UNIKASSEL
VERSITÄT

Submitted by
Christoph Heidemann

Submitted at
Universität Kassel

Supervisor
Prof. Dr. Georg Regensburger

Co-Supervisor
Dr. Clemens Hofstadler

December 2024

Noncommutative Gröbner bases over Euclidean domains and the F4 algorithm

Master Thesis
to obtain the academic degree of
Master of Science
in the Master's Program
Mathematics

UNIVERSITY Kassel
Heinrich-Plett-Straße 40
34132 Kassel
www.uni-kassel.de

Abstract

Gröbner bases were introduced in 1965 by Bruno Buchberger for commutative polynomial rings over fields along with the Buchberger algorithm to compute them. Later, Jean-Charles Faugère developed the F4 algorithm, which improved efficiency by translating polynomials into a matrix and computing and reducing multiple S-polynomials simultaneously using Gaussian elimination and fast linear algebra.

In this thesis, we discuss the theory of strong noncommutative Gröbner bases over Euclidean domains. Supplementary to S-polynomials, we explain the necessity of additional G-polynomials, which are used to minimize leading coefficients and also why there are infinitely many of both. Furthermore, we discuss the Diamond Lemma to algorithmically verify Gröbner bases, which leads to the Buchberger algorithm for enumerating strong Gröbner bases.

We then extend the F4 algorithm to our setting of strong Gröbner bases over Euclidean domains, highlighting the key differences from the coefficient field case. Furthermore, we present an adaptation of the SageMath package `operator_gb`, by Clemens Hofstadler, to compute strong Gröbner bases over the integers, along with documentation.

Zusammenfassung

Gröbnerbasen wurden 1965 von Bruno Buchberger für kommutative Polynomringe über Körpern zusammen mit dem Buchberger-Algorithmus zu deren Berechnung eingeführt. Später entwickelte Jean-Charles Faugère den F4-Algorithmus, der die Effizienz verbesserte, indem er Polynome in eine Matrix übersetzt und mehrere S-Polynome gleichzeitig mit Hilfe von Gauß-Elimination und schneller linearer Algebra berechnet und reduziert.

In dieser Arbeit diskutieren wir die Theorie der starken nichtkommutativen Gröbnerbasen über Euklidischen Ringen. Ergänzend zu den S-Polynomen erklären wir die Notwendigkeit zusätzlicher G-Polynome, welche verwendet werden, um Leitkoeffizienten zu minimieren und auch, warum es unendlich viele von beiden gibt. Darüber hinaus diskutieren wir das Diamond Lemma, um Gröbnerbasen algorithmisch zu verifizieren, was zum Buchberger-Algorithmus für die Aufzählung starker Gröbnerbasen führt.

Anschließend erweitern wir den F4-Algorithmus auf unsere Situation mit starken Gröbnerbasen über Euklidischen Ringen, wobei wir die wichtigsten Unterschiede zum Fall der Koeffizientenkörper hervorheben. Außerdem stellen wir eine Anpassung des SageMath-Pakets `operator_gb` von Clemens Hofstadler vor, um starke Gröbnerbasen über den ganzen Zahlen zu berechnen, zusammen mit einer Dokumentation.

Acknowledgements

I want to express my heartfelt gratitude to my supervisors, Georg Regensbuger and Clemens Hofstadler, for their endless efforts and support. Their lecture "Commutative and noncommutative polynomials" immediately ignited my interest in noncommutative Gröbner bases with their engaging manner. I am very grateful for the countless hours of their time which were invaluable in completing this thesis. Every of the many discussions left me with more knowledge and motivation than before. I also want to express my gratitude for their time spent proofreading this thesis.

Finally, I want to thank my family for their unwavering support and sympathy throughout my whole life and the opportunities they have provided me.

Christoph Heidemann
Kassel, December 2024

Eidesstattliche Erklärung

Hiermit bestätige ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken (dazu zählen auch Internetquellen) entnommen sind, wurden unter Angabe der Quelle kenntlich gemacht.

Ort, Datum	Christoph Heidemann

Contents

Chapter 1

Introduction

The topic of Gröbner bases has been studied extensively ever since their introduction by Bruno Buchberger in 1965 as part of his dissertation [Buc65]. Gröbner bases have numerous applications in the field of computer algebra, including solving systems of polynomial equations, analyzing algebraic structures, or robotics, see for example [CLO15]. The extension of Gröbner basis theory to noncommutative polynomials, see for example [Mor85], expands their utility to areas such as automating proofs and certifying operator statements using Gröbner bases, we refer to [Hof23] and references therein. Much of the research done in this field focuses on coefficient fields. However, allowing for coefficients from a Euclidean domain further broadens the applicability of Gröbner bases. In particular, in the context of operator statements it is natural to consider noncommutative polynomials with integer coefficients.

When Buchberger initially introduced Gröbner bases, he did so for commutative polynomial rings over fields. He also presented *Buchberger's algorithm*, which produces a Gröbner basis of an ideal, given a finite generating set. Jean-Charles Faugère developed the *F4 algorithm* as an improvement of Buchberger's algorithm, detailed in [Fau99]. The main idea of the algorithm is to translate polynomials into a matrix and to then reduce multiple S-polynomials simultaneously using Gaussian elimination. A generalization of the F4 algorithm to the noncommutative case over fields was first formulated in [Xiu12].

Methods for computing strong Gröbner bases over Euclidean domains have been explored in works such as [Lic12]. The main difference to the coefficient field case is that not only S-polynomials are used to cancel leading terms but also G-polynomials are used to minimize leading coefficients. A noncommutative version of Buchberger's algorithm for strong Gröbner bases over the integers has been presented, for instance, in [LMAZ23]. Another difference to the coefficient field case is that there are infinitely many S-polynomials and G-polynomials to consider.

The aim of this thesis is to extend the F4 algorithm to compute noncommutative Gröbner bases over Euclidean domains. We discuss the necessity of G-polynomials and the infinite amount of S-polynomials and G-polynomials. Furthermore, Gaussian elimination is not applicable in Euclidean domains, therefore we use the Hermite normal form instead.

In addition, this thesis presents an adaptation of a SageMath implementation of the F4 algorithm to enumerate strong Gröbner bases over the integers.

The overall structure of this thesis is divided into five chapters. The following Chapter 2 establishes the basic concepts and notations that are used throughout this thesis. It begins with the key properties of Euclidean domains since many of the results in this thesis depend on them. Since matrices over a Euclidean domain and their normal forms play a central role in the F4 algorithm, we briefly discuss the Hermite normal form. Additionally, we recall the definition of noncommutative polynomials and monomial orders.

Polynomial division is an important tool in Gröbner basis theory over fields, but is not directly applicable to noncommutative polynomials over Euclidean domains. In the context of Euclidean domains, coefficients are no longer inherently invertible but we can still perform division with remainder. In Chapter 3, we discuss a reduction relation that focuses on the remainder. To this end, we first explore general term rewriting and some abstract characteristics. Then we define the polynomial reduction and demonstrate that it satisfies certain desired characteristics. In general, our reduction does not yield a unique remainder. This leads to the definition of strong Gröbner bases, which are special sets for which this reduction is confluent.

In Chapter 4, we state the definition of a strong Gröbner basis, which serves as the main framework of this thesis. We present different characterizations and discuss S-polynomials and G-polynomials and why there are infinitely many. Furthermore, we state the Diamond Lemma, a constructive characterization to verify strong Gröbner bases. This lemma then leads to the Buchberger algorithm to enumerate strong Gröbner bases.

Finally, in Chapter 5, we discuss a procedure to efficiently compute strong Gröbner bases in the setting of coefficient rings. We adapt the F4 method to perform multiple reductions simultaneously through fast linear algebra. We then provide documentation for an adaptation of a SageMath package designed to compute (partial) strong Gröbner bases over the integers. The original package, `operator_gb`, developed by Clemens Hofstadler, supports noncommutative Gröbner basis computations over fields. Furthermore, we highlight key differences from coefficient fields to coefficients from a Euclidean domain.

Notations and conventions

For this thesis, we adopt the following conventions:

- $\mathbb{N} = \{0, 1, 2, \dots\}$ denotes the set of nonnegative integers, and $\mathbb{N}_{>0} = \{1, 2, \dots\}$ denotes the positive integers.
- By a ring, we always mean a ring with a unit element 1 that is not necessarily commutative.

Chapter 2

Preliminaries

In this chapter, we establish the basic concepts and notations that are used and built upon throughout this thesis. We assume familiarity with basic algebraic structures such as monoids, rings, and modules.

2.1 Euclidean domains

In our study of noncommutative Gröbner bases over $\mathbb{Z}$, the reduction of polynomials is an important tool for determining whether a polynomial can be generated by a combination of others. The division of coefficients is a crucial part of the reduction. Since ring elements are not inherently invertible, we introduce Euclidean domains. In that sense, we first recall the well known division with remainder over the integers.

Proposition 2.1. *Let $a, b \in \mathbb{Z}$, if $b \neq 0$, there exist integers q, r such that*

$$a = qb + r, \qquad 0 \leq r < |b|.$$

The values q and r are unique.

The values q and r are commonly referred to as *quotient* and *remainder*.

Definition 2.2. *Let $a, b \in \mathbb{Z}$. We say b* divides *a if there exists $c \in \mathbb{Z}$ such that $a = cb$ and denote it by $b \mid a$. For any $a, b \in \mathbb{Z}$ with $a \neq 0$ or $b \neq 0$, we say*

1. *$d \in \mathbb{Z}$ is a* greatest common divisor *of a and b if (i) $d \mid a$, $d \mid b$ and (ii) for any $c \in \mathbb{Z}$, $c \mid a$, $c \mid b$ implies $c \mid d$.*
2. *$m \in \mathbb{Z}$ is a* least common multiple *of a and b if (i) $a \mid m$, $b \mid m$ and (ii) for any $c \in \mathbb{Z}$, if $a \mid c$, $b \mid c$, then $m \mid c$.*

Both the greatest common divisor and least common multiple of two integers are unique up to the sign. By always choosing the positive value, we achieve uniqueness and can therefore define a function. For two integers $a, b \in \mathbb{Z}$ and $a \neq 0$ or $b \neq 0$, we denote the greatest common divisor by

$$\gcd(a, b)$$

and the least common multiple by

$$\operatorname{lcm}(a, b).$$

To compute the greatest common divisor of two integers we can use the *Euclidean algorithm*, named after Euclid of Alexandria (ca. 300 BC). See, for example, [vzGG03, Section 3.1] for a more detailed examination of the topic. Using the greatest common divisor, we can easily determine the least common multiple with the following relation. Let $a, b \in \mathbb{Z}$ with either $a \neq 0$ or $b \neq 0$, then the following equation holds:

$$\operatorname{lcm}(a, b) = \frac{|ab|}{\gcd(a, b)}.$$

The crucial component in all these operations is division with remainder. The ability to perform some form of division with remainder is the central part of an important class of rings which we now introduce.

Definition 2.3. *An integral domain R is called* Euclidean *if there exists a function*

$$|\cdot| \colon R \setminus \{0\} \to \mathbb{N}$$

with the following properties:

1. *for any $a, b \in R$, $b \neq 0$, there exist $q, r \in R$ such that*

$$a = qb + r, \qquad \text{with } r = 0 \text{ or } |r| < |b|;$$

2. *for all $a, b \in R \setminus \{0\}$, it holds $|ab| \geq |a|$.*

Clearly, the ring of integers is a Euclidean domain. The product of two nonzero integers is always nonzero meaning $\mathbb{Z}$ is an integral domain. Additionally, the absolute value function $|\cdot|$ for integers satisfies the conditions of Definition 2.3. Another common Euclidean domain is the ring of univariate polynomials over a field K, denoted by $K[x]$. The function of Definition 2.3 corresponds to the degree of polynomials.

The definition of greatest common divisor and least common multiple from Definition 2.2 naturally extends to Euclidean domains. They are again unique only up to multiplication with a

unit. For integers, we obtain uniqueness by always choosing the positive option. Similarly, we assume to have a canonical choice for every Euclidean domain. We then denote the greatest common divisor and least common multiple for two elements a, b of a Euclidean domain R by $\gcd(a, b)$ and $\operatorname{lcm}(a, b)$. The following equation yields a characterization of the greatest common divisor of two elements as a linear combination of them.

Lemma 2.4. *Let R be a Euclidean domain and $a, b \in R$ with $a \neq 0$ or $b \neq 0$. Then there exist $u, v \in R$ with*

$$\gcd(a, b) = ua + vb. \tag{1}$$

Proof. See, for example, [Coh01, Theorem 3.8]. □

Equation (1) is called *Bézout's identity* (after Étienne Bézout, 1730–1783) with u, v being referred to as *Bézout coefficients*. They can be computed with the *extended Euclidean algorithm*. See, for example, [vzGG03, Section 3.2]. Notably, the Bézout coefficients are not unique.

2.2 Hermite normal form

In this section, we recall the Hermite normal form (after Charles Hermite, 1822–1901) of an integer matrix. In particular, we discuss its existence, uniqueness, and construction. We begin with the notion of *row-equivalence*.

Definition 2.5. *Let A and B be $m \times n$ matrices with entries in $\mathbb{Z}$. We say that A and B are* row-equivalent *over $\mathbb{Z}$ if there exists an $m \times m$ matrix U which is invertible over $\mathbb{Z}$ and which satisfies $B = UA$.*

A matrix U is said to be invertible over $\mathbb{Z}$ if the entries of both U and U^{-1} are integers. If two matrices A and B are row-equivalent, they can be transformed into each other by elementary row operations, which involve

1. row switching,
2. row multiplication, by a nonzero integer, and
3. adding an integer multiple of one row to another.

The sequence of elementary row operations can be represented by an invertible integer matrix U.

Definition 2.6. *An $m \times n$ matrix H with integer coefficients is in* Hermite normal form *if the following conditions are satisfied:*

1. *H is upper triangular and any zero rows are below any other row;*
2. *the leading coefficient of a nonzero row, also called the* pivot *of a row, is always strictly to the right of the leading coefficient of the row above it, it is also positive;*
3. *the elements below pivots are zero and elements above pivots are nonnegative and strictly smaller than the pivot.*

Example 2.7. *The following matrix is in Hermite normal form:*

$$H = \begin{pmatrix} 1 & 2 & 0 & 0 \\ 0 & 3 & 4 & 0 \\ 0 & 0 & 5 & 6 \end{pmatrix}.$$

Theorem 2.8. *If A is an $m \times n$ matrix over $\mathbb{Z}$, then there exists a unique $m \times n$ matrix H over $\mathbb{Z}$, which satisfies the following two conditions:*

1. *H is row-equivalent to A over $\mathbb{Z}$.*
2. *H is in Hermite normal form.*

Proof. See [AW12, 3, §5.2] □

Definition 2.9. *The matrix H in Theorem 2.8 is referred to as the* Hermite normal form *of the matrix A and is denoted by* $\mathrm{HNF}(A)$.

The Hermite normal form of an integer matrix can be computed using a modified version of Gaussian elimination that preserves integer entries and exclusively uses elementary row operations. See [Bre11, Section 14.2] for more details. However, modern algorithms have been developed that deviate from this approach and are more efficient.

The Hermite normal form not only exists for integer matrices but also for every matrix over a Euclidean domain. A thorough exploration of the topic lies beyond the scope of this thesis so we refer the reader to [AW12, Section 5.2].

Let A be a matrix over a Euclidean domain. We extend the notion from Definition 2.9 to matrices over Euclidean domains. The Hermite normal form of A is also denoted by $\mathrm{HNF}(A)$.

2.3 Noncommutative polynomials

When we discuss noncommutative polynomials, we specifically refer to noncommutativity regarding variables, i.e., $xy \neq yx$, while the coefficients remain commutative. Noncommutative

polynomials are able to model matrix computations and can also be used to prove operator statements, see [BHR]. The structure of this section closely follows that of [HR23, Chapter 1].

Definition 2.10. *An* alphabet *is a set X and a* word *over X is a finite sequence $w = x_1 \dots x_d$ with $d \in \mathbb{N}$ and $x_i \in X$ for $i = 1, \dots, d$. For $d = 0$, we obtain the empty sequence, also called the* empty word*, which we denote by 1. The quantity d is called the* length *of w and is denoted by $|w|$. The set of all words over X is denoted by $\langle X \rangle$.*

Example 2.11. *Let $X = \{x, y, z\}$. There are infinitely many words w over this alphabet so let us consider only words of length $|w| = d \leq 2$. We have for the length*

- *$d = 0$: the empty word 1;*
- *$d = 1$: the words x, y, and z;*
- *$d = 2$: the words $xx, yy, zz, xy, xz, yx, yz, zx$, and zy.*

The number of words that have the length d is $|X|^d$.

A fundamental operation in $\langle X \rangle$ is the concatenation, defined as follows:

$$(x_1 \dots x_d, x'_1 \dots x'_{d'}) \mapsto x_1 \dots x_d x'_1 \dots x'_{d'}.$$

For instance, to concatenate the words xy and yxz, we write $xy \cdot yxz = xyyxz$, which can also be expressed as $xyyxz = xy^2xz$. It is easy to see that concatenation is associative. The empty word 1 acts as the identity. Equipped with this operation, the set $\langle X \rangle$ forms a *monoid*.

Definition 2.12. *Let X be a set of variables. The set $\langle X \rangle$ together with the binary operation of concatenation is called the* free monoid *or* word monoid *over X.*

To simplify notation, when the elements of X are explicitly listed, we write $\langle x_1, \dots, x_n \rangle$ instead of $\langle \{x_1, \dots, x_n\} \rangle$. The concatenation as a multiplication yields an intuitive understanding of divisibility within $\langle X \rangle$.

Definition 2.13. *Let X be a set and $w, w' \in \langle X \rangle$. We say that w'* divides *w, or w* is divisible by *w', if there exist $a, b \in \langle X \rangle$ such that*

$$w = aw'b.$$

In this case, we call a, b the cofactors *of the division. If w' divides w, we also equivalently say that w is a* multiple *of w'.*

Example 2.14. *Let $X = \{x, y, z\}$ and $w \in \langle X \rangle$ with $w = xzyz$. Then we have the following relations:*

- *w is divisible by $w' = zy$ because there exist $a, b \in \langle X \rangle$ such that $w = aw'b$. With the cofactors $a = x$ and $b = z$, we have $w = xzyz = aw'b$.*
- *w is also divisible by x, y, and z, as well as xz, yz, xzy, zyz, and, of course, by itself.*
- *We have multiple proper prefixes and suffixes. For example, xzy is a proper prefix and yz is one proper suffix. Furthermore, w is both a prefix and a suffix of itself, although not a proper one.*

Note that w is divisible by $w' = z$ in two ways, namely $w = xzyz = xw'yz = xzyw'$.

The words of the free monoid $\langle X \rangle$ are the building blocks of noncommutative polynomials and serve as monomials. To formalize this, we construct a ring with words as the elements over another ring, which will be the coefficients.

For an arbitrary monoid M over a commutative ring with 1, we define two fundamental operations: addition and multiplication. In the following, R denotes a commutative ring with (with 1).

Definition 2.15. *Let M be a monoid. Consider the set*

$$RM = \left\{ \sum_{m \in M} c_m m \mid c_m \in R \text{ such that } c_m = 0 \text{ for almost all } m \right\}.$$

We define addition on RM by

$$\sum_{m \in M} c_m m + \sum_{m \in M} c'_m m = \sum_{m \in M} (c_m + c'_m) m,$$

and multiplication by

$$\sum_{k \in M} c_k k \cdot \sum_{l \in M} c'_l l = \sum_{m \in M} \Big(\sum_{kl = m} c_k c'_l \Big) m.$$

The set RM equipped with the defined addition and multiplication forms a ring, which is called the *monoid ring* of M over R. This yields the ring of noncommutative polynomials as the monoid ring of the free monoid $\langle X \rangle$ over R.

Definition 2.16. *We refer to the monoid ring of $\langle X \rangle$ over R, denoted by $R\langle X \rangle$, as the* ring of noncommutative polynomials *in the indeterminates X with coefficients in R. In other contexts, $R\langle X \rangle$ is also called the* free (associative) algebra *or the* free monoid ring *generated by X over R. Elements in $R\langle X \rangle$ are called* (noncommutative) polynomials.

Remark 2.17. *If X consists of only one indeterminate x, the elements of $R\langle X\rangle$ behave in a commutative manner. We can write $\langle X\rangle = \{x^n \mid n \in \mathbb{N}\}$ and concatenation works just like multiplication in $[x]$ by adding the exponents. $R\langle x\rangle$ is equal to the univariate polynomial ring $R[x]$. For any X with more than one element we lose commutativity.*

Example 2.18. *Let $f_1, f_2 \in \mathbb{Z}\langle x, y, z\rangle$ with $f_1 = xy + z$ and $f_2 = yx - 2z$. Then we can do the following computations:*

$$f_1 + f_2 = xy + z + yx - 2z = xy + yx - z;$$

$$f_1 \cdot f_2 = (xy + z)(yx - 2z) = xyyx - 2xyz + zyx - 2zz;$$

$$f_2 \cdot f_1 = (yx - 2z)(xy + z) = yxxy + yxz - 2zxy - 2zz.$$

The products $f_1 \cdot f_2$ and $f_2 \cdot f_1$ only have one term in common.

Definition 2.19. *Let $f = \sum_{w\in\langle X\rangle} c_w w \in R\langle X\rangle$ be a polynomial. The* coefficient $c_w \in R$ *of the* monomial w *in f is denoted by*

$$\operatorname{coeff}(f, w) = c_w.$$

Furthermore, the support *of f is the set*

$$\operatorname{supp}(f) = \{w \in \langle X\rangle \mid \operatorname{coeff}(f, w) \neq 0\}.$$

Example 2.20. *Let $f = 2xyyz + 5zyx + 3zx + y \in \mathbb{Z}\langle x, y, z\rangle$. The support of f is given by*

$$\operatorname{supp}(f) = \{xyyz, zyx, zx, y\} \subset \langle x, y, z\rangle,$$

and one coefficient is $\operatorname{coeff}(f, zyx) = 5$.

To proceed towards the theory of Gröbner bases, we now recall the concept of *ideals*. They provide the fundamental algebraic structure upon which Gröbner bases are constructed. Throughout this section, let S denote a ring, not necessarily commutative, and let R denote a commutative ring (with 1). We distinguish between left and right ideals.

Definition 2.21. *A subset $I \subseteq S$ is a* two-sided ideal *of S if*

1. *I is nonempty;*
2. *I is closed under addition, that is, $f + g \in I$, for all $f, g \in I$;*
3. *I is closed under left and right multiplication by arbitrary ring elements, that is, $sf \in I$, $fs \in I$ for all $s \in S$ and $f \in I$.*

If I is a two-sided ideal of S, we write $I \trianglelefteq S$.

The sets $I = \{0\}$ and $I = S$ are both examples of ideals of S. They are referred to as the *trivial ideals*. Since the only ideals appearing in this thesis are two-sided ideals, we simply refer to them as ideals.

Definition 2.22. *Let $F \subseteq S$, we denote by (F) the ideal generated by F, that is,*

$$(F) = \left\{ \sum_{i=1}^{d} r_i f_i s_i \mid d \in \mathbb{N}, r_i, s_i \in S, f_i \in F \right\}. \tag{1}$$

Example 2.23. *Let $F \subseteq \mathbb{Z}\langle x, y, z\rangle$ with $F = \{x + y, z\}$. We can construct elements of (F) by adding elements of F and by multiplying them with elements of $\mathbb{Z}\langle x, y, z\rangle$. Let $f_1 = x + y$ and $f_2 = z$, then*

1. $f_1 + f_2 \;=\; x + y + z \;\in (F)$;
2. $yf_1 - xf_2z \;=\; yx + y^2 - xz^2 \;\in (F)$;
3. $(3y + z)(x + y - z) + 4f_1 \;=\; 3yx + 3y^2 - 3yz + zx + zy - z^2 + 4x + 4y \;\in (F)$.

Lemma 2.24. *Let $F \subseteq S$, then (F) is the smallest ideal containing the set F.*

Proof. We prove this by showing that (F) is a subset of all ideals containing the set F. Let $I \subseteq S$ be an ideal such that $F \subseteq I$, and let $f \in (F)$. By (1), we can write f as the sum $\sum_{i=1}^{d} r_i f_i s_i$ for some $d \in \mathbb{N}$, $r_i, s_i \in S$, and $f_i \in F$ for all i. Since all f_i's are in I, and I is closed under left and right multiplication by arbitrary ring elements, and I is closed under addition with ideal elements, f is also an element of I. Therefore, $(F) \subseteq I$. □

If $(F) = I \trianglelefteq S$, then the set F is called a *generating set* for I. Furthermore, an ideal is said to be *finitely generated* if it has a finite generating set. This is not guaranteed for every ideal $I \trianglelefteq \mathbb{Z}\langle x, y\rangle$. For example, the ideal $I = (yx^n y \mid n \in \mathbb{N}) \trianglelefteq \mathbb{Z}\langle x, y\rangle$ has no finite generating set.

The existence of finite generating sets is related to the ascending chain condition, which states that every ascending chain of ideals $I_0 \subseteq I_1 \subseteq I_2 \subseteq \dots$ in S eventually stabilizes, i.e., there exists a $d \in \mathbb{N}$, such that:

$$\dots I_{d-1} \subseteq I_d = I_{d+1} = I_{d+2} \dots.$$

In $\mathbb{Z}\langle x, y\rangle$, however, we have the following infinitely ascending chain of ideals:

$$(yxy) \subset (yxy, yx^2y) \subset (yxy, yx^2y, yx^3y) \subset (yxy, yx^2y, yx^3y, yx^4y) \subset \dots.$$

If a ring satisfies the ascending chain condition it is called *Noetherian*. The ring of integers, for example, is Noetherian and, by Hilbert's basis theorem, $\mathbb{Z}[x_1, \ldots, x_n]$ (commutative polynomial ring) is as well for $n \in \mathbb{N}$. However, we have just seen that Hilbert's basis theorem does not work for $\mathbb{Z}\langle x, y\rangle$. In fact, for any Noetherian ring R and free monoid $\langle X\rangle$, an ideal of the free algebra $R\langle X\rangle$ does not have to be finitely generated, provided that $|X| > 1$.

We no longer consider infinitely generated ideals from now on. Thus, whenever we refer to an ideal, we assume it has a finite generating set. This property is important when asking whether an element is contained in an ideal, or alternatively, deciding the *ideal membership problem*.

Problem 2.25 (Ideal membership)**.**
Input: $r, r_1, \ldots, r_s \in S$
Output: *True if* $r \in (r_1, \ldots, r_s)$ *and* *False otherwise*

The ideal membership problem is a central problem in ideal theory. It arises in both univariate and multivariate, as well as commutative and noncommutative, settings. Its decidability varies between the commutative and noncommutative case. Recall that a problem is decidable if there is an algorithm that can determine whether a statement is true or false, in a finite amount of time. While the ideal membership problem is decidable in the commutative case, it becomes undecidable for $R\langle X\rangle$ if $|X| > 1$, see [Xiu12, Rem. 2.2.12]. However, we introduce a procedure that can at least verify that an element is contained in an ideal, but not disprove it.

One way to verify the ideal membership of an element is to give a representation similar to those of elements in the ideal generated by a finite set, see (1). This type of representation is called a *cofactor representation*.

Definition 2.26. *Let $F \subseteq R\langle X\rangle$. A representation of an element $f \in (F)$ of the form*

$$f = \sum_{i=1}^{d} a_i f_i b_i, \tag{2}$$

with $d \in \mathbb{N}$, $a_i, b_i \in R\langle X\rangle$ and $f_i \in F$ is called a cofactor representation *of f with respect to F. We refer to the elements a_i and b_i as the* cofactors *of f with respect to the representation* (2)*.*

Example 2.27. *Let $(f_1, f_2) \trianglelefteq \mathbb{Z}\langle x, y, z\rangle$ with $f_1 = 2yx + z$ and $f_2 = xy - 3z$. Also, let $f = 5xyxy + x^2y^2 + 3xyz$. Then f has the cofactor representation*

$$f = 3xf_1y + xf_2y - xyf_2,$$

because

$$3xf_1y + xf_2y - xyf_2 = 6xyxy + 3xzy + x^2y^2 - 3xzy - xyxy + 3xyz = f.$$

Note that a cofactor representation is not necessarily unique. Consider $(f_1, f_2) \trianglelefteq \mathbb{Z}\langle x, y\rangle$ with $f_1 = x$ and $f_2 = y$. Then $f = xy + yx$ has multiple cofactor representations. For example,

$$f = f_1 y + f_2 x \quad \text{or} \quad f = x f_2 + f_2 x.$$

The task of providing a cofactor representation is nontrivial. For the purpose of ideal membership verification, it suffices to know that there exists a cofactor representation, we do not need to state it explicitly. In the commutative case, the existence of a cofactor representation can be verified via polynomial division. If the remainder is zero, the polynomial has a cofactor representation. Our goal is to develop a similar procedure for the free algebra $R\langle X\rangle$. For that purpose we require a criterion to order the monomials of a polynomial.

2.4 Monomial orders

In this thesis, we confine the examination of monomial orders to the essential results. We follow the chapter "Monomial orders" of [HR23] and [Hof23], and refer to them for a deeper exploration of the topic.

A natural way to order monomials of $\langle X\rangle$ is to view them as words once more, where the variables have an order just like in the English alphabet. Imagine an infinite dictionary where all monomials in $\langle X\rangle$ are listed as words in a lexicographic manner. To compare two monomials we simply check which one is listed in the dictionary before the other and denote that as the lesser. This monomial order is called the *lexicographic order*.

Definition 2.28. *Let $X = \{x_1, x_2, \ldots, x_n\}$. Order the elements of X as*

$$x_1 \prec_{\text{lex}} x_2 \prec_{\text{lex}} \cdots \prec_{\text{lex}} x_n.$$

Then $w \preceq_{\text{lex}} w'$ for $w, w' \in \langle X\rangle$ if one of the following conditions holds:

1. *there exist $l, r, r' \in \langle X\rangle$ and $x_i, x_j \in X$ such that $w = l x_i r$, $w' = l x_j r'$ and $x_i \prec_{\text{lex}} x_j$;*
2. *there exists $r \in \langle X\rangle$ such that $w' = wr$.*

Example 2.29. *Let $X = \{x, y, z\}$ with the ordering of indeterminates $x \prec_{\text{lex}} y \prec_{\text{lex}} z$. Then the*

following comparisons are true for the lexicographic order:

$$\begin{aligned} xy &\prec_{\text{lex}} yx, \\ x^3y &\prec_{\text{lex}} y, \\ y &\prec_{\text{lex}} yx, \\ yxz &\prec_{\text{lex}} yz, \\ xz &\prec_{\text{lex}} z, \\ xxz &\prec_{\text{lex}} xz. \end{aligned}$$

Remark 2.30. *While the lexicographic order can be used for polynomial division of elements in $R[X]$ (commutative polynomial ring), it lacks certain attributes which are required to do polynomial reduction in $R\langle X\rangle$ properly. For one, it is not compatible with multiplication. We saw in Example 2.29 that $y \prec_{\text{lex}} yx$, but multiplying both sides with z from the right flips the relation because $yz \succ_{\text{lex}} yxz$.*

Another disadvantage of the lexicographic order is that it has infinite strictly decreasing chains. In Example 2.29 we see the start of such a sequence, which would follow the infinite pattern

$$z \succ_{\text{lex}} xz \succ_{\text{lex}} xxz \succ_{\text{lex}} xxxz \succ_{\text{lex}} \dots.$$

Nevertheless, the lexicographic order appears in many monomial orders in some form. We encounter it as part of the *degree lexicographic order*. Some basic properties which the lexicographic order also has, are captured in the term *total order*.

Definition 2.31. *A* total order $\preceq$ *on a set A is a binary relation on A that satisfies the following properties for all $a, b, c \in A$:*

1. *if $a \preceq b$ and $b \preceq a$, then $a = b$;* (antisymmetry)
2. *if $a \preceq b$ and $b \preceq c$, then $a \preceq c$;* (transitivity)
3. *$a \preceq b$ or $b \preceq a$;* (connexity)

We discussed two essential properties of orders in Remark 2.30. Together with the definition of a total order they make the notion of a *monomial order*.

Definition 2.32. *A total order $\preceq$ on $\langle X\rangle$ is called a* monomial order *or an* admissible order *on $\langle X\rangle$ if it satisfies the following two conditions:*

1. *$w \preceq w'$ implies $awb \preceq aw'b$ for all $a, b, w, w' \in \langle X\rangle$;* (compatibility with multiplication)

2. every nonempty subset of $\langle X \rangle$ has a least element; (well-order)

We can characterize well-orders by the non-existence of infinite strictly decreasing sequences of elements with the following lemma.

Lemma 2.33. *Let $\preceq$ be a total order on a set A. Every nonempty subset of A has a least element if and only if every decreasing sequence of elements in A eventually stabilises, that is, if*

$$a_0 \succeq a_1 \succeq \cdots$$

is an infinite decreasing sequence of elements $a_0, a_1, \ldots \in A$, then there exists $n \in \mathbb{N}$ such that $a_n = a_{n+k}$ for all $k \in \mathbb{N}$.

We present the proof from [HR23, Lemma 2.4].

Proof. For the first implication, assume that every nonempty subset of A has a least element and let $a_0, a_1, \ldots \in A$ be an infinite decreasing sequence. By assumption, the set $B = \{a_i \mid i \in \mathbb{N}\}$ has a least element, say a_n for some $n \in \mathbb{N}$. By definition of a least element, $a_n \preceq a_i$ for all $i \in \mathbb{N}$, and so, in particular, $a_n \preceq a_{n+k}$ for all $k \in \mathbb{N}$. Since $a_0 \succeq a_1 \succeq \cdots$ is a decreasing sequence, we additionally have $a_n \succeq a_{n+k}$ for all $k \in \mathbb{N}$. Now the antisymmetry of $\preceq$ implies that $a_n = a_{n+k}$ for all $k \in \mathbb{N}$ and so the sequence stabilizes.

For the second implication, assume that every decreasing sequence stabilizes and assume, for contradiction, that there exists a nonempty subset of A without a least element. Let $B \subseteq A$ be such a set. Then an infinite strictly deceasing sequence can be constructed as follows: since B is nonempty, choose $a_0 \in B$ arbitrary. Now, for $n \in \mathbb{N}$, choose $a_{n+1} \in B$ such that $a_n \succ a_{n+1}$. Note that such a choice is possible because no a_n is a least element of B. This yields the infinite strictly decreasing sequence $a_0 \succ a_1 \succ \cdots$, contradicting the assumption. □

Definition 2.34. *Let $X = \{x_1, x_2, \ldots, x_n\}$. The* degree lexicographic order *is defined as follows: $w \preceq_{\text{deglex}} w'$ for $w, w' \in \langle X \rangle$ if one of the following conditions holds:*

1. *$|w| < |w'|$;*
2. *$|w| = |w'|$ and $w \preceq_{\text{lex}} w'$;*

Proposition 2.35. *The degree lexicographic order $\preceq_{\text{deglex}}$ is a monomial order.*

Proof. See [Hof23, Theorem 2.4.13] □

Example 2.36. *Let $\langle x, y, z\rangle$ with the ordering of indeterminates $x \prec_{\text{lex}} y \prec_{\text{lex}} z$. Then the following relations are true for the degree lexicographic order:*

$$\begin{aligned} y &\prec_{\text{deglex}} x^2, \\ x^2 &\prec_{\text{deglex}} z^2, \\ yz^2x &\prec_{\text{deglex}} xyxzy, \\ xy &\prec_{\text{deglex}} yx, \\ y^2z^3 &\prec_{\text{deglex}} x^6. \end{aligned}$$

Definition 2.37. *Let $f \in R\langle X\rangle \setminus \{0\}$. The* leading monomial $\operatorname{lm}(f)$ *of f is the $\preceq$-maximal element in* $\operatorname{supp}(f)$*, and the* leading coefficient $\operatorname{lc}(f)$ *of f is the coefficient of* $\operatorname{lm}(f)$*. The* leading term $\operatorname{lt}(f)$ *of f is* $\operatorname{lt}(f) = \operatorname{lc}(f) \cdot \operatorname{lm}(f)$*. Furthermore, we define* $\operatorname{lc}(0) = \operatorname{lt}(0) = 0$*. A polynomial with leading coefficient* 1 *is called* monic*.*

For the sake of simplicity we extend the monomial order to $\langle X\rangle \cup \{0\}$. We set $\operatorname{lm}(0) = \operatorname{lt}(0) = 0$ and $0 \prec w$ for all $w \in \langle X\rangle$. The following properties follow directly from Definition 2.37.

Lemma 2.38. *Let $f, g \in R\langle X\rangle$.*

1. $\operatorname{lm}(f+g) \preceq \max_{\preceq}\{\operatorname{lm}(f), \operatorname{lm}(g)\}$ *with equality if and only if* $\operatorname{lt}(f) + \operatorname{lt}(g) \neq 0$*;*
2. $\operatorname{lm}(fg) = \operatorname{lm}(f)\operatorname{lm}(g)$*,* $\operatorname{lc}(fg) = \operatorname{lc}(f)\operatorname{lc}(g)$*, and thus,* $\operatorname{lt}(fg) = \operatorname{lt}(f)\operatorname{lt}(g)$*;*

Note that the second result only holds if R is an integral domain. To illustrate this, let $f, g \in \mathbb{Z}_4\langle x\rangle$ with $f = 2x^2 + x$ and $g = 2x$. Then

$$\operatorname{lm}(fg) = \operatorname{lm}((2x^2 + x)2x) = \operatorname{lm}(4x^3 + 2x^2) = \operatorname{lm}(2x^2) = x^2,$$

which is not equal to $\operatorname{lm}(f)\operatorname{lm}(g) = x^2 \cdot x = x^3$. Furthermore, $\operatorname{lc}(fg) = 2 \neq 0 = \operatorname{lc}(f)\operatorname{lc}(g)$.

Definition 2.39. *The* tail *of $f \in R\langle X\rangle$ is defined as*

$$\operatorname{tail}(f) = f - \operatorname{lt}(f).$$

Example 2.40. *Let $f, g \in \mathbb{Z}\langle x, y, z\rangle$, equipped with the order $\preceq_{\text{deglex}}$ such that $x \prec_{\text{lex}} y \prec_{\text{lex}} z$. Let*

$$f = 3xyz^2 + 2yx + z, \qquad g = 2z^3 + 5yxy + x.$$

Then we have the leading monomials $\operatorname{lm}(f) = xyz^2$ and $\operatorname{lm}(g) = z^3$ with the leading coefficients $\operatorname{lc}(f) = 3$ and $\operatorname{lc}(g) = 2$. We therefore have the leading terms $\operatorname{lt}(f) = 3xyz^2$ and $\operatorname{lt}(g) = 2z^3$. The following also hold:

1. $\mathrm{lm}(f+g) = xyz^2 + z^3 \preceq xyz^2 \max_{\preceq}\{\mathrm{lm}(f), \mathrm{lm}(g)\}$;
2. $\mathrm{lm}(fg) = \mathrm{lm}(f)\,\mathrm{lm}(g) = xyz^2z^3$;
3. $\mathrm{tail}(f) = f - \mathrm{lt}(f) = 2yx + z$;
4. $\mathrm{tail}(g) = 5yxy + x$.

We end this section with an understanding of monomial orders and their properties. In particular, we established the degree lexicographic order, which we use in the rest of this thesis to order monomials.

Chapter 3

Reduction relation

Polynomial division in commutative polynomial rings is a vital tool for computing Gröbner bases and deciding the ideal membership problem, see for example [CLO15]. However, it is not directly applicable to noncommutative polynomials over Euclidean domains because the division of two coefficients not necessarily yields an element of the Euclidean domain. We can still use the division with remainder. We discuss a reduction relation that focuses on the remainder and begin with a brief exploration of abstract rewriting.

3.1 Abstract rewriting

An abstract rewriting system is a formal framework to study the transformation of objects under a set of rules. We start with its simplest form, a set with a binary relation.

The structure of this section follows [HR23, Section 3] and [BN98, Chapter 2].

Definition 3.1. *Let A be a set and $\to \subseteq A \times A$ be a binary relation on A. The pair $(A, \to)$ is called an* abstract reduction system*, and $\to$ is a* reduction relation *or simply a* reduction*. We write $x \to y$ for $(x, y) \in \to$.*

Reducing one element by another is useful. However, what we really are interested in is reducing one element by another, and then another, and so on, until we can no longer reduce it. For this purpose we introduce the following notions

Definition 3.2. *We define the following notions:*

$\xrightarrow{0}$	$\coloneqq$	$\{(x,x) \mid x \in A\}$	identity
$\xrightarrow{n+1}$	$\coloneqq$	$\xrightarrow{n} \circ \to$	$(n+1)$-fold composition, $n \in \mathbb{N}$
$\xrightarrow{*}$	$\coloneqq$	$\bigcup_{n\in\mathbb{N}} \xrightarrow{n}$	reflexive transitive closure
$\leftarrow$	$\coloneqq$	$\{(y,x) \mid x \to y\}$	inverse
$\leftrightarrow$	$\coloneqq$	$\to \cup \leftarrow$	symmetric closure
$\xleftrightarrow{*}$	$\coloneqq$	$(\leftrightarrow)^*$	reflexive transitive symmetric closure

Remark 3.3. *Notions such as $\xleftrightarrow{*}$ or $\leftarrow$ should only be used for arrow-like symbols. Note that a common notation for the inverse of an arbitrary relation $R \subseteq A \times A$ is R^{-1} rather than $\leftarrow$.*

While all notions in Definition 3.2 are non-trivial, the most significant is $\xrightarrow{*}$, as it features prominently in the remainder of this chapter. We fix an abstract reduction system $(A, \to)$.

Definition 3.4. *Let $x, y, z \in A$.*

1. *x is* reducible *if there exists y such that $x \to y$;*
2. *x is* irreducible *or* in normal form *if it is not reducible. If x has a unique normal form, the latter is denoted by $x{\downarrow}$;*
3. *y is* a normal form *of x if $x \xrightarrow{*} y$ and y is in normal form;*
4. *x and y are* joinable*, denoted by $x \downarrow y$, if there exists z such that $x \xrightarrow{*} z \xleftarrow{*} y$;*

To get familiar with the terminology, we look at Example 2.1.2 from [BN98] and extend it slightly.

Example 3.5. *1. Let $A = \mathbb{N}_{>0} \setminus \{1\}$ and $\to = \{(m,n) \mid m > n \ \wedge \ n \text{ divides } m\}$. Then*

   (a) m is in normal form iff m is prime.
   (b) p is a normal form of m iff p is a prime factor of m.
   (c) $m \downarrow n$ iff m and n are not relatively prime.
   (d) $\xrightarrow{+} = \to$ because $>$ and "divides" are already transitive.
   (e) $\xleftrightarrow{*} = A \times A$.
   (f) m has a unique normal form $m{\downarrow}$ iff m is a prime power.

2. Let $A = \langle a, b \rangle$ (the set of words over the alphabet $\{a, b\}$) and $\to = \{(ubav, uabv) \mid u, v \in A\}$. Then

(a) w is in normal form iff w is sorted, i.e., all a's are before all b's.

(b) Every w has a unique normal form $w{\downarrow}$, the result of sorting w.

(c) $w_1 \downarrow w_2$ iff $w_1 \overset{*}{\leftrightarrow} w_2$ iff w_1 and w_2 contain the same number of a's and b's.

Our goal is to develop a reduction system capable of solving the ideal membership problem for noncommutative ideals with coefficients in a ring. With respect to that reduction system, two elements should be equivalent if they both belong to the ideal. This task falls within the class of a "word problem".

Problem 3.6 (Word problem)**.**
Input: $(A, \to)$ *abstract reduction system,* $x, y \in A$
Output: True *if* $x \overset{*}{\leftrightarrow} y$ *and* False *otherwise*

One method to check whether $x \overset{*}{\leftrightarrow} y$ is to compute the normal forms of x and y, and return True if they are equal. The reliability of this method depends on two properties, namely the existence of normal forms and their uniqueness. While both reductions in Example 3.5 ensure the existence of normal forms, only the second one guarantees their uniqueness. For instance, under the first reduction system, both 2 and 3 are normal forms of 6 since they divide 6 and are prime. To classify reduction systems by these properties, we require additional terminology.

Definition 3.7. *A reduction $\to$ is*

1. Church-Rosser *if $x \overset{*}{\leftrightarrow} y$ implies $x \downarrow y$;*
2. confluent *if $y_1 \overset{*}{\leftarrow} x \overset{*}{\to} y_2$ implies $y_1 \downarrow y_2$;*
3. normalizing *if every element has a normal form;*
4. terminating *if there is no infinite descending chain $a_0 \to a_1 \to \dots$;*

Upon revisiting Example 3.5, we observe that while both reductions are terminating and normalizing, only the second is both confluent and Church-Rosser. In fact, it is not a coincidence that neither reduction in Example 3.5 is exclusively confluent without being Church-Rosser, or vice versa.

Theorem 3.8. *A reduction $\to$ is Church-Rosser if and only if it is confluent.*

Proof. See [BN98, Theorem 2.15] □

Corollary 3.9. *If $\to$ is confluent and $x \overset{*}{\leftrightarrow} y$, then*

1. *$x \overset{*}{\to} y$ if y is in normal form;*

2. $x = y$ if both x and y are in normal form;

With this result, we know that in a confluent reduction system two elements are equivalent if and only if they have the same (unique) normal form. However, this equivalence is decidable only if both elements actually have a normal form. Reductions that are normalizing guarantee, per Definition 3.2, the existence of a normal form for every element.

Theorem 3.10. *If $\to$ is normalizing and confluent, then every element has a unique normal form. Furthermore, $x \overset{*}{\leftrightarrow} y$ if and only if $x{\downarrow} = y{\downarrow}$.*

Finally, we can algorithmically decide whether two elements are equivalent or not. If the reduction system is both confluent and normalizing, we can compute their unique normal forms and simply check for equality. Since a terminating reduction is inherently normalizing, we are going to rather focus on that property, since it is often easier to establish. Confluence remains the second required property.

3.2 Polynomial reduction

In this section, we define a polynomial reduction and show that it possesses the desired property of termination. Let R be a Euclidean domain for the rest of the chapter.

Definition 3.11. *Let $a, b \in \langle X \rangle$, $f, g \in R\langle X \rangle \setminus \{0\}$ with $|\mathrm{lc}(g)| \leq |\mathrm{coeff}(f, agb)|$, and $G \subseteq R\langle X \rangle$. Furthermore, let $q, r \in R$ with $q \neq 0$, $\mathrm{coeff}(f, agb) = q\,\mathrm{lc}(g) + r$ and $0 \leq |r| < |\mathrm{lc}(g)|$. We define the following reduction relations on $R\langle X \rangle$:*

$$\begin{aligned} f \to_{a,g,b} f' &\iff \mathrm{lm}(agb) \in \mathrm{supp}(f) \text{ and } f' = f - qagb \\ \to_g &= \bigcup_{a,b \in \langle X \rangle} \to_{a,g,b} \\ \to_G &= \bigcup_{g \in G \setminus \{0\}} \to_g \end{aligned}$$

These relations are called the polynomial reduction relation *with respect to a, g, b, with respect to g, and with respect to G respectively.*

Example 3.12. *Let $f = 6zxy^3 + 8zxyz$, and $G \subseteq \mathbb{Z}\langle x, y, z \rangle$ with $G = \{2xy + z, 3z^2y^2 + 4z^3\}$, equipped with the monomial order $\preceq_{\mathrm{deglex}}$ such that $x \prec_{\mathrm{lex}} y \prec_{\mathrm{lex}} z$. Then*

1. for $g = 2xy + z$, we have $\mathrm{lm}(agb) \in \mathrm{supp}(f)$, with $a = z$ and $b = y^2$, and additionally $q = 3$.

This results in the reduction

$$\begin{aligned} 6zxy^3 + 8zxyz \rightarrow_{z,g,y^2} & 6zxy^3 + 8zxyz - 3z \cdot g \cdot y^2 \\ = & 6zxy^3 + 8zxyz - (6zxy^3 + 3z^2y^2) \\ = & 8zxyz - 3z^2y^2. \end{aligned}$$

2. *for* $g = 2xy + z$*, we also have* $\mathrm{lm}(agb) \in \mathrm{supp}(f)$*, with* $a = z$ *and* $b = z$*, and* $q = 4$*. This gives the reduction*

$$\begin{aligned} 6zxy^3 + 8zxyz \rightarrow_{z,g,z} & 6zxy^3 + 8zxyz - 4z \cdot g \cdot z \\ = & 6zxy^3 + 8zxyz - (8zxyz + 4z^3) \\ = & 6zxy^3 - 4z^3. \end{aligned}$$

3. *again for* $g = 2xy + z$*, we can compute* $f \rightarrow_g f'$ *(reduction with respect to* g*, not* a, g, b*) by reducing the result of either of the above reductions again by* g*. Considering the first result* $(6zxy^3 + 8zxyz \rightarrow_{z,g,y^2} 8zxyz - 3z^2y^2)$*, we have* $\mathrm{lm}(agb) \in \mathrm{supp}(8zxyz - 3z^2y^2)$*, with* $a = z$*,* $b = z$*, and* $q = 4$*. This yields the reduction*

$$\begin{aligned} 8zxyz - 3z^2y^2 \rightarrow_{z,g,z} & 8zxyz - 3z^2y^2 - 4z \cdot g \cdot z \\ = & 8zxyz - 3z^2y^2 - (8zxyz + 4z^3) \\ = & 8zxyz - 3z^2y^2 - 8zxyz - 4z^3 \\ = & -3z^2y^2 - 4z^3. \end{aligned}$$

This result can be reduced no further by g*. Therefore, we have computed one possibility for* $f \rightarrow_g f'$*, namely* $6zxy^3 + 8zxyz \rightarrow_g -3z^2y^2 - 4z^3$*. This is only one possibility because we have already seen two ways for one reduction step of* f *by* g*.*

4. *one possible reduction of* f *with respect to the set* G *can be computed by continuing the reduction in 3.. While* $-3z^2y^2 - 4z^3$ *is no longer reducible with respect to* $2xy + z$*, we can continue reducing it with other elements of* G*. Let* $g' = 3z^2y^2 + 4z^3$*, then* $\mathrm{lm}(ag'b) \in \mathrm{supp}(f)$ *with* $a = b = 1$ *and* $q = -1$*. The reduction is as follows:*

$$\begin{aligned} -3z^2y^2 - 4z^3 \rightarrow_{1,g',1} & -3z^2y^2 - 4z^3 - (-1)a \cdot g' \cdot b \\ = & -3z^2y^2 - 4z^3 + 3z^2y^2 + 4z^3 = 0. \end{aligned}$$

We have found one reduction of f *with respect to* G*, so* $f \rightarrow_G 0$*. This reduction is*

$$f \to_{z,g,y^2} f' \to_{z,g,z} f'' \to_{1,g',1} 0.$$

An important characteristic of the reduction relations from Definition 3.11 is, that in no possible reduction the leading monomial can *increase*.

Lemma 3.13. *If $f \to_G f'$, then $\mathrm{lm}(f) \succeq \mathrm{lm}(f')$. Moreover, if $f \to_{a,g,b} f'$, then also $\mathrm{lm}(f) \succeq \mathrm{lm}(agb)$.*

It follows that for all $g \in R\langle X\rangle \setminus \{0\}$, $\mathrm{tail}(g)$ is irreducible with respect to $\to_g$.

Remark 3.14. *During a reduction $f \to_{a,g,b}$ from Definition 3.11 at lest one term in the support of f is either canceled or has its coefficient decreased. If that term is the leading term it is referred to as a* top reduction *and otherwise a* tail reduction.

It is straightforward to see that the reduction $\to_{a,g,b}$ from Definition 3.11 is terminating. For fixed $a, b \in \langle X\rangle$, $\mathrm{lm}(agb)$ can only appear once in $\mathrm{supp}(f)$. Furthermore, $q \in R$ is chosen such that the coefficient of $\mathrm{lm}(agb)$ in f is canceled or minimized during the reduction $f \to_{a,g,b} f'$. Therefore every reduction $\to_{a,g,b}$ results in a normal form. This normal form is unique. However, for the reductions $\to_g$ and $\to_G$, we can construct a simple example to show that these normal forms are not unique: let

$$f = y^3 \text{ and } G = \{y^2 - x\},$$

then

$$f \to_G yx, \text{ and also } f \to_G xy$$

are both normal forms. Since G has only one element, $\to_g$ has by extension also no unique normal form.

We now extend the notion of a monomial order from Definition 2.32 to a term order.

Definition 3.15. *Let $\preceq$ be a monomial order on $\langle X\rangle$ and let R be a Euclidean. Furthermore, let $c, c' \in R$ and $w, w' \in \langle X\rangle$. We define the* term order *$cw \prec c'w'$ if one of the following conditions holds:*

1. *$w \prec w'$;*
2. *$w = w'$ and $|c| < |c'|$;*

Lemma 3.16. *The term order from Definition 3.15 is a well order.*

Proof. Suppose, for contradiction, that there exists an infinite strictly decreasing sequence

$$c_0 w_0 \succ c_1 w_1 \succ \dots$$

of elements $c_0w_0, c_1w_1, \ldots \in R\langle X\rangle$ where $c_0, c_1, \ldots \in R$ and $w_0, w_1, \ldots \in \langle X\rangle$ such that w_0 is minimal with respect to $\preceq$ among all $w \in \langle X\rangle$ starting an finite sequence, and $|c_0|$ is minimal among all of those. Choosing such a term c_0w_0 is possible because $\preceq$ and the natural order on $\mathbb{N}$ are both well orders. Since the sequence is strictly decreasing, in every step $j \in \mathbb{N}$ we either have $w_j \succ w_{j+1}$ or $|c_j| > |c_{j+1}|$. Consider only the w's. Since $\preceq$ is a well order and therefore no infinite strictly decreasing sequences of elements can exist, there must exist an $i \in \mathbb{N}$ such that $w_i = w_{i+1} = \ldots$. For all steps after i, the $|c|$'s strictly decrease in every step. However, the natural order on $\mathbb{N}$ is also a well order. Therefore there must exist a $k \in \mathbb{N}$ such that $c_kw_k = c_{k+1}w_{k+1} = \ldots$.. This is a contradiction to the assumption. □

As a result of the properties of monomial orders, we can show that $\to_G$ (and therefore $\to_g$) is terminating. We prove this by extending our notion of a term order $\preceq$ to a strict partial order $\lll$ on polynomials, following the method outlined in [Win96].

Definition 3.17. *Let $f, g \in R\langle X\rangle$. Then, we define recursively $f \lll g$ if one of the following conditions holds:*

1. $\mathrm{lt}(f) \prec \mathrm{lt}(g)$*;*
2. $\mathrm{lt}(f) = \mathrm{lt}(g)$ *and* $\mathrm{tail}(f) \lll \mathrm{tail}(g)$*;*

Proposition 3.18. *There exist no infinite strictly decreasing sequences of elements in $R\langle X\rangle$ with respect to $\lll$.*

We adapt the proof of [HR23, Proposistion 4.4].

Proof. Assume, for contradiction, that there exists an infinite strictly decreasing sequence

$$f_0 \ggg f_1 \ggg \ldots$$

of elements $f_0, f_1, \ldots \in R\langle X\rangle$ such that $\mathrm{lt}(f_0)$ is minimal with respect to $\preceq$ among all $f \in R\langle X\rangle$ starting an infinite sequence. Choosing such f_0 is possible since $\preceq$ is a well-order. Note that $f_n \neq 0$ for all $n \in \mathbb{N}$ as 0 is minimal with respect to $\lll$. Hence, we can consider $t_n = \mathrm{lt}(f_n)$, which produces the following infinite decreasing sequence of terms:

$$t_0 \succeq t_1 \succeq \ldots.$$

Because $\preceq$ is a well-order, by Lemma 3.16, this sequence stabilizes at some point $n \in \mathbb{N}$, which means that $\mathrm{lt}(f_n) = \mathrm{lt}(f_{n+k})$ for all $k \in \mathbb{N}$. Thus, by definition of $\lll$, we must have

$$\mathrm{tail}(f_n) \ggg \mathrm{tail}(f_{n+1}) \ggg \ldots,$$

which is still an infinite sequence. For the same reason as before, $\text{tail}(f_{n+k}) \neq 0$ for all $k \in \mathbb{N}$. But then $\text{lt}(\text{tail}(f_n)) \prec \text{lt}(f_0)$, which is a contradiction to the minimality of $\text{lt}(f_0)$. □

Corollary 3.19. *The reduction $\to_G$ is terminating.*

Proof. Any reduction $f \to_G f'$ implies $f \gg f'$. An infinite sequence $f_0 \to_G f_1 \to_G \dots$ would imply an infinite strictly decreasing sequence $f_0 \gg f_1 \gg \dots$. Such a sequence cannot exist according to Proposition 3.18. □

If a sequence of reductions is found that results in zero for a polynomial f, we can express f as a *cofactor representation* of the reducers. Formally, let $f \in R\langle X\rangle \setminus \{0\}$ and $G \subseteq R\langle X\rangle$. If $f \xrightarrow{*}_G 0$, there exists a reduction sequence

$$f = h_0 \ \to_{a_1,g_1,b_1} \ h_1 \ \to_{a_2,g_2,b_2} \ \dots \ \to_{a_d,g_d,b_d} \ h_d = 0$$

with $h_1, \dots, h_d \in R\langle X\rangle$, $a_i, b_i \in \langle X\rangle$, and $g_i \in G$. We can obtain a representation of f in terms of the a_i, g_i, b_i by expanding the reduction sequence. To illustrate, consider a reduction to zero in only two steps. Let $f = h_0 \to_{a_1,g_1,b_1} \ h_1 \ \to_{a_2,g_2,b_2} \ h_2 = 0$. Then, by Definition 3.11 (Polynomial reduction),

$$h_2 = h_1 - q_2 a_2 g_2 b_2 = 0 \iff h_1 = q_2 a_2 g_2 b_2,$$

where the q_i are obtained by Euclidean division of the respective coefficients as described in the definition. We can then write $f = h_0$ as

$$h_1 = h_0 - q_1 a_1 g_1 b_1 \iff q_2 a_2 g_2 b_2 = h_0 - q_1 a_1 g_1 b_1 \iff h_0 = q_1 a_1 g_1 b_1 + q_2 a_2 g_2 b_2.$$

This pattern naturally extends to d steps, yielding the following cofactor representation of f with respect to G:

$$f = \sum_{i=1}^{d} q_i a_i g_i b_i. \tag{1}$$

The cofactor representation is clearly an element of the ideal generated by G, since an ideal is closed under addition with ideal elements and closed under multiplication with ring elements. Therefore, we have a first connection of the polynomial reduction to the ideal membership problem in $R\langle X\rangle$.

Lemma 3.20. *If $f \xrightarrow{*}_G 0$, then $f \in (G)$. More generally, $f \xrightarrow{*}_G f'$ implies $f - f' \in (G)$.*

If f reduces to zero with respect to G, the resulting cofactor representation is *bounded* by the below definition.

Definition 3.21. *Let $G \subseteq R\langle X\rangle$. For a nonzero $f \in (G)$, a cofactor representation*

$$f = \sum_{i=1}^{d} a_i g_i b_i,$$

with nonzero $a_i, b_i \in R\langle X\rangle$ and $g_i \in G$, of f with respect to G is called bounded *if $\mathrm{lm}(a_i g_i b_i) \preceq \mathrm{lm}(f)$ for all $i = 1, \ldots, d$.*

Corollary 3.22. *If $f \xrightarrow{*}_G 0$, then f has a bounded cofactor representation with respect to G. More precisely, the cofactor representation (1) is bounded.*

Proof. Follows from Lemma 3.13 which says the leading monomial can not increase as a result of the polynomial reduction. □

With Lemma 3.20 we also present a sufficient condition which states that if f reduces to 0 with respect to G, then $f \in (G)$. The converse is not necessarily true as we see in the following example.

Example 3.23. *Let $G = (f, g) \trianglelefteq \mathbb{Z}\langle x, y\rangle$ with $f = 3xy$ and $g = 4xy$. Then*

$$xy = g - f \in G.$$

However, this element is not reducible with respect to G since $|\mathrm{lc}(xy)| < |\mathrm{lc}(f)| < |\mathrm{lc}(g)|$.

The following results will be of use when we characterize the ideal membership problem with the polynomial reduction of Definition 3.11.

Lemma 3.24. *Let $G \subseteq R\langle X\rangle$ and $f, f', g, h \in R\langle X\rangle$. Then the following hold:*

1. *$f \xrightarrow{*}_G f'$ implies $afb \xrightarrow{*}_G af'b$ for all $a, b \in \langle X\rangle$;*
2. *$f \to_g 0$ implies $cf \to_g 0$ for all nonzero $c \in R$;*
3. *$f \to_g 0$ implies $f + h \downarrow_g h$;*

The proof is an adaptation of [BN98, Lemma 8.2.6].

Proof. The first statement follows directly from Definition 3.11. If $f \to_{u,g,v} f'$, then for any $a, b \in \langle X\rangle$, we have $afb \to_{au,g,vb} af'b$.

For the second statement, let $f \to_g 0$. This implies $0 = f - q \cdot agb$ for some $a, b \in \langle X\rangle$ and $q \in R$ such that $\mathrm{lc}(f) = q \cdot \mathrm{lc}(g)$ (note that r has to be zero). Then, for the choice $q' = cq$, we have $c \cdot 0 = c \cdot (f - q \cdot agb) = cf - q' \cdot agb$ and $\mathrm{lc}(cf) = q' \cdot \mathrm{lc}(g)$. This shows that $cf \to_g 0$.

To show the third statement, let again $f \to_g 0$. Then $0 = f - q \cdot agb$ for some $a, b \in \langle X \rangle$ and $q \in R$ such that $\mathrm{lc}(f) = q \cdot \mathrm{lc}(g)$ (note that r has to be zero again). Furthermore, denote by c_h the coefficient of $\mathrm{lm}(agb)$ in h. We distinguish between the following cases:

Case 1: $c_h = 0$. Then $f + h \to_g h$ (because h does not affect the reduction).

Case 2: $c_h \neq 0$ and $c_h + \mathrm{lc}(f) = 0$. Then we have $c_h = -\mathrm{lc}(f) = -q \cdot \mathrm{lc}(g)$. Thus, if we reduce h by agb, we obtain

$$h - (-q \cdot agb) = h + q \cdot agb + \underbrace{f - q \cdot agb}_{=0} = h + f.$$

This means that $h \to_g f + h$.

Case 3: $c_h \neq 0$ and $c_h + \mathrm{lc}(f) \neq 0$. Divide c_h by $\mathrm{lc}(g)$ to get

$$c_h = q' \,\mathrm{lc}(g) + r$$

with q', r as specified by the Euclidean function. Then, reducing h by agb yields

$$h - q'agb.$$

On the other hand, reducing $f + h$ by agb yields

$$f + h - (q + q') \cdot agb = \underbrace{(f - q \cdot agb)}_{=0} + (h - q' \cdot agb) = h - q' \cdot agb.$$

Thus, we can see that

$$f + h \to_g h - q' \cdot agb \leftarrow_g h.$$

All three cases yield that $f + h \downarrow_g h$. □

Lemma 3.25. *$f_1 + f_2 \xrightarrow{*}_G 0$ implies $f_1 \downarrow_G f_2$.*

We adapt the proof of [BN98, Lemma 8.3.3].

Proof. By induction on the length of the reduction sequence. If $f_1 + f_2 = 0$, then $f_1 = -f_2$ and $f_1 \downarrow_G f_2$ trivially holds. For the induction step, assume that $f_1 + f_2 \to_g h \xrightarrow{*}_G 0$ for some polynomial $g \in G$. Assume that the reduction is applied to the monomial w in $f_1 + f_2$. Let a, b be such that $w = a\,\mathrm{lm}(g)b$ and let $q_1, q_2, r_1, r_2 \in R$ be such that

1. $\mathrm{coeff}(f_1, w) = q_1 \,\mathrm{lc}(g) + r_1$ with $0 \leq |r_1| < |\mathrm{lc}(g)|$;
2. $\mathrm{coeff}(f_2, w) = q_2 \,\mathrm{lc}(g) + r_2$ with $0 \leq |r_2| < |\mathrm{lc}(g)|$.

Furthermore, let $\tilde{q}, \tilde{r} \in R$ be such that

$$r_1 + r_2 = \tilde{q}\,\mathrm{lc}(g) + \tilde{r} \quad \text{with} \quad 0 \leq |\tilde{r}| < |\mathrm{lc}(g)|.$$

Then

$$\begin{aligned}\mathrm{coeff}(f_1 + f_2, w) =& (q_1 + q_2 + \tilde{q}) \cdot \mathrm{lc}(g) + \tilde{r} \\ =& (q_1 + q_2) \cdot \mathrm{lc}(g) + (r_1 + r_2) \\ =& q_1\,\mathrm{lc}(g) + r_1 + q_2\,\mathrm{lc}(g) + r_2.\end{aligned}$$

Therefore, we can choose the reduction $f + g \to_{g_i} h$, such that

$$\begin{aligned}h =& (f_1 + f_2) - (q_1 + q_2)agb \\ =& (f_1 - q_1 agb) + (f_2 - q_2 agb).\end{aligned}$$

Depending on whether q_1, q_2 are zero or not, we have the following zero- or one-step reductions:

$$f_1 \to_g f_1 - q_1 agb \quad \text{and} \quad f_2 \to_g f_2 - q_2 agb.$$

Since $h = (f_1 - q_1 agb) + (f_2 - q_2 agb)$, the induction hypothesis yields $f_1 - q_1 agb \downarrow_G f_2 - q_2 agb$. □

For a full characterization of the ideal membership problem using polynomial reduction, we also need to allow the inverse reduction.

Theorem 3.26. *Let $G \subseteq R\langle X\rangle$ and $f, f' \in R\langle X\rangle$. Then $f \in (G)$ if and only if $f \overset{*}{\leftrightarrow}_G 0$. More generally, $f - f' \in (G)$ if and only if $f \overset{*}{\leftrightarrow}_G f'$.*

We adapt the proof of [HR23, Theorem 4.12].

Proof. The first assertion clearly follows from the second one. Thus, we show the latter.

For the "if"-direction, define a relation on $R\langle X\rangle$ by $f \sim_G f'$ if $f - f' \in (G)$. This is an equivalence relation. Furthermore, note that $\overset{*}{\leftrightarrow}_G$ is the smallest equivalence relation containing $\to_G$. Thus, $\to_G \subseteq \sim_G$ implies $\overset{*}{\leftrightarrow}_G \subseteq \sim_G$, which in turn yields the desired result. But $\to_G \subseteq \sim_G$ follows from Lemma 3.20 and the fact that $\to_G \subseteq \overset{*}{\to}_G$.

For the "only if"-direction, write $f - f'$ as $f - f' = \sum_{i=1}^{d} c_i a_i g_i b_i$ with nonzero $c_i \in R$, $a_i, b_i \in \langle X\rangle$ and $g_i \in G$. We show that $f \overset{*}{\leftrightarrow}_G f'$ holds for the case $d = 1$. Then the general statement for $d \geq 1$ follows by induction on d. Clearly $g_1 \to_G 0$, and thus, by the first part of Lemma 3.24, also $f - f' = c_1 a_1 g_1 b_1 \to_G 0$. With this, the second part of Lemma 3.24 yields $f \downarrow_G f'$, and consequently, $f \overset{*}{\leftrightarrow}_G f'$. □

Under this relation, $f \in (G)$ is equivalent to $f \overset{*}{\leftrightarrow}_G 0$. However, the inverse relation is not terminating and therefore neither is $\overset{*}{\leftrightarrow}_G$. Only if $f \overset{*}{\rightarrow}_G 0$ is confluent, Theorem 3.26 is applicable to solve the ideal membership problem algorithmically.

Corollary 3.27. *If $\rightarrow_G$ is confluent, then $f \in (G)$ if and only if $f \overset{*}{\rightarrow}_G 0$.*

To guarantee the confluence of the reduction $\rightarrow_G$, we have to chose G with some care. If chosen appropriately, these sets are called *(strong) Gröbner bases* of the ideal (G) and their characterizations are the topic of the following chapter.

Chapter 4

Strong Gröbner bases

Bruno Buchberger first introduced Gröbner bases in 1965 as part of his dissertation, see [Buc65]. He constructed these special bases as sets that ensure that every polynomial has a unique remainder when divided by the Gröbner basis. Buchberger defined Gröbner bases for polynomial rings over a field. We state a definition for noncommutative polynomial rings over a Euclidean domain R. Many of the results of this chapter are in essence similar to [LMAZ23], a study of noncommutative Gröbner bases over the integers.

Definition 4.1. *Let $I \trianglelefteq R\langle X\rangle$. A subset $G \subseteq I$ is a* (strong) Gröbner basis *of I if $(G) = I$ and $\rightarrow_G$ is confluent.*

Recall from Definition 2.13 that we say $w \in \langle X\rangle$ is divisible by another $w' \in \langle X\rangle$, if there exist $a, b \in \langle X\rangle$ such that $w = aw'b$. This notion allows us to determine whether one leading monomial divides the other. We now extend this concept to leading terms.

Definition 4.2. *Let $f, g \in R\langle X\rangle$. We say the leading term of g* divides *the leading term of f if*

1. $\mathrm{lm}(g)$ *divides* $\mathrm{lm}(f)$ *and*
2. $\mathrm{lc}(g)$ *divides* $\mathrm{lc}(f)$.

In a Gröbner basis G over a field, for every $f \in I$, there exists a $g \in G$, such that $\mathrm{lm}(g)$ divides $\mathrm{lm}(f)$. Just divisibility of leading monomials is not sufficient in the context of Gröbner bases over Euclidean domains because divisibility of leading coefficients is not a given as in the case of coefficient fields. However, if G is a strong Gröbner basis of I, for every nonzero $f \in I$, there exists a $g \in G$ such that $\mathrm{lt}(g)$ divides $\mathrm{lt}(f)$. This is part of the following characterization.

Theorem 4.3. *Let $I \trianglelefteq R\langle X\rangle$ and $G \subseteq I$. Then the following are equivalent:*

1. *G is a Gröbner basis of I;*
2. *$f \xrightarrow{*}_G 0$ for all $f \in I$;*
3. *for all nonzero $f \in I$, there exists $g \in G$ such that $\mathrm{lt}(g)$ divides $\mathrm{lt}(f)$.*

We adapt the proof of [Hof23, Theorem 2.4.27].

Proof. We show $1 \iff 2 \iff 3$:

$1 \implies 2$ Follows form Corollary 3.27.

$2 \implies 1$ Note that $(G) = I$ follows from Lemma 3.20. For the confluence, let $f, f_1, f_2 \in R\langle X\rangle$ be such that $f_1 \xleftarrow{*}_G f \xrightarrow{*}_G$. Then clearly $f_1 \xleftrightarrow{*}_G f_2$, and thus $f_1 - f_2 \in I$ by Theorem 3.26. Our assumption implies $f_1 - f_2 \xrightarrow{*}_G 0$. With this, Lemma 3.25 yields $f_1 \downarrow_G f_2$, showing that $\downarrow_G$ is confluent.

$2 \implies 3$

Assume, for contradiction, that there exists a nonzero $f \in I$ such that $f \xrightarrow{*}_G 0$ and there is no $g \in G$ such that $\mathrm{lt}(g)$ divides $\mathrm{lt}(f)$. Then, by assumption, $f \xrightarrow{*}_G 0$, but this is only possible if f is top reducible by G. Let $g' \in G$ be the element with minimal leading coefficient among those which can be used for a top reduction of f. Since G is a Gröbner basis of I (because $1 \iff 2$) we can assume, without loss of generality, that $f \to_{a,g',b} f'$ is the first step of $f \xrightarrow{*}_G 0$ for some $a, b \in \langle X\rangle$. According to the assumption, the leading term is not canceled in that step, and therefore the leading coefficient of f' is smaller than the leading coefficient of g'. Since the leading coefficient of g' is minimal among all $g \in G$ that can be used to top reduce f, there exists no possible reducer to top reduce f' further. Therefore f cannot reduce to zero with respect to G, this is a contradiction.

$3 \implies 2$

Assume, for contradiction, that not all elements in I can be reduced to zero with respect to G. Let $f \in I$ be such an element. Without loss of generality, we can assume that f is irreducible with respect to $\to_G$. By assumption, there exists $g \in G$ such that $\mathrm{lt}(g)$ divides $\mathrm{lt}(f)$, but then f is (top) reducible by g – a contradiction. □

The previous characterizations are not constructive, meaning they cannot be used to verify algorithmically whether a set forms a Gröbner basis.

4.1 Ambiguities

To formulate a constructive characterization of Gröbner bases, we begin by analyzing minimal situations where different reductions can be applied to the same monomial. This leads to the notion of *ambiguities*.

Definition 4.4. *Let $p, q \in \langle X \rangle$. If there exists $a, b \in \langle X \rangle \setminus \{1\}$ with $|a| < |q|$ and $|b| < |p|$ such that*

1. *$pa = bq$ (resp. $ap = qb$), then we call the tuple $(1 \otimes a, b \otimes 1, p, q)$ (resp. $(a \otimes 1, 1 \otimes b, p, q)$) an* overlap ambiguity *of p and q.*
2. *$p = aqb$ (resp. $apb = q$), then we call the tuple $(1 \otimes 1, a \otimes b, p, q)$ (resp. $(a \otimes b, 1 \otimes 1, p, q)$) an* inclusion ambiguity *of p and q.*

Additionally, for $w \in \langle X \rangle$, we call the tuple $(1 \otimes wq, pw \otimes 1, p, q)$ (resp. $(qw \otimes 1, 1 \otimes wp, p, q)$) an external ambiguity *of p and q.*

External ambiguities exist for any two elements in $\langle X \rangle$ and do not have special requirements. For $f, g \in R\langle X \rangle \setminus \{0\}$ and R a Euclidean domain, an ambiguity is also denoted by

$$(a \otimes b, c \otimes d, f, g),$$

where $(a \otimes b, c \otimes d, \mathrm{lm}(f), \mathrm{lm}(g))$ is an ambiguity of $\mathrm{lm}(f)$ and $\mathrm{lm}(g)$. In the case of $f = g$, at least one element of a, b, c, d is not 1 and the set of all ambiguities of f and g is denoted by

$$\mathrm{amb}(f, g).$$

The set of ambiguities of a set $G \subseteq R\langle X \rangle$ is denoted by

$$\mathrm{amb}(G) = \bigcup_{f,g \in G} \mathrm{amb}(f, g).$$

Lemma 4.5. *Let $f, g \in R\langle X \rangle \setminus \{0\}$. If $(a \otimes b, c \otimes d, f, g)$ is an ambiguity of f and g then*

$$\mathrm{lm}(afb) = \mathrm{lm}(cgd).$$

Definition 4.6. *Let $f, g \in R\langle X \rangle \setminus \{0\}$ and $\mathfrak{a} = (a \otimes b, c \otimes d, f, g) \in \mathrm{amb}(f, g)$. The* leading monomial *of $\mathfrak{a}$ is defined as*

$$\mathrm{lm}(\mathfrak{a}) = \mathrm{lm}(afb) = \mathrm{lm}(cgd).$$

Example 4.7. *Let $f, g \in \mathbb{Z}\langle x, y\rangle$ with $f = 2xyx + x$ and $g = 3yx - y$ equipped with the monomial order $\preceq_{\text{deglex}}$ such that $x \prec_{\text{lex}} y$, we have the following overlap ambiguity (note that we only consider the leading monomials in search of ambiguities):*

$$\begin{array}{c|c|c} & x & yx \\ y & x & \end{array} \quad \textit{yields } \mathfrak{a} = (y \otimes 1, 1 \otimes yx, f, g);$$

and the inclusion ambiguity

$$\begin{array}{c|c|} x & yx \\ & yx \end{array} \quad \textit{yields } \mathfrak{b} = (1 \otimes 1, x \otimes 1, f, g).$$

Now, let $w \in \langle x, y\rangle$ be arbitrary. Then the external ambiguities, which require neither an actual overlap nor an inclusion of f and g, have the general form of

$$\begin{array}{c|c|c} xyx & w & \\ & w & yx \end{array} \quad \textit{yields } \mathfrak{c}_1 = (1 \otimes wyx, xyxw \otimes 1, f, g);$$

and respectively

$$\begin{array}{c|c|c} & w & xyx \\ yx & w & \end{array} \quad \textit{yields } \mathfrak{c}_2 = (yxw \otimes 1, 1 \otimes wxyx, f, g).$$

The leading monomials of these ambiguities are

$$\operatorname{lm}(\mathfrak{a}) = yxyx, \quad \operatorname{lm}(\mathfrak{b}) = xyx, \quad \operatorname{lm}(\mathfrak{c}_1) = xyxwyx, \quad \operatorname{lm}(\mathfrak{c}_2) = yxwxyx.$$

Note that w is arbitrary, meaning there are infinitely many variations of $\mathfrak{c}_1$ and $\mathfrak{c}_2$. To illustrate this, let $d \in \mathbb{N}$ be the maximum length of the leading monomials of the external ambiguities and let $d = 7$. Since $|\operatorname{lm}(f)| = |xyx| = 3$ and $|\operatorname{lm}(g)| = |yx| = 2$, only $w \in \{1, x, y, xx, xy, yx, yy\}$ are valid choices. The external ambiguities then are:

$$\begin{aligned}
\mathfrak{c}_1 &= (1 \otimes yx, xyx \otimes 1, f, g), & \mathfrak{c}_2 &= (yx \otimes 1, 1 \otimes xyx, f, g),\\
\mathfrak{c}_3 &= (1 \otimes xyx, xyxx \otimes 1, f, g), & \mathfrak{c}_4 &= (yxx \otimes 1, 1 \otimes xxyx, f, g),\\
\mathfrak{c}_5 &= (1 \otimes yyx, xyxy \otimes 1, f, g), & \mathfrak{c}_6 &= (yxy \otimes 1, 1 \otimes yxyx, f, g),\\
\mathfrak{c}_7 &= (1 \otimes xxyx, xyxxx \otimes 1, f, g), & \mathfrak{c}_8 &= (yxxx \otimes 1, 1 \otimes xxxyx, f, g),\\
\mathfrak{c}_9 &= (1 \otimes xyyx, xyxxy \otimes 1, f, g), & \mathfrak{c}_{10} &= (yxxy \otimes 1, 1 \otimes xyxyx, f, g),\\
\mathfrak{c}_{11} &= (1 \otimes yxyx, xyxyx \otimes 1, f, g), & \mathfrak{c}_{12} &= (yxyx \otimes 1, 1 \otimes yxxyx, f, g),\\
\mathfrak{c}_{13} &= (1 \otimes yyyx, xyxyy \otimes 1, f, g), & \mathfrak{c}_{14} &= (yxyy \otimes 1, 1 \otimes yyxyx, f, g).
\end{aligned}$$

Generating external ambiguities is a combinatorial problem that does not require polynomial arithmetic. In programming, it is clearly not possible to consider the infinitely many external ambiguities of two polynomials. A common practice is to fix a maximum length $d \in \mathbb{N}$ for the leading monomials. Our SageMath adaptation of the package `operator_gb` generates external ambiguities by firstly determining the maximum length of the external part and then generating all possible words up to that length. Proceeding, it generates ambiguities for all pairs of polynomials and for all valid words.

The following SageMath code continues Example 4.7. We begin by importing the necessary data structures from `operator_gb_Z` and then define the ideal. The function

`generate_external_ambs`

yields all external ambiguities up to `maxdeg` length.

```
from operator_gb_Z import *
F.<x,y> = FreeAlgebra(ZZ,2)
A = MyFreeAlgebra(ZZ,F.gens())
I = NCIdeal([2*x*y*x + x, 3*y*x - y])
words = [str(A(g).lm()) for g in I.gens()]
ex_ambs = Ambiguity.generate_external_ambs(words, I.parent(), maxdeg=7)
output = A.translator()(str(ex_ambs[0]), to_internal=False)
print('external ambiguities for maxdeg = ' + str(7) + ': ' + str(output))
```

We can examine the following output in our SageMath environment.

```
#Output
external ambiguities for maxdeg = 7: [(xyxyx, , xyx, xyx, , (0, 1)),
(yxxyx, , yx, yx, , (1, 0)), (xyxxyx, , xxyx, xyxx, , (0, 1)), (yxxxyx,
, xyx, yxx, , (1, 0)), (xyxyyx, , xyyx, xyxy, , (0, 1)), (yxyxyx, ,
xyx, yxy, , (1, 0)), (xyxxxyx, , xxxyx, xyxxx, , (0, 1)), (yxxxxyx, ,
xxyx, yxxx, , (1, 0)), (xyxxyyx, , xxyyx, xyxxy, , (0, 1)),
(yxxyxyx, , yxyx, yxxy, , (1, 0)), (xyxyxyx, , xyxyx, xyxyx, , (0,
1)), (yxyxxyx, , xxyx, yxyx, , (1, 0)), (xyxyyyx, , xyyyx, xyxyy,
, (0, 1)), (yxyyxyx, , yxyx, yxyy, , (1, 0))]
```

4.2 S-polynomials and G-polynomials

Using the notion of ambiguities allows us to capture the different possibilities one monomial can be reduced by another. Each reduction possibility can be applied in multiple ways through the

multiplication of different cofactors. One approach is to multiply the cofactors such that the leading terms cancel out. This idea is captured in the following definition.

Definition 4.8. *Let $G \subseteq R\langle X\rangle$ and $f, g \in G$ with $\mathfrak{a} = (a \otimes b, c \otimes d, f, g) \in \operatorname{amb}(f, g)$ and $q = \operatorname{lcm}(\operatorname{lc}(f), \operatorname{lc}(g))$ such that $q = q_f \operatorname{lc}(f) = q_g \operatorname{lc}(g)$. Then the* S-Polynomial *of $\mathfrak{a}$ is*

$$\text{S-Pol}(\mathfrak{a}) = q_f afb - q_g cgd.$$

Remark 4.9. *The definition of the S-polynomial differs in the Euclidean domain case compared to the field case in the sense of how the leading coefficients are canceled out. In the field case, both terms are simply divided by the respective leading coefficient to produce leading coefficients of 1. However, the quotient of two elements does not necessarily exist in a Euclidean domain. The least common multiple is therefore used to produce equal leading coefficients that then cancel out.*

For coefficient fields, it suffices to consider only the S-polynomials. However, for Euclidean domains, we can easily show that we need an additional reduction.

Example 4.10. *Let $I = (3xy, 2xy) \trianglelefteq \mathbb{Z}\langle x, y\rangle$ with $f = 3xy$ and $g = 2xy$. We have the ambiguity*

$$\mathfrak{a} = (1 \otimes 1, 1 \otimes 1, f, g)$$

which yields the S-polynomial

$$\text{S-Pol}(\mathfrak{a}) = 2 \cdot 3xy - 3 \cdot 2xy = 0.$$

We can also compute S-polynomials for external ambiguities. Let $w \in \langle x, y\rangle$ then we have two general forms of external ambiguities:

$$\mathfrak{b} = (1 \otimes wxy, xyw \otimes 1, f, g) \text{ and } \mathfrak{c} = (xyw \otimes 1, 1 \otimes wxy, f, g).$$

which have the same corresponding S-polynomial

$$\text{S-Pol}(\mathfrak{b}) = \text{S-Pol}(\mathfrak{c}) = 2 \cdot 3xy \cdot w \cdot xy - 3 \cdot xy \cdot w \cdot 2xy = 0.$$

For the coefficient field case, this would mean $G = \{3xy, 2xy\}$ is a Gröbner basis of I. Every nonzero $p \in I$ has a leading term which is divisible by $\operatorname{lt}(f)$, and also by $\operatorname{lt}(g)$. One such element is xy which we receive by subtracting g from f, so it is clearly an element of I. However, since we have coefficients in $\mathbb{Z}$, $\operatorname{lt}(xy)$ is not divisible by any leading term of G. By adding it to G, we get the Gröbner basis

$$G = \{3xy, 2xy, xy\}.$$

We can even omit f and g, so $G = \{xy\}$ is also a Gröbner basis of I.

Example 4.10 shows that we also need to consider reductions that minimize the leading coefficient.

Definition 4.11. *Let $G \subseteq R\langle X\rangle$ and $f, g \in G$ with $\mathfrak{a} = (a \otimes b, c \otimes d, f, g) \in \operatorname{amb}(f, g)$ and $p = \gcd(\operatorname{lc}(f), \operatorname{lc}(g))$ with the Bézout coefficients u, v such that $p = u \operatorname{lc}(f) + v \operatorname{lc}(g)$. Then the* G-Polynomial *of $\mathfrak{a}$ is*

$$\text{G-Pol}(\mathfrak{a}) = uafb + vcgd.$$

Lemma 4.12. *Let $G \subseteq R\langle X\rangle$. For $\mathfrak{a} \in \operatorname{amb}(G)$, we have*

$$\operatorname{lm}(\text{S-Pol}(\mathfrak{a})) \prec \operatorname{lm}(\mathfrak{a}) \text{ and } \text{S-Pol}(\mathfrak{a}) \in (G).$$

Furthermore, we have $\text{G-Pol}(\mathfrak{a}) \in (G)$.

With the notion of S-polynomials we can easily show why we also need to consider external ambiguities in addition to inclusion and overlap ambiguities.

Example 4.13. *Let $(f, g) \subseteq \mathbb{Z}\langle x, y\rangle$ with $f = 2y + x$ and $g = 2x$. The set* $\operatorname{amb}(f, g)$ *has no inclusion or overlap ambiguities. One external ambiguities is*

$$\mathfrak{a} = (1 \otimes x, y \otimes 1, f_1, f_2)$$

which has the corresponding S-polynomial

$$\begin{aligned}\text{S-Pol}(\mathfrak{a}) &= f \cdot x - y \cdot g \\ &= (2y + x) \cdot x - y \cdot 2x \\ &= 2yx + x^2 - 2yx \\ &= x^2.\end{aligned}$$

According to Lemma 4.12, $\text{S-Pol}(\mathfrak{a}) \in (f, g)$ *but there exists no $g \in G$ such that* $\operatorname{lt}(g)$ *divides x^2. Therefore, even though there are no inclusion or overlap ambiguities in* $\operatorname{amb}(f, g)$*, G is not a Gröbner basis.*

If the coefficients in Example 4.13 were from a field, G would be a Gröbner basis because $\operatorname{lt}(g)$ would divide x^2.

Our primary interest is in whether the S-polynomial and G-polynomial of an ambiguity can be reduced to zero with respect to G. This is an indication of whether the reduction introduces a new element to the Gröbner basis.

Definition 4.14. *Let $G \subseteq R\langle X\rangle$. An ambiguity $\mathfrak{a} \in \operatorname{amb}(G)$ is called* resolvable *if*

1. $\text{S-Pol}(\mathfrak{a}) \xrightarrow{*}_G 0$ *and*
2. $\text{G-Pol}(\mathfrak{a}) \xrightarrow{*}_G 0$.

Using this concept, we can provide another characterization of Gröbner bases. The following theorem presents a version of Buchberger's criterion, as detailed in [CLO15, Theorem 2.6.6], which provides a constructive characterization of Gröbner bases. The noncommutative version was first introduced by Bergmann in [Ber78] and generalized to Euclidean domains in [LMAZ23].

Theorem 4.15 (Bergmann's diamond lemma)**.** *Let $G \subseteq R\langle X\rangle \setminus \{0\}$. Then the following conditions are equivalent:*

1. *G is a Gröbner basis of (G);*
2. *for every pair $f, g \in G$, their ambiguities are resolvable.*

Proof. See [LMAZ23, Lemma 13]. □

With Theorem 4.15, we have a criterion to construct a Gröbner basis of an ideal $(F) \trianglelefteq R\langle X\rangle$, given F. We can set $G = F$, and check the resolvability of all ambiguities $\operatorname{amb}(G)$. If their normal forms with respect to G are not zero, we simply add them to G so they become resolvable. However, because the set G was enlarged, we must reassess the resolvability of all ambiguities. This strategy was already applied in Example 4.10 to compute a finite Gröbner basis. A finite Gröbner basis generally only exists in such special cases as we illustrate now.

Example 4.16. *Let $(f, g) \trianglelefteq \mathbb{Z}\langle x, y, z\rangle$ with $f = 3xy + z$ and $g = 5zz + y$ be equipped with the degree lexicographic order such that $x \prec_{\text{lex}} y \prec_{\text{lex}} z$. There are no overlap or inclusion ambiguities in $\operatorname{amb}(G)$, only the infinitely many external ambiguities for $w \in \langle x, y, z\rangle$, which are*

$$\mathfrak{a} = (1 \otimes wzz, xyw\otimes, f, g) \text{ and } \mathfrak{b} = (zzw \otimes 1, 1 \otimes wxy, f, g).$$

The corresponding S-polynomials are

$$\begin{aligned}
\text{S-Pol}(\mathfrak{a}) &= 5 \cdot f \cdot wzz - 3 \cdot xyw \cdot g \\
&= 5 \cdot (3xy + z) \cdot wzz - 3 \cdot xyw \cdot (5zz + y) \\
&= 15xywzz + 5zwzz - 15xywzz - 3xywy \\
&= 5zwzz - 3xywy
\end{aligned}$$

and

$$\begin{aligned}\text{S-Pol}(\mathfrak{b}) &= 5 \cdot zzw \cdot f - 3 \cdot g \cdot wxy \\ &= 5 \cdot zzw \cdot (3xy + z) - 3 \cdot (5zz + y) \cdot wxy \\ &= 15zzwxy + 5zzwz - 15zzwxy - 3ywxy \\ &= 5zzwz - 3ywxy.\end{aligned}$$

We can reduce S-Pol($\mathfrak{a}$) *by* g *in the following manner:*

$$\begin{aligned}\text{S-Pol}(\mathfrak{a}) \to_{zw,g,1} &(5zwzz - 3xywy) - zw \cdot (5zz + y) \\ =&5zwzz - 3xywy - 5zwzz - zwy \\ =&-3xywy - zwy\end{aligned}$$

the only way to continue reducing S-Pol($\mathfrak{a}$) *by* G *is to choose* f *as the next reducer which leads to*

$$\begin{aligned}-3xywy - zwy \to_{1,f,wy} &(-3xywy - zwy) - (3xy + z) \cdot wy \\ =&-3xywy - zwy - 3xywy - zwy = -2zwy.\end{aligned}$$

Switching the reduction steps leads to the same normal form of S-Pol($\mathfrak{a}$) *with respect to* G*. We cannot reduce* $-2zwy$ *any further so we need to add it to* G*. Even for* $w = x^n$ *with* $n \in \mathbb{N}$*, we need to add all infinitely many* $-2zx^ny$ *to* G *because for no* $n \in \mathbb{N}$*, there exists a leading term in* G *which divides* $-2zx^ny$ *other than itself.*

Even though for most ideals there exists no finite Gröbner basis, we can still formulate a procedure to enumerate a Gröbner basis. For that matter, we need a way to deal with the infinitely many external ambiguities each pair of elements has. Therefore, we introduce the enumeration strategy from [Hof23] to iterate ambiguities between two polynomials.

Definition 4.17. *Let* $f, g \in R\langle X\rangle$*. We assume the existence of a method to enumerate ambiguities of* f *and* g*, which is provided by the following two functions:*

- `firstamb` *taking as input two polynomials* f *and* g *and returning an ambiguity* $\mathfrak{a}_0$ *between them;*
- `nextamb` *taking as input an ambiguity* $\mathfrak{a}$ *of two polynomials* f *and* g*, and returning another ambiguity* $n(\mathfrak{a})$ *between them;*

with the property that $\{\mathfrak{a}_0, n(\mathfrak{a}_0), n(n(\mathfrak{a}_0)), \dots\} = \text{amb}(f, g)$*.*

This enumeration strategy enables us to handle infinitely many ambiguities by iteratively working with finite subsets. If one ambiguity is checked for resolvability, its successor is added to the set of ambiguities that await processing. To guarantee that every ambiguity in this ever-changing set is eventually processed, we must select them *fairly*.

Definition 4.18. *A selection strategy, choosing from an ever-changing set S an element at a time, is called* fair *if every element that is present in S at some point is selected eventually.*

The *first-in first-out* strategy is fair. It chooses the element which has been in S the longest, ensuring that every element is eventually chosen. Choosing the $n \in \mathbb{N}$ elements that have been in S the longest is also a fair selection strategy. We now have all tools to formulate a procedure to enumerate a Gröbner basis.

Procedure 1 Buchberger

Input: a finite set $F \subseteq R\langle X \rangle$

Enumerates: $G \subseteq R\langle X \rangle$ such that G is a Gröbner basis of (F)

1: $G \leftarrow F$;
2: spool $\leftarrow \{\text{firstamb}(f, g) \mid f, g \in G\}$;
3: **while** spool $\neq \emptyset$ **do**
4: choose $\mathfrak{a} \in$ spool fairly;
5: spool $\leftarrow (\text{spool} \setminus \{\mathfrak{a}\}) \cup \{\text{nextamb}(\mathfrak{a})\}$;
6: $f' \leftarrow$ S-Pol($\mathfrak{a}$) reduced w.r.t. G;
7: $g' \leftarrow$ G-Pol($\mathfrak{a}$) reduced w.r.t. G;
8: **if** $f' \neq 0$ **then**
9: $G \leftarrow G \cup \{f'\}$;
10: spool $\leftarrow$ spool $\cup \{\text{firstamb}(f', g) \mid g \in G\}$;
11: **if** $g' \neq 0$ **then**
12: $G \leftarrow G \cup \{g'\}$;
13: spool $\leftarrow$ spool $\cup \{\text{firstamb}(g', g) \mid g \in G\}$;
14: **return** G;

Theorem 4.19. *Let $F \subseteq R\langle X \rangle$ be finite and, for $n \in \mathbb{N}$, let G_n be the value of G in Procedure 1 after n iterations of the "while" loop given F as input. Then $G = \bigcup_{n \in \mathbb{N}} G_n$ is a Gröbner basis of $I = (F)$. In this sense, Procedure 1 enumerates a Gröbner basis of G of I.*

We adapt the proof of [Hof23, Theorem 2.4.59].

Proof. Let $F = \{f_1, \ldots, f_r\}$. Note that $(G) = I$ because $f_1, \ldots, f_r \in G$ and all normal forms f' and g', which are added to G, are elements of I by Lemma 3.20 and Lemma 4.12. With

Theorem 4.15 it remains to show that all ambiguities of G are resolvable. To this end, let $\mathfrak{a} \in \mathrm{amb}(G)$. Since the selection strategy in line 4 is fair, the ambiguity $\mathfrak{a}$ is chosen eventually from spool, say, at the $(n+1)$th iteration. Then its S-polynomial and its G-polynomial are reduced with respect to $\to_{G_n}$. If the normal forms f' and g' of these reductions are both zero, the ambiguity $\mathfrak{a}$ is resolvable with respect to G_n, and thus with respect to G since $G_n \subseteq G$. Otherwise if either normal form is not zero, it is added to G_n, so that $G_{n+1} = G_n \cup \{f', g'\}$. Then $\mathfrak{a}$ becomes resolvable with respect to G_{n+1}, implying again that $\mathfrak{a}$ is resolvable with respect to G since $G_{n+1} \subseteq G$. □

Remark 4.20. *Procedure 1 terminates as soon as the set* spool *is empty. However, because there are infinitely many external ambiguities for each pair of polynomials, a new ambiguity is added in every iteration, preventing termination. This means that the algorithm cannot terminate. In practice we need some measure to enforce termination. One approach is to simply stop after a certain amount of time. Another is to bound the maximum length of the leading monomials of external ambiguities, which ensures that only finitely many ambiguities are considered.*

For a SINGULAR:LETTERPLACE implementation of the Buchberger algorithm to compute Gröbner bases over the integers, we refer to [LMAZ23].

Chapter 5

F4 algorithm

In this chapter, we formulate an algorithm to compute multiple normal forms of S-polynomials and G-polynomials simultaneously. We divide the chapter into two sections, the theory and a documentation for the SageMath package with examples.

5.1 Theory

A significant computational challenge in the Buchberger algorithm is the reduction of S-polynomials and G-polynomials. In [Fau99], Jean-Charles Faugère introduced a new algorithm to compute commutative Gröbner bases, where a matrix is constructed from polynomials to then perform multiple reductions simultaneously using fast linear algebra. Xingqiang Xiu formulates the F4 algorithm for noncommutative Gröbner bases in [Xiu12]. In this section, we present a modified version of Faugère's algorithm to fit the context of noncommutative polynomials and coefficient rings. A study of the F4 algorithm for noncommutative polynomials over coefficient fields can be found in [Hof20, Section 4.3].

We begin by defining the translation of a finite list of polynomials to a matrix and then converting it back. For that matter, we first recall the *linear span*. We denote by R again a Euclidean domain.

Definition 5.1. *Let $P \subseteq R\langle X\rangle$ be a set of polynomials. The* linear span *of P over R, denoted by $\mathrm{span}_R(P)$, is the set of all R-linear combinations of elements in P, which means*

$$\mathrm{span}_R(P) = \{\sum_{i=1}^{m} c_i p_i \mid m \in \mathbb{N}, c_i \in R, p_i \in P\} \subseteq R\langle X\rangle.$$

In [Xiu12], polynomials and matrices are related via an isomorphism for a coefficient field. We adapt this idea by defining a similar isomorphism for a coefficient ring. For that matter, we

use free modules which are analogous to vector spaces, but where scalars only form a ring. Free modules also have a basis and every element of a free module can be uniquely represented as a linear combination of its basis elements. Refer to [Keo75, Chapter 1.3], for an overview of free modules.

Definition 5.2. *Let $T = \{t_1, \dots, t_m\} \subseteq \langle X \rangle$ be a set of monomials such that $t_1 \succ \cdots \succ t_m$. Furthermore, let R^m be the free module of dimension m over R equipped with the canonical basis $e_1, \dots, e_m \in R^m$. We consider T as a subset of $R\langle X \rangle$ and define an isomorphism*

$$\varphi_T : R^m \to \operatorname{span}_R(T)$$

given by $\varphi_T(e_i) = t_i$ for $i = 1, \dots, m$.

This definition enables us to associate each finite set of polynomials with a matrix and, conversely, associate each matrix with a corresponding set of polynomials, given a fixed set of monomials $T = \{t_1, \dots, t_m\} \subseteq \langle X \rangle$.

Definition 5.3. *Let $M \in R^{m \times n}$ be a matrix and $T = \{t_1, \dots, t_m\} \subseteq \langle X \rangle$ be a set of monomials. We call the set*

$$P_{M,T} = \{\varphi_T(r_1), \dots, \varphi_T(r_n)\} \subseteq \operatorname{span}_R(T),$$

where $r_i \in R^m$ denotes the i-th row of M, the polynomial form *of M with respect to T.*

Conversely, given a finite set of polynomials $P \subseteq R\langle X \rangle$, we let $T = \operatorname{supp}(P)$, and we call the matrix

$$M_P \in R^{m \times n},$$

whose i-th row is given by $\varphi_T^{-1}(p_i)$, the matrix form *of P.*

Example 5.4. *Let us consider $\mathbb{Z}\langle x, y, z \rangle$ where we use the monomial order $\preceq_{\text{deglex}}$ with $x \prec_{\text{lex}} y \prec_{\text{lex}} z$. Let $P = \{p_1, p_2, p_3\} \subseteq \mathbb{Z}\langle x, y, z \rangle$ with*

$$p_1 = 2xyz + zy, \qquad p_2 = zy + x - y, \qquad p_3 = xyz + 2y.$$

The corresponding matrix M_P is then given by

$$M_P = \begin{array}{c} \begin{matrix} xyz & zy & y & x \end{matrix} \\ \begin{pmatrix} 2 & 1 & 0 & 0 \\ 0 & 1 & -1 & 1 \\ 1 & 0 & 2 & 0 \end{pmatrix} \end{array} \begin{matrix} p_1 \\ p_2. \\ p_3 \end{matrix}$$

Staying with $T = \{xyz, zy, y, x\}$*, for the matrix*

$$M = \begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 1 & 2 & 2 \\ 0 & 0 & 3 & 1 \end{pmatrix}$$

we have the polynomial form $P_{M,T} = \{xyz + 2y, zy + 2y + 2x, 3y + x\}$.

Recall from Section 2.2 that a matrix H over the integers is in Hermite normal form if

1. H is upper triangular and any zero rows are below any other row;
2. the leading coefficient of a nonzero row, also called the pivot of a row, is always strictly to the right of the leading coefficient of the row above it, it is also positive;
3. the elements below pivots are zero and elements above pivots are non negative and strictly smaller than the pivot.

The matrix M from Example 5.4 is the Hermite normal form of M_P. Recall that every integer matrix has a unique Hermite normal form, as do all matrices over Euclidean domains. This form can be achieved through a sequence of row operations. Translating a set of polynomials into a matrix has the great advantage of replacing polynomial reductions by fast row operations. The following example demonstrates that connection.

Example 5.5. *Let* $p_1, p_2 \in \mathbb{Z}\langle x, y\rangle$ *with* $p_1 = xy + x$ *and* $p_2 = 2xy - y$ *where we again use the monomial order* $\preceq_{\text{deglex}}$ *with* $x \prec_{\text{lex}} y$*. For the set* $P = \{p_1, p_2\}$ *we have the matrix*

$$M_P = \begin{pmatrix} 1 & 0 & 1 \\ 2 & -1 & 0 \end{pmatrix}.$$

We can obtain the Hermite normal form of M_P *by first subtracting a multiple of the first row from the second which yields* $\begin{pmatrix} 1 & 0 & 1 \\ 0 & -1 & -2 \end{pmatrix}$*. By multiplying the second row with* -1*, we get*

$$\text{HNF}(M_P) = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \end{pmatrix}.$$

With $T = \{xy, y, x\}$*, the polynomial form of* $\text{HNF}(M_P)$ *is given by*

$$P_{\text{HNF}(M_P),T} = \{xy + x, 2x + y\}.$$

Let us now consider the polynomial reduction of p_2 by p_1:

$$\begin{aligned} 2xy - y &\to_{2,p_1,1} 2xy - y - 2(xy + x) \\ &= -2x - y. \end{aligned}$$

Apart from multiplication with -1, both methods yield the same result.

We can extend this example from a reduction of a single polynomial by another to the reduction of a finite set of polynomials $P \subseteq R\langle X\rangle$ by another finite set of polynomials $G \subseteq R\langle X\rangle$. We begin by constructing the matrix $M_{P\cup G}$ and then computing $\mathrm{HNF}(M_{P\cup G})$. Prior to this, we need to identify the suitable elements of G that can be used in a reduction of P and multiply them by the corresponding cofactors. An algorithm for choosing the appropriate elements of G that can be used to reduce P is formulated as follows.

Algorithm 2 Symbolic Preprocessing

Input: a finite set $P \subseteq R\langle X\rangle$ and $G \subseteq R\langle X\rangle$
Output: $G' \subseteq \{agb \mid a, b \in \langle X\rangle, g \in G\}$

$G' \leftarrow \emptyset$;
$T \leftarrow \mathrm{supp}(P)$;
$done \leftarrow \emptyset$;
while $T \neq \emptyset$ **do**
    select $t \in T$;
    $T \leftarrow T \setminus \{t\}$;
    $done \leftarrow done \cup \{t\}$;
    if there exists $g \in G, a, b \in \langle X\rangle$ such that $\mathrm{lm}(agb) = t$ **then**
        $G' \leftarrow G' \cup \{agb\}$;
        $T \leftarrow T \cup (\mathrm{tail}(agb) \setminus done)$;
return G';

Proposition 5.6. *Let $P \subseteq R\langle X\rangle$ be finite and $G \subseteq R\langle X\rangle$. Then Algorithm 2 terminates with input P and G.*

Proof. We first note that since P is finite, so is the set $\mathrm{supp}(P)$. Let $\preceq$ be the monomial order for $R\langle X\rangle$. Assume, for the sake of contradiction, that the algorithm does not terminate. In each iteration, we remove one $t \in T$ but may also add new elements to T. We only add new elements to T if there exist $g \in G$, $a, b \in \langle X\rangle$ such that $\mathrm{lm}(agb) = t$. In this case, we add $\mathrm{tail}(agb)$ to T, where every element is strictly smaller than t. With the variable *done* we ensure that no term gets added to T that has already been processed. Since we always remove t from T and add only elements to it which are strictly smaller than the one we removed, the algorithm would produce

a strictly decreasing sequence of monomials in T if it were to run indefinitely. According to the well ordering property of $\preceq$, this is not possible. Therefore, the algorithm must terminate. □

Remark 5.7. *In step 8 of Algorithm 2 there may exist multiple choices for $g \in G$, $a, b \in \langle X \rangle$ such that $\mathrm{lm}(agb) = t$. In that case, we choose g such that $|\mathrm{lc}(g)|$ is minimal among all candidates.*

Example 5.8. *In this example we apply Algorithm 2 to find suitable reducers in $G = \{f_1, f_2, f_3\} \subseteq \mathbb{Z}\langle x, y, z\rangle$ with*

$$f_1 = xyx + zx, \qquad f_2 = yyz + xy, \qquad f_3 = z + 1,$$

for the set $P = \{p_1, p_2, p_3, p_4, p_5, p_6\} \subseteq \mathbb{Z}\langle X\rangle$ with

$$\begin{aligned} p_1 &= yyz + zy, & p_2 &= xyz - yz, & p_3 &= xyx + zz, \\ p_4 &= xyz, & p_5 &= y - x, & p_6 &= zzy + yx. \end{aligned}$$

We start by initializing $G' = \emptyset$, $done = \emptyset$ and $T = \mathrm{supp}(P)$, such that

$$T = \{yyz, zy, xyz, yz, xyx, zz, x, y, zzy, yx\}.$$

After entering the loop, we select $t = yyz$, remove it from T and add it to done. Since $\mathrm{lm}(f_2) = yyz$, we can use f_2 to reduce t and therefore add it to G'. Note that a and b are both the empty word in this case. In the next step we add $\mathrm{tail}(f_2)$ to T without elements that are in done. That means

$$T = T \cup \{xy\}$$

and we are finished with the first iteration. In the second iteration we select $t = zy$, again remove it from T and add it to done. We choose f_3 to reduce t because $\mathrm{lm}(f_3 y) = t$ and add it to G'. We would also add $\mathrm{tail}(f_3 y) = y$ to T, if it were not already contained. Next we have $t = xyz$, for which we have xyf_3 as a reducer. We add xyf_3 to G' and $\mathrm{tail}(xyf_3) = xy$ to T. For $t = yz$, we add yf_3 to G' but $tail(yf_3) = y$ is already in T. We now have

$$T = \{xyx, zz, x, y, zzy, yx, xy\}$$

$$done = \{yyz, zy, xyz, yz\}$$

$$G' = \{f_2, f_3 y, xyf_3, yf_3\}.$$

For $t = xyx$ we again remove it from T and add it to done. We also choose f_1 as a reducer and add it to G'. Since $\mathrm{tail}(f_1) \in done$, we do not add it to T. For $t = zz$ we add zf_3 to G' and $\mathrm{tail}(zf_3) = z$ to T. Note that we could also have chosen $f_3 z$. For $t = x$ and $t = y$ there are not

suitable reducers and they are moved from T to done. For $t = zzy$ we have f_3zy as a reducer and add it to G'. Since $\mathrm{tail}(f_3zy) \in$ *done, we skip to the next iteration. There we have $t = yx$, which does not have a suitable reducer in G. The same goes for $t = xy$. For $t = z$ we have f_3 as a reducer and therefore add it to G'. The tail of f_3 is 1 which can not be reduced so it gets added to T once and then moved to done. There is no element left in T to process which leaves us with*

$$T = \emptyset,$$

$$done = \{yyz, zy, xyz, yz, xyx, zz, x, y, zzy, yx, xy, 1\},$$

$$G' = \{f_2, f_3y, xyf_3, yf_3, f_1, zf_3, f_3zy, f_3\},$$

and the algorithm terminates.

For a finite set of polynomials P, this algorithm identifies all possible reducers in G and multiplies them by the corresponding cofactors. The computation of the Hermite normal form of $M_{P \cup G'}$ is therefore equivalent to a reduction of P by G.

Recall that to apply the diamond lemma, we need to verify the resolvability of the S-polynomials and G-polynomials. To minimize polynomial operations, we compute the S- and G-polynomials in the matrix form via row operations. For that purpose, we introduce the *critical pairs* of an ambiguity.

Definition 5.9. *Let $G \subseteq R\langle X \rangle$ and let $\mathfrak{a} = (a \otimes b, c \otimes d, f, g)$ be an ambiguity for two polynomials $f, g \in G$. Given $q = \mathrm{lcm}(lc(f), \mathrm{lc}(g))$, such that $q = q_f lc(f) = q_g \,\mathrm{lc}(g)$, we call the tuple*

$$\mathrm{cp}_S(\mathfrak{a}) = (q_f afb, q_g cgd)$$

the critical pair *of $\mathfrak{a}$ corresponding to the S-polynomial of $\mathfrak{a}$. The* critical pair *corresponding to the G-polynomial of $\mathfrak{a}$ is*

$$\mathrm{cp}_G(\mathfrak{a}) = (uafb, vcgd),$$

with u, v being the Bézout coefficients of $lc(f)$ and $\mathrm{lc}(g)$.

Definition 5.10. *Let $G \subseteq R\langle X \rangle$ and let A be a set of ambiguities of G. Then we denote the set of critical pairs of A corresponding to S-polynomials by*

$$\mathrm{cp}_S(A) = \{\mathrm{cp}_S(\mathfrak{a}) \mid \mathfrak{a} \in A\},$$

and the set of critical pairs corresponding to G-polynomials by

$$\mathrm{cp}_G(A) = \{\mathrm{cp}_G(\mathfrak{a}) \mid \mathfrak{a} \in A\}.$$

The general set of critical pairs of A is defined as

$$\mathrm{cp}(A) = \mathrm{cp}_S(A) \cup \mathrm{cp}_G(A).$$

For brevity, we extend the notions of Definition 5.10 to sets $G \subseteq R\langle X\rangle$, where $\mathrm{cp}(G)$ denotes the set of critical pairs of ambiguities of G.

For practical applications, it is beneficial to have some measure to order critical pairs, allowing an algorithm to choose appropriate subsets. To that end, we extend the notion of the degree of an ambiguity to its associated critical pairs.

Definition 5.11. *Let $G \subseteq R\langle X\rangle$ and let $\mathfrak{a} = (a\otimes b, c\otimes d, f, g)$ be an ambiguity for two polynomials $f, g \in G$. We define the degree of $\mathrm{cp}_S(\mathfrak{a})$ and $\mathrm{cp}_G(\mathfrak{a})$ as the degree of $\mathfrak{a}$.*

Let $G \subseteq R\langle X\rangle$. Consider the critical pair $\mathrm{cp}_S(\mathfrak{a}) = (f_{\mathfrak{a}_S}, g_{\mathfrak{a}_S})$ of an ambiguity $\mathfrak{a} \in \mathrm{cp}(G)$. Let $P = \{f_{\mathfrak{a}_S}, g_{\mathfrak{a}_S}\}$. Using Algorithm 2, we gather the finitely many elements in G required for the reduction of P. We then construct the matrix

$$M_{P\cup G'} = \begin{pmatrix} q & * & \cdots & * \\ q & * & \cdots & * \\ \hline * & & \cdots & * \\ & \ddots & & \vdots \\ & & \ddots & \vdots \\ & & * & * \end{pmatrix} \begin{matrix} f_{\mathfrak{a}_S} \\ g_{\mathfrak{a}_S} \\ g_1 \\ \vdots \\ g_m \end{matrix}$$

and compute its Hermite normal form. During the computation, at some step, we subtract the first row from the second row. Without loss of generality, let that be the first step. The transformed matrix is then given by

$$\begin{pmatrix} q & * & \cdots & * \\ 0 & * & \cdots & * \\ \hline * & & \cdots & * \\ & \ddots & & \vdots \\ & & \ddots & \vdots \\ & & * & * \end{pmatrix} \begin{matrix} f_{\mathfrak{a}_S} \\ g_{\mathfrak{a}_S} - f_{\mathfrak{a}_S} \\ g_1 \\ \vdots \\ g_m \end{matrix} .$$

Apparently, the second row corresponds to the S-polynomial of $\mathfrak{a}$. Thus, one row of the Hermite normal form of this matrix represents the normal form of the S-polynomial of $\mathfrak{a}$ with respect to G. Since switching rows is a common operation during the process of computing the Hermite normal form, we cannot guarantee that the second row still corresponds to the normal form of the S-polynomial, after the process is complete.

However, we know that the S-polynomial can either be reduced to zero by G, or its normal form with respect to G must introduce a new leading term. In the first case, the row corresponding to the normal form of the S-polynomial must be a zero row. If the normal form does not correspond to a zero row, we can identify it by converting the matrix $\mathrm{HNF}(M_{P\cup G'})$ back into polynomial form and checking each polynomial whether its leading term lies in $\mathrm{lt}(P \cup G')$.

This procedure works in the same way for the critical pair of $\mathfrak{a}$ regarding the G-polynomial. Indeed, we can initialize P with multiple critical pairs, allowing us to compute any number of normal forms simultaneously. Algorithm 3 describes this procedure.

Algorithm 3 Reduction

Input: a finite set $P \subseteq R\langle X\rangle$ and $G \subseteq R\langle X\rangle$

Output: $\tilde{P} \subseteq R\langle X\rangle \setminus \{0\}$

1: $G' \leftarrow \mathrm{SymbolicPreprocessing}(P, G)$;
2: $P' \leftarrow P \cup G'$;
3: $\tilde{P} \leftarrow \{p \in P_{\mathrm{HNF}(M_{P'})} \mid p \neq 0 \text{ and } \mathrm{lt}(p) \notin \mathrm{lt}(P')\}$;
4: **return** $\tilde{P}$;

The following results are essential for formulating an algorithm to compute Gröbner bases using matrices.

Lemma 5.12. *Let $P \subseteq R\langle X\rangle$ be a finite set and $G \subseteq R\langle X\rangle$. Furthermore, let $\tilde{P} = \mathrm{Reduction}(P, G)$ and let P' be obtained as in Algorithm 3 during the computation of $\tilde{P}$. Then all elements in $\mathrm{span}_R(P')$ can be reduced to zero by $P' \cup \tilde{P}$.*

We present an analogous proof to [Hof20, Lemma 4.26], tailored to the setting of coefficient rings.

Proof. Suppose, for contradiction, there exists an element $p \in \mathrm{span}_R(P')$ that cannot be reduced to zero by $P' \cup \tilde{P}$. Furthermore, let $\mathrm{lt}(p)$ be minimal among all choices for p. Obviously, $p \neq 0$. Since the rows of $M_{P'}$ and $\mathrm{HNF}(M_{P'})$ generate the same module over R, it holds that $\mathrm{span}_R(P') = \mathrm{span}_R(P'')$ with $P'' = P_{\mathrm{HNF}(M_{P'})}$. Therefore, we can express p as an R-linear combination of elements $p_1, \ldots, p_n \in P'' \setminus \{0\}$. The leading monomials of $p_1, \ldots, p_n$ must all be different since each of the elements corresponds to one row of a matrix, which is in Hermite normal form. It follows that we must have $\mathrm{lt}(p) = c \cdot \mathrm{lt}(p_{i_0})$ for some $1 \leq i_0 \leq n$ and $c \in R^\times$ where $R^\times$ denotes the set of units of R. By the construction of $\tilde{P}$ we know that either $p_{i_0} \in \tilde{P}$ or $\mathrm{lt}(p_{i_0}) \in \mathrm{lt}(P')$. In both cases, there exists some $g \in P' \cup \tilde{P}$ such that $\mathrm{lt}(g) = \mathrm{lt}(p_{i_0})$. Furthermore, since $\tilde{P} \subseteq P'' \subseteq \mathrm{span}_R(P'') = \mathrm{span}_R(P')$, it follows that $g \in \mathrm{span}_R(P')$. With this g we can

reduce p and obtain $p' \in \operatorname{span}_R(P')$ with $\operatorname{lt}(p') \prec \operatorname{lt}(p)$ and since p was assumed to not be reducible to zero, the same must hold for p'. This contradicts the minimality of $\operatorname{lt}(p)$. □

Theorem 5.13. *Let $G \subseteq R\langle X\rangle$ and let A be a finite set of ambiguities of G. Furthermore,*

$$P = \bigcup_{(f,g)\in \operatorname{cp}(A)} \{f, g\}$$

is the set of all polynomials appearing in $\operatorname{cp}(A)$ and let $\tilde{P} = \mathit{Reduction}(P, G)$. Then the ambiguities corresponding to the critical pairs in $\operatorname{cp}(A)$ are all resolvable with respect to $G \cup \tilde{P}$, i.e.,

$$\text{S-Pol}(\mathfrak{a}) \xrightarrow{*}_{G\cup\tilde{P}} 0 \text{ and}$$

$$\text{G-Pol}(\mathfrak{a}) \xrightarrow{*}_{G\cup\tilde{P}} 0,$$

for all $\mathfrak{a} \in A$.

We modify the proof of [Hof20, Theorem 4.27] to include G-polynomials.

Proof. Let $\mathfrak{a} \in A$ such that $\operatorname{cp}_S(\mathfrak{a}), \operatorname{cp}_G(\mathfrak{a}) \in \operatorname{cp}(A)$ with $\operatorname{cp}_S(\mathfrak{a}) = (f_{\mathfrak{a}_S}, g_{\mathfrak{a}_S})$ and $\operatorname{cp}_G(\mathfrak{a}) = (f_{\mathfrak{a}_G}, g_{\mathfrak{a}_G})$. Let P' be the set obtained during the execution of Algorithm 3 in the computation of $\tilde{P}$. Since $f_{\mathfrak{a}_S}, g_{\mathfrak{a}_S}, f_{\mathfrak{a}_G}, g_{\mathfrak{a}_S} \in P'$ and $\text{S-Pol}(\mathfrak{a}) = f_{\mathfrak{a}_S} - g_{\mathfrak{a}_S}$ and $\text{G-Pol}(\mathfrak{a}) = f_{\mathfrak{a}_G} + g_{\mathfrak{a}_G}$, it follows that $\text{S-Pol}(\mathfrak{a}), \text{G-Pol}(\mathfrak{a}) \in \operatorname{span}_R(P')$. By Lemma 5.12, we can reduce $\text{S-Pol}(\mathfrak{a})$ and $\text{G-Pol}(\mathfrak{a})$ to zero using $P' \cup \tilde{P}$. Note that we can write every element of a critical pair of an ambiguity of G as $cwgw'$ for some $c \in R$, $w, w' \in \langle X\rangle$ and $g \in G$. Furthermore, all elements of P can be written in this way, as well as all elements of $G' = \text{SymbolicPreprocessing}(P, G)$, which are added to P to produce P'. As a result, any reduction performed using an element $p \in P'$, can equivalently be performed by some $g \in G$. This implies that $\text{S-Pol}(\mathfrak{a})$ and $\text{G-Pol}(\mathfrak{a})$ can be reduced to zero by $G \cup \tilde{P}$. □

With the previous results, we can formulate the F4 algorithm for $R\langle X\rangle$.

Procedure 4 F4

Input: a finite set $F \subseteq R\langle X\rangle$

Enumerates: $G \subseteq R\langle X\rangle$ such that G is a Gröbner basis of (F)

$G \leftarrow F$;
spool $\leftarrow \{\text{firstamb}(f,g) \mid f,g \in G\}$;
while spool $\neq \emptyset$ **do**
    choose $A \subseteq$ spool fairly;
    spool $\leftarrow$ (spool $\setminus A$) $\cup \{\text{nextamb}(\mathfrak{a}) \mid \mathfrak{a} \in A\}$;
    $P = \bigcup_{(f,g)\in \text{cp}(A)} \{f,g\}$;
    $\tilde{P} \leftarrow \text{Reduction}(P,G)$;
    for $p \in \tilde{P}$ **do**
        $G \leftarrow G \cup \{p\}$;
        spool $\leftarrow$ spool $\cup \{\text{firstamb}(p,g) \mid g \in G\}$;
return G;

Theorem 5.14. *Let $F \subseteq R\langle X\rangle$ be finite and, for $n \in \mathbb{N}$, let G_n be the value of G in Procedure 4 after n iterations of the "while" loop given F as input. Then $G = \bigcup_{n\in\mathbb{N}} G_n$ is a Gröbner basis of $I = (F)$. In this sense, Procedure 4 enumerates a Gröbner basis of G of I.*

Proof. Analogous to the proof of Theorem 4.19. The single difference is in the number of ambiguities processed in each iteration of the "while" loop. The reductions applied are also the same, only that they are performed by row operations in Procedure 4. □

We examine an example, that deviates slightly from the steps in Procedure 4, instead following the methodology implemented in our SageMath package `operator_gb_Z`. The functions `firstamb` and `nextamb` are designed to handle infinitely many ambiguities. Here, we present an example where we only consider ambiguities with a certain maximum degree. In that context, it is more efficient to compute ambiguities in generations. This approach involves computing all ambiguities of the input set in the first iteration and in all subsequent iterations, we compute all new ambiguities. Our selection strategy is to always choose the ambiguities of lowest degree to process next. For this selection strategy, it is advantageous to have more ambiguities available, which is the case in our implementation compared to Procedure 4.

Example 5.15. *Let $F = \{f_1, f_2\} \subseteq \mathbb{Z}\langle x,y\rangle$ where $\mathbb{Z}\langle x,y\rangle$ is equipped with the monomial order $\preceq_{\text{deglex}}$ and $x \prec_{\text{lex}} y$. Furthermore, let*

$$f_1 = 2yx + x, \qquad f_2 = 3xyx.$$

We use Algorithm 4 to compute a Gröbner basis of (F) *considering ambiguities up to degree* 4. *The first step is to initiate* $G = F$ *and compute all ambiguities of* G. *We have the inclusion ambiguity*

$$\mathfrak{a}_1 = (x \otimes 1, 1 \otimes 1, f_1, f_2)$$

and the overlap ambiguity

$$\mathfrak{a}_2 = (1 \otimes yx, y \otimes 1, f_1, f_2).$$

As the maximum length of ambiguities we consider is 4, *we do not have any external ambiguities in this step. The selection strategy we apply is to always choose the critical pairs of lowest degree. Since the degree of a critical pair depends on its corresponding ambiguity, we can look at the ambiguities and compute only the critical pairs for the ambiguities of lowest degree, which is* $\mathfrak{a}_1$. *We have the corresponding critical pairs*

$$\mathrm{cp}_S(\mathfrak{a}_1) = (6xyx + 3x^2, 6xyx), \qquad \mathrm{cp}_G(\mathfrak{a}_1) = (-2xyx - x^2, 3xyx).$$

From these critical pairs we get

$$P = \{6xyx + 3x^2, 3xyx, -2xyx - x^2, 6xyx\}.$$

The symbolic preprocessing of P *with respect to* G *returns* $G' = \{2xyx + xx\}$. *In* G' *we have the reducers in* G *that can be used to reduce* P. *In the next step, we form the matrix* $M_{P'}$ *for* $P' = P \cup G'$. *It is given by*

$$M_{P'} = \begin{array}{c} \begin{array}{cc} xyx & x^2 \end{array} \\ \begin{pmatrix} 6 & 3 \\ 3 & 0 \\ -2 & -1 \\ 6 & 0 \\ 2 & 1 \end{pmatrix} \end{array} \begin{array}{l} p_1 \\ p_2 \\ p_3. \\ p_4 \\ p_5 \end{array}$$

The computation of the Hermite normal form of $M_{P'}$ *yields*

$$\mathrm{HNF}(M_{P'}) = \begin{pmatrix} 1 & 2 \\ 0 & 3 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix},$$

which corresponds to the polynomial form $P_{\mathrm{HNF}(M_{P'})} = \{xyx + 2x^2, 3x^2\}$. *Since* $\mathrm{lt}(xyx + 2x^2) \notin$

$\mathrm{lt}(P')$ *and* $\mathrm{lt}(3x^2) \notin \mathrm{lt}(P')$*, we add them both to* G*. With* $f_3 = 3x^2$ *and* $f_4 = xyx + 2x^2$*, we now have*

$$G = \{f_1, f_2, f_3, f_4\}.$$

In the next iteration, we start by obtaining the new ambiguities, which are

$$\begin{aligned}
\mathfrak{a}_3 &= (x \otimes 1, 1 \otimes x, f_3, f_3), & \mathfrak{a}_4 &= (x \otimes 1, 1 \otimes yx, f_2, f_3),\\
\mathfrak{a}_5 &= (1 \otimes x, y \otimes 1, f_1, f_3), & \mathfrak{a}_6 &= (1 \otimes x, xy \otimes 1, f_2, f_3),\\
\mathfrak{a}_7 &= (xy \otimes 1, 1 \otimes x, f_3, f_4), & \mathfrak{a}_8 &= (1 \otimes yx, x \otimes 1, f_3, f_4),\\
\mathfrak{a}_9 &= (1 \otimes yx, y \otimes 1, f_1, f_4), & \mathfrak{a}_{10} &= (1 \otimes 1, 1 \otimes 1, f_2, f_4),\\
\mathfrak{a}_{11} &= (x \otimes 1, 1 \otimes 1, f_1, f_4),
\end{aligned}$$

and additionally, the external ambiguities

$$\mathfrak{a}_{12} = (1 \otimes x^2, yx \otimes 1, f_1, f_3), \qquad \mathfrak{a}_{13} = (x^2 \otimes 1, 1 \otimes yx, f_1, f_3).$$

Since we already processed $\mathfrak{a}_1$*, we now have*

$$\mathrm{amb} = \{\mathfrak{a}_2, \mathfrak{a}_3, \mathfrak{a}_4, \mathfrak{a}_5, \mathfrak{a}_6, \mathfrak{a}_7, \mathfrak{a}_8, \mathfrak{a}_9, \mathfrak{a}_{10}, \mathfrak{a}_{11}, \mathfrak{a}_{12}, \mathfrak{a}_{13}\}.$$

Next, we again choose the ambiguities of lowest degree to process. The ambiguities $\mathfrak{a}_3, \mathfrak{a}_5$*,* $\mathfrak{a}_{10}$*, and* $\mathfrak{a}_{11}$ *all have degree three, which is the lowest. The S-polynomial for* $\mathfrak{a}_3$ *does not need to be considered because S-polynomials of ambiguities between the same elements always yield zero. G-polynomials of ambiguities between the same elements are also just that element so they trivially reduce to zero and can therefore be discarded. Then, we get the critical pairs*

$$\begin{aligned}
\mathrm{cp}_S(\mathfrak{a}_5) &= (6yxx + 3x^2, 6yxx), & \mathrm{cp}_G(\mathfrak{a}_5) &= (-2yx^2 - x^2, 3yx^2),\\
\mathrm{cp}_S(\mathfrak{a}_{10}) &= (3xyx, 3xyx + 6x^2), & \mathrm{cp}_S(\mathfrak{a}_{10}) &= (0, xyx + 2x^2),\\
\mathrm{cp}_S(\mathfrak{a}_{11}) &= (2xyx + x^2, 2xyx + 4x^2), & \mathrm{cp}_G(\mathfrak{a}_{11}) &= (0, xyx + 2x^2),
\end{aligned}$$

From these critical pairs we get

$$P = \{6yx^2, 3yx^2, 4x^2 + 2xyx, -x^2 - 2yxx, 3x^2 + 6yx^2, x^2 + 2xyx, 2x^2 + xyx, 3xyx, 6x^2 + 3xyx\}.$$

The symbolic preprocessing of this new P *with respect to* G *returns*

$$G' = \{x^2 + 2yx^2, 3x^2, x^2 + 2xyx\}.$$

Next, we form the matrix $M_{P'}$ for $P' = P \cup G'$:

$$M_{P'} = \begin{array}{c} \begin{matrix} yx^2 & xyx & x^2 \end{matrix} \\ \begin{pmatrix} 6 & 0 & 3 \\ 0 & 2 & 1 \\ 0 & 2 & 4 \\ 0 & 1 & 2 \\ 6 & 0 & 0 \\ 0 & 3 & 6 \\ 0 & 3 & 0 \\ 3 & 0 & 0 \\ -2 & 0 & -1 \\ 0 & 2 & 1 \\ 0 & 0 & 3 \\ 2 & 0 & 1 \end{pmatrix} \end{array} \begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ p_8 \\ p_9 \\ p_{10} \\ p_{11} \\ p_{12} \end{matrix}$$

with the Hermite normal form

$$\mathrm{HNF}(M_{P'}) = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Translating that matrix back into polynomial form yields

$$P_{\mathrm{HNF}(M_{P'})} = \{3x^2, 2x^2 + xyx, 2x^2 + yx^2\}.$$

Of these three polynomials, we add only $f_5 = 2x^2 + yx^2$ to G. In the next iteration, for every element $f_i \in G \setminus \{f_5\}$, we add $\mathrm{amb}(f_i, f_5)$ to the set of ambiguities we still need to process.

For brevity, we omit the next iterations since their presentation is cumbersome. After seven iterations, no new elements are added to G and all ambiguities have been processed. Therefore, the algorithm terminates. The final Gröbner basis is given by

$$G = \{2yx + x, 3xyx, 3x^2, xyx + 2x^2, yx^2 + 2x^2, 3x^3, \\ x^2yx + 2x^3, xyx^2 + 2x^3, yx^3 + 2x^3, yxyx + 2x^2, 3x^4\}.$$

5.2 Software

We present the SageMath package `operator_gb_Z` to compute noncommutative Gröbner bases over the integers up to a certain maximum degree of ambiguities we consider. The package is a modification of the package `operator_gb` by Clemens Hofstadler, which supports Gröbner basis computations over fields.

The package `operator_gb` already provides data structures for working with noncommutative polynomials including ideals, ambiguities, and critical pairs. The existing implementation of the F4 algorithm was also largely suitable for integer coefficients, with some changes.

The first modification occurs when generating ambiguities. While Gröbner basis computations over coefficient fields consider overlap and inclusion ambiguities, we additionally need to consider external ambiguities, as shown in Example 4.13. Since there are infinitely many external ambiguities for even one polynomial, no terminating procedure can handle all of them. A common strategy, which we also use, is to only consider ambiguities where the degree is less than a predetermined maximum degree $d \in \mathbb{N}$, see for example [LMAZ23].

To generate external ambiguities, we do not need to perform any polynomial arithmetic. The only computational task is combinatorial, as it involves identifying all possible external overlaps such that the degrees of the corresponding ambiguities are no longer than the maximum degree d.

Another difference from integer coefficients to field coefficients lies in the critical pairs. As seen in Example 4.10, we have to consider S-polynomials and G-polynomials. The SageMath package `operator_gb` does not work directly with S-polynomials but rather with their critical pairs. For our package, we extended the critical pair data structure to additionally correspond to the G-polynomial of an ambiguity.

In the F4 algorithm for noncommutative Gröbner bases over fields, Gaussian elimination is used to compute normal forms of matrices. In our package, however, we employ the Hermite normal form for that purpose.

The latest version of our SageMath package can be downloaded from

https://github.com/ChristophHeidemann/operator_gb_Z.

The following examples demonstrate how to use the package. We use the degree lexicographic monomial order in our implementation.

Example 5.16. *Let* $(F_1) \trianglelefteq \mathbb{Z}\langle x, y\rangle$ *with*

$$F_1 = \{2xyx + x, 3yx - y\}.$$

We compute a Gröbner basis of (F_1) *and consider ambiguities up to a maximum degree of* 4*:*

```
from operator_gb_Z import *
F.<x,y> = FreeAlgebra(ZZ,2)
A = MyFreeAlgebra(ZZ,F.gens())
I = NCIdeal([2*x*y*x + x, 3*y*x - y])
I.groebner_basis(4)
```

```
#Output
Gröbner basis: [x + 2*y*x, 3*x*y*x, 3*x^2, 2*x^2 + x*y*x, 3*x^3, 2*x^2 +
y*x^2, 3*x^4, 2*x^3 + x^2*y*x, 2*x^3 + x*y*x^2, 2*x^3 + y*x^3, 2*x^2 +
y*x*y*x]
```

Example 5.17. *Let* $(F_2) \trianglelefteq \mathbb{Z}\langle x, y, z\rangle$ *with*

$$F_2 = \{6xy + z, 5yz + 2x, 3x^2\}.$$

We compute a Gröbner basis of (F_2) *and consider ambiguities up to a maximum degree of* 5*:*

```
from operator_gb_Z import *
F.<x,y,z> = FreeAlgebra(ZZ,3)
B = MyFreeAlgebra(ZZ,F.gens())
J = NCIdeal([6*x*y + z, 5*y*z + 2*x, 3*x*x])
J.groebner_basis(5)
```

```
#Output
Gröbner basis: [z + 6*x*y, 2*x + 5*y*z, 3*x^2, x*z, 5*z^2, 3*x^3,
3*x^2*y, x^2 + z^2 + x*y*z, x^2*z, x*z^2, 5*y*z^2, 5*z^3, z*x + 6*x*y*x,
x*z*x, z*x^2, z*x*z, 2*z*x + 5*z*y*z, z^2*x, 3*x^4, 3*x^3*y, x^3*z, x^3 +
x^2*y*z, x^2*z^2, 3*x*y*x^2, x*y*x*z, z*x + 4*x*y*x + z*y*z + x*y^2*z,
z^3 + x*y*z^2, x*z*x^2, x*z*x*y, x*z*y*z, x*z^3, x^3 + y*z*x^2, 4*x^2*y +
y*z^2 + y*z*x*y, y*z*x*z, 5*y*z^3, z^2*x^2, z^3 + z^2*x*y, z^2*x*z,
5*z^2*y*z, 5*z^4, x^3, x^2 + z^2 + y*z*x, 2*x^2 + 2*z^2 + x*y*z + z*x*y,
```

```
z^3, 3*x^2*y*x, x^2*z*x, x*y*z + 6*x*y*x*y, x*z*x*z, x*z^2*x, x^2 + 3*z^2
+ 5*y*z*y*z, y*z^2*x, z*x^3, z*x^2*y, z*x^2*z, z*x*z*x, z*x*z^2,
5*z*y*z^2, z^3*x, 3*y*x^2, 4*y*z^2 + z^2*y, x^4, x^3*y, x*y*z*x, 2*x^2*y
+ 2*y*x^2 + y*z^2 + y*x*y*z, y*z^3, z*x*y*x, z*x*y*z, z*y*z*x, z^4,
x*z*y, 2*x^2*y + y*x^2, y*x*z, x*z^2*y, y*x^3, 3*y*x^2*y, y*x^2*z,
5*y^2*z^2, 4*y^2*z^2 + y*z^2*y, z*y*z^2, z^2*y*z, z^3*y]
```

Bibliography

[AW12] William A. Adkins and Steven H. Weintraub. *Algebra: an approach via module theory*. Springer Science & Business Media, 2012.

[Ber78] George M. Bergman. The diamond lemma for ring theory. *Advances in Mathematics*, 29(2):178–218, 1978.

[BHR] Klara Bernauer, Clemens Hofstadler, and Georg Regensburger. How to automatise proofs of operator statements: Moore–penrose inverse; a case study. In *Proceedings of CASC (2023)*, pages 39–68.

[BN98] Franz Baader and Tobias Nipkow. *Term rewriting and all that*. Cambridge University Press, 1998.

[Bre11] Murray Bremner. *Lattice basis reduction*. CRC Press New York, 2011.

[Buc65] Bruno Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal*. PhD thesis, Universität Innsbruck, Austria, 1965.

[CLO15] David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms*. Undergraduate Texts in Mathematics. Springer, Cham, fourth edition, 2015.

[Coh01] Paul M. Cohn. *Introduction to ring theory*. Springer Science & Business Media, 2001.

[Fau99] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases (F4). *Journal of pure and applied algebra*, 139(1-3):61–88, 1999.

[Hof20] Clemens Hofstadler. Certifying operator identities and ideal membership of noncommutative polynomials. Master's thesis, Johannes Kepler University, Linz, Austria, 2020.

[Hof23] Clemens Hofstadler. *Noncommutative Gröbner bases and automated proofs of operator statements*. PhD thesis, Johannes Kepler University, Linz, Austria, 2023.

[HR23] Clemens Hofstadler and Georg Regensburger. Noncommutative polynomials and Gröbner bases. Lecture notes, 2023.

[Keo75] R. Keown. *An introduction to group representation theory*. Academic press, 1975.

[Lic12] Daniel Lichtblau. Effective computation of strong Gröbner bases over Euclidean domains. *Illinois Journal of Mathematics*, 56(1):177–194, 2012.

[LMAZ23] Viktor Levandovskyy, Tobias Metzlaff, and Karim Abou Zeid. Computing free noncommutative Gröbner bases over Z with Singular: Letterplace. *Journal of Symbolic Computation*, 115:201–222, 2023.

[Mor85] Ferdinando Mora. Gröbner bases for non-commutative polynomial rings. In *International Conference on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes*, pages 353–362. Springer, 1985.

[vzGG03] Joachim von zur Gathen and Jürgen Gerhard. *Modern computer algebra*. Cambridge University Press, 2003.

[Win96] Franz Winkler. *Polynomial algorithms in computer algebra*. Springer Verlag, Wien, 1996.

[Xiu12] Xingqiang Xiu. *Non-commutative Gröbner bases and applications*. PhD thesis, University of Passau, Germany, 2012.