GRÖBNER BASES OVER $\mathbb{K}\langle X\rangle$ AND $\mathbb{Z}\langle X\rangle$ IN THEORY AND PRACTICE

TOBIAS METZLAFF, VIKTOR LEVANDOVSKYY, HANS SCHÖNEMANN, AND KARIM ABOU ZEID

Abstract. We report on the recent theoretical adoption and implementation of non-commutative Gröbner bases both over fields and over the ring $\mathbb{Z}$ with more principal ideal rings in mind, which was accomplished in parallel. In both cases we are based on the Letterplace correspondence of La Scala and Levandovskyy. The corresponding subsystem of Singular is called Letterplace. In addition to division with remainder algorithm, we also provide algorithms for syzygy bimodules and lifting. Despite a certain similarity, the case of rings as coefficients demonstrates crucial intrinsic differences, compared to the case of fields. We also remark on models of computation and on decidability.

Introduction

The aim of this article is to introduce the subsystem Letterplace [10] of a computer algebra system Singular for computations with finitely presented algebras over fields and over the ring $\mathbb{Z}$. The name Letterplace has been selected since the subsystem implements Letterplace technology, initiated by La Scala and Levandovskyy in [8, 9] and further extended in [12, 7]. This technology is based on the correspondence between two-sided ideals of a free associative algebra and certain ideals in countably generated commutative polynomial ring. It extends to the correspondence between generating sets of ideals and to Gröbner bases as well. Key observations on the Letterplace technology are the following:

- Since free algebras in more than one variable are not Noetherian, one has to work with infinite Gröbner bases; a typical **model of computation** is to perform computations of Gröbner bases (also in the algorithms, which use them) *up to a degree (length) bound*.
- A proper implementation of an algorithm for computing a Gröbner basis over the free algebra over a ring is able to find a finite Gröbner basis (if it exists with respect to the fixed monomial ordering) in finitely many steps; it was formulated by Mora and Pritchard [15, 14].
- Another important result of Pritchard which we use is on the **decidability**: notably, over the free algebra over a ring, verifying ideal membership while knowing the positive result in advance can be done in finitely many steps.
- Because of the bound given in advance, Letterplace computations might be seen as happening in a commutative polynomial ring in finitely many variables. In particular, data structures of such rings and their ideals (up to the lowest levels of operations with them) can be used.
- As indicated in [12], the algorithm computing a Letterplace Gröbner basis can be put exactly in the form of the generalized Buchberger's algorithm for computing Gröbner basis over the free algebra. This one, in turn, can be formulated as augmented Buchberger's algorithm for commutative rings. This makes the implementation much easier.

In order to give a reader an impression 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 speaking on the commutative case!), consider the following example.

Example 0.1. The generating set $\{2x, 3y\} \subset \mathbb{Z}\langle x, y\rangle$ delivers a **finite strong** Gröbner basis $\{2x, 3y, yx, xy\}$. However, when considered over $\mathbb{Z}\langle x, y, z_1, \ldots, z_m\rangle$ for any $m \geq 1$ the generating set $\{2x, 3y\}$ has an **infinite** Gröbner basis, containing e. g. $xz_i^k y$, $yz_i^k x$ for any $k \in \mathbb{N}$.

In the book [14] Teo Mora has presented "a Do-It-Yourself manual for creating your own Gröbner bases theory" over *effective* associative rings. It is a unifying theoretical framework for handling very general rings. However procedures (and 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 [6]. We are based on SINGULAR and utilize its' long and successful experience with data structures and algorithms in commutative algebra. Fortunately, 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. [4, 5, 3]. 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.

1. Non-commutative Gröbner Bases over rings

We start with a commutative coefficient ring R, which is supposed to be unital and Euclidean, thus principal. By $P \coloneqq R\langle X\rangle$ we denote the free R-algebra of a finite alphabet $X = \{x_1, \ldots, x_n\}$ with coefficients in R. The R-basis B of this algebra is the monoid consisting of all finite words on X (including the empty word identified with $1 \in R$), which we call *monomials*. Let us fix a monomial ordering $\prec$ on X, that is a well-ordering, which is compatible with the multiplication on the monoid, we can define the leading coefficient "LC", leading monomial "LM", leading term "LT" and tail "tail" for every non-zero polynomial in P. The *degree* of a non-zero polynomial $f \in P$ is the maximal length of a monomial occurring in f.

A non-empty set $F \subseteq P$ of polynomials yields a leading ideal $\mathrm{LT}(F) = \langle \mathrm{LT}(f) : f \in F\rangle$. A Gröbner basis G for an ideal I of $R\langle X\rangle$ is a generating set of I satisfying the property $\mathrm{LT}(G) = \mathrm{LT}(I)$.

It is known for the classical case with coefficients in fields, that this definition suffices to obtain unique normal forms after a reduction process, which is the consecutive division with remainder by leading terms. This holds, regardless, whether the polynomials are commutative or non-commutative. In [4, 5] the commutative case of $\mathbb{Z}[X]$ was studied with the conclusion, that one cannot obtain a *unique* normal form with the above definition.

For instance, consider the ideal I generated by $G = \{2x, 3y\} \subset \mathbb{Z}[X]$ over the commutative polynomial ring. The S-polynomial of $2x$ and $3y$ is zero. Thus G is a Gröbner basis. However, $I \ni -2x \cdot y + x \cdot 3y = xy$, so xy must have normal form zero w.r.t. G. But its leading term is not divisible by any term of G. Notably, the same example reveals even more anomalies in the non-commutative case.

A Gröbner basis G for an ideal I is called *strong Gröbner basis*, if for every $f \in I$ its leading term $\mathrm{LT}(f)$ is divisible by $\mathrm{LT}(g)$ for some $g \in G$. In the non-commutative case divisibility means subword containment or equivalently full overlap.

Lichtblau proved in [13] for the commutative case over $\mathbb{Z}$, that having a strong Gröbner basis is equivalent to obtaining a unique normal form after reduction. He introduces an algorithm in theory to obtain a strong Gröbner basis using, next to S-polynomials, also G-polynomials. Furthermore, he gives criteria for discarding critical pairs. Eder, Pfister and Popescu tested this theory in practice in [5] and also addressed the problem of coefficient swell, that occurs over $\mathbb{Z}$.

We have proven a generalized version of the characterization given by Lichtblau for the case of non-commutative Gröbner bases in [11]. Moreover, we have studied the criteria that are essential for a practical implementation of Buchberger's algorithm. We recall some of these results below.

Theorem 1.1. ([11], 2.2) Let $G \subseteq P \setminus \{0\}$ and $\{0\} \neq I \subseteq P$ be an ideal. Then the following statements are equivalent with respect to G and $\prec$.

(1) G is a strong Gröbner basis for I.
(2) Every $f \in I \setminus \{0\}$ has a strong Gröbner representation, that is a representation of f by the elements of G, where every leading monomial of the representation is, except for one, strictly smaller than $\mathrm{LM}(f)$.
(3) Every $f \in P \setminus \{0\}$ has a unique normal form after reduction by G.

To construct a Gröbner basis we introduce S- and G-polynomials. For two polynomials $f, g \in P \setminus \{0\}$, find minimal $a_f, a_g, b_f, b_g \in R$ with

(1) $a_f \,\mathrm{LC}(f) = a_g \,\mathrm{LC}(g)$ and
(2) $c \coloneqq \gcd(\mathrm{LC}(f), \mathrm{LC}(g)) = b_f \,\mathrm{LC}(f) + b_g \,\mathrm{LC}(g)$ (Bézout identity for leading coefficients).

Next, we can find a monomial $t \in B$ that is divisible by both leading monomials and minimal with this property.

The S-polynomial is constructed with the syzygy relation corresponding to t of the leading monomials. The G-polynomial is constructed in a way, such that it has leading term ct. Let $t = t_f^1 \,\mathrm{LM}(f)\, t_f^2 = t_g^1 \,\mathrm{LM}(g)\, t_g^2$ for $t_f^1, t_f^2, t_g^1, t_g^2 \in B$. Then

$$\mathrm{spoly}(f, g) \coloneqq a_f \, t_f^1 \, f \, t_f^2 - a_g \, t_g^1 \, g \, t_g^2 \qquad \text{and} \qquad \mathrm{gpoly}(f, g) \coloneqq b_f \, t_f^1 \, f \, t_f^2 + b_g \, t_g^1 \, g \, t_g^2.$$

Since the monomial t is not unique, we distinguish between two kinds of S- and G-polynomials. The first kind is related to non-trivial one-sided overlap relations of the leading monomials, the second kind to trivial ones. In the field case Buchberger's product criterion tells us, that we do not need to consider trivial overlap relations. Notably, this is not true over rings!

Consider the ideal $I = \langle 2x, 3y\rangle \subseteq \mathbb{Z}\langle x, y, z\rangle$ from Example 3.1 with $m = 1$. The monomials generators have only trivial overlaps xy and yx, which are also contained in the ideal. But we also construct $xz^k y = -2x \cdot z^k y + xz^k \cdot 3y \in I$ for every power z^k, $k \geq 2$. Each such new element is neither reducible by any generator nor by xy or yx. In other words, we have to add every $xz^k y$, $k \geq 0$, to the generating set in order to obtain a strong Gröbner basis. This leads to the next problem, especially for an implementation: The strong Gröbner basis is infinite and we can only compute up to a degree bound, unless criteria for the finiteness Gröbner bases can be applied.

The two kinds of S- and G-polynomial only use one-sided overlap relations. We prove, that this suffices to construct a Gröbner basis. The argument goes back to Pritchard [15].

Theorem 1.2. ([11], 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 kind S- and G-polynomials reduce to zero with respect to G.

2. Forming Pairs

In practice, one often deals with large sets of hundreds or thousands of polynomials. The (critical) pairs have to be built, then S- and G-polynomials to have be computed for each pair, which, after the reduction, might add more polynomials to the set. Strategies on how to choose two polynomials among all the given ones are therefore of big importance for the overall performance. We studied the commutative ring version and non-commutative field version of *product* and of *chain* criteria and generalized or derived new criteria for the non-commutative ring case.
A problem in the non-commutative case is, that we have to deal with not just unique, but several S- and G-polynomials for one pair of polynomials. This results from several overlap relations of the leading monomials.
A trivial criterion, that holds in both the commutative and non-commutative case is the following one for G-polynomials: If one leading coefficient divides the other, then every first and second kind G-polynomial of the pair reduces to zero. By this, G-polynomials are redundant in the field case.
As we have already mentioned, there are only non-trivial overlap relations to be considered in the field case. This results from the product criterion. We found an analog statement for the ring case, but the conditions are much more specific.

Lemma 2.1. ([11], 3.3) Let $f, g \in P \setminus \{0\}$ and $w \in B$, with $w = 1$ possible, such that

(1) $\mathrm{LC}(f)$ and $\mathrm{LC}(g)$ are coprime over R,
(2) $\mathrm{LM}(f)$ and $\mathrm{LM}(g)$ have only trivial overlaps and
(3) for all $i, j \geq 1$, $\mathrm{LM}(\mathrm{tail}^i(f))w\,\mathrm{LM}(g) = \mathrm{LM}(f)w\,\mathrm{LM}(\mathrm{tail}^j(g))$ is not satisfied by w, where, tail^i denotes the tail of tail^{i-1} for $i \geq 2$ and $\mathrm{tail}^1 = \mathrm{tail}$.

Then $\mathrm{spoly}_2^w(f, g)$ reduces to zero w.r.t. $\{f, g\}$.

A chain criterion is very important tool in practice. Under certain conditions it allows us to use two pairs of three, built on the set of polynomials $\{f, g, h\}$, to conclude that the third pair one is useless. We present one version for S-polynomials and one for G-polynomials in [11]. Similarly to the above product criterion, both of them are criteria for specific overlap relations and cannot be generalized in order to be applied to all S- or G-polynomials.

Lemma 2.2 ([11])**.** Suppose that $\mathrm{LM}(f) = \mathrm{LM}(g) =: t$, then we can write

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

and the matrix at the right side has determinant 1 over $\mathbb{Z}$. Then we can replace a pair (f, g) by the pair, consisting of their S- and G-polynomials of the first kind corresponding to t.

3. Examples

In this section we present the code in the Singular C-like language for examples, illustrating what and how one can compute with Singular:Letterplace over fields and over $\mathbb{Z}$. Also, we provide information on previously covered technical questions.

Example 3.1. Consider the ideal $I = \langle yx - 3xy - 3z,\ zx - 2xz + y,\ zy - yz - x\rangle$ from $\mathbb{Z}\langle x, y, z\rangle$ with respect to the degree left lexicographical ordering $z \succ y \succ x$.
At first, we analyze this ideal over the field $\mathbb{Q}$, thus in $\mathbb{Q}\langle x, y, z\rangle$:

```
LIB "freegb.lib"; // import library for 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; // print generators of J
```

The output is a Gröbner basis, which is finite (the length bound is 7 while the maximal length of a generator is 3 and $7-1 \geq 2\cdot 3$ see e. g. [12])

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

Note, that the original generators have decomposed. In order to compute their presentations in terms of the elements of the Gröbner basis above, one can use the command `lift` or `liftstd`. In particular, $yx-3xy-3z = -\frac{3}{4}(4xy+3z)+\frac{1}{4}(4yx-3z)$. From the form of leading monomials, we conjecture that $\mathbb{Q}\langle x,\, y,\, z\rangle/J$ is a 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 returns $\{1,\, z,\, y,\, x,\, x^2\}$, so $\dim_{\mathbb{Q}} \mathbb{Q}\langle x,\, y,\, z\rangle/J = 5$.

Now, we proceed to work over $\mathbb{Z}$ and restart a SINGULAR session. We declare $\mathbb{Z}$ in the definition of a ring, the rest is the same as above:

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

The output has elements in each degree, thus a Gröbner basis is rather infinite (what we confirm below) and the elements, which can be subsequently constructed, are

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

Indeed, we can show that $\forall i \geq 2$ I contains an element with the leading monomial xy^iz. Therefore this Gröbner basis is infinite, but can be presented in finite terms. Note, that the original generators have been preserved in a Gröbner basis, while over $\mathbb{Q}$ they were decomposed.

Intermezzo. Free bimodules over the supported rings are constructed as follows: the rank of a free bimodule has to be specified in advance. The i-th canonical generator is denoted by `ncgen(i)` and it commutes only with constants. The usual SINGULAR's i-th canonical basis vector (commuting with everything) is denoted by `gen(i)`. Their combination allows manipulations with submodules of free bimodules, in particular Gröbner bases.

Example 3.2. Let $I = \langle f_1 = yx-3xy-z,\, f_2 = zx-xz+y,\, f_3 = zy-yz-x\rangle \subset \mathbb{Z}\langle x,\, y,\, z\rangle$. Then 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. We wish to compute the ideal $I:2^\infty = \{r \in \mathbb{Z}\langle x,y,z\rangle \mid \exists n \in \mathbb{N}_0\ 2^n\cdot r \in I\}$, called the *saturation ideal*of I at 2. Since the ground ring commutes with all variables, we can adopt the classical method (by Caboara and Traverso) for computing saturation via colon (or quotient)

ideal to our situation. Though the `quotient` command of the commutative part of SINGULAR has not yet been generalized in the implementation of LETTERPLACE, 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 is the 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;
```

The output is a list of vectors, and pretty-printed it 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)]
...
```

This output is sorted according to the ordering. We gather all vectors with 0 in the first component `ncgen(1)` like `SN[9]` above, 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 $L = I : 2^\infty \subset \mathbb{Z}\langle x,\ y,\ z\rangle$. Notably $2 \cdot L \subset I \subset L$ holds.

Example 3.3. 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 that we have computed a complete finite Gröbner basis. We use degree right lexicographical ordering, while its left version and elimination orderings do not result in finite sets

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

There have been created 17068 effective critical pairs, and internally the length of intermediate elements grew to 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.

Now, the data on the Gröbner basis computation with the same input over $\mathbb{Q}$ shows a *huge* difference. Namely, only 14 critical pairs were considered, the maximal length of the intermediate elements was 6. We used no product criterion and only 9 times the chain criterion with less than 1 MB of allocated memory. The result is

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

This shows once again, how technically involved computations with free algebras over rings as coefficients are.

4. Implementation

We have created a powerful implementation called LETTERPLACE [10] in the framework of SINGULAR. Its extension to coefficient rings like $\mathbb{Z}$ addresses ideals and subbimodules of a free bimodule of a finite rank. For these, the following functions are provided.

- `twostd`: a two-sided Gröbner basis; when run with respect to an elimination ordering, it allows to eliminate variables, and thus to compute kernels of ring morphisms and preimages of ideals under such;
- `reduce` (or `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;
- `rightStd`: computation of a right Gröbner basis of a module, also effective over the factor algebra. At the moment it is only available over fields.

All of these respect the specified ordering which can be degree left/right lexicographical, weighted degree left lexicographical, left/right total elimination or an extra weight ordering extension. For modules, position-over-term and term-over-position constructions are available.

It is crucial for the understanding of the following to differentiate between polynomials and *Letterplace polynomials*, the latter being the encoding of the former in a commutative data structure as described in [8]. Let f be a polynomial and let `f` be the corresponding Letterplace polynomial, we define

$$\text{letterplace}(f) := \texttt{f} \qquad \text{and} \qquad \text{poly}(\texttt{f}) := f.$$

In the context of Singular, Letterplace polynomials are multiplied like their corresponding non-commutative polynomials, but every other operation on them, in particular the divisibility, behaves like their commutative encoding:

$$x(1)y(2)\cdot y(1)x(2) := x(1)y(2)y(3)x(4), \quad x(1)y(2) \nmid y(1)x(2)y(3), \text{while} \quad x(2)y(3) \mid y(1)x(2)y(3).$$

The implementation of the `reduce` function was the easiest, not least because in theory the non-commutative reduction algorithm can be formulated in the same form as the commutative one. But since we use Letterplace to encode non-commutative polynomials in commutative data structures, the implementation needs to be adjusted. The divisibility check in Singular [1] treats monomials as if they were from a commutative ring. Instead of modifying the divisibility check directly, we went with the following slightly unconventional approach. For every Letterplace polynomial `f` in the set of reductors (used in the normal form algorithm), all finitely many possible shifts of LM(`f`) are created, and their copies added to the set of reductors. Note that only the copied leading monomial is shifted, the rest of the polynomial is not copied but only referenced. If LM(poly(`f`)) divides a monomial m, then, because of the shifted copies, the set of reductors contains a Letterplace polynomial which divides letterplace(m). This way, when a divisor is found, we can rely on the commutative implementation again to perform the reduction step.

Implementing the `twostd` function was more involved because of significant theoretical differences to the commutative case. During the `twostd` computation, Singular stores a set of polynomials S which contains the candidate for Gröbner basis; the same S will contain the (possibly reduced) Gröbner basis in the end of the execution. Elements are removed from S whenever possible because the pairs are created with respect to this set. Also the superset $T \supseteq S$ of reductors for the reduction procedure is stored, and no element gets removed from it. We applied the same trick as in the `reduce` function for the divisibility check to the set T and left the set S as is. When a new Gröbner basis generator candidate `f` is found, critical pairs have to be created. While in the commutative case a critical pair yields precisely one S-polynomial, in the non-commutative case a critical pair can lead to more than one overlap S-polynomial (or overlap relation). Therefore, working over fields,

for every element $\mathtt{s} \in S$ we create a pair with `f`, a pair for every shift of LM(`f`) up to the length of LM(`s`) and a pair for every shift of LM(`s`) up to the length of LM(`f`). This way, S-polynomials of those pairs now correspond to possible overlap relations of LM(poly(`s`)) and LM(poly(`f`)). To filter out the pairs whose S-polynomials do not correspond to valid overlap relations, we created a new criterion that simply checks whether poly(`lcm`) is well defined, where `lcm` is the least common multiple of the leading monomials in the pair.

The generalization to bimodules (see Section 3) introduced the generators of free bimodule `ncgen(i)`. Furthermore, `syz`, `lift`, `liftstd` and `modulo` were modified along the lines, decribed above for `twostd`.

The development of the version with coefficients over rings required yet more efforts, since more pairs needed to be built and both S- and G-polynomials of the first and of the second kind have to be constructed (we shortly described this in Section 1 and Section 2 above). The existing implementation of commutative Gröbner bases over $\mathbb{Z}$ in the kernel of Singular was very helpful and inspirational for us.

5. Conclusion and Future Work

We dealt with non-commutative Gröbner bases for ideals and bimodules over $\mathbb{Z}\langle X\rangle$ and $\mathbb{Q}\langle X\rangle$. We have presented an implementation in a Singular subsystem Letterplace, which offers a rich functionality at a good speed. It is a part of the next release of Singular. Therefore the way of integration and interoperation with other systems like SAGE, OSCAR, and HomAlg [2] (to name a few) is wide open.

6. 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 work of the first 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). The second and fourth 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).

References

[1] O. Bachmann and H. Schönemann. Monomial representations for gröbner bases computations. In *Proc. of the International Symposium on Symbolic and Algebraic Computation (ISSAC'98)*, pages 309–316. ACM Press, 1998.

[2] M. Barakat, S. Gutsche, and M. Lange-Hegermann. homalg - a homological algebra meta-package for computable abelian categories. `https://homalg-project.github.io/homalg_project/homalg/`, 2019.

[3] C. Eder and T. Hofmann. Efficient Gröbner bases computation over principal ideal rings. `https://arxiv.org/abs/1906.08543`, 2019.

[4] C. Eder, G. Pfister, and A. Popescu. New strategies for standard bases over Z. `https://arxiv.org/abs/1609.04257`, 2016.

[5] C. Eder, G. Pfister, and A. Popescu. Standard bases over Euclidean domains. `https://arxiv.org/abs/1811.05736`, 2018.

[6] H. Kredel. Parametric solvable polynomial rings and applications. In V. P. Gerdt, W. Koepf, W. M. Seiler, and E. V. Vorozhtsov, editors, *Proc. CASC'15*, pages 275–291, Cham, 2015. Springer International Publishing.

[7] R. La Scala. Extended letterplace correspondence for nongraded noncommutative ideals and related algorithms. *Int. J. Algebra Comput.*, 24(8):1157–1182, 2014.

[8] R. La Scala and V. Levandovskyy. Letterplace ideals and non-commutative Gröbner bases. *Journal of Symbolic Computation*, 44(10):1374–1393, 2009.

[9] R. La Scala and V. Levandovskyy. Skew polynomial rings, Gröbner bases and the letterplace embedding of the free associative algebra. *Journal of Symbolic Computation*, 48(1):110–131, 2013.
[10] V. Levandovskyy, K. Abou Zeid, and H. Schönemann. SINGULAR:LETTERPLACE — A SINGULAR 4-1-2 subsystem for non-commutative finitely presented algebras. `http://www.singular.uni-kl.de`, 2020.
[11] V. Levandovskyy, T. Metzlaff, and K. Abou Zeid. Computation of free non-commutative Gröbner bases over $\mathbb{Z}$ with SINGULAR:LETTERPLACE. 2020. submitted.
[12] V. Levandovskyy, G. Studzinski, and B. Schnitzler. Enhanced computations of Gröbner bases in free algebras as a new application of the Letterplace paradigm. In M. Kauers, editor, *Proc. of the International Symposium on Symbolic and Algebraic Computation (ISSAC'13)*, pages 259 – 266. ACM Press, 2013.
[13] D. Lichtblau. Effective computation of strong Gröbner bases over Euclidean domains. *Illinois Journal of Mathematics*, 56:177–194, 2012.
[14] T. Mora. *Solving Polynomial Equation Systems IV: Volume 4, Buchberger Theory and Beyond.* Cambridge University Press, Cambridge, 2016.
[15] F. L. Pritchard. The ideal membership problem in non-commutative polynomial rings. *J. Symb. Comput.*, 22(1):27–48, 1996.

AROMATH, INRIA, UNIVERSITÉ CÔTE D'AZUR, 2004 ROUTE DES LUCIOLES SOPHIA ANTIPOLIS 06902 FRANCE
Email address: `tobias.metzlaff@inria.fr`

LEHRSTUHL D FÜR MATHEMATIK, RWTH AACHEN UNIVERSITY, TEMPLERGRABEN 64, 52062 AACHEN, GERMANY
Email address: `Viktor.Levandovskyy@math.rwth-aachen.de`

TU KAISERSLAUTERN, ERWIN-SCHRÖDINGER-STR. 48, 67663 KAISERSLAUTERN, GERMANY
Email address: `hannes@mathematik.uni-kl.de`

LEHRSTUHL D FÜR MATHEMATIK, RWTH AACHEN UNIVERSITY, TEMPLERGRABEN 64, 52062 AACHEN, GERMANY
Email address: `karim.abou.zeid@rwth-aachen.de`