm{LC}(g_1), \mathrm{LC}(g_2)) = b_1\mathrm{LC}(g_1) + b_2\mathrm{LC}(g_2) \in \mathcal{R}$ (the Bézout identity for the leading coefficients). Now, if T corresponds to a nontrivial overlap, then we can compute $\mathrm{spoly}_1^T(g_1, g_2)$, $\mathrm{gpoly}_1^T(g_1, g_2)$ or $\mathrm{spoly}_1^T(g_2, g_1)$, $\mathrm{spoly}_1^T(g_2, g_1)$, respectively. Otherwise there exists $w \in \mathcal{B}$, such that $T = \mathrm{LM}(g_1)w\mathrm{LM}(g_2)$ or $T = \mathrm{LM}(g_2)w\mathrm{LM}(g_1)$. In this case we are interested in $\mathrm{spoly}_2^w(g_1, g_2)$, $\mathrm{gpoly}_2^w(g_1, g_2)$ or $\mathrm{spoly}_2^w(g_2, g_1)$, $\mathrm{spoly}_2^w(g_2, g_1)$, respectively. This shows that $\mathrm{spoly}(g_1, g_2) := a_1\tau_1 g_1 - a_2\tau_2 g_2$ and $\mathrm{gpoly}(g_1, g_2) := b_1\tau_1 g_1 + b_2\tau_2 g_2$ are first or second type S- and G-polynomials and $\mathrm{LM}(h_1) = \tau\tau_1$, $\mathrm{LM}(h_2) = \tau\tau_2$.

As d is a greatest common divisor, there exists $a \in \mathcal{R} \setminus \{0\}$, such that $ad = \mathrm{LC}(h_1)\mathrm{LC}(g_1) + \mathrm{LC}(h_2)\mathrm{LC}(g_2)$ or equivalently $\mathrm{LC}(h_1)\mathrm{LC}(g_1) = b_1\mathrm{LC}(g_1) + b_2\mathrm{LC}(g_2) - \mathrm{LC}(h_2)\mathrm{LC}(g_2)$, i. e. there is $b \in \mathcal{R} \setminus \{0\}$ with $\mathrm{LC}(h_1) = ab_1 + ba_1$, $\mathrm{LC}(h_2) = ab_2 - ba_2$.

Then, since $|a_1\mathrm{LC}(g_1) + a_2\mathrm{LC}(g_2)| > 0$ and by the triangle inequality, we have

$$\begin{aligned}
&|\mathrm{LC}(h_1)\mathrm{LC}(g_1)| + |\mathrm{LC}(h_2)\mathrm{LC}(g_2)| \\
=&|(ab_1 + ba_1)\mathrm{LC}(g_1)| + |(ab_2 - ba_2)\mathrm{LC}(g_2)| \\
\geq&|ab_1\mathrm{LC}(g_1)| + |ba_1\mathrm{LC}(g_1)| + |ab_2\mathrm{LC}(g_2)| + |ba_2\mathrm{LC}(g_2)| \\
>&|ab_1\mathrm{LC}(g_1)| + |ab_2\mathrm{LC}(g_2)| \geq |ab_1\mathrm{LC}(g_1) + ab_2\mathrm{LC}(g_2)| = |ad|,
\end{aligned}$$

thus $|ad|$ is strictly smaller than $|\mathrm{LC}(h_1)\mathrm{LC}(g_1)| + |\mathrm{LC}(h_2)\mathrm{LC}(g_2)|$. Furthermore, we have

$$\begin{aligned}
&h_1 g_1 + h_2 g_2 \\
=&(\mathrm{LC}(h_1)\mathrm{LM}(h_1)\,\mathrm{tail}(h_1))g_1 + (\mathrm{LC}(h_2)\mathrm{LM}(h_2)\,\mathrm{tail}(h_2))g_2 \\
=&(ab_1 + ba_1)\tau\tau_1 g_1 + \mathrm{tail}(h_1)g_1 + (ab_2 - ba_2)\tau\tau_2 g_2 + \mathrm{tail}(h_2)g_1 \\
=&a\tau(b_1\tau_1 g_1 + b_2\tau_2 g_2) + b\tau(a_1\tau_1 g_1 - a_2\tau_2 g_2) + \mathrm{tail}(h_1)g_1 + \mathrm{tail}(h_2)g_1 \\
=&a\tau\,\mathrm{gpoly}(g_1, g_2) + b\tau\mathrm{spoly}(g_1, g_2) + \mathrm{tail}(h_1)g_1 + \mathrm{tail}(h_2)g_1.
\end{aligned}$$

Since the S- and the G-polynomials are of first or second type, they reduce to zero w.r.t. $\mathcal{G}$. Hence, we can write $h_1 g_1 + h_2 g_2 = \sum_j h_j' g_j$ for $h_j' \in \mathcal{P}^e$ and define $M' := \{j \in \mathbb{N} \mid \mathrm{LM}(h_j' g_j) = t\}$. Since $\mathrm{LM}(\tau\,\mathrm{spoly}(g_1, g_2)) \prec t$, $\mathrm{LM}(\mathrm{tail}(h_1)g_1) \prec t$ and $\mathrm{LM}(\mathrm{tail}(h_2)g_1) \prec t$, we have

$$\begin{aligned}
&\sum_{j \in M'} |\mathrm{LC}(h_j')\mathrm{LC}(g_j)| = \sum_{j \in M'} |\mathrm{LC}(h_j' g_j)| = \\
=&|\mathrm{LC}(d\tau\,\mathrm{gpoly}(g_1, g_2))| = |ad| < |\mathrm{LC}(h_1)\mathrm{LC}(g_1)| + |\mathrm{LC}(h_2)\mathrm{LC}(g_2)|,
\end{aligned}$$

which contradicts our assumption that the leading coefficient of our original representation are minimal. Therefore, M contains exactly one element and thus we have a strong Gröbner representation of f w.r.t. $\mathcal{G}$, i. e. $\mathcal{G}$ is a strong Gröbner basis for $\mathcal{I}$. □

This is similar to a statement over fields which can be found in [21]. The point is that these overlap relations or "obstructions" $t_f\mathrm{LM}(f)t_f' = t_g\mathrm{LM}(g)t_g'$ 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 loose any information. Now let $\tau_f = t_f \otimes t_f'$, $\tau_g = t_g \otimes t_g' \in \mathcal{B}^e$, $t \in \mathrm{cm}(\mathrm{LM}(f), \mathrm{LM}(g))$ with $t = \tau_f\mathrm{LM}(f) = \tau_g\mathrm{LM}(g)$. Then there exists a $t' \in \mathrm{cm}(\mathrm{LM}(f), \mathrm{LM}(g))$ that satisfies one of the above four cases (1) – (4) and τ, τ_f', $\tau_g' \in \mathcal{B}^e$, such that $t = \tau t' = \tau_f'\mathrm{LM}(f) = \tau_g'\mathrm{LM}(g)$ and $\tau_f = \tau\tau_f'$, $\tau_g = \tau\tau_g'$. Let

- $\mathrm{spoly}(f, g) = a_f\tau_f f - a_g\tau_g g$
- $\mathrm{gpoly}(f, g) = b_f\tau_f f + b_g\tau_g g$
- $\mathrm{spoly}'(f, g) = a_f\tau_f' f - a_g\tau_g' g$
- $\mathrm{gpoly}'(f, g) = b_f\tau_f' f + b_g\tau_g' g$

be the corresponding S- and G-polynomials. Clearly $\mathrm{spoly}'(f, g)$, $\mathrm{gpoly}'(f, g)$ are first or second type S- and G-polynomials and we have $\mathrm{spoly}(f, g) = \tau\,\mathrm{spoly}'(f, g)$ and $\mathrm{gpoly}(f, g) = \tau\,\mathrm{gpoly}'(f, g)$. Therefore, if $\mathrm{spoly}'(f, g)$, $\mathrm{gpoly}'(f, g)$ reduce to zero w.r.t. $\mathcal{G}$, then so do $\mathrm{spoly}(f, g)$ and $\mathrm{gpoly}(f, g)$.

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 $\mathcal{G} \subseteq \mathcal{P} \setminus \{0\}$. Then $\mathcal{G}$ is called a *reduced Gröbner basis*, if

(1) every $g \in \mathcal{G}$ has leading coefficient with signum 1,
(2) $\mathrm{L}(\mathcal{G} \setminus \{g\}) \subsetneq \mathrm{L}(\mathcal{G})$ for every $g \in \mathcal{G}$ and
(3) $\mathrm{LT}(\mathrm{tail}(g)) \notin \mathrm{L}(\mathcal{G})$ for every $g \in \mathcal{G}$.

The first condition states that, in the case of $\mathcal{R} = \mathbb{Z}$, 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 $\mathcal{I} = \langle f, g \rangle$. Then $\{f, g\}$ is a Gröbner basis for $\mathcal{I}$ with respect to any ordering $x \succ y$ 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 $\mathcal{I}$. 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 $\mathcal{G} \subset R\langle X\rangle$ 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 $L(\mathcal{G})$.*

Proof. It suffices to prove the "if" part. Assume that $\mathcal{G}'$ is a result of a computation up to degree $2d-1$ and $\mathrm{L}(\mathcal{G}) = \mathrm{L}(\mathcal{G}')$. 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 $\mathrm{L}(\mathcal{G})$, then this means that the trivial overlap relations $\mathrm{LM}(f)\mathrm{LM}(g)$, 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 $\mathrm{LM}(f)w\mathrm{LM}(g)$ with $w \in \mathcal{B}$ to be irrelevant. Moreover, we need to take the divisibility condition $\mathrm{LT}(g) \mid \mathrm{LT}(f)$ into account. As a consequence we could replace "Gröbner basis" with "strong Gröbner basis" in Lemma 2.9.

3 CRITICAL PAIRS

To improve BuchbergerAlgorithm, 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 $\mathcal{R}\langle X\rangle$. 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 $t := \mathrm{LM}(f)$ is divisible by (or even equals to) $\mathrm{LM}(g)$. Then $\mathrm{lcm}(\mathrm{LM}(f), \mathrm{LM}(g))$ contains exactly one element, namely t, because it is the only minimal element that is divisible by both leading monomials. Therefore, $\mathrm{spoly}_1^t(f, g)$ and $\mathrm{gpoly}_1^t(f, g)$ are the only first type S- and G-polynomials. However, these are not uniquely determined, we might have more overlap relations of $\mathrm{LM}(f)$, $\mathrm{LM}(g)$, 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 $LC(f) \mid LC(g)$ in $\mathcal{R}$, then every G-polynomial of f and g is redundant.*

Proof. By the hypothesis we have $b = \mathrm{lcm}(\mathrm{LC}(f), \mathrm{LC}(g)) = \mathrm{LC}(f)$. Let $r \in \mathcal{R}$, such that $r\mathrm{LC}(f) = \mathrm{LC}(g)$. Then $\mathrm{LC}(f) = (nr+1)\mathrm{LC}(f) - n\mathrm{LC}(g)$ yields any possible Bézout identity for b, where $n \in \mathbb{Z}$. Thus, with $t = \tau_f\mathrm{LM}(f) = \tau_g\mathrm{LM}(g)$, every G-polynomial of f and g has shape $\mathrm{gpoly}(f, g) = (nr+1)\tau_f f - n\tau_g g = \mathrm{LC}(f)t + n(r\tau_f\mathrm{tail}(f) - \tau_g\mathrm{tail}(g)) + \tau_f\mathrm{tail}(f)$. Subtracting $\tau_f f$, we can reduce this to $n(r\tau_f\mathrm{tail}(f) - \tau_g\mathrm{tail}(g))$. Note that $r\tau_f\mathrm{tail}(f) - \tau_g\mathrm{tail}(g)$ 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 $\mathrm{tail}^0(f) := f$ and $\mathrm{tail}^i(f) := \mathrm{tail}(\mathrm{tail}^{i-1}(f))$ for $i \geq 1$ when $\mathrm{tail}^{i-1}(f) \neq 0$.

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

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

Then $s := spoly_2^w(f, g)$ reduces to zero w.r.t. $\{f, g\}$.

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

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

Remark 3.4. The commutative version of Buchberger's product (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 $w \in \mathcal{B}$, 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 $\mathrm{LM}(f)$ and $\mathrm{LM}(g)$ do not overlap and the leading coefficients are not coprime, i. e. $\mathrm{lcm}(\mathrm{LC}(f), \mathrm{LC}(g)) \neq 1$, then we can make no *a priori* statement about reduction. This only applies to second type S- and G-polynomials. Take for example $f = 4xy + x$, $g = 6zy+z \in \mathbb{Z}\langle X\rangle = \mathbb{Z}\langle x, y, z\rangle$ in the degree left lexicographical ordering with $x \succ y \succ z$. Then $\mathrm{spoly}_2^1(f, g) = 3fzy - 2xyg = 3xzy - 2xyz$ and $\mathrm{gpoly}_2^1(f, g) = (-1)fzy + 1xyg = 2xyzy + xyz - xzy$ 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 $\mathrm{spoly}_1^{xyz}(f, g) = 3fz - 2xg = 3yz - 2xy$ and $\mathrm{gpoly}_1^{xyz}(f, g) = (-1)fz + 1xg = 2xyz - yz + xy$ 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 $\tau_f = \tau_g = 1 \otimes 1$ and therefore $\text{spoly}_1^t(f, g) = a_f f - a_g g$ and $\text{gpoly}_1^t(f, g) = b_f f + b_g g$. This yields a linear equation

$$\begin{pmatrix} \text{spoly}_1^t(f, g) \\ \text{gpoly}_1^t(f, g) \end{pmatrix} = \begin{pmatrix} a_f & -a_g \\ b_f & b_g \end{pmatrix} \begin{pmatrix} f \\ g \end{pmatrix},$$

where the defining matrix has determinant $a_f b_g + a_g b_f = 1$, and thus is invertible over $\mathcal{R}$. 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 pairs 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 $a, b \in \{f, g, h\}$ let $\text{lcm}(LM(a), LM(b)) \neq \emptyset$ and fix $T_{ab} \in \text{lcm}(LM(a), LM(b))$ and choose $\tau_{ab} \in \mathcal{B}^e$ with $\tau_{ab} LM(a) = T_{ab}$. There exist $\tau_{ba} \in \mathcal{B}^e$, such that $\tau_{ba} LM(b) = T_{ab}$. We assume that $T_{ab} = T_{ba}$. Furthermore, let*

1. *$T_{hg} = T_{gh}$ be divisible by both T_{hf} and T_{gf} with $\delta_{gf} T_{hf} = T_{hg}$ and $\delta_{hf} T_{gf} = T_{gh}$ for some $\delta_{gf}, \delta_{hf} \in \mathcal{B}^e$,*
2. *$LC(f) \mid lcm(LC(g), LC(h))$ over $\mathcal{R}$ and*
3. *$\text{spoly}_1^{T_{fg}}(f, g)$ and $\text{spoly}_1^{T_{fh}}(f, h)$ both have strong Gröbner representations w.r.t. $\mathcal{G}$.*

Then $\text{spoly}_1^{T_{gh}}(f, g)$ has a strong Gröbner representation w.r.t. $\mathcal{G}$.

Proof. Let $c_{ab} := \dfrac{\text{lcm}(\text{LC}(a), \text{LC}(b))}{\text{LC}(a)}$ for $a, b \in \{f, g, h\}$. Then

$$\begin{aligned}
&\frac{c_{hg}}{c_{hf}} \delta_{gf} \text{spoly}_1^{T_{fh}}(f, h) - \frac{c_{gh}}{c_{gf}} \delta_{hf} \text{spoly}_1^{T_{fg}}(f, g) \\
=&\frac{c_{hg}}{c_{hf}} \delta_{gf} (c_{fh} \tau_{fh} f - c_{hf} \tau_{hf} h) - \frac{c_{gh}}{c_{gf}} \delta_{hf} (c_{fg} \tau_{fg} f - c_{gf} \tau_{gf} g) \\
=&c_{gh} \delta_{hf} \tau_{gf} g - c_{hg} \delta_{gf} \tau_{hf} h + \left(\frac{c_{hg} c_{fh}}{c_{hf}} \delta_{gf} \tau_{fh} - \frac{c_{gh} c_{fg}}{c_{gf}} \delta_{hf} \tau_{fg} \right) f
\end{aligned}$$

and with $\tau_{hg} \text{LM}(h) = T_{hg} = \delta_{gf} T_{hf} = \delta_{gf} \tau_{hf} \text{LM}(h)$ we have $\delta_{gf} \tau_{hf} = \tau_{hg} \in \mathcal{B}^e$. Analogously $\delta_{hf} \tau_{gf} = \tau_{gh}$ and thus the first term equals $\text{spoly}_1^{T_{gh}}(g, h)$. Moreover, one can observe that $c_{fh} c_{hg} c_{gf} = c_{fg} c_{gh} c_{hf}$. Finally, $\delta_{gf} T_{fh} = \delta_{gf} T_{hf} = T_{hg}$ and $T_{gh} = \delta_{hf} T_{gf} = \delta_{hf} T_{fg}$ implies $\delta_{gf} \tau_{fh} = \delta_{hf} \tau_{fg}$ in $\mathcal{P}^e$, hence

$$\frac{c_{hg}}{c_{hf}} \delta_{gf} \text{spoly}_1^{T_{fh}}(f, h) - \frac{c_{gh}}{c_{gf}} \delta_{hf} \text{spoly}_1^{T_{fg}}(f, g) = \text{spoly}_1^{T_{gh}}(g, h),$$

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 $\text{spoly}_2^{\tilde{w}}(h, g)$ if we choose w or $\tilde{w}$, such that either $\text{LM}(g) w \text{LM}(h) = T_{gh}$ or $\text{LM}(h) \tilde{w} \text{LM}(g) = T_{hg}$. □

We give a similar criterion for G-polynomials

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 τ_{ab} from Buchberger's S-chain criterion. Let*

1. *$T_{hg} = T_{gh}$ be divisible by both T_{hf} and T_{gf} with $\delta_{gf} T_{hf} = T_{hg}$ and $\delta_{hf} T_{gf} = T_{gh}$ for some $\delta_{gf}, \delta_{hf} \in \mathcal{B}^e$ and*
2. *$LC(f) \mid gcd(LC(g), LC(g))$ with $d := \dfrac{gcd(LC(g), LC(g))}{LC(f)}$.*

Then $\text{gpoly}_1^{T_{gh}}(g, h)$ has a strong Gröbner representation w.r.t. $\mathcal{G}$.

Proof. First of all note that

$$\begin{aligned}
\text{gpoly}_1^{T_{gh}} =& \text{gcd}(\text{LC}(g), \text{LC}(h)) T_{gh} + b_g \tau_{gh} \text{ tail}(g) + b_h \tau_{hg} \text{ tail}(h), \\
\text{spoly}_1^{T_{fg}} =& \frac{\text{LC}(g)}{\text{LC}(f)} \tau_{fg} f - \tau_{gf} g = \frac{\text{LC}(g)}{\text{LC}(f)} \tau_{fg} \text{ tail}(f) - \tau_{gf} \text{ tail}(g), \\
\text{spoly}_1^{T_{fh}} =& \frac{\text{LC}(h)}{\text{LC}(f)} \tau_{fh} f - \tau_{hf} h = \frac{\text{LC}(h)}{\text{LC}(f)} \tau_{fh} \text{ tail}(f) - \tau_{hf} \text{ tail}(h)
\end{aligned}$$

(leaving out arguments of S- and G-polynomials). Since T_{fh} divides T_{gh}, there exists $w \in \mathcal{B}^e$ with $w \text{LM}(f) = T_{gh}$. Then

$$w \text{LM}(f) = T_{gh} = \delta_{gf} T_{fh} = \delta_{gf} T_{fh} \text{LM}(f).$$

Hence, $w = \delta_{gf} \tau_{fh}$ and analogously $w = \delta_{hf} \tau_{fg}$.

Moreover, $dw \text{LC}(f) \text{LM}(f) = \text{gcd}(\text{LC}(g), \text{LC}(h)) T_{gh}$ and finally we obtain

$$\begin{aligned}
&\text{gpoly}_1^{T_{gh}} - dwf + b_g \delta_{hf} \text{spoly}_1^{T_{fg}} + b_h \delta_{gf} \text{spoly}_1^{T_{fh}} \\
=& \text{gcd}(\text{LC}(g), \text{LC}(h)) T_{gh} - (\text{gcd}(\text{LC}(g), \text{LC}(h)) T_{gh} + dw \text{ tail}(f)) \\
&+ b_g \tau_{gh} \text{ tail}(g) + b_g \delta_{hf} \left(\frac{\text{LC}(g)}{\text{LC}(f)} \tau_{fg} \text{ tail}(f) - \tau_{gf} \text{ tail}(g) \right) \\
&+ b_h \tau_{hg} \text{ tail}(h) + b_h \delta_{gf} \left(\frac{\text{LC}(h)}{\text{LC}(f)} \tau_{fh} \text{ tail}(f) - \tau_{hf} \text{ tail}(h) \right) \\
=& b_g \tau_{gh} \text{ tail}(g) + b_h \tau_{hg} \text{ tail}(h) - dw \text{ tail}(f) + b_g \frac{\text{LC}(g)}{\text{LC}(f)} \delta_{hf} \tau_{fg} \text{ tail}(f) \\
&- b_g \underbrace{\delta_{hf} \tau_{gf}}_{=\tau_{gh}} \text{ tail}(g) + b_h \frac{\text{LC}(h)}{\text{LC}(f)} \underbrace{\delta_{gf} \tau_{fh}}_{=\delta_{hf} \tau_{fg}} \text{ tail}(f) - b_h \underbrace{\delta_{gf} \tau_{hf}}_{=\tau_{hg}} \text{ tail}(h) \\
=& \left(\frac{b_g \text{LC}(g) + \text{LC}(h)}{\text{LC}(f)} \delta_{hf} \tau_{fg} - dw \right) \text{tail}(f) = d(\delta_{hf} \tau_{fg} - w) \text{ tail}(f) = 0.
\end{aligned}$$

Thus,

$$\text{gpoly}_1^{T_{gh}} = dwf - b_g \delta_{hf} \text{spoly}_1^{T_{fg}} - b_h \delta_{gf} \text{spoly}_1^{T_{fh}}$$

is a strong Gröbner representation of $\text{gpoly}_1^{T_{gh}}(g, h)$. □

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 $\mathcal{P} = \mathbb{Z}\langle x, y, z\rangle$ with the degree left lexicographical ordering and $x \succ y \succ z$.

Example 4.1. (cf. [1]) We consider the ideal $\mathcal{I} = \langle f_1 = yx - 3xy - 3z,\ f_2 = zx - 2xz + y,\ f_3 = zy - yz - x\rangle \subset \mathcal{P}$.
At first, we analyze this ideal over the field $\mathbb{Q}$:

```
LIB "freegb.lib"; // initialization of free algebras
ring r = 0,(z,y,x),Dp; // degree left lex ord on z>y>x
ring R = freeAlgebra(r,7); // length bound is 7
ideal I = y*x - 3*x*y - 3*z, z*x - 2*x*z +y, z*y-y*z-x;
option(redSB); option(redTail); // for minimal reduced GB
option(intStrategy); // avoid divisions by coefficients
ideal J = twostd(I); // compute a two-sided GB of I
J; // prints generators of J
```

The output is a finite Gröbner basis

$$\{4xy + 3z,\ 3xz - y,\ 4yx - 3z,\ 2y^2 - 3x^2,\ 2yz + x,\\ 3zx + y, 2zy - x,\ 3z^2 - 2x^2,\ 4x^3 + x\}.$$

As we see, original generators have decomposed. In order to compute their expressions in the Gröbner basis above, one can use the `lift` command. In particular, $yx - 3xy - 3z = -\frac{3}{4}(4xy + 3z) + \frac{1}{4}(4yx - 3z)$. Now, it seems from the form of leading monomials, that $\mathbb{Q}\langle x, y, z\rangle/J$ is finite dimensional vector space. Let us check it:

```
LIB "fpadim.lib"; // load the library for K-dimensions
lpMonomialBasis(7,0,J); // compute all monomials
// of length up to 7 in Q<x,y,z>/J
```

which results in $\{1, z, y, x, x^2\}$.

Now, we proceed to work over $\mathbb{Z}$. For doing this, we need just one change in the code above, namely in the definition of a ring

```
ring r = integer,(z,y,x),Dp;
```

The output has plenty of elements in each degree (which is the same as length because of the degree ordering), what hints at potentially infinite Gröbner basis (what we confirm below) and the elements, which can be subsequently constructed, are

$$\{f_1,\ f_2,\ f_3,\ 12xy + 9z,\ 9xz - 3y,\ 6y^2 - 9x^2,\ 6yz + 3x,\\ 3z^2 + 2y^2 - 5x^2,\ 6x^3 - 3yz,\ 4x^2y + 3xz,\ 3x^2z + 3xy + 3z,\\ 2xy^2 + 3x^3 + 3yz + 3x,\ 3xyz + 3y^2 - 3x^2,\ 2y^3 + x^2y + 3xz,\\ 2x^4 + y^2 - x^2,\ 2x^3y + 3y^2z + 3xy + 3z,\ x^2yz + xy^2 - x^3,\\ xy^2z - y^3 + x^2y,\ x^5 - y^3z - xy^2 + x^3,\ y^3z^2 - x^4y,\\ x^4z + x^3y + 2y^2z + x^2z + 3xy + 3z,\ xy^3z - y^4 + x^4 - y^2 + x^2,\\ xy^4z - y^5 + x^2y^3,\ xy^5z - y^6 + x^4y^2 + y^4 + x^4 + 2y^2 - 2x^2\}.$$

We can show that for every $2 \le i \in \mathbb{N}$ $\mathcal{I}$ contains an element $xy^iz +$ l.o.t. and these are the only polynomials which have to be added to the above set in order to obtain a Gröbner basis for $\mathcal{I}$. 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 4.2. Let $\mathcal{I} = \langle f_1 = yx - 3xy - z,\ f_2 = zx - xz + y,\ f_3 = zy - yz - x\rangle \subset \mathcal{P}$. Then $\mathcal{I}$ has a finite strong Gröbner basis, namely

$$\{f_1,\ f_2,\ f_3,\ 8xy + 2z,\ 4xz - 2y,\ 4yz + 2x,\\ 2x^2 - 2y^2,\ 4y^2 - 2z^2,\ 2z^3 - 2xy\}.$$

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 $M := \mathbb{Z}\langle x, y, z\rangle/I$ might have nontrivial 2-torsion, i. e. there is a nonzero submodule $T_2(M) := \{p \in M : \exists n \in \mathbb{N}_0\ 2^n \cdot p \in I\}$. 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:

```
LIB "freegb.lib";
ring r = integer,(x,y,z),(c,dp); // position-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;
option(redSB); option(redTail);
ideal J = twostd(I);    module N;
N = 2*ncgen(1)*gen(1)+ncgen(2)*gen(2),J*ncgen(1)*gen(1);
module SN = twostd(N);   SN;
```

Above, `gen(i)` stands for the i-th canonical basis vector (commuting with everything) and `ncgen(i)` - 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:

```
...
SN[9]=[0,z*z*z*ncgen(2)-x*y*ncgen(2)]
SN[10]=[2*ncgen(1),ncgen(2)]
SN[11]=[z*y*ncgen(1)-y*z*ncgen(1)-x*ncgen(1)]
...
```

From this output we gather all vectors with 0 in the first component `ncgen(1)`, which results into an ideal, whose Gröbner basis is

$$\{zy - yz - x,\ zx - xz + y,\ yx + xy,\ 2yz + x,\\ 2xz - y,\ 2y^2 - z^2,\ 4xy + z,\ x^2 - y^2,\ z^3 - xy\}.$$

Another step of the colon computation terminates, therefore we have computed the saturation ideal of I at 2, denoted by $L = I : 2^\infty \subset \mathbb{Z}\langle x, y, z\rangle$. It is the presentation for the 2-torsion submodule $T_2(M) = \mathbb{Z}\langle x, y, z\rangle L/I$ and, moreover, $2 \cdot L \subset I \subset L$ holds.

Example 4.3. Another ideal that has a finite Gröbner basis is $\mathcal{I} = \langle f_1 = yx - 3xy,\ f_2 = zx + y^2,\ f_3 = zy - yz + z^2\rangle$. A Gröbner basis of $\mathcal{I}$ is given by

$$\{f_1,\ f_2,\ f_3,\ 2y^3 + y^2z - 2yz^2 + 2z^3,\ 14yz^3 - 28z^4,\\ y^2z^2 - 4yz^3 + 6z^4,\ 27xy^2z - 54xyz^2 + 54xz^3 + y^4,\ 14z^5,\\ 2yz^4 - 6z^5,\ y^4z,\ y^5,\ 2xyz^3 - 4xz^4,\ 27xy^3z,\ 2z^6,\ 2xz^5\}.$$

Example 4.4. In this example we have to run a Gröbner basis of $\langle f_1 = zy - yz + z^2,\ f_2 = zx + y^2,\ f_3 = yx - 3xy\rangle$ 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 left version and elimination orderings do not result in finite sets

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

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 $\mathbb{Q}$ 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

$$\{f_1,\ f_2,\ f_3,\ 2y^3 + y^2z - 2yz^2 + 2z^3,\ yz^3 - 2z^4,$$
$$y^2z^2 - 2z^4,\ xy^2z - 2xyz^2 + 2xz^3,\ z^5\}.$$

5 IMPLEMENTATION

We have created a powerful implementation called Letterplace [13] in the framework of Singular. Its' extension to coefficient rings like $\mathbb{Z}$ addresses the following functions with the current release for ideals and subbimodules of a free bimodule of a finite rank. All these can be executed with respect to the orderings like degree right and left lexicographical, weighted degree left lexicographical, left and right total elimination, extra weight ordering extension. For modules, position-over-term and term-over-position constructions are available.

`twostd`: a two-sided Gröbner basis; run with respect to an elimination ordering, it allows t eliminate variables, and thus to compute kernels of ring morphisms and preimages of ideals under such;
`reduce (NF)`: a normal form of a vector or a polynomial with respect to a two-sided Gröbner basis;
`syz`: a generating set of a syzygy bimodule of an input;
`lift`: computation of a transformation matrix between a module and its submodule, in other words expressing generators of a submodule in terms of generators of a module;
`liftstd`: computation of a two-sided Gröbner basis and a transformation matrix of a given ideal or subbimodule and, optionally, a syzygy bimodule.

6 CONCLUSION AND FUTURE WORK

Following Mora's "manual for creating own Gröbner basis theory" [18], we have considered the case of free non-commutative Gröbner bases for ideals and bimodules over $\mathbb{Z}\langle X\rangle$. We have derived novel information on the building critical pairs and on criteria to discard them when possible. Armed with this theoretical and algorithmic knowledge, we have created an implementation in a Singular subsystem Letterplace, which offers a rich functionality at a decent speed.

In this paper we have demonstrated several important applications of our algorithms and their implementation, in particular the determination of torsion submodules with respect to natural numbers.

A further adaptation of our implementation to the explicitly given $\mathbb{Z}/m\mathbb{Z}$ is planned, as well as the development (also a theoretical) of one-sided Gröbner bases in factor algebras (over fields, Letterplace already offers `rightStd`). More functions for dealing with matrices will make possible the usage of our implementation as a backend from the system HomAlg [4]. This system performs homological algebra computations within computable Abelian categories and uses other computer algebra systems as backends for concrete calculations with matrices over rings.

7 ACKNOWLEDGEMENTS

The authors are grateful to Gerhard Pfister (Kaiserslautern), Anne Frühbis-Krüger (Oldenburg), Leonard Schmitz, Eva Zerz (RWTH Aachen) and Evelyne 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

[1] J. Apel. 2000. Computational ideal theory in finitely generated extension rings. *Theor. Comput. Sci.* 244, 1-2 (2000), 1–33.
[2] J. Apel and U. Klaus. 1991. *FELIX* – an assistant for algebraists. In *Proc. of the International Symposium on Symbolic and Algebraic Computation (ISSAC'91)*. ACM Press, 382–389.
[3] J. Apel and U. Klaus. 1998. FELIX, a Special Computer Algebra System for the Computation in Commutative and Non-commutative Rings and Modules. http://felix.hgb-leipzig.de
[4] Mohamed Barakat, Sebastian Gutsche, and Markus Lange-Hegermann. 2019. homalg - A homological algebra meta-package for computable Abelian categories. https://homalg-project.github.io/homalg_project/homalg/.
[5] Christian Eder and Tommy Hofmann. 2019. Efficient Gröbner Bases Computation over Principal Ideal Rings. https://arxiv.org/abs/1906.08543.
[6] Christian Eder, Gerhard Pfister, and Adrian Popescu. 2016. New Strategies for Standard Bases over Z. https://arxiv.org/abs/1609.04257.
[7] Christian Eder, Gerhard Pfister, and Adrian Popescu. 2018. Standard Bases over Euclidean Domains. https://arxiv.org/abs/1811.05736.
[8] Heinz Kredel. 2015. Parametric Solvable Polynomial Rings and Applications. In *Proc. CASC'15*, Vladimir P. Gerdt, Wolfram Koepf, Werner M. Seiler, and Evgenii V. Vorozhtsov (Eds.). Springer International Publishing, Cham, 275–291. https://doi.org/10.1007/978-3-319-24021-3_21
[9] Roberto La Scala. 2014. Extended letterplace correspondence for nongraded noncommutative ideals and related algorithms. *Int. J. Algebra Comput.* 24, 8 (2014), 1157–1182.
[10] Roberto La Scala and Viktor Levandovskyy. 2009. Letterplace ideals and non-commutative Gröbner bases. *Journal of Symbolic Computation* 44, 10 (2009), 1374–1393. https://doi.org/10.1016/j.jsc.2009.03.002
[11] Roberto La Scala and Viktor Levandovskyy. 2013. Skew polynomial rings, Gröbner bases and the letterplace embedding of the free associative algebra. *JSC* 48, 1 (2013), 110–131. http://dx.doi.org/10.1016/j.jsc.2012.05.003
[12] V. Levandovskyy. 2005. *Non–commutative computer algebra for polynomial algebras: Gröbner bases, applications and implementation.* Doctoral thesis, Universität Kaiserslautern.
[13] Viktor Levandovskyy, Karim Abou Zeid, and Hans Schönemann. 2020. Singular:Letterplace — A Singular 4-1-2 Subsystem for Non-commutative Finitely Presented Algebras. http://www.singular.uni-kl.de.
[14] Viktor Levandovskyy, Grischa Studzinski, and Benjamin Schnitzler. 2013. Enhanced Computations of Gröbner Bases in Free Algebras as a New Application of the Letterplace Paradigm. In *Proc. of the International Symposium on Symbolic and Algebraic Computation (ISSAC'13)*, Manuel Kauers (Ed.). ACM Press, 259 – 266.

[15] Huishi Li. 2012. Algebras Defined by Monic Gröbner Bases over Rings. *International Mathematical Forum* 7 (2012), 1427–1450.
[16] Daniel Lichtblau. 2012. Effective computation of strong Gröbner bases over Euclidean domains. *Illinois Journal of Mathematics* 56 (2012), 177–194.
[17] Thomas Markwig, Yue Ren, and Oliver Wienand. 2015. Standard Bases in mixed Power Series and Polynomial Rings over Rings. *Journal of Symbolic Computation* 79 (09 2015). https://doi.org/10.1016/j.jsc.2016.08.009
[18] Teo Mora. 2016. *Solving Polynomial Equation Systems IV: Volume 4, Buchberger Theory and Beyond* (1st ed.). Cambridge University Press.
[19] Franz Pauer. 2007. Gröbner bases with coefficients in rings. *Journal of Symbolic Computation* 42 (2007), 1003 – 1011. https://doi.org/10.1016/j.jsc.2007.06.006
[20] F. Leon Pritchard. 1996. The ideal membership problem in non-commutative polynomial rings. *J. Symb. Comput.* 22, 1 (1996), 27–48. https://doi.org/10.1006/jsco.1996.0040
[21] G. Studzinski. 2013. *Implementation and Applications of Fundamental Algorithms relying on Gröbner Bases in Free Associative Algebras.* Doctoral thesis, RWTH Aachen University.