UNIVERSITAT DE BARCELONA

UAB
Universitat Autònoma de Barcelona

ADVANCED MATHEMATICS
MASTER'S FINAL PROJECT

Gröbner Bases on Σ-Algebras

Author:
Iker Pérez Arrese-Igor

Supervisors:
Carlos D'Andrea
Ana Belén de Felipe

Facultat de Matemàtiques i Informàtica

June 12, 2026

Abstract

In this text, we develop a theory of Gröbner bases for Σ-algebras. To do so, we first give introductory notions on Gröbner bases, and on Σ-algebras. Moreover, we propose a method to represent certain monomial orderings of Σ-polynomial algebras via infinite matrices. Within this setting, and despite the non-Noetherianity, we present an algorithm to compute Gröbner bases whenever feasible. Finally, we illustrate these concepts with an example and provide implementations of the proposed algorithms.

Contents

Abstract 3
Introduction 5
1. Gröbner Bases on Polynomial Rings 7
2. Polynomial Difference Algebras 15
3. Gröbner Bases of Σ-Ideals 19
3.1. Monomial Orders and Σ-Orders 19
3.2. Gröbner Bases on Σ-Polynomial Algebras 30
3.3. Order Function and Truncation 37
4. An Illustrative Example 43
5. Appendix: Implementations 46
References 57

Introduction

Gröbner bases were introduced in 1965 in Bruno Buchberger's PhD thesis, together with an algorithm for computing them. In it, Bruno Buchberger and his advisor, Wolfgang Gröbner, presented an algorithmic method for computing a "K-basis" of the K-algebra $\frac{K[x_1,\ldots,x_n]}{I}$, for any ideal I. The necessity of introducing Gröbner bases is based on the study of polynomial ideals. In fact, an apparently simple question like whether a polynomial belongs to a given polynomial ideal was not easy to answer without Gröbner bases. Since their introduction, Gröbner bases have been deeply studied due to their good properties and applications. Indeed, they have been generalized to broader settings, such as non-commutative algebras and path algebras. The feasibility of developing an algorithm that computes Gröbner bases relies fundamentally on the Noetherianity of $K[x_1,\ldots,x_n]$, which poses a problem when trying to present an algorithm that computes Gröbner bases in more general settings, where Noetherianity is not necessarily guaranteed.

Difference algebras were introduced by J.F. Ritt in the 1930s to develop an algebraic approach to the study of systems of difference equations. Ritt worked out a general algebraic framework (Σ-algebras) in which the algebraic properties of difference and differential algebras are reflected, namely, the algebraic properties of shift and derivation operators of arguments of analytic functions. Ritt's work was continued by many other researchers, serving as a background to introduce differential Galois theory, and, more generally, introducing an algebraic-geometric approach to difference and differential algebras. One can find applications of this theory in the study of numerical approximations to solutions of differential equations via finite difference methods. Moreover, developing a Gröbner-bases-like theory on difference algebras will help verify the good properties of these finite difference approximations.

In this text, we introduce Gröbner bases on some concrete Σ-algebras (which we call Σ-polynomial algebras), give some algorithms for their computation along with their implementations, and show how we can apply these concepts in practice. One can find some reasons for developing such a theory in this context, the most immediate of which is to provide a solution to the Σ-ideal membership problem for Σ-polynomials. Other applications include giving a way to construct difference schemes for partial differential equations [6], providing a reduction method of Feynman integrals to master integrals [10], and offering a certification method for whether a finite difference approximation with respect to a P.D.E. is strongly consistent [1]. This last application will be mentioned in Section 5.

In the first section, we introduce the standard Gröbner bases theory on polynomial rings over a field K, emphasizing the importance of the Noetherianity of $K[x_1,\ldots,x_n]$ to develop such a theory, and introducing algorithms on how to compute these bases. We also give an algorithmic solution to the membership problem. In the next section, we introduce the algebraic framework given by Ritt, and define Σ-polynomial algebras, which are the structures where we develop a theory of Gröbner bases. Moreover, we see difference and differential algebras as instances of these Σ-polynomial algebras and present a motivation for developing Gröbner bases on them. The third section is divided into three

parts. In the first part, inspired by Robbiano's characterization of monomial orderings in $K[x_1, \ldots, x_n]$ using real matrices [9, Theorem 4], we present some results that allow us to describe monomial orderings in Σ-polynomial algebras involving infinite matrices. This part constitutes original work by the author, with the help of the tutors. In the second part, we develop a theory of Gröbner bases in Σ-polynomial algebras. In particular, we define Σ-bases, which serve as the analogue of Gröbner bases within this Σ-polynomial setting. Furthermore, we see how the non-Noetherian nature of the Σ-polynomial algebras poses problems when trying to give an algorithm to compute these bases. In the third part, we provide a partial solution to the problem posed by the non-Noetherianity by introducing a notion of order on the variables of the Σ-polynomial algebra. We present an algorithm that, if possible, computes a finite Σ-basis of a Σ-ideal. In addition, we propose an algorithmic solution to the membership problem of a polynomial p in a Σ-ideal. Finally, we conclude with an example in which we put into practice the introduced theory and compute a Σ-basis. In the appendix, we implement the algorithms that appear in the third section.

In this project, we have mainly followed the approach proposed by V. Gerdt and S. La Scala in [5] to introduce Gröbner bases in Σ-algebras. For a proper understanding, some basic notions of commutative algebra are needed, such as polynomial rings over a field K, K-algebras, K-algebra homomorphisms, and the free abelian monoid of rank r.

1. Gröbner Bases on Polynomial Rings

This is a preliminary section where we present some basics about the standard theory of Gröbner bases on polynomial rings $K[x_1, \ldots, x_n]$ where K is a field. Throughout this section, M denotes the set of monomials of $K[x_1, \ldots, x_n]$ (including 1). Furthermore, throughout this text, we consider $\mathbb{N}$ to include 0, and all rings to be commutative and unitary. Recall that a well-ordering on a set A is a total order relation such that any non-empty subset of A has a minimum.

We first introduce monomial orderings and give a result for representing them via matrices.

Notation 1.1. We denote the monomial $x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n} \in K[x_1, \ldots, x_n]$ as x^α, where $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$. We call α the exponent tuple of x^α.

Definition 1.2. The relation $>$ is a monomial ordering on $K[x_1, \ldots, x_n]$ if $>$ is a well-ordering on M and it satisfies the multiplicative property:

$$x^\alpha > x^\beta \implies x^\nu x^\alpha > x^\nu x^\beta$$

for any $x^\alpha, x^\beta, x^\nu \in M$.

We now give some examples of monomial orderings on $K[x_1, \ldots, x_n]$.

Example 1.3. Lexicographical ordering: Let $x^\alpha, x^\beta \in K[x_1, \ldots, x_n]$. Then, $x^\alpha >_{\text{lex}} x^\beta$ if and only if the first non-zero coordinate of $\alpha - \beta \in \mathbb{Z}^n$ is positive.

Example 1.4. Degree Lexicographical ordering: Let $x^\alpha, x^\beta \in K[x_1, \ldots, x_n]$. Then, $x^\alpha >_{\text{deglex}} x^\beta$ if:

$$\begin{cases} \sum_{i=1}^n \alpha_i > \sum_{i=1}^n \beta_i & \text{or} \\ \sum_{i=1}^n \alpha_i = \sum_{i=1}^n \beta_i \text{ and } x^\alpha >_{\text{lex}} x^\beta. \end{cases}$$

Example 1.5. Inverse lexicographical ordering: Let $x^\alpha, x^\beta \in K[x_1, \ldots, x_n]$. Then, $x^\alpha >_{\text{invlex}} x^\beta$ if and only if the last non-zero entry of $\alpha - \beta$ is positive.

Example 1.6. Degree reverse lexicographical ordering: Let $x^\alpha, x^\beta \in K[x_1, \ldots, x_n]$. Then, $x^\alpha >_{\text{degrlex}} x^\beta$ if:

$$\begin{cases} \sum_{i=1}^n \alpha_i > \sum_{i=1}^n \beta_i & \text{or} \\ \sum_{i=1}^n \alpha_i = \sum_{i=1}^n \beta_i \text{ and } x^\beta >_{\text{invlex}} x^\alpha. \end{cases}$$

Remark 1.7. Any monomial ordering $>$ on $K[x_1, \ldots, x_n]$ induces an additive well ordering on $\mathbb{N}^n$ (by abuse of notation, we denote it also by $>$), as follows

$$x^\alpha > x^\beta \iff \alpha > \beta.$$

By additive, we mean that

$$\alpha > \beta \implies \nu + \alpha > \nu + \beta \qquad \forall \alpha, \beta, \nu \in \mathbb{N}^n.$$

Conversely, any additive well ordering on $\mathbb{N}^n$ induces a monomial ordering on $K[x_1, \ldots, x_n]$. The orderings on $\mathbb{N}^n$ induced by the lexicographical and the inverse lexicographical orderings can be "extended" to additive total orderings on $\mathbb{R}^n$ (embedding $\mathbb{N}$ in $\mathbb{R}$) as follows. Let $\alpha, \beta \in \mathbb{R}^n$:

- We write $\alpha >'_{\text{lex}} \beta$ if and only if the first non-zero coordinate of $\alpha - \beta$ is positive.

- We write $\alpha >'_{\text{invlex}} \beta$ if and only if the last non-zero coordinate of $\alpha - \beta$ is positive.

The following theorem provides conditions to express a monomial ordering on $K[x_1, \ldots, x_n]$ via a matrix A' and the ordering $>'_{\text{lex}}$.

Theorem 1.8. *Let $A' \in \mathcal{M}_{n\times m}(\mathbb{R})$ be a rank n matrix. Then, the ordering on M defined by*

$$x^\alpha > x^\beta \iff \alpha A' >'_{\text{lex}} \beta A'$$

is a monomial ordering if and only if the first non-zero element of each row of A' is positive.

The proof of this result can be found in [7] (Proposition 1.4, 12) with a different notation.

Example 1.9. Let us consider the degree reverse lexicographical ordering on $K[x, y, z]$. Then, the matrix A' that describes this ordering, like in the theorem above, is the following

$$A' = \begin{pmatrix} 1 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 \\ 1 & -1 & 0 & 0 \end{pmatrix}.$$

Take, for instance, $x^\alpha = x^2yz \in M$ and $x^\beta = xy^3 \in M$. Let us compare both monomials with respect to the orderings $>_{A'}$ and $>_{\text{degrlex}}$. On one hand, by definition, $x^2yz <_{\text{degrlex}} xy^3$. On the other hand, their respective exponent tuples are $\alpha = (2, 1, 1)$ and $\beta = (1, 3, 0)$. We compute $\alpha A'$ and $\beta A'$ and get:

$$\alpha A' = (2, 1, 1)A' = (4, -1, -1, -2), \qquad \beta A' = (1, 3, 0)A' = (4, 0, -3, -1).$$

Since $(4, 0, -3, -1) >_{\text{lex}} (4, -1, -1, -2)$, we get that $x^2yz >_{A'} xy^3$.

To define Gröbner bases of polynomial ideals, we need the following concepts.

Definition 1.10. Let $I \subset K[x_1, \ldots, x_n]$ be an ideal. Then, I is a monomial ideal if it can be generated by a set of monomials $S \subset K[x_1, \ldots, x_n]$.

Definition 1.11. Let $>$ be a monomial ordering on $K[x_1, \ldots, x_n]$ and $f = \sum_{i=1}^k c_i m_i \in K[x_1, \ldots, x_n]$ different than zero, where $c_i \in K$, $c_i \neq 0$, and $m_i \in M$. We define:

- $\text{lm}(f) := \max_i\{m_i\}$.
- $\text{lc}(f) := c_i$ where $m_i = \text{lm}(f)$.
- $\text{lt}(f) := \text{lc}(f)\ \text{lm}(f)$.
- $\text{tail}(f) := f - \text{lt}(f)$.

Note that all of these notions depend on the monomial ordering $>$.

Notation 1.12. Let $>$ be a monomial ordering on $K[x_1, \ldots, x_n]$ and $G \subset K[x_1, \ldots, x_n]$, we denote $\text{lm}(G) = \{\text{lm}(f) \mid 0 \neq f \in G\}$.

We observe that if $I \subseteq K[x_1, \ldots, x_n]$ is an ideal, then $\langle \text{lm}(I) \rangle$ is a monomial ideal.

We can now define Gröbner bases.

Definition 1.13. Let $>$ be a monomial ordering on $K[x_1, \ldots, x_n]$, let I be an ideal of $K[x_1, \ldots, x_n]$, and let $G \subset I$. Then, G is a Gröbner basis of I if $\langle \text{lm}(G) \rangle = \langle \text{lm}(I) \rangle$.

To compute a finite Gröbner basis of a polynomial ideal I from a finite set of generators, we use a criterion to check whether a set of polynomials is a Gröbner basis (Theorem 1.21), which allows us to develop an algorithm for the computation. First, we introduce a concept that is crucial to our approach.

Definition 1.14. A commutative unitary ring R is Noetherian if for any increasing chain of ideals of R

$$I_1 \subseteq I_2 \subseteq I_3 \subseteq \dots$$

there exists some $i_0 \in \mathbb{N}$ such that $\forall i \geq i_0$, $I_{i_0} = I_i$.

A well-known result states that a ring R is Noetherian if and only if every ideal of R is finitely generated (see Proposition 6.2 in [2]).

Thanks to Hilbert's Basis Theorem (see [4], Chapter 2, Theorem 4), we know that the polynomial ring $K[x_1, \dots, x_n]$ is Noetherian. Consequently, all of its ideals are finitely generated. In particular, for any ideal I, $\langle \mathrm{lm}(I) \rangle$ is finitely generated. This ensures the existence of a finite Gröbner basis for any polynomial ideal I. Therefore, the Noetherianity of $K[x_1, \dots, x_n]$ is crucial if we want to actually compute a finite Gröbner basis.

We now proceed with some necessary definitions.

Definition 1.15. Let $>$ be a monomial ordering on $K[x_1, \dots, x_n]$, $f, g \in K[x_1, \dots, x_n]$ where $f \neq 0$ and $g \neq 0$, and $L(f, g) := \mathrm{lcm}(\mathrm{lm}(f), \mathrm{lm}(g))$. We define the S-polynomial of f and g as:

$$\mathrm{S}(f, g) = \frac{L(f, g)}{\mathrm{lt}(f)} f - \frac{L(f, g)}{\mathrm{lt}(g)} g.$$

Since the leading monomials of f and g both divide $L(f, g)$, $\mathrm{S}(f, g)$ is actually a polynomial.

Definition 1.16. Let $>$ be a monomial ordering on $K[x_1, \dots, x_n]$. A polynomial $f \in K[x_1, \dots, x_n]$ has a Gröbner representation with respect to a subset $G \subset K[x_1, \dots, x_n]$ if $f = 0$, or if $f \neq 0$ and

$$f = \sum_{i=1}^{k} f_i g_i$$

with $\mathrm{lm}(f) \geq \mathrm{lm}(f_i)\,\mathrm{lm}(g_i)$, where $f_i \in G$ and $g_i \in K[x_1, \dots, x_n]$ for any $i \in \{1, \dots, k\}$.

To compute Gröbner representations with respect to some finite set of polynomials $G \subset K[x_1, \dots, x_n]$ (when possible), we present the following algorithm.

Algorithm 1: Reduce

Input: A finite set of polynomials $G = \{g_1, \ldots, g_k\} \subset K[x_1, \ldots, x_n]$, a monomial ordering $>$ on $K[x_1, \ldots, x_n]$, and a non-zero polynomial $p \in K[x_1, \ldots, x_n]$

Output: A set of polynomials $\{f_1, \ldots, f_k, r\} \subset K[x_1, \ldots, x_n]$ such that $\mathrm{lm}(p) \geq \mathrm{lm}(f_i)\,\mathrm{lm}(g_i)$, $p = r + \sum_{i=1}^{k} f_i g_i$, and that no monomial of r is divisible by $\mathrm{lm}(g_i)$ for any $i \in \{1, \ldots, k\}$.

$f_1, \ldots, f_k = 0$
$r = 0$
while $p \neq 0$ **do**
  if $\mathrm{lm}(g_i) \mid \mathrm{lm}(p)$ *for some* $i \in \{1, \ldots, k\}$ **then**
    $f_i \leftarrow f_i + \frac{\mathrm{lt}(p)}{\mathrm{lt}(g_i)}$
    $p \leftarrow p - \frac{\mathrm{lt}(p)}{\mathrm{lt}(g_i)} g_i$
  else
    $r \leftarrow r + \mathrm{lt}(p)$
    $p \leftarrow p - \mathrm{lt}(p)$
return $\{f_1, \ldots, f_k, r\}$

The polynomial r obtained in Algorithm 1 is called the remainder.

Remark 1.17. Algorithm 1 stops after a finite number of steps because in each iteration the leading monomial of p decreases strictly with respect to $>$, and because the monomial ordering $>$ on $K[x_1, \ldots, x_n]$ is a well-ordering on M. Moreover, if we obtain a zero remainder after the computation ($r = 0$), then p has a Gröbner representation with respect to $G = \{g_1, \ldots, g_k\}$.

Lemma 1.18. *If G is a Gröbner basis of an ideal I, then any polynomial in I has a Gröbner representation with respect to G. In particular, G generates I.*

Proof. Let $G = \{g_1, \ldots, g_k\} \subset I$ be a finite Gröbner basis of I, and let $p \in I$. Let also r be the remainder of reducing p with respect to G and suppose that $r \neq 0$. Observe that

$$r = p - \sum_{i=1}^{k} f_i g_i,$$

for some $f_i \in \mathbb{K}[x_1, \ldots, x_n]$, hence, $r \in I$. Consequently, $\mathrm{lm}(r)$ belongs to $\langle \mathrm{lm}(g_1), \ldots, \mathrm{lm}(g_k) \rangle$ because G is a Gröbner basis and so, $\mathrm{lm}(r) \mid \mathrm{lm}(g_i)$ for some $i \in \{1, \ldots, k\}$, which is a contradiction. Therefore, reducing p with respect to G yields a zero remainder; furthermore, a Gröbner representation of p with respect to G. □

We now prove some useful results leading to the characterization of Gröbner bases via Buchberger's Criterion.

Proposition 1.19 (Product criterion)**.** *Let $>$ be a monomial ordering on $K[x_1, \ldots, x_n]$ and $f, g \in K[x_1, \ldots, x_n]$. If $\gcd(\mathrm{lm}(f), \mathrm{lm}(g)) = 1$, then $\mathrm{S}(f, g)$ has a Gröbner representation with respect to $\{f, g\}$.*

Proof. If $\mathrm{S}(f,g) = 0$, it is trivial by definition. Let us suppose $\mathrm{S}(f,g) \neq 0$. Since $\gcd(\mathrm{lm}(f), \mathrm{lm}(g)) = 1$, $\mathrm{lcm}(\mathrm{lm}(f), \mathrm{lm}(g)) = \mathrm{lm}(f)\ \mathrm{lm}(g)$, and therefore:

$$\mathrm{S}(f,g) = \frac{\mathrm{lm}(f)\ \mathrm{lm}(g)}{\mathrm{lt}(f)} f - \frac{\mathrm{lm}(f)\ \mathrm{lm}(g)}{\mathrm{lt}(g)} g = \frac{\mathrm{lm}(g)}{\mathrm{lc}(f)} f - \frac{\mathrm{lm}(f)}{\mathrm{lc}(g)} g. \tag{1}$$

Denoting $c = \mathrm{lc}(f)\,\mathrm{lc}(g)$, we get that $c\ \mathrm{S}(f,g) = f\ \mathrm{lt}(g) - g\ \mathrm{lt}(f)$.
We now observe that

$$0 = fg - gf = f\ \mathrm{lt}(g) + f\ \mathrm{tail}(g) - g\,\mathrm{lt}(f) - g\,\mathrm{tail}(f) = \\ = c\ \mathrm{S}(f,g) + f\ \mathrm{tail}(g) - g\ \mathrm{tail}(f).$$

so

$$c\ \mathrm{S}(f,g) = g\ \mathrm{tail}(f) - f\ \mathrm{tail}(g)$$

We prove that this is a Gröbner representation of $c\ \mathrm{S}(f,g)$. To this end, we first prove that the leading terms of $g\ \mathrm{tail}(f)$ and $f\ \mathrm{tail}(g)$ do not vanish in $g\ \mathrm{tail}(f) - f\ \mathrm{tail}(g)$. Suppose they do, then

$$\mathrm{lm}(g)\,\mathrm{lm}(\mathrm{tail}(f)) = \mathrm{lm}(f)\,\mathrm{lm}(\mathrm{tail}(g))$$

which means that $\mathrm{lm}(g) \mid \mathrm{lm}(f)\,\mathrm{lm}(\mathrm{tail}(g))$. Since $\mathrm{lcm}(\mathrm{lm}(g), \mathrm{lm}(f)) = 1$, then $\mathrm{lm}(g) \mid \mathrm{lm}(\mathrm{tail}(g))$. This implies that $\mathrm{lm}(g) \leq \mathrm{lm}(\mathrm{tail}(g))$, which is a contradiction. Therefore,

$$\mathrm{lt}(g)\,\mathrm{lt}(\mathrm{tail}(f)) \neq \mathrm{lt}(f)\,\mathrm{lt}(\mathrm{tail}(g)).$$

This means that $\mathrm{lm}(g\ \mathrm{tail}(f) - f\ \mathrm{tail}(g)) = \max\{\mathrm{lm}(g\ \mathrm{tail}(f)), f\ \mathrm{tail}(g)\}$. Finally, we observe that:

$$\mathrm{lm}(S(f,g)) = \mathrm{lm}(g\ \mathrm{tail}(f) - f\ \mathrm{tail}(g)) = \\ = \max\{\mathrm{lm}(g\ \mathrm{tail}(f)), \mathrm{lm}(f\ \mathrm{tail}(g))\} \geq \mathrm{lm}(g)\ \mathrm{lm}(\mathrm{tail}(f)), \mathrm{lm}(f)\ \mathrm{lm}(\mathrm{tail}(g))$$

So $c\ \mathrm{S}(f,g)$ has a Gröbner representation with respect to $\{f,g\}$, hence $\mathrm{S}(f,g)$ as well. □

Lemma 1.20. *Let $>$ be a monomial ordering on $K[x_1,\ldots,x_n]$, $q_1,\ldots,q_s \in K[x_1,\ldots,x_n]$ be non-zero polynomials such that $\mathrm{lm}(q_i) = m \in M\ \forall i \in \{1,\ldots,s\}$ and some $c_1,\ldots,c_s \in K$. Then, $\mathrm{lm}(\mathrm{S}(q_i,q_{i+1})) < m$ for $i \in \{1,\ldots,s-1\}$ such that $\mathrm{S}(q_i,q_{i+1}) \neq 0$. Moreover, if $\mathrm{lm}\left(\sum_{i=1}^{s} c_i\,q_i\right) < m$, then $\sum_{i=1}^{s} c_i\,q_i$ is a K-linear combination of the S-polynomials $\mathrm{S}(q_i,q_{i+1})$ for $i = 1,\ldots,s-1$.*

Proof. The first part follows from the fact that $\mathrm{lm}(q_i) = m$, $\forall i \in \{1,\ldots,s\}$ and by definition of the S-polynomials. For the second part, we write each q_i as

$$q_i = a_i\,m + \text{ terms with monomials lower than } m\ ,$$

with $a_i \in K$. Since $\mathrm{lm}\left(\sum_{i=1}^{s} c_i\,q_i\right) < m$, we have that the sum $c_1\,a_1 + c_2\,a_2 + \ldots + c_s\,a_s = 0$. Note that the S-polynomial of q_i and q_{i+1} with $i \in \{1,\ldots,s-1\}$ is of the form $\mathrm{S}(q_i,q_{i+1}) = \frac{1}{a_i}\,q_i - \frac{1}{a_{i+1}}\,q_{i+1}$ since $\mathrm{lm}(q_i) = m\ \forall i \in \{1,\ldots,s\}$. We observe that

$$c_1\,q_1 + c_2\,q_2 + \ldots + c_s\,q_s = c_1\,a_1\left(\frac{1}{a_1}\right)q_1 + c_2\,a_2\left(\frac{1}{a_2}\right)q_2 + \ldots + c_s\,a_s\left(\frac{1}{a_s}\right)q_s.$$

After adding and substracting $\sum_{i=1}^{s-1}\left(\sum_{j=1}^{i} c_j\, a_j\right)\frac{1}{a_{i+1}}\, q_{i+1}$, and reordering the previous expression, we have

$$
\begin{aligned}
& c_1\, a_1\left(\frac{1}{a_1}\, q_1 - \frac{1}{a_2}\, q_2\right) + \\
& (c_1\, a_1 + c_2\, a_2)\left(\frac{1}{a_2}\, q_2 - \frac{1}{a_3}\, q_3\right) + \\
& (c_1\, a_1 + c_2\, a_2 + c_3\, a_3)\left(\frac{1}{a_3}\, q_3 - \frac{1}{a_4}\, q_4\right) + \\
& \vdots \\
& (c_1\, a_1 + \ldots + c_{s-1}\, a_{s-1})\left(\frac{1}{a_{s-1}}\, q_{s-1} - \frac{1}{a_s}\, q_s\right) + \\
& (c_1\, a_1 + \ldots + c_s\, a_s)\frac{1}{a_s}\, q_s = \sum_{i=1}^{s-1}\left(\sum_{j=1}^{i} c_j\, a_j\right)\mathrm{S}(q_i, q_{i+1}).
\end{aligned}
$$

The last equality holds because $c_1\, a_1 + c_2\, a_2 + \ldots + c_s\, a_s = 0$. □

Theorem 1.21 (Buchberger's criterion)**.** *Let $I \subseteq K[x_1, \ldots, x_n]$ be an ideal, $>$ be a monomial ordering on $K[x_1, \ldots, x_n]$ and $G = \{q_1, \ldots, q_s\}$ a set that generates I, where $q_i \neq 0$ for $i = 1, \ldots, s$. Then, G is a Gröbner basis of I if and only if for any different $q_i, q_j \in G$ such that $\gcd(\mathrm{lm}(q_i), \mathrm{lm}(q_j)) \neq 1$, $\mathrm{S}(q_i, q_j)$ has a Gröbner representation with respect to G.*

Proof.
⇒) Suppose that G is a Gröbner basis. Since $\mathrm{S}(q_i, q_j) \in I\ \forall i, j \in \{1, \ldots, s\}$ with $i \neq j$, there exists a Gröbner representation for each one of them with respect to G by Lemma 1.18.
⇐) By the Product criterion 1.19, we can assume that every S-polynomial $\mathrm{S}(q_i, q_j)$ has a Gröbner representation with respect to G. Since $G \subset I$, we have that $\langle \mathrm{lm}(G)\rangle \subset \langle \mathrm{lm}(I)\rangle$.

To prove the other inclusion, consider $p \in I$ different than zero, we prove that $\mathrm{lm}(p) \in \langle \mathrm{lm}(G)\rangle$. Let us choose $f_i \in K[x_1, \ldots, x_n]$ non-zero such that

$$p = \sum_{i=1}^{k} f_i q_i$$

where $k \leq s$ and that minimizes $m = \max_i\{\mathrm{lm}(f_i)\ \mathrm{lm}(q_i)\}$. This combination exists because $>$ is well-ordered. Observe that $\mathrm{lm}(p) \leq m$.

Let us prove that $\mathrm{lm}(p) = m$. Suppose $\mathrm{lm}(p) < m$. Denoting $m_i = \mathrm{lm}(f_i)\,\mathrm{lm}(q_i)$, we get that

$$
\begin{aligned}
p = \sum_{i=1}^{k} f_i\, q_i &= \sum_{i\geq 1:\, m_i < m} f_i\, q_i + \sum_{i\geq 1: m_i = m} f_i\, q_i \\
&= \sum_{i\geq 1: m_i < m} f_i\, q_i + \sum_{i\geq 1: m_i = m} \mathrm{lt}(f_i)\, q_i + \sum_{i\geq 1: m_i = m} \left(f_i\, q_i - \mathrm{lt}(f_i)\, q_i\right). \qquad (2)
\end{aligned}
$$

By hypothesis $\mathrm{lm}(p) < m$, hence $\sum_{i\geq 1: m_i = m} \mathrm{lt}(f_i)\, q_i < m$. We consider now $A = \{i \in \{1,\ldots,k\} \mid m_i = m\}$, and label it $A = \{i_1, \ldots, i_l\}$. By the previous Lemma 1.20 we can write

$$\sum_{k=1}^{l} \mathrm{lt}(f_{i_k})\, q_{i_k} = \sum_{k=1}^{l-1} c_{i_k}\, \mathrm{S}(\mathrm{lt}(f_{i_k})\, q_{i_k}, \mathrm{lt}(f_{i_k+1})\, q_{i_{k+1}})$$

with $c_{i_k} \in K$, because $m_{i_k} = m$ for all $k \in \{1,\ldots,l-1\}$. Moreover, thanks to the Lemma 1.20 we get that

$$\mathrm{lm}(\mathrm{S}(\mathrm{lt}(f_{i_k})\, q_{i_k}, \mathrm{lt}(f_{i_{k+1}})\, q_{i_{k+1}})) < m$$

for $k \in C := \{k \in \{1\ldots, l-1\} : \mathrm{S}(\mathrm{lt}(f_{i_k})\, q_{i_k},\, \mathrm{lt}(f_{i_{k+1}})\, q_{i_{k+1}}) \neq 0\}$.

Note that the S-polynomials of $\mathrm{lt}(f_{i_k})\, q_{i_k}$ and $\mathrm{lt}(f_{i_{k+1}})\, q_{i_{k+1}}$ are as follows:

$$\frac{m}{\mathrm{lt}(f_{i_k})\,\mathrm{lt}(q_{i_k})}\ \mathrm{lt}(f_{i_k})\, q_{i_k} - \frac{m}{\mathrm{lt}(f_{i_{k+1}})\ \mathrm{lt}(q_{i_{k+1}})}\ \mathrm{lt}(f_{i_{k+1}})\, q_{i_{k+1}}.$$

If we multiply and divide by $x^{\gamma_k} = \mathrm{lcm}(\mathrm{lm}(q_{i_k}), \mathrm{lm}(q_{i_{k+1}}))$ we get:

$$\begin{aligned}\mathrm{S}(\mathrm{lt}(f_{i_k})\, q_{i_k}, \mathrm{lt}(f_{i_{k+1}})\, q_{i_{k+1}}) &= m\, x^{-\gamma_k} \left(\frac{x^{\gamma_k}}{\mathrm{lt}(q_{i_k})}\, q_{i_k} - \frac{x^{\gamma_k}}{\mathrm{lt}(q_{i_{k+1}})}\, q_{i_{k+1}} \right) \\ &= m\, x^{-\gamma_k}\, \mathrm{S}(q_{i_k}, q_{i_{k+1}}),\end{aligned}$$

where $x^{\gamma_k} \mid m$ by definition of m. We observe that $\mathrm{lm}(m\, x^{-\gamma_k}\, \mathrm{S}(q_{i_k}, q_{i_{k+1}})) < m$ because $\mathrm{lm}(\mathrm{S}(\mathrm{lt}(f_{i_k})\, q_{i_k}, \mathrm{lt}(f_{i_{k+1}})\, q_{i_{k+1}})) < m$.

By hypothesis, there exists a Gröbner representation of $\mathrm{S}(q_{i_k}, q_{i_{k+1}})$ with respect to G, hence:

$$\mathrm{S}(q_{i_k}, q_{i_{k+1}}) = \sum_{j \in D_k} g_{j,k}\, q_j,$$

where $D_k = \{j \in \{1,\ldots,s\} : g_{j,k} \neq 0\}$ and $g_{j,k} \in K[x_1,\ldots,x_n]$ such that $\mathrm{lm}(g_{j,k}\, q_j) \leq \mathrm{lm}(\mathrm{S}(q_{i_k}, q_{i_{k+1}}))$ for every $j \in D_k$ and $k \in C$. Consequently, we get the following equalities:

$$\begin{aligned}\sum_{i\in A} \mathrm{lt}(f_i)\, q_i &= \sum_{k\in C} c_{i_k}\, \mathrm{S}(\mathrm{lt}(f_{i_k})\, q_{i_k}, \mathrm{lt}(f_{i_{k+1}})\, q_{i_{k+1}}) \\ &= \sum_{k\in C} c_{i_k} m\, x^{-\gamma_k}\, \mathrm{S}(q_{i_k}, q_{i_{k+1}}) = \sum_{k\in C} c_{i_k} m\, x^{-\gamma_k} \sum_{j\in D_k} g_{j,k}\, q_j \\ &= \sum_{k\in C} c_{i_k} m\, x^{-\gamma_k} \sum_{j=1}^{s} g_{j,k}\, q_j = \sum_{j=1}^{s} q_j \left(\sum_{k\in C} c_{i_k} m\, x^{-\gamma_k}\, g_{j,k} \right) \\ &= \sum_{j=1}^{s} q_j \left(\sum_{k\in C^j} c_{i_k} m\, x^{-\gamma_k}\, g_{j,k} \right) = \sum_{j\in D} q_j h_j,\end{aligned}$$

where $C^j = \{k \in C : g_{j,k} \neq 0\}$, $D = \{j \in \{1,\ldots,s\} : h_j \neq 0\}$ and $h_j = \sum_{k\in C^j} c_{i_k} m\, x^{-\gamma_k}\, g_{j,k} \in K[x_1,\ldots,x_n]$.

Let us prove that each polynomial $q_j h_j$, for $j \in D$, has a leading monomial strictly smaller than m. Observe that

$$\mathrm{lm}(q_j h_j) = \mathrm{lm}\left(q_j \sum_{k\in C^j} c_{i_k} m\, x^{-\gamma_k}\, g_{j,k} \right) \leq \max_{k\in C^j} (\mathrm{lm}(c_{i_k}\, m\, x^{-\gamma_k}\, q_j\, g_{j,k})).$$

Since $\mathrm{lm}(g_{j,k}\, q_j) \leq \mathrm{lm}(\mathrm{S}(q_{i_k}, q_{i_{k+1}}))$ for every $j \in D_k$ and $k \in C$, the same inequality holds for every $k \in C^j$ where $j \in D$. Therefore:

$$\mathrm{lm}(q_j h_j) \leq \max_{k \in C^j}(\mathrm{lm}(\mathrm{S}(m\, x^{-\gamma_k} q_{i_k}, m\, x^{-\gamma_k} q_{i_{k+1}}))) = \\ = \max_{k \in C^j}(\mathrm{lm}(\mathrm{S}(\mathrm{lt}(f_{i_k})\, q_{i_k}, \mathrm{lt}(f_{i_{k+1}})\, q_{i_{k+1}}))) < m$$

for every $j \in D_k$ and $k \in C$.

Returning to (2) and replacing $\sum_{i \in A} \mathrm{lt}(f_i q_i)$ with $\sum_{j \in D} q_j h_j$, we obtain an expression of p as combination of polynomials in $\{q_1, \ldots, q_s\}$ whose leading terms are strictly less than m. This contradicts the minimality of m. Hence $\mathrm{lm}(p) = m \in \langle \mathrm{lm}(G) \rangle$. □

This characterization allows us to introduce the following algorithm for computing a Gröbner basis.

Algorithm 2: Buchberger's algorithm

Input: A polynomial ideal $I = \langle q_1, \ldots, q_s \rangle \subset K[x_1, \ldots, x_n]$, and a monomial ordering $>$ in $K[x_1, \ldots, x_n]$.

Output: A Gröbner basis of $I = \langle q_1, \ldots, q_s \rangle$.

```
1  G = {q_1, ..., q_s}
2  S = {S(q_i, q_j) | q_i ≠ q_j and q_i, q_j ∈ G}
3  G' := G
4  repeat
5  |  G ← G'
6  |  for s ∈ S do
7  |  |  r ← Reduce'(s,>,G)
8  |  |  if r ≠ 0 then
9  |  |  |  G' ← G' ∪ {r}
10 |  S ← {S(p, q) | p ≠ q, and p, q ∈ G'}
11 until G = G'
12 return G
```

The method "Reduce'$(s,>,G)$" does the same thing as Algorithm 1, but instead of returning the whole set $\{f_1, \ldots, f_k, r\}$ it only returns the remainder r. The algorithm returns a set $G \subset I$ such that it generates I, and for any $f, g \in G$, the remainder of reducing $\mathrm{S}(f, g)$ with respect to G is zero. Thus, by Buchberger's Criterion, it returns a Gröbner Basis.

Proposition 1.22. *Buchberger's Algorithm (Algorithm 2) stops after a finite number of steps.*

Proof. For it to stop at some point, there must be an iteration within the **repeat** in which all the reminders "r" are zero. We suppose we can obtain new non-zero reminders all the time (equivalently, that the algorithm does not finish). Then, denoting G_i the set "G" in the i-th iteration inside the **repeat**, by the reduction Algorithm, for any $i \geq 0$, there exist some $r_i \in I$ such that $\mathrm{lm}(r_i) \notin \langle \mathrm{lm}(G_i) \rangle$ and $\mathrm{lm}(r_i) \in \langle \mathrm{lm}(G_{i+1}) \rangle$. Hence, we have a strictly

increasing chain

$$\langle \mathrm{lm}(G_0)\rangle \subsetneq \langle \mathrm{lm}(G_1)\rangle \subsetneq \langle \mathrm{lm}(G_2)\rangle \subsetneq \dots$$

But this contradicts the Noetherianity of $K[x_1, \dots, x_n]$, therefore the algorithm stops after a finite number of steps. □

Remark 1.23 (Memebership problem)**.** Without this theoretical framework, determining whether a polynomial $p \in K[x_1, \dots, x_n]$ belongs to an ideal $I = \langle f_1 \dots f_l\rangle$, might not be an easy task. Fortunately, with the theory developed above, we proceed as follows:

(1) We fix any monomial ordering $>$, and compute a Gröbner Basis of I with respect to $>$. Let $G = \{g_1, \dots, g_s\}$ be such basis.
(2) We reduce p with respect to G and the monomial ordering $>$. Since G is a Gröbner basis, $p \in \langle G\rangle = I$ if and only if the remainder of the reduction is zero.

Let us finish this section by highlighting the importance of the Noetherianity of $K[x_1, \dots, x_n]$ for the computation of a finite Gröbner basis. It ensures, on the one hand, the termination of the proposed algorithm and, on the other hand, guarantees the existence of a finite Gröbner basis (as mentioned earlier).

At the end of Section 3, we present the computation of some kind of finite bases with special properties analogous to those of a Gröbner basis, but within a different setting where the Noetherianity is no longer guaranteed. Consequently, we require alternative tools to compute these bases.

2. Polynomial Difference Algebras

In this section, we define and specify some structures so that, later on, we can develop a Gröbner basis theory on them. We start by defining what a Σ-algebra is. For the remainder of this text, we denote by $\Sigma = \langle \sigma_1, \dots, \sigma_r\rangle$ the free abelian monoid of rank r generated by $\{\sigma_1, \dots, \sigma_r\}$. For its operation, we use the additive notation.

Notation 2.1. Let R be a ring. We denote by $\mathrm{End}(R)$ the monoid of endomorphisms of R given by the composition operation, that is, $(\mathrm{End}(R), \circ)$.

Definition 2.2. Let K be a field. Then, K is a Σ-field if there exists a monoid homomorphism

$$\rho\colon \Sigma \longrightarrow \mathrm{End}(K).$$

This monoid homomorphism induces the following action of Σ on K:

$$\begin{aligned} K \times \Sigma &\longrightarrow K \\ (c, \sigma) &\longrightarrow \sigma \cdot c := \rho(\sigma)(c) \end{aligned}$$

Definition 2.3. Let K be a Σ-field, and A be a K-algebra. Then, A is a Σ-algebra if there is a monoid homomorphism

$$\rho'\colon \Sigma \longrightarrow \mathrm{End}(A)$$

such that it extends $\rho\colon \Sigma \to \mathrm{End}(K)$, meaning that $\rho'(\sigma)(c) = \rho(\sigma)(c)$ for $c \in K$ and $\sigma \in \Sigma$. As before, this monoid homomorphism induces an action of Σ on A:

$$\begin{aligned} A \times \Sigma &\longrightarrow A \\ (a, \sigma) &\longrightarrow \sigma \cdot a := \rho'(\sigma)(a). \end{aligned}$$

Since ρ' extends ρ, the action induced by ρ' extends the action induced by ρ. Hence, we use the same notation for both of them.

Definition 2.4. Let A, B be two Σ-algebras and let $\varphi\colon A \to B$ be a K-algebra homomorphism. Then, φ is a Σ-algebra homomorphism if for any $\sigma \in \Sigma$, $a \in A$, we have that

$$\varphi(\sigma \cdot a) = \sigma \cdot \varphi(a).$$

After these definitions, we can consider the category of Σ-algebras denoted by $\mathcal{C}_\Sigma$, where the objects are Σ-algebras and morphisms are Σ-algebra homomorphisms.

Because the standard Gröbner basis theory works over polynomial rings $K[x_1, \ldots, x_n]$ where K is a field and $n \in \mathbb{N}$, we aim to develop a similar theory for Σ-algebras with a comparable "polynomial ring"-kind structure. This motivates the following definitions.

Notation 2.5. Let $\cdot : A \times \Sigma \to A$ be a monoid action on a set A. If $X \subseteq A$, we denote $\Sigma \cdot X := \{\sigma \cdot x \mid \sigma \in \Sigma, x \in X\}$.

Definition 2.6. Let K be a Σ-field, and $K \hookrightarrow L$ be a field extension. If L is also a Σ-field and there is compatibility between the action of Σ on L and the action of Σ on K, then, K is a sub-Σ-field of L, or equivalently, L is a Σ-field extension of K.

Definition 2.7. Let $K \hookrightarrow L$ be a Σ-field extension, and X be a subset of L. Then, X is Σ-algebraically independent over K if for any distinct $\tau_1 \cdot x_1, \ldots, \tau_k \cdot x_k \in \Sigma \cdot X$ where $\tau_1, \ldots, \tau_k \in \Sigma$, there is no non-zero polynomial p with coefficients in K such that $p(\tau_1 \cdot x_1, \ldots, \tau_k \cdot x_k) = 0$. In this case, we denote $x(\sigma) := \sigma \cdot x$ and $X(\Sigma) := \Sigma \cdot X$.

Definition 2.8. Let L be a Σ-field extension of K, and $X = \{x_1, \ldots, x_n\} \subset L$ be a finite Σ-algebraically independent set over K. Then, we call $P = K[X(\Sigma)]$ a Σ-polynomial algebra in the Σ-indeterminates (or variables) X.

In [8], these kinds of structures with an action by a free monoid Σ (Σ-fields, Σ-algebras, Σ-polynomial algebras...) are also known as partial difference structures (partial difference fields, partial difference algebras, etc...). However, we reserve this terminology for specific instances of Σ-algebras, which are introduced in Example 2.14.

Proposition 2.9. *The Σ-polynomial algebra P is a Σ-algebra.*

Proof. We consider the monoid homomorphism between Σ and $\mathrm{End}(P)$ given by

$$\begin{aligned} \rho'\colon \Sigma &\longrightarrow \mathrm{End}(P) \\ \sigma &\longrightarrow \overline{\sigma} \end{aligned}$$

where, given $c \in K$ and $\tau \in \Sigma$,

$$\begin{aligned} \overline{\sigma}\colon P &\longrightarrow P \\ c\,x(\tau) &\longrightarrow (\sigma \cdot c)x(\sigma + \tau) \end{aligned}$$

is defined over the variables, and extended to polynomials in P. In this way, by definition, $\overline{\sigma}$ preserves sums and products in P. Hence $\overline{\sigma} \in \mathrm{End}(P)$, and

ρ' is well-defined. Also by definition, ρ' extends $\rho\colon \Sigma \to \mathrm{End}(K)$ because $\rho'(\sigma)(c) = \overline{\sigma}(c) = \sigma \cdot c = \rho(\sigma)(c)$ for every $c \in K$. We now prove that it is a monoid homomorphism. Let $\sigma_1, \sigma_2 \in \Sigma$. Then,

$$\begin{aligned}\overline{\sigma_1 + \sigma_2}(cx(\tau)) &= ((\sigma_1 + \sigma_2) \cdot c)x((\sigma_1 + \sigma_2) + \tau) \\ &= (\rho(\sigma_1 + \sigma_2)(c))x(\sigma_1 + (\sigma_2 + \tau)) \\ &= (\rho(\sigma_1) \circ \rho(\sigma_2)(c))x(\sigma_1 + (\sigma_2 + \tau)) \\ &= \overline{\sigma_1} \circ \overline{\sigma_2}(cx(\tau)).\end{aligned}$$

□

We prove now that it is a free object in the category of Σ-algebras. We first recall the definition of a free object.

Definition 2.10. Let $\mathcal{C}$ be a category, and X be a set. Then, $F(X)$ is the free object of $\mathcal{C}$ generated by X if there exists an map $h\colon X \to F(X)$ such that for any $A \in \mathrm{Obj}(\mathcal{C})$ and any $f\colon X \to A$ map, there exists a unique morphism of $\mathcal{C}$

$$g\colon F(X) \to A$$

such that $g \circ h = f$.

Proposition 2.11. *The Σ-polynomial algebra $P = K[X(\Sigma)]$ is the free object of $\mathcal{C}_\Sigma$ generated by X.*

Proof. Let A be an Σ-algebra and ρ_A be the monoid homomorphism that induces the action of Σ on A. Consider as well the map

$$\begin{aligned} h\colon X &\longrightarrow P \\ x &\longmapsto x(0). \end{aligned}$$

We give the unique Σ-algebra homomorphism. Let $f\colon X \to A$ be a map. We define the K-algebra homomorphism

$$\begin{aligned} g\colon P &\longrightarrow A \\ x(\sigma) &\longmapsto \sigma \cdot f(x). \end{aligned}$$

Observe that it is a homomorphism because it is defined in each variable, and extends to polynomials in P preserving multiplication and addition. We prove it commutes with the action by Σ: let $\sigma, \tau \in \Sigma$, then

$$g(\sigma \cdot cx(\tau)) = g((\sigma \cdot c)x(\sigma + \tau)) = (\sigma \cdot c)((\sigma + \tau) \cdot f(x)) =$$
$$(\rho'_A(\sigma)(c))(\rho'_A(\sigma) \circ \rho'_A(\tau)(f(x))) = \sigma \cdot c(\tau \cdot f(x)) = \sigma \cdot (cg(x(\tau))) = \sigma \cdot g(cx(\tau)).$$

Note that for any $x \in X$, $f(x) = 0 \cdot f(x) = g(x(0)) = g(h(x))$. Therefore, g is a Σ-algebra homomorphism such that $g \circ h = f$. To show the uniqueness, suppose that there is another Σ-algebra homomorphism $\psi\colon P \to A$ such that $\psi(h(x)) = f(x)$ for all $x \in X$. Then, as $\sigma \cdot x(0) = x(\sigma)$ for every $\sigma \in \Sigma$ and $x \in X$; we have that:

$$\psi(x(\sigma)) = \psi(\sigma \cdot x(0)) = \sigma \cdot \psi(x(0)) = \sigma \cdot \psi(h(x)) = \sigma \cdot f(x)$$

for every $x(\sigma) \in X(\Sigma)$. Hence $g = \psi$. □

To develop a Gröbner basis theory for Σ-polynomial algebras, we introduce sub-structures that play the same role as ideals in the standard polynomial ring case.

Definition 2.12. Let A be a Σ-algebra. Then,

- I is a Σ-ideal of A if it is an ideal of A invariant with respect to the action of Σ on A. That is,
$$a \in I \iff \sigma \cdot a \in I \quad \forall \sigma \in \Sigma.$$
Let $X \subseteq A$ be a subset of A such that I is generated by the set $\Sigma \cdot X$, then, X Σ-generates I and we denote it by $I = \langle X \rangle_\Sigma$.
- B is a Σ-subalgebra of A if it is a K-subalgebra of A invariant with respect to the action of Σ on A. Let $X \subseteq A$ be a subset of A such that $\Sigma \cdot X$ generates B as a K-algebra, then, X Σ-generates the Σ-subalgebra B.

Let us give some examples of Σ-polynomial algebras.

Example 2.13. Let $\Sigma = \mathbb{N}$, and consider the fraction field $\mathbb{C}(z)$. Both $\mathbb{C}$ and $\mathbb{C}(z)$ are $\mathbb{N}$-fields considering the monoid homomorphism
$$\begin{aligned} \rho \colon \mathbb{N} &\longrightarrow \mathrm{End}(\mathbb{C}(z)) \\ n &\longmapsto \overline{n} \end{aligned}$$
where $\overline{n} \colon \mathbb{C}(z) \to \mathbb{C}(z)$ is defined by $\overline{n}(f(z)) := f(z+n)$, for each $f(z) \in \mathbb{C}(z)$. We observe that the restricted action given by ρ to $\mathbb{C}$ is the trivial one, that is, $\overline{n}(c) = c$ for every $c \in \mathbb{C}$ and $n \in \mathbb{N}$. Hence, $\mathbb{N} \cdot \mathbb{C} = \mathbb{C}$, and therefore, $\mathbb{C} \hookrightarrow \mathbb{C}(z)$ is a Σ-field extension. We fix now $x \in \mathbb{C}(z)$ to be a Σ-algebraically independent rational function over $\mathbb{C}$, for instance $\frac{1}{z}$. Then, we can consider the Σ-polynomial algebra $P = \mathbb{C}[X(\mathbb{N})]$ where $X = \{x\}$. The variables of P are $x(n)$ for every $n \in \mathbb{N}$ and we write $x(n)(z) = \frac{1}{z+n}$. For instance, an element of P would be
$$(1+i)x(2)^3x(5) - \frac{1}{1-2i}x(3)^4 \in P.$$

The following example is especially interesting because it introduces concrete Σ-algebras that appear in Section 4.

Example 2.14. Let $K = \mathbb{Q}(t_1, \ldots, t_r)$ be a fraction field over $\mathbb{Q}$, $h_1, \ldots, h_r \in \mathbb{Q}$ be some fixed elements and $(\mathbb{N}^r, +)$ be a free abelian monoid. Consider the following monoid homomorphism:
$$\begin{aligned} \rho \colon \mathbb{N}^r &\longrightarrow \mathrm{End}(K) \\ \alpha &\longmapsto \overline{\alpha} \end{aligned}$$
where, if $\alpha = (\alpha_1, \ldots, \alpha_r)$, then, $\overline{\alpha}$ is defined as follows
$$\begin{aligned} \overline{\sigma} \colon K &\longrightarrow K \\ f(t_1, \ldots, t_r) &\longmapsto \sigma \cdot f := f(t_1 + \alpha_1 h_1, \ldots, t_r + \alpha_r h_r). \end{aligned}$$
That is, the action induced by ρ is to shift each variable t_i of $f(t_1, \ldots, t_r) \in K$, some amount $\alpha_i\, h_i$, which depends on $\alpha = (\alpha_1, \ldots, \alpha_r)$ and on the previously fixed $h_1, \ldots, h_r \in \mathbb{Q}$. Each $h_i \in \mathbb{Q}$ is the basic shifting unit of each i-th component of f, and it is denoted "mesh step". Furthermore, α specifies how many times we shift each h_i. Considering this action, K is a Σ-field.

Moreover, let us fix $V = \{v_1, \ldots, v_n\}$ a finite set of elements in a Σ-field extension L of K, such that V is Σ-algebraically independent over K. Then, $\tilde{P} = K[V(\Sigma)]$ is a Σ-polynomial algebra, and is denoted difference algebra in the variables $v_1, \ldots, v_n$ with mesh steps $h_1, \ldots, h_r$.

3. Gröbner Bases of Σ-Ideals

In this section, we develop a Gröbner basis theory in the Σ-polynomial algebras $P = K[X(\Sigma)]$. To do so, we first introduce monomial orders compatible with the Σ-ideals defined in the previous section.

3.1. **Monomial Orders and Σ-Orders.** In what follows, let $P = K[X(\Sigma)]$ be a Σ-polynomial algebra, with $X = \{x_1, \ldots, x_n\}$ and $\Sigma = \langle \sigma_1, \ldots, \sigma_r \rangle$ an abelian free monoid. Moreover, we denote by M the set of monomials of P (including 1). In this part, we define monomial orders and Σ-monomial orders on P and give a way to express some of them via infinite matrices.

Definition 3.1. A monomial ordering on P is a well-ordering $\succ$ on M that is also multiplicative, that is,

$$m \succ m' \implies l\, m \succ l\, m' \qquad \forall l, m, m' \in M.$$

Notation 3.2. Occasionally, given a collection of distinct variables $x_{i_1}(\tau_1)$, $x_{i_2}(\tau_2), \ldots, x_{i_k}(\tau_k) \in X(\Sigma)$, we simplify the notation to $x(\tau)_1, x(\tau)_2, \ldots, x(\tau)_k$. This notation emphasizes that each $x(\tau)_j$ is an individual variable, with the outer index serving to distinguish it from the rest. Because the variables are different, the elements $\tau \in \Sigma$ and $x \in X$ do not need to be the same for each $x(\tau)_j$.

Definition 3.3. Let $m = x(\tau)_1^{\alpha_1}\, x(\tau)_2^{\alpha_2} \ldots x(\tau)_k^{\alpha_k} \in M$ be a monomial of P. The support of m, denoted by $\operatorname{supp}(m)$, is defined as:

$$\operatorname{supp}(m) = \{x(\sigma) \in X(\Sigma) \text{ such that } x(\sigma) \mid m\} \subset X(\Sigma).$$

We define the degree of m as $\deg(m) = \sum_{i=1}^{k} \alpha_i$.

Notation 3.4. Let $>$ be a monomial ordering on $K[y_1, \ldots, y_r]$. Then, $>$ induces an additive well-ordering on Σ as follows:

$$\sigma = \sum_{i=1}^{r} \alpha_i \sigma_i > \tau = \sum_{i=1}^{r} \beta_i \sigma_i \iff \prod_{i=1}^{r} y_i^{\alpha_i} > \prod_{i=1}^{r} y_i^{\beta_i}$$

where $\alpha_i, \beta_i \in \mathbb{N}$ for all $i \in \{1, \ldots, r\}$. In fact, this definition establishes a one-to-one correspondence between monomial orderings on $K[y_1, \ldots, y_r]$ and additive well-orderings on Σ. Throughout this section, we denote both orders with the same symbol.

Next, we show how to introduce well orderings on $X(\Sigma)$, and we introduce an order relation that will be useful later on.

Definition 3.5. Let $>$ be any monomial ordering on $Q = K[\sigma_1, \ldots, \sigma_r]$ and $\triangleright$ be any total order relation on X. We define the ordering $\succ$ on $X(\Sigma)$ induced by $>$ and $\triangleright$ as follows:

$$x(\sigma) \succ y(\tau) \iff \begin{cases} \sigma > \tau \\ \sigma = \tau \text{ and } x \triangleright y. \end{cases}$$

Note that $\succ$ is a well ordering in $X(\Sigma)$ because $>$ and $\triangleright$ are so in Q and X respectively.

Notation 3.6. We denote by $>_u$ the usual order relation on $\mathbb{N}$ or on $\mathbb{R}$, depending on the context. We also denote $\mathbb{N}^\infty := \bigoplus_{i\in\mathbb{N}} \mathbb{N}$, and $\mathbb{R}^\infty := \bigoplus_{i\in\mathbb{N}} \mathbb{R}$.

Definition 3.7. Let $v, w \in \mathbb{R}^\infty$, we denote by $\succeq_u$ the relation in $\mathbb{R}^\infty$ given as follows:

$$v \succeq_u w \iff \begin{cases} \text{the last non-zero coordinate of } v - w \text{ is positive or} \\ v = w. \end{cases}$$

Proposition 3.8. *The relation $\succeq_u$ is an additive total ordering on $\mathbb{R}^\infty$*

Proof. We show that the relation is reflexive, antisymmetric, transitive, total, and additive.

- Reflexive: If $v = v$, by definition $v \succeq_u v$.
- Antisymmetric: Suppose that $v \succeq_u w$ and $w \succeq_u v$. Suppose also that $v \neq w$. Then, the last non-zero coordinate of $v - w$ has to be strictly positive, while that of $w - v$ has to be also strictly positive, which leads to a contradiction. Hence $v = w$.
- Transitive: Suppose $v \succeq_u w$, $w \succeq_u u$. If $u = v$, then $v \succeq_u u$. Assume $u \neq v$, since $0 \neq v - u = (v - w) + (w - u)$, the last non-zero coordinate of $v - u$ is either the last non-zero coordinate of $v - w$, that of $w - u$, or the sum of both. They all must be positive by hypothesis, hence $v \succeq u$.
- Total ordering: By definition, either $w \succeq_u v$ or $v \succeq_u w$.
- Additive: Let $v \succeq w$. On one hand, if $v = w$, then $u + v = u + w$ for any $u \in \mathbb{R}$. On the other hand, if the last non-zero coordinate of $v - w$ is positive, then the last non-zero coordinate of $u + v - (u + w)$ must also be positive for any $u \in \mathbb{R}^\infty$.

□

Any monomial ordering on $K[x_1, \ldots, x_n]$ is characterized by a matrix $A' \in \mathcal{M}_{n\times n'}(\mathbb{R})$ (see [9] Theorem 5). We would also like to define monomial orderings on P by infinite matrices. To do so, we must be able to represent the monomials of P as infinite tuples of $\mathbb{N}^\infty$, and to introduce this notation, we require the following concept and proposition:

Definition 3.9. Let $(Y_1, >_1)$ and $(Y_2, >_2)$ be two partially ordered sets. Then, $\phi\colon (Y_1, >_1) \to (Y_2, >_2)$ is an order isomorphism if it is a bijection and given $y_1, y_2 \in Y_1$, we have that $y_1 >_1 y_2$ if and only if $\phi(y_1) >_2 \phi(y_2)$.

Proposition 3.10. *Let P be a Σ-polynomial algebra and $\succ$ be the induced ordering on $X(\Sigma)$ by $>$ and $\triangleright$. Then, there exists an order isomorphism*

$$\phi\colon (X(\Sigma), \succ) \longrightarrow (\mathbb{N}, >_u)$$

if and only if for any $\sigma_0 \in \Sigma$, $\{\sigma \in \Sigma \mid \sigma < \sigma_0\}$ is finite.

Proof. We first prove that the existence of ϕ implies that $\forall \sigma_0 \in \Sigma$, $\#\{\sigma \in \Sigma \mid \sigma < \sigma_0\} < \infty$. Suppose there exists $\sigma_0 \in \Sigma$ such that $\#\{\sigma \in \Sigma \mid \sigma < \sigma_0\} = \infty$. Consider $x_1 \in X$, we have that $\#\{x(\sigma) \in X(\Sigma) \mid x(\sigma) \prec x_1(\sigma_0)\} = \infty$. If ϕ exists, then $\{n \in \mathbb{N} \mid n <_u \phi(x_1(\sigma_0))\}$ is infinite, which leads to a contradiction.

For the other implication, suppose that for any $\sigma_0 \in \Sigma$, $\{\sigma \in \Sigma \mid \sigma < \sigma_0\}$ is finite. Then, $\{x(\sigma) \in X(\Sigma) \mid x(\sigma) \prec x_0(\sigma_0)\}$ is finite for any $x_0 \in X$ and $\sigma_0 \in \Sigma$. We define the order isomorphism ϕ. To do so, since $\succ$ is a well ordering on $X(\Sigma)$, we consider:

$$\begin{aligned} x(\gamma)_1 &= \min_{\succ}\{X(\Sigma)\} \\ x(\gamma)_2 &= \min_{\succ}\{X(\Sigma) \setminus \{x(\gamma)_1\}\} \\ x(\gamma)_3 &= \min_{\succ}\{X(\Sigma) \setminus \{x(\gamma)_1, x(\gamma)_2\}\} \\ &\vdots \end{aligned}$$

And so, obtain the infinite increasing chain

$$x(\gamma)_1 \prec x(\gamma)_2 \prec x(\gamma)_3 \prec \ldots \prec x(\gamma)_j \prec \ldots \tag{3}$$

In this way, we order the entire $X(\Sigma)$, because if we do not, this contradicts the fact that for any $x_0 \in X$ and $\sigma_0 \in \Sigma$, $\{x(\sigma) \in X(\Sigma) \mid x(\sigma) \prec x_0(\sigma_0)\}$ is finite.

Hence, we define

$$\begin{aligned} \phi\colon (X(\Sigma), \succ) &\longrightarrow (\mathbb{N}, >_u) \\ x(\gamma)_j &\longmapsto j - 1. \end{aligned}$$

By the previous, ϕ is well-defined, injective, and surjective. Note that it also preserves the ordering, so it is an order isomorphism. □

Notation 3.11. From now on, whenever we consider the well ordering $\succ$ on $X(\Sigma)$ induced by a monomial ordering $>$ on Q and a total ordering $\triangleright$ on X, we assume that $>$ satisfies the following property: for any $\sigma_0 \in \Sigma$, $\{\sigma \in \Sigma \mid \sigma < \sigma_0\}$ is finite. Hence, each time we consider an ordering of $X(\Sigma)$ in this way, we also have the order isomorphism ϕ constructed in Proposition 3.10.

Using the previous Proposition 3.10, we introduce the following concepts.

Notation 3.12. Let P be a Σ-polynomial algebra and let $\succ$ be the well ordering on $X(\Sigma)$ induced by $>$ and $\triangleright$. Then, $x(\sigma)$ is the i-th variable of $X(\Sigma)$ if $\phi(x(\sigma)) = i - 1 \in \mathbb{N}$, where ϕ is the order isomorphism defined in Proposition 3.10.

Definition 3.13. Let P be a Σ-polynomial algebra, $\succ$ be the well ordering on $X(\Sigma)$ induced by $>$ and $\triangleright$, and $m \in M$. We denote by (m) the tuple $(\alpha_1, \alpha_2, \ldots) \in \mathbb{N}^\infty$, where α_i is the exponent of the i-th variable of $X(\Sigma)$. If $x(\sigma)$ is the i-th variable of $X(\Sigma)$, we may refer to α_i as the i-th coordinate of (m) or as the $x(\sigma)$-th coordinate of (m).

Note that only finitely many coordinates of (m) are non-zero.

If P is a Σ-polynomial algebra and $\succ$ is the well ordering on $X(\Sigma)$ induced by $>$ and $\triangleright$, there exists a monoid isomorphism $\phi'\colon (M, \cdot) \to (\mathbb{N}^\infty, +)$ given by $\phi'(m) = (m)$, where $\cdot$ denotes the multiplication in M.

Example 3.14. Let $X = \{x_1, x_2\}$ and $\Sigma = \mathbb{N}^2$, and consider the ordering $\triangleright$ in X such that $x_2 \triangleleft x_1$, and the graded lexicographical order $>_{\text{deglex}}$ in $K[(1,0),(0,1)]$. We consider $\succ$ the well ordering on $X(\mathbb{N}^2)$ induced by $>_{\text{deglex}}$ and $\triangleright$. Since

$>_{\text{deglex}}$ satisfies that for any $\sigma_0 \in \Sigma$, $\{\sigma \in \Sigma \mid \sigma <_{\text{deglex}} \sigma_0\}$ is finite, by Proposition 3.10, there is a order isomorphism:

$$\begin{aligned}\phi\colon (X(\mathbb{N}^2), \succ) &\longrightarrow (\mathbb{N}, >_u)\\ x_2(0,1) &\longmapsto 0\\ x_1(0,1) &\longmapsto 1\\ x_2(1,0) &\longmapsto 2\\ x_1(1,0) &\longmapsto 3\\ &\vdots\end{aligned}$$

We observe that, $\succ$ orders $X(\Sigma)$ as follows

$$x_2(0,1) \prec x_1(0,1) \prec x_2(1,0) \prec x_1(1,0) \prec x_2(0,2) \prec \dots$$

Therefore, $x_2(0,1)$ is the first variable, $x_1(0,1)$ is the second variable, $x_2(1,0)$ is the third variable, and so on. We define

$$\begin{aligned}\phi'\colon (M, \cdot) &\longrightarrow (\mathbb{N}^\infty, +)\\ m_0 = 1 &\longmapsto (1) = (0, \dots)\\ m_1 = x_1(0,1) &\longmapsto (x_1(0,1)) = (0,1,0,\dots)\\ m_2 = x_2(0,2)^4 x_2(1,0) x_1(0,1)^5 &\longmapsto (m_2) = (0,5,1,0,4,0,\dots).\end{aligned}$$

Observe that the exponent in each m_i of the j-th variable appears exactly in the j-th position of the tuple. For instance, in m_2, $x_2(0,2)$ is the 5-th variable and its exponent "4" appears in the 5-th position of the tuple.

We need the following notions about infinite matrices.

Definition 3.15. We denote by $\mathcal{M}_{\infty\times\infty}(\mathbb{R})$ the set of matrices with real entries whose rows and columns are countably infinite and indexed starting from 1. So, the matrices $A \in \mathcal{M}_{\infty\times\infty}(\mathbb{R})$ are as follows:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots \\ a_{21} & a_{22} & & \\ a_{31} & & \ddots & \\ \vdots & & & \end{pmatrix}.$$

Definition 3.16. Let $v = (v_1, v_2, \dots) \in \mathbb{R}^\infty$ and $A \in \mathcal{M}_{\infty\times\infty}(\mathbb{R})$, where the entries of A are labeled a_{ij} with i being the row and j being the column, and $i, j \in \mathbb{N} \setminus \{0\}$. Then, the product vA is defined as follows:

$$vA := \left(\sum_{i=1}^{\infty} a_{i1} v_i,\ \sum_{i=1}^{\infty} a_{i2} v_i,\ \sum_{i=1}^{\infty} a_{i3} v_i, \dots \right) \in \prod_{i=1}^{\infty} \mathbb{R}.$$

Since v has only a finite number of non-zero entries, each infinite sum has only a finite number of non-zero terms.

Notation 3.17. Let $v \in \mathbb{R}^\infty$, we denote the t-th coordinate of v as v_t. Moreover, if $A \in \mathcal{M}_{\infty\times\infty}(\mathbb{R})$ the the t-th coordinate of vA is denoted as vA_t

Lemma 3.18. *Let $v = (v_1, v_2, \dots) \in \mathbb{R}^\infty$ and $A \in \mathcal{M}_{\infty\times\infty}(\mathbb{R})$. If A has a finite number of non-zero entries in each row, then $vA \in \mathbb{R}^\infty$.*

Proof. Since $v = (v_1, v_2, \ldots) \in \mathbb{R}^\infty$, by hypothesis, there exists some $t \in \mathbb{N}$ such that for every $t' > t$, $v_{t'} = 0$. Then, we take

$$s = \max\{j \in \mathbb{N} \mid a_{ij} \neq 0, i \leq t\}$$

and we get that:

$$vA = (v_1, \ldots, v_t, 0\ldots) \begin{pmatrix} a_{11} & \ldots & a_{1s} & 0\ldots \\ \vdots & \ddots & \vdots & \vdots \\ a_{t1} & \ldots & a_{ts} & 0\ldots \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix} \in \mathbb{R}^\infty$$

□

These definitions yield Proposition 3.20, which enables us to define monomial orderings on M via a matrix A under certain conditions. For this result, we need the following lemma.

Lemma 3.19. *Let $\succ$ be the well ordering on $X(\Sigma)$ induced by $>$ and $\triangleright$, $\{m_i\}_{i=1}^\infty \subset M$, and $A \in \mathcal{M}_{\infty\times\infty}(\mathbb{R})$ such that every row of A has a finite number of non-zero entries and the last non-zero entry is positive. If*

$$(m_1)A \succ_u (m_2)A \succ_u (m_3)A \succ_u (m_4)A \succ_u \ldots$$

Then, there exists some coordinate t for which $\{(m_i)A_t\}_{i=1}^\infty \subset \mathbb{R}$ has a strictly decreasing subsequence $\{(m_{i_k})A_t\}_{i=1}^\infty$ such that $(m_{i_k})A_s = (m_{i_{k'}})A_s$ $\forall s > t$.

Proof. Let us denote by l_i the position of the last non-zero coordinate of $(b_i)A$. The coordinate of $(m_i)A$ in the l_i-th position must be strictly positive because the last non-zero entry of each row of A is positive. Therefore, since $(m_1)A \succ_u (m_i)A$ for every $i > 1$, we have that $l_i \leq l_1$ for every $i \geq 1$.

Furthermore, since $(m_i)A \succ_u (m_{i+1})A$ for every $i \in \mathbb{N}$, by definition of $\succ_u$, for each $i \geq 1$ there exists a coordinate $t(i) \in \{1, \ldots, l_i\}$ such that $(m_i)A_{t(i)} >_u (m_{i+1})A_{t(i)}$ and that for any $s \geq t(i)$, $(m_i)A_s = (m_{i+1})A_s$. Then, for any $i \geq 1$, $1 \leq t(i) \leq l$. Hence, there must exists a coordinate $t \in \{1, \ldots, l\}$ such that $\#\{i \in \mathbb{N} \mid t(i) = t\} = \infty$. Let $\{i_k\}_{k=1}^\infty \subset \mathbb{N}$ be the sequence such that $t(i_k) = t$ and indexed such that

$$i_1 <_u i_2 <_u i_3 <_u i_4 < \ldots$$

By construction, we have that

$$(m_{i_1})A_t >_u (m_{i_1+1})A_t \geq (m_{i_2})A_t >_u (m_{i_2+1})A_t \geq (m_{i_3})A_t >_u (m_{i_3+1})A_t \geq \ldots$$

and that $\forall s \geq t$, $(m_{i_k})A_s = (m_{i_{k'}})A_s$ which concludes the proof. □

Proposition 3.20. *Let P be a Σ-polynomial algebra, $\succ$ be a well ordering on $X(\Sigma)$ induced by $>$ and $\triangleright$, and $A \in \mathcal{M}_{\infty\times\infty}(\mathbb{R})$ such that every row of A has a finite number of non-zero entries, and the last non-zero entry is positive. If*

$$\begin{aligned} \varphi_A \colon \mathbb{R}^\infty &\longrightarrow \mathbb{R}^\infty \\ (m) &\longmapsto (m)A \end{aligned}$$

is injective, then, $\succ_A$ defined as

$$m \succ_A m' \iff (m)A \succ_u (m')A$$

is a monomial ordering on P.

Proof. We start by proving it is multiplicative, let $m, m' \in M$ and suppose $m \succ_A m'$. Then, $(m)A \succ_u (m')A$, and so for any $l \in M$, $(l)A + (m)A \succ_u (l)A + (m')A$ because of additivity of $\succ_u$. Hence, $((l)+(m))A \succ_u ((l)+(m'))A$, and therefore $lm \succ_A lm'$.

We prove that it is a total ordering. Since $\succ_u$ is transitive and reflexive, $\succ_A$ is also transitive and reflexive. Moreover, because φ_A is injective and $\phi' \colon M \to \mathbb{N}^\infty$ is a monoid isomorphism, we have $(m)A = (n)A \iff (m) = (n) \iff m = n$. Thus, $\succ_A$ is antisymmetric because $\succ_u$ is antisymmetric. Furthermore, since $\succ_u$ is a total ordering, $\succ_A$ is also a total ordering.

To prove $\succ_A$ is well-ordered, we have to prove that any nonempty set of M has a minimum. Let us suppose there exists $\{m_i\}_{i=1}^\infty \subseteq M$ such that

$$(m_1)A \succ_u (m_2)A \succ_u (m_3)A \succ_u \dots$$

By the previous Lemma, there exist a subsequence $\{m_{i_k}\}_{k=1}^\infty$ and a coordinate t such that

$$(m_{i_1})A_t >_u (m_{i_2})A_t >_u (m_{i_3})A_t >_u \dots$$

and that $\forall s \geq t$, $(m_{i_k})A_s = (m_{i_{k'}})A_s$. For clarity, let us denote this subsequence by $\{(m_i)A\}_{i=1}^\infty$. It follows that there must exist some coordinate $c \in \mathbb{N}$ such that $(m_i)A_s = 0$ for every $s \geq c$ and $i \geq 1$. Therefore, φ_A maps every (m_i) to a sub-vector space of $\mathbb{R}^\infty$ of dimension c. By injectivity of φ_A, the vector space spanned by $\{(m_i)\}_{i=1}^\infty$ in $\mathbb{R}^\infty$ has dimension c, thus, there exists a coordinate $c' \in \mathbb{N}$ such that $(m_i)_s = 0$ $\forall s \geq c'$ and $i \geq 1$. Denoting by $(m_i^f) \in \mathbb{N}^{c'}$ the tuple coinciding with the first c' coordinates of (m_i), and by A^f the top left matrix of A with c' rows and such that

$$A = \left(\begin{array}{c|c} A^f & 0 \\ \hline * & * \end{array}\right),$$

it follows that $(m_i)A$ coincides with $(m_i^f)A^f$ in all non-zero coordinates. Hence, we have the strictly decreasing sequence:

$$(m_1^f)A^f \succ_u (m_2^f)A^f \succ_u (m_3^f)A^f \succ_u \dots$$

Because φ_A is injective, A^f must be of full rank, and because the last non-zero coordinate of A^f is positive, we get a contradiction with Theorem 1.8. □

This proposition gives us a lot of possible ways to consider monomial orders on P. However, we are not interested in all monomial orders on P; we would like to consider only those that behave well with respect to our notion of Σ-ideal.

Definition 3.21. Let P be a Σ-polynomial algebra, and $\succ$ be a monomial ordering on P. Then, $\succ$ is a Σ-ordering if for any $\sigma \in \Sigma$ and m, m' monomials in P

$$m \succ m' \implies \sigma \cdot m \succ \sigma \cdot m'$$

Not all monomial orderings on P induced by a matrix $A \in \mathcal{M}_{\infty\times\infty}(\mathbb{R})$ as in the Proposition 3.20 are Σ-orderings.

Example 3.22. Let $X = \{x_1, x_2\}$, $\Sigma = \langle\sigma\rangle$, we consider $\triangleright$ to be the order relation on X such that $x_1 \triangleright x_2$, and $>$ to be the monomial ordering on $K[\sigma]$ given by the grading. Let also P be the Σ-polynomial algebra $K[\Sigma(X)]$. We

define the well ordering $\succ$ on $X(\Sigma)$ induced by $>$ and $\triangleright$. Observe that $\succ$ orders the variables as follows:

$$x_2(0) \prec x_1(0) \prec x_2(\sigma) \prec x_1(\sigma) \prec x_2(\sigma^2) \prec x_1(\sigma^2) \prec \dots$$

Now let $A \in \mathcal{M}_{\infty\times\infty}(\mathbb{R})$ be as follows

$$A = \left(\begin{array}{cccc|c|c} 1 & 0 & 0 & 0 & 1 & 0\dots \\ 0 & 1 & 0 & 0 & 2 & 0\dots \\ 0 & 0 & 1 & 0 & 2 & 0\dots \\ 0 & 0 & 0 & 1 & 1 & 0\dots \\ \hline 0 & 0 & 0 & 0 & 0 & \\ \vdots & \vdots & \vdots & \vdots & \vdots & Id_{\infty\times\infty} \\ & & & & & \end{array}\right)$$

The induced map φ_A is injective by definition of A. Consider $m = x_1(0)^3x_2(0)^5$, $m' = x_1(0)^2x_2(0)^6$ two monomials in P. Then, we have that

$$(m) = (5, 3, 0, \dots) \qquad (m') = (6, 2, 0, \dots)$$
$$(m)A = (5, 3, 0, 0, 11, 0, \dots) \qquad (m')A = (6, 2, 0, 0, 10, 0, \dots).$$

Then, $(m)A \succ_u (m')A \implies m \succ_A m'$. However,

$$(\sigma \cdot m) = (0, 0, 5, 3, 0, \dots) \qquad (\sigma \cdot m') = (0, 0, 6, 2, 0, \dots)$$
$$(\sigma \cdot m)A = (0, 0, 5, 3, 13, \dots) \qquad (\sigma \cdot m')A = (0, 0, 6, 2, 14, \dots).$$

Therefore, $\sigma \cdot m \prec_A \sigma \cdot m'$, which means that $\succ_A$ is not a Σ-ordering.

In Proposition 3.27, we give some sufficient conditions in A to ensure that $\succ_A$ is a Σ-ordering, but to give this result, we need the following definitions.

Definition 3.23. Let $A \in \mathcal{M}_{\infty\times\infty}(\mathbb{R})$, and $A' \in \mathcal{M}_{n\times n'}(\mathbb{R})$. Then, A is defined by blocks of A' if

$$A = \left(\begin{array}{c|c|c|c} A' & 0 & 0 & 0\dots \\ \hline 0 & A' & 0 & 0\dots \\ \hline 0 & 0 & A' & 0\dots \\ \hline 0 & 0 & 0 & \ddots \\ \vdots & \vdots & \vdots & \end{array}\right).$$

Note that if A is defined by blocks of $A' \in \mathcal{M}_{n\times n'}(\mathbb{R})$, the last non-zero coordinate of each row of A' is positive, and if A' is of rank n, then it satisfies all the hypotheses of Proposition 3.20.

Definition 3.24. Let $\succ$ be the well ordering on $X(\Sigma)$ induced by $>$ and $\triangleright$, $\gamma \in \Sigma$ and $(m) \in \mathbb{N}^\infty$. Consider $\alpha_1, \dots, \alpha_n \in \mathbb{R}$ to be the entries in the positions $x_1(\gamma), \dots, x_n(\gamma)$ of (m), and also $A \in \mathcal{M}_{\infty\times\infty}(\mathbb{R})$ to be defined by blocks of $A' \in \mathcal{M}_{n\times n'}(\mathbb{R})$. Then, we define:

- The γ-block of (m) as the tuple $(m)_\gamma \in \mathbb{N}^\infty$ that agrees with (m) in the coordinates corresponding to $x_1(\gamma), \dots, x_n(\gamma)$ and is zero elsewhere. Note that the coordinates $x_1(\gamma), \dots, x_n(\gamma)$ in (m) are adjacent to one another.
- The γ-block of $(m)A$ as $(m)A_\gamma := (m)_\gamma A$.

Remark 3.25. In the setting of the previous definition, since A is defined by blocks of A', and the number of rows of A' is $n = \#X$, the γ-blocks of $(m)A$ are infinite tuples with only n' non-zero entries adjacent to one another:

$$(0,\dots 0,\overbrace{*,\dots,*}^{n},0,\dots)A = (0,\dots,0,\overbrace{*,\dots,*}^{n'},0\dots)$$

Moreover, we observe that any two $(m)A_\gamma$, $(m)A_{\gamma'}$ such that $\gamma \neq \gamma'$ are "disjoint". That is, if some t-th coordinate of $(m)A_\gamma$ is non-zero, then for any $\gamma \neq \gamma' \in \Sigma$, the t-th coordinate of $(m)A_{\gamma'}$ must be zero.

The following lemma follows from the definition of $\succ_u$ and this remark. It will be useful for the following propositions.

Lemma 3.26. *Let $\succ$ be the well ordering on $X(\Sigma)$ induced by $>$ and $\triangleright$, and $A \in \mathcal{M}_{\infty\times\infty}(\mathbb{R})$ be a matrix defined by blocks of $A' \in \mathcal{M}_{n\times n'}(\mathbb{R})$. Then, $(m)A \succ_u (m')A$ if and only if there exists $\delta \in \Sigma$ such that for any $\delta < \gamma \in \Sigma$, $(m)A_\gamma = (m')A_\gamma$ and $(m)A_\delta \succ_u (m')A_\delta$.*

Proposition 3.27. *Let $\succ$ be the well ordering on $X(\Sigma)$ induced by $>$ and $\triangleright$, and $A \in \mathcal{M}_{\infty\times\infty}(\mathbb{R})$ be a matrix defined by blocks of $A' \in \mathcal{M}_{n\times n'}(\mathbb{R})$ where A' has rank n and the last non-zero element of each row is positive. Then, $\succ_A$ defined by $\succ_u$ and $\succ$ as in the Proposition 3.20 is a Σ-ordering.*

Proof. Thanks to Proposition 3.20, we only have to prove that $\succ_A$ behaves well with respect to the action of Σ. Let $\sigma \in \Sigma$, and suppose $m \succ_A m'$.

We note that for any $\gamma \in \Sigma$, by definition of γ-block, $(\sigma \cdot m)_{\sigma+\gamma} = (m)_\gamma$ and that $(m)_\gamma A = (m)A_\gamma$.

Suppose that $(m)A \succ_u (m')A$. By Lemma 3.26, this is equivalent to stating that exists a $\delta \in \Sigma$ such that for any $\delta < \gamma \in \Sigma$ $(m)A_\gamma = (m')A_\gamma$ and $(m)A_\delta \succ_u (m')A_\delta$. Therefore, we have that for every $\gamma > \delta$,

$$(\sigma{\cdot}m)A_{\sigma+\gamma} = (\sigma{\cdot}m)_{\sigma+\gamma}A = (m)_\gamma A = (m')_\gamma A = (\sigma{\cdot}m')_{\sigma+\gamma}A = (\sigma{\cdot}m')A_{\sigma+\gamma}.$$

and that

$$(\sigma{\cdot}m)A_{\sigma+\delta} = (\sigma m)_{\sigma+\delta}A = (m)_\delta A \succ_u (m')_\delta A = (\sigma{\cdot}m')_{\sigma+\delta}A = (\sigma{\cdot}m')A_{\sigma+\delta}$$

Hence for $\sigma+\delta \in \Sigma$, $(m)A_{\sigma+\delta} \succ_u (m')A_{\sigma+\delta}$ and for any $\gamma > \sigma+\delta$, $(m)A_\gamma = (m')A_\gamma$. Therefore $(\sigma \cdot m) \succ_A (\sigma \cdot m')$. □

In this way, we define Σ-monomial orders in P using matrices.

However, in [5], one finds a different way for constructing Σ-monomial orderings. Before presenting it, we need the following definition.

Definition 3.28. Let P be a Σ-polynomial algebra, and $\sigma \in \Sigma$. We define $X(\sigma) := \{x(\sigma) \mid x \in X\} \subset X(\Sigma)$, and $P(\sigma) := K[X(\sigma)] \subset P$.

Notation 3.29. Observe that $P(\delta) \cong K[X]$ as a K-algebra for any $\delta \in \Sigma$. Via this isomorphism, any monomial ordering $\triangleright$ on $K[X]$, induces a monomial ordering on each $P(\delta)$. Making an abuse of notation, we denote all these induced orderings with the symbol $\triangleright$.

Definition 3.30. Let $>$ be a monomial ordering on Q (so that it does not necessarily have to satisfy that $\forall\sigma_0 \in \Sigma\ \#\{\sigma \in \Sigma \mid \sigma < \sigma_0\} <_u \infty$) and $\triangleleft$ be a monomial ordering on $K[X]$. We define the block ordering $\succ_b$ on P induced by $>$ and $\triangleright$, following the next steps:

(1) Consider m, m' two monomials in P. Then, we write

$$m = m_1 \dots m_k \qquad m' = m'_1 \dots m'_k$$

where each $m_i, m'_i \in P(\delta_i)$ for some $\delta_i \in \Sigma$, in such a way that $\delta_1 > \dots > \delta_k$ (some m_i and m'_i can be 1).

(2) Then, $m \succ_b m'$ if and only if there exists an $i \in \{1, \dots, k\}$ such that for all $1 \leq j < i$, $m_j = m'_j$ and $m_i \triangleright m'_i$.

Proposition 3.31. *The block ordering $\succ_b$ on P induced by $>$ and $\triangleright$ is a Σ-ordering.*

Proof. We first prove that $\succ_b$ is a monomial ordering on P. To prove the multiplicative property, suppose $m \succ_b m'$, and take an $l \in M$. We write m, m' and l as mentioned in Definition 3.30:

$$\begin{aligned} m &= m_1 \dots m_k \\ m' &= m'_1 \dots m'_k \\ l &= l_1 \dots l_k. \end{aligned}$$

Since $m \succ_b m'$, there exists an i such that for every $1 \leq j < i$, $m_j = m'_j$ and $m_i \triangleright m'_i$. Therefore, since $\triangleright$ is multiplicative in $P(\delta_i)$, for the same i we have that for each $1 \leq j < i$, $l_j m_j = l_j m'_j$ and $l_i m_i \triangleright l_i m'_i$. Hence $lm \succ_b lm'$.

Since $>$ and $\triangleright$ are total orderings, $\succ_b$ is also a total ordering on M. To prove that $\succ_b$ is well-ordered, let $A \subseteq M$. Suppose it is not well-ordered. Then, there exists $\{m_i\}_{i=1}^{\infty} \subseteq A$ such that

$$m_1 \succ_b m_2 \succ_b m_3 \succ_b \dots$$

By definition of $\succ_b$, this implies that either $>$ or $\triangleright$ is not a well ordering, which is a contradiction.

We finally have to prove that $\succ_b$ is a Σ-ordering. Let $\sigma \in \Sigma$, and suppose $m \succ_b m'$. This means that there exists i such that for each $1 \leq j < i$, $m_j = m'_j$ and $m_i \triangleright m'_i$. We observe that

$$\begin{aligned} \sigma \cdot m &= (\sigma \cdot m_1)(\sigma \cdot m_2) \dots (\sigma \cdot m_k) \\ \sigma \cdot m' &= (\sigma \cdot m'_1)(\sigma \cdot m'_2) \dots (\sigma \cdot m'_k) \end{aligned}$$

where each $\sigma \cdot m_i, \sigma \cdot m'_i \in P(\sigma + \delta_i)$ and $\sigma + \delta_1 > \dots > \sigma + \delta_k$ by additivity of $>$. Since the monomial ordering on each $P(\delta)$ for $\delta \in \Sigma$, is the same ($\triangleright$ it does not depending on the δ), we get that for the same i and for each $1 \leq j < i$, $\sigma \cdot m_j = \sigma \cdot m'_j$ and that $\sigma \cdot m_i \triangleright \sigma \cdot m'_i$. Hence, $\sigma \cdot m \succ \sigma \cdot m'$. □

We know that any monomial ordering on $K[x_1, \dots, x_n]$ can be defined in terms of a matrix with real entries (see [9] Theorem 5). Hence, it seems natural to aim for a similar result for certain block orderings. To this end, we need the following notation.

Notation 3.32. Let $A' \in \mathcal{M}_{n \times n'}(\mathbb{R})$ be a matrix. Then, we denote by:

- A'_c the matrix obtained by reversing the columns of A', where the entry $a_{i,j}$ of A' is the entry $a_{i,n'-j+1}$ of A'_c.
- A'_r the matrix obtained by inverting the rows of A', where the entry $a_{i,j}$ of A' is the entry $a_{n-i+1,n'}$ of A'_r.

Combining these notations, we denote by A'_{rc} the matrix obtained by inverting the rows and columns.

Proposition 3.33. *Let $>$ and $\triangleright$ be monomial orderings of Q and $K[x_1, \dots, x_n]$ respectively, where $\#\{\sigma \in \Sigma \mid \sigma < \sigma_0\} <_u \infty \ \forall \sigma_0 \in \Sigma$. Let $A' \in \mathcal{M}_{n\times n'}(\mathbb{R})$ such that $\triangleright$ and $>_{A'}$ are equal on $K[x_1, \dots, x_n]$, and $A \in \mathcal{M}_{\infty\times\infty}(\mathbb{R})$ be defined by blocks over A'_{rc}. Furthermore, let $\succ$ be the well ordering on $X(\Sigma)$ induced by $>$ and the ordering $x_1 \triangleright_u x_2 \triangleright_u \dots \triangleright_u x_n$ on X. Under this conditions, the block ordering $\succ_b$ on P induced by $>$ and $\triangleright$ equals the Σ-ordering $\succ_A$ defined in Proposition 3.20.*

Proof. First, note that since $>_{A'}$ is a monomial ordering, A' must have maximum rank. Moreover, by Theorem 1.8, the first non-zero entry of each row of A' must be positive. Therefore, A'_{rc} is also of rank n and the last non-zero entry of each row must be positive. Hence, $\succ_A$ is well defined.

Recovering the notation in Definition 3.30, we write

$$m = m_1 \dots m_k$$
$$m' = m'_1 \dots m'_k$$

where $m_j, m'_j \in P(\delta_j)$ for some $\delta_1 > \delta_2 > \dots > \delta_k \in \Sigma$.

By Lemma 3.26, it suffices to show that $m \succ_b m'$ if and only if there exists some $i \in \{1, \dots, k\}$ such that $\forall \gamma > \delta_i$, $(m)A_\gamma = (m')A_\gamma$ and $(m)A_{\delta_i} \succ_u (m')A_{\delta_i}$.

Let $i \in \{1, \dots, k\}$, we first prove that $m_j = m'_j$ for any $1 \le j < i$ if and only if $(m)_\gamma = (m)_\gamma$ for any $\gamma > \delta_i$. By definition of $(m)_{\delta_j}$, $m_j = m'_j \iff (m)_{\delta_j} = (m')_{\delta_j}$ for $1 \le j < i$. Moreover, $\forall \gamma > \delta_i$ such that $\gamma \notin \{\delta_1, \dots, \delta_{i-1}\}$, $(m)_\gamma = (m')_\gamma$ because they are the zero tuple. Hence, $m_j = m'_j$ for any $1 \le j < i$ if and only if $(m)_\gamma = (m)_\gamma$ for any $\gamma > \delta_i$.

It suffices to prove that $m_i \triangleright m'_i \iff (m)A_{\delta_i} \succ_u (m)A_{\delta_i}$. Since m_i, m'_i are monomials of $P(\delta_i)$, and $P(\delta_i) \cong K[x_1, \dots, x_n]$ as a K-algebra, we consider their exponent vectors $\alpha = (\alpha_1, \dots, \alpha_n), \beta = (\beta_1, \dots, \beta_n) \in \mathbb{N}^n$, where α_j and β_j denote the exponents of the variable $x_j(\delta_i)$ in m_i and m'_i respectively. In this way, we have that $m_i \triangleright m'_i \iff \alpha A' >_{\text{lex}} \beta A'$. Let us denote by $\overline{(m)_{\delta_i}} \in \mathbb{N}^n$ the tuple formed by the coordinates $x_1(\delta_i), \dots, x_n(\delta_i)$ of $(m)_{\delta_i}$. Since $x_1 \triangleright_u x_2 \triangleright_u \dots \triangleright_u x_n$, $x_1(\delta_i) \succ x_2(\delta_i) \succ \dots \succ x_n(\delta_i)$. Hence $\overline{(m)_{\delta_i}} = (\alpha_n, \dots, \alpha_1) = \alpha_c$ and $\overline{(m')_{\delta_i}} = (\beta_n, \dots, \beta_1) = \beta_c$. Finally, using all of this, we have that

$$\begin{aligned}\alpha A' >'_{\text{lex}} \beta A' &\iff \alpha_c A'_r >'_{\text{lex}} \beta_c A'_r \iff \\ &\iff \alpha_c A'_{rc} >'_{\text{invlex}} \beta_c A'_{rc} \iff \overline{(m)_{\delta_i}} A'_{rc} >'_{\text{invlex}} \overline{(m')_{\delta_i}} A'_{rc} \iff \\ &\iff (m)_{\delta_i} A \succ_u (m)_{\delta_i} A \iff (m)A_{\delta_i} \succ_u (m)A_{\delta_i}.\end{aligned}$$

The equivalence $\overline{(m)_{\delta_i}} A'_{rc} >_{\text{invlex}} \overline{(m')_{\delta_i}} A'_{rc} \iff (m)_{\delta_i} A \succ_u (m)_{\delta_i} A$ follows from the definition of $>_{\text{invlex}}$ in $\mathbb{R}^{n'}$ in Remark 1.7. □

Let us give an example of a Σ-ordering, and show the two equivalent ways of defining it.

Example 3.34. Let $X = \{x_1, x_2, x_3\}$ and $\Sigma = \mathbb{N}^3$. We consider the degree reverse lexicographical ordering $\triangleright$ in $K[X]$, and the degree lexicographical ordering on $Q = K[(1,0,0),(0,1,0),(0,0,1)]$. Consider the monomials

$$m = x_1(2,1,3)^4\, x_2(0,1,1)^2\, x_3(0,1,1)^3$$
$$m' = x_1(2,1,3)^4\, x_1(0,1,1).$$

Let also $\succ_b$ be the block monomial ordering obtained by $>$ and $\triangleright$. Then, considering

$$m_1 = x_1(2,1,3)^4 \in P(2,1,3), \qquad m_2 = x_2(0,1,1)^2 x_3(0,1,1) \in P((0,1,1)),$$
$$m_1' = x_1(2,1,3)^4 \in P(2,1,3), \qquad m_2' = x_1(0,1,1) \in P(0,1,1),$$

we take $i = 2$ such that $m_1 = m_1'$ and $m_2 \triangleleft m_2'$. Hence, $m \succ_b m'$.

We show that $m \succ_A m'$ for a matrix A like in Proposition 3.33. To this end, let $\triangleright_u$ be the ordering in X satisfying $x_1 \triangleright_u x_2 \triangleright_u \ldots \triangleright_u x_n$ and $\succ$ be the well ordering on $X(\Sigma)$ induced by $>$ and $\triangleright_u$. $\succ$ orders $X(\Sigma)$ as follows:

$$x_3(0,0,1) \prec x_2(0,0,1) \prec x_1(0,0,1) \prec x_3(0,1,0) \prec x_2(0,1,0) \prec x_1(0,1,0)$$
$$\prec x_3(1,0,0) \prec x_2(1,0,0) \prec x_1(1,0,0) \prec x_3(0,0,2) \prec x_2(0,0,2) \prec$$
$$x_1(0,0,2) \prec x_3(0,1,1) \prec x_2(0,1,1) \prec x_1(0,1,1) \prec \ldots$$

Using this order, we express m and m' as tuples (m) and (m') where the exponent of the i-th variable appears in the i-th coordinate of the tuple. Since $x_1(0,1,1), x_2(0,1,1), x_3(0,1,1)$ are the 12-th, 11-th and 10-th variables respectively, and $x_1(2,1,3)$ is the 195-th variable, we get the tuples:

$$(m) = (0,\ldots,0,3,2,0,0,\ldots,0,4,0,\ldots)$$
$$(m') = (0,\ldots,0,0,0,1,0,\ldots,0,4,0,\ldots).$$

Consider A' the matrix of the degree reverse lexicographical ordering given in the Example 1.9 and compute:

$$A'_{rc} = \begin{pmatrix} 0 & 0 & -1 & 1 \\ 0 & -1 & 0 & 1 \\ -1 & 0 & 0 & 1 \end{pmatrix}$$

Let $A \in \mathcal{M}_{\infty\times\infty}(\mathbb{R})$ be the matrix defined by blocks of A'_{rc}:

$$A = \left(\begin{array}{cccc|cccc|cc} 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & \ldots \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \ldots \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \ldots \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & \ldots \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & \ldots \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & \ldots \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \ddots \end{array}\right).$$

Thus, we have that

$$(m)A = (0,\ldots,0,0,-2,-3,5,0,\ldots,0,-4,0,0,4,0,\ldots)$$
$$(m')A = (0,\ldots,0,-1,0,0,1,0,\ldots,0,-4,0,0,4,0,\ldots).$$

where $-2, -3$ and 5 occupy the 14-th, 15-th, and 16-th entries, -4 and 4 occupy the 257-th and 260-th position in both tuples, and -1 and 1 occupy the 13-th and the 16-th positions. Therefore, by the definition of $\succ_u$, we get that $(m)A \succ_A (m')A$ because $5 > 1$.

To finish this part, we discuss the implications of Proposition 3.33. Note that any Σ-block ordering obtained by $>$ (monomial ordering on Q) and $\triangleright$ (monomial ordering on $K[X]$), $\succ_b$, can be expressed in terms of an infinite matrix A, as long as:

- The monomial ordering $>$ in Q satisfies that $\forall \sigma_0 \in \Sigma \ \#\{\sigma \in \Sigma \mid \sigma < \sigma_0\} <_u \infty$.
- We already know a matrix $A' \in \mathcal{M}_{n\times n'}(\mathbb{R})$ such that $>_A$ is equal to $\triangleright$.

To do this, first, we consider the ordering $\triangleright_u$ in X such that $x_1 \triangleright_u x_2 \triangleright_u \ldots \triangleright_u x_n$, and consider the well ordering on $X(\Sigma)$ induced by $>$ and $\triangleright_u$. Second, let A to be the matrix defined by blocks of A'_{rc}. Finally, defining $\succ_A$ like in the Proposition 3.20 from $\succ_u$ and $\succ$, Proposition 3.33 stablishes that $\succ$ and $\succ_A$ are equal.

3.2. **Gröbner Bases on Σ-Polynomial Algebras.** In this subsection, we present a theory of Gröbner bases for Σ-polynomial algebras P. We start by giving similar definitions to those given in Section 1.

Definition 3.35. Let $\succ$ be a Σ-ordering on P, and $0 \neq f = \sum_{i=1}^{l} c_i m_i \in P$ where $c_i \in K$, $c_i \neq 0$, and $m_i \in M$. Then, we define:

- The leading monomial of f as $\text{LM}(f) := \max_{\succ}\{m_i\}$.
- The leading coefficient of f by $\text{LC}(f) := c_k$ where $\text{LM}(f) = m_k$ for some $k \in \{1, \ldots, n\}$.
- The leading term of f by $\text{LT}(f) := \text{LC}(f)\ \text{LM}(f)$.
- The tail of f by $\text{TAIL}(f) := f - \text{LT}(f)$.

Example 3.36. Let $\Sigma = \mathbb{N}^3$, $X = \{x_1, x_2\}$, $K = \mathbb{C}$, and $Q = \mathbb{C}[(1,0,0), (0,1,0), (0,0,1)]$. Consider

- $\succ_{\text{deglex}}$ the block Σ-ordering defined by the degree lexicographical orderings $>_{\text{deglex}}$ in Q and $\triangleright_{\text{deglex}}$ in $\mathbb{C}[X]$
- $\succ_{\text{lex}}$ the block Σ-ordering obtained by the lexicographical orderings $>_{\text{lex}}$ in Q and $\triangleright_{\text{lex}}$ in $\mathbb{C}[x_1, x_2]$
- $\succ$ the block Σ-ordering obtained by $>_{\text{lex}}$ in Q and $\triangleright_{\text{deglex}}$ in $\mathbb{C}[x_1, x_2]$.

Finally, we define the polynomial $f = 3x_1(0,2,0)^3 - 5x_3(1,0,0)^2x_2(0,1,0) + x_2(0,2,0)^2x_3(0,2,0)^2$. Since

$$\begin{gathered}(0,2,0) >_{\text{deglex}} (1,0,0), (0,1,0)\\ (1,0,0) >_{\text{lex}} (0,2,0)\\ x_2(0,2,0)^2x_3(0,2,0)^2 \triangleright_{\text{deglex}} 3x_1(0,2,0)^3\\ 3x_1(0,2,0)^3 \triangleright_{\text{lex}} x_2(0,2,0)^2x_3(0,2,0)^2.\end{gathered}$$

We have that for the different Σ-orderings $\succ_{\text{lex}}, \succ_{\text{deglex}}, \succ$, the leading coefficient, leading monomial, and leading term of f are different.

Σ-Ordering/ Terms	$\mathrm{LM}(f)$	$\mathrm{LC}(f)$	$\mathrm{LT}(f)$
$\succ_{\mathrm{lex}}$	$x_3(1,0,0)^2x_2(0,1,0)$	3	$3x_3(1,0,0)^2x_2(0,1,0)$
$\succ_{\mathrm{deglex}}$	$x_1(0,2,0)$	-5	$-5x_1(0,2,0)$
$\succ$	$x_2(0,2,0)^2x_3(0,2,0)^2$	1	$x_2(0,2,0)^2x_3(0,2,0)^2$

Since LM, LC, LT, TAIL depend on the fix Σ-ordering. Each time we consider one of them, we implicitly fix some Σ-ordering.

Remark 3.37. Let $\succ$ be a Σ-ordering on P, and $f = \sum_{i=1}^{l} c_i m_i \in P$ where $f \neq 0$. For any $\sigma \in \Sigma$, we have that $\sigma \cdot f = \sum_{i=1}^{l} (\sigma \cdot c_i)(\sigma \cdot m_i)$, hence:

$$\begin{aligned}
&\mathrm{LM}(\sigma \cdot f) = \max\{\sigma \cdot m_i\} = \sigma \cdot \max\{m_i\} = \sigma\, \mathrm{LM}(f)\\
&\mathrm{LC}(\sigma \cdot f) = \sigma \cdot \mathrm{LC}(f)\\
&\mathrm{LT}(\sigma \cdot f) = (\sigma \cdot \mathrm{LC}(f))(\sigma \cdot \mathrm{LM}(f)) = \sigma \cdot (\mathrm{LC}(f)\, \mathrm{LM}(f)) = \sigma \cdot \mathrm{LT}(f)\\
&\mathrm{TAIL}(\sigma \cdot f) = \sigma f - \mathrm{LT}(\sigma \cdot f) = \sigma \cdot \mathrm{TAIL}(f)
\end{aligned}$$

That is, the action of Σ on P commutes with LM, LC, LT and TAIL.

Notation 3.38. Let $\succ$ be a Σ-ordering on P and $G \subseteq P$, we denote

$$\mathrm{LM}(G) = \{\mathrm{LM}(f) \mid 0 \neq f \in G\}.$$

If $I \subset P$ is a Σ-ideal of P, that is, $\Sigma \cdot I \subset I$, then $\langle \mathrm{LM}(I) \rangle$ is also a Σ-ideal of P by Remark 3.37:

$$\Sigma \cdot \langle \mathrm{LM}(I) \rangle = \langle \Sigma \cdot \mathrm{LM}(I) \rangle = \langle \mathrm{LM}(\Sigma \cdot I) \rangle \subset \langle \mathrm{LM}(I) \rangle.$$

We are now ready to define a Σ-basis of a Σ-ideal. This concept in our setting of Σ-polynomial algebras and Σ-ideals is equivalent to the concept of Gröbner basis in the setting of polynomial rings.

Definition 3.39. Let $\succ$ be a Σ-ordering on P, I a Σ-ideal of P, and $G \subset I \subset P$. Then, G is a Σ-basis of I if $\mathrm{LM}(G)$ Σ-generates $\langle \mathrm{LM}(I) \rangle$ as a Σ-ideal, that is, if:

$$\langle \mathrm{LM}(G) \rangle_\Sigma = \langle \Sigma \cdot \mathrm{LM}(G) \rangle = \langle \mathrm{LM}(I) \rangle.$$

To build some intuition, let us give an example of a Σ-basis.

Example 3.40. Let $\Sigma = \mathbb{N}$, $X = \{x\}$ and $P = \mathbb{Q}[\mathbb{N}(X)]$ be the Σ-polynomial algebra given by the action

$$\begin{aligned}
\mathbb{N} \times P &\longrightarrow P\\
(n, c\,x(k)) &\longmapsto n \cdot c\,x(k) := c\,x(k+n)
\end{aligned}$$

Let also $\succ$ be the block ordering on $P = \mathbb{Q}[\mathbb{N}(X)]$ given by the degree orderings $>$ and $\triangleright$ on $\mathbb{Q}[\sigma]$ and on $\mathbb{Q}[x]$ respectively. Consider the Σ-ideal $I = \langle f_1, f_2 \rangle_\Sigma$ where

$$\begin{aligned}
f_1 &= x(2) + x(1)^3\\
f_2 &= x(1)^3 - x(1)^2.
\end{aligned}$$

We check that $\{f_1, f_2\}$ is not a Σ-basis of I, because

$$2x(1)^2 = 1 \cdot f_2 - x(2)^2 f_1 - (f_1\, q_1 + f_2\, q_2) \in I$$

where

$$q_1 = x(2)x(1)^3 + x(2) - x(1)^6 - x(1)^3$$
$$q_2 = x(1)^6 + x(1)^5 + x(1)^4 + 2\,x(1)^3 + 2\,x(1)^2 + 2\,x(1) + 2,$$

but $x(1)^2 \notin \langle \mathrm{LM}(f_1), \mathrm{LM}(f_2)\rangle_\Sigma = \langle x(2), x(1)^3\rangle_\Sigma$.

Moreover, $\{f_1, f_2, 2\,x(1)^2\}$ is a Σ-basis of I with respect to $\succ$. We will prove this in Example 3.67. We note that Σ-bases are in general not unique for a given Σ-ideal and Σ-ordering. For instance, in this example, if $\{f_1, f_2, 2\,x(1)^2\}$ is a Σ-basis, it follows by definition that the family of sets $\{f_1, f_2, a\,x(1)^2\}_{a\in\mathbb{Q}}$ are all Σ-bases of I.

Similarly to what we did in the previous section for Gröbner basis, we would like to develop an algorithm such that, given a finite Σ-generating set of a Σ-ideal, it computes a finite Σ-basis. However, this time, not all Σ-ideals of P are finitely Σ-generated because P is not Noetherian. We show this fact in the following example.

Example 3.41. Let P be the Σ-polynomial algebra as in the Example 3.40. For simplicity, we denote $x_i := x(i)$. Consider the family of Σ-ideals:

$$\begin{aligned} I_1 &= \langle x_0x_1\rangle_\Sigma \\ I_2 &= \langle x_0x_1, x_0x_2\rangle_\Sigma \\ I_3 &= \langle x_0x_1, x_0x_2, x_0x_3\rangle_\Sigma \\ &\vdots \end{aligned}$$

Let us prove that $I_{i-1} \subsetneqq I_i$. By definition $I_{i-1} \subseteq I_i$ for $i \geq 2$. We take $x_0x_i \in I_i$, and suppose that $x_0x_i \in I_{i-1}$. Then, there must exist some $1 \neq m \in \langle x_0x_1, \ldots, x_0x_{i-1}\rangle$ and some $k \in \mathbb{N}$ such that $k \cdot m \mid x_0x_i$. Since $k \cdot m \neq x_0$ and $k \cdot m \neq x_i$, the only possible option is $k \cdot m = x_0x_i$. Therefore, m must have only two variables, which means that $m \in \{x_0x_1, \ldots, x_0x_{i-1}\}$. Note that for any $j \in \{1, \ldots, i-1\}$ and any $k \in \mathbb{N}$, $k \cdot x_0x_j = x_kx_{j+k}$. Therefore

$$\{k \cdot x_0x_j \mid k \in \mathbb{N}, 1 \leq j \leq i-1\} = \{x_rx_s \mid r, s \in \mathbb{N}, 0 \leq s - r \leq i-1\}.$$

But $x_0x_i \notin \{x_rx_s \mid r, s \in \mathbb{N}, 0 \leq s - r \leq i-1\}$, which means that $x_0x_i \neq k \cdot m$ for any $k \in \mathbb{N}$ and $m \in \{x_0x_1, \ldots, x_0x_{i-1}\}$. Hence, we reached a contradiction, and we have the following strictly ascending chain of Σ-ideals.

$$I_1 \subsetneqq I_2 \subsetneqq I_3 \subsetneqq I_4 \subsetneqq \ldots$$

This implies that we can not ensure the existence of a finite Σ-basis for any Σ-ideal I, nevertheless, we continue with an approach similar to that of the first section. In Section 1, we developed an algorithm which computed a finite Gröbner basis given by Buchberger's criterion (Theorem 1.21). In this setting, we also seek a similar characterization to compute Σ-bases. To this end, we present the following results and definitions.

Proposition 3.42. *Let $\succ$ be a Σ-ordering on P, I be a Σ-ideal of P, and G be a Σ-basis of I. Then, G Σ-generates I.*

Proof. Since $G \subset I$ and I is a Σ-ideal, $\langle G\rangle_\Sigma \subset I$. Let us prove that $I \subset \langle G\rangle_\Sigma$. Consider $f \in I$ and suppose $f \notin \langle G\rangle_\Sigma$. Then, we choose $h \in \langle G\rangle_\Sigma$ such that:

$$\mathrm{LM}(f - h) = \min_{\succ}\{\mathrm{LM}(f - h') \mid h' \in \langle G\rangle_\Sigma\}.$$

Note that $f - h \neq 0$, $\forall h \in \langle G \rangle_\Sigma$ by hypothesis. Since $f, h \in I$, $f - h \in I$. So, because G is a Σ-basis of I, there exists some $g \in G$, $\sigma \in \Sigma$ and $m \in M$ such that $\operatorname{LM}(f - h) = m \operatorname{LM}(\sigma \cdot g)$. Hence, we have that $\operatorname{LM}(f - (h + m(\sigma \cdot g))) < \operatorname{LM}(f - h)$ and that $h + m(\sigma \cdot g) \in \langle G \rangle_\Sigma$, which contradicts the minimality of h. Therefore G must Σ-generate I. □

Definition 3.43. Let $\succ$ be a Σ-ordering on P, $f, g \in P$, and $L(f, g) = \operatorname{lcm}(\operatorname{LM}(f), \operatorname{LM}(g))$. We define the S-polynomial of f and g as follows:

$$\operatorname{SPOL}(f, g) := \frac{L(f, g)}{\operatorname{LT}(f)} f - \frac{L(f, g)}{\operatorname{LT}(g)} g.$$

We observe that it actually belongs to P since $\operatorname{LM}(f), \operatorname{LM}(g) \mid L(f, g)$.

Remark 3.44. Let $f, g \in P$, $\succ$ be a Σ-ordering on P, and $L(f, g) = \operatorname{lcm}(\operatorname{LM}(f), \operatorname{LM}(g))$. We observe that for any $\sigma \in \Sigma$ we have that

$$L(\sigma \cdot f, \sigma \cdot g) = \operatorname{lcm}(\sigma \cdot \operatorname{LM}(f), \sigma \cdot \operatorname{LM}(g)) = \sigma \cdot \operatorname{lcm}(\operatorname{LM}(f), \operatorname{LM}(g)),$$

because applying does not change any exponent of $\operatorname{LM}(f)$ and $\operatorname{LM}(g)$. Thus, we have that:

$$\operatorname{SPOL}(\sigma \cdot f, \sigma \cdot g) = \frac{\sigma \cdot L(f, g)}{\sigma \cdot \operatorname{LT}(f)} (\sigma \cdot f) - \frac{\sigma \cdot L(f, g)}{\sigma \cdot \operatorname{LT}(g)} (\sigma \cdot g) = \sigma \cdot \operatorname{SPOL}(f, g)$$

Definition 3.45. Let $\succ$ be a Σ-ordering on P, $G \subset P$ and $f \in P$ such that $f \neq 0$. If $f = \sum_{i=1}^{l} f_i g_i$, where $f_i \in P$, $g_i \in G$ and such that $\operatorname{LM}(f) \succeq \operatorname{LM}(f_i) \operatorname{LM}(g_i)$ for every $i \in \{1, \ldots, l\}$, then f has a Gröbner representation with respect to G.

Similarly as we did in the first section, we present a reduction algorithm that, when possible, provides a Gröbner representation of some $f \in P$, with respect to a set $G \subset P$.

Algorithm 3: Σ-Reduce

Input: A subset $G \subseteq P$, a Σ-ordering $\succ$ on P and $f \in P$ where $f \neq 0$.
Output: A set $\{f_1, \ldots, f_l, g_1, \ldots, g_l, r\} \subset P$ such that $\{g_1, \ldots, g_l\} \subset G$, $f = r + \sum_{i=1}^{l} f_i g_i$, $\operatorname{LM}(f) \succeq \operatorname{LM}(g_i) \operatorname{LM}(f_i)$, and that no $\operatorname{LM}(g_i)$ divides any monomial of r.

```
1  F := ∅, G' := ∅
2  r := 0, i := 0
3  while f ≠ 0 do
4      if ∃g ∈ G such that LM(g) | LM(f) then
5          if g ∉ G' then
6              i ← #G' + 1
7              g_i := g, f_i := 0
8              G' ← G' ∪ {g_i}
9              F ← F ∪ {f_i}
10         else
11             i ← i ∈ {1, ..., #G'} such that g = g_i
12         f_i ← f_i + LT(f)/LT(g)
13         f ← f − LT(f)/LT(g) g
14     else
15         r ← r + LT(f)
16         f ← f − LT(f)
17 return F ∪ G' ∪ {r}
```

Let $f \in P$, $G \subset P$, and $\succ$ a Σ-ordering on P. To reduce f with respect to G refers to computing Σ-Reduce(f,$\succ$,G). Moreover, the remainder of reducing f with respect to G refers to the "r" yield by computing Σ-Reduce(f,$\succ$,G).

Remark 3.46. It is worth noting a few things about Algorithm 3.

- The condition in line 4 can be computed even if G is not finite: since $\operatorname{LM}(f) \in M$ has finitely many divisors in M, we can perform this step by traversing G according to a Σ-ordering. This permits us to restrict the research to just a finite set of monomials smaller than $\operatorname{LM}(f)$.
- The algorithm stops after a finite number of steps because in each iteration inside the **while** $\operatorname{LM}(f)$ gets strictly reduced and because $\succ$ is a well ordering on M.
- By definition of the algorithm, no monomial of the remainder can divide $\operatorname{LM}(g_i)$ for any $i \in \{1, \ldots, l\}$. So, no monomial of r is in $\langle \operatorname{LM}(G) \rangle$. Moreover, if the set G is Σ-invariant, that is, if $\Sigma \cdot G \subset G$. Then, no monomial of r is in $\langle \operatorname{LM}(\Sigma \cdot G) \rangle = \langle \operatorname{LM}(G) \rangle_\Sigma$.
- If G is a Σ basis of I and we reduce $f \in I$ with respect to $\Sigma \cdot G$, we obtain $r = 0$. This follows from the fact that, if $r \neq 0$, since $r \in I$, $\operatorname{LM}(r) \in \langle \operatorname{LM}(I) \rangle = \langle \operatorname{LM}(G) \rangle_\Sigma$. This contradicts that no monomial of r is in $\langle \operatorname{LM}(G) \rangle_\Sigma$. Conversely, if we get a zero remainder after reducing $f \in P$ with respect to $\Sigma \cdot G$, then $p \in I$. Therefore, if G is a Σ-basis of

I, the remainder of reducing f with respect to $\Sigma \cdot G$ is 0 if and only if $p \in I$.

By the previous points, if $f \in I$ and G is a Σ-basis of I, then f has a Gröbner representation with respect to $\Sigma \cdot G$ given by Algorithm 3.

Notation 3.47. Let $\sigma, \tau \in \Sigma = \langle \sigma_1, \ldots, \sigma_r \rangle$, where

$$\sigma = \sum_{i=1}^{r} \alpha_i \sigma_i, \qquad \tau = \sum_{i=1}^{r} \beta_i \sigma_i,$$

and $\alpha_i, \beta_i \in \mathbb{N}$. Consider $\gamma_i = \min\{\alpha_i, \beta_i\}$ for any $i \in \{1, \ldots, r\}$. We denote

$$\gcd(\sigma, \tau) = \sum_{i=1}^{r} \gamma_i \sigma_i.$$

It is clear that most of the introduced concepts this far (LM, LT, LC, TAIL, SPOL) are completely analogous to those introduced in the first section (lm, lt, lc, tail, S). Moreover, we have shown that all of these operations commute with the action by Σ. This is useful to prove the following theorem, which is very similar to Buchberger's Criterion (Theorem 1.21).

Theorem 3.48 (Σ-Criterion)**.** *Let $\succ$ be a Σ-ordering on P, and $\langle G \rangle_\Sigma = I \subset P$ a Σ-ideal finitely Σ-generated by G. Then, G is a Σ-basis of I if and only if for every non-zero $f, g \in G$ and $\sigma, \tau \in \Sigma$ such that $\gcd(\sigma, \tau) = 0$ and $\gcd(\sigma \cdot \mathrm{LM}(f), \tau \cdot \mathrm{LM}(g)) \neq 1$ we have that $\mathrm{SPOL}(\sigma \cdot f, \tau \cdot g)$ has a Gröbner representation with respect to $\Sigma \cdot G$.*

Proof. $\Longrightarrow$) Suppose that G is a Σ-basis of I. Since any $\mathrm{SPOL}(\sigma \cdot f, \tau \cdot g) \in I$ for any non-zero $g, f \in I$ and $\sigma, \tau \in \Sigma$. There exists a Gröbner representation of $\mathrm{SPOL}(\sigma \cdot f, \tau \cdot g)$ with respect to $\Sigma \cdot G$ given by the Algorithm 3.

$\Longleftarrow$) Since $\succ$ is a Σ ordering, LM, LT, SPOL, $\max_\succ$, $\min_\succ$, and TAIL commute with respect to the action given by Σ. Proceeding like in the proofs of the Lemma 1.20, of the Product Criterion (Proposition 1.19) and of the implication "$\Leftarrow$)" in Buchberger's Criterion (Theroem 1.21), we get that: if for every non-zero $f, g \in G$ and $\sigma, \tau \in \Sigma$ such that $\gcd(\sigma \cdot \mathrm{LM}(f), \tau \cdot \mathrm{LM}(g)) \neq 1$, $\mathrm{SPOL}(\sigma \cdot f, \tau \cdot g)$ has a Gröbner representation with respect to $\Sigma \cdot G$, then, G is a Σ-basis of I.

Hence, we now want to prove that if this condition holds for $\sigma, \tau \in \Sigma$ such that $\gcd(\sigma, \tau) = 0$, then it also holds for any $\sigma, \tau \in \Sigma$. Suppose that we have a Gröbner representation of $\mathrm{SPOL}(\sigma \cdot f, \tau \cdot g)$ with respect to $\Sigma \cdot G$, for every $f, g \in G$ and $\sigma, \tau \in \Sigma$ such that $\gcd(\sigma, \tau) = 0$. Consider $\sigma', \tau' \in \Sigma$ such that $\gcd(\sigma', \tau') \neq 0$. Then, there exists $\delta \in \Sigma$ non-zero, such that $\sigma' = \delta + \sigma''$, $\tau' = \delta + \tau''$ and $\gcd(\sigma'', \tau'') = 0$. By hypothesis, there is a Gröbner representation of $\mathrm{SPOL}(\sigma'' \cdot f, \tau'' \cdot g)$ with respect to $\Sigma \cdot G$:

$$\mathrm{SPOL}(\sigma'' \cdot f, \tau'' \cdot g) = \sum_{i=1}^{l} f_i(\gamma_i \cdot g_i)$$

where $\gamma_i \in \Sigma, g_i \in G$, $f_i \in P$ and such that

$$\mathrm{LM}(\mathrm{SPOL}(\sigma'' \cdot f, \tau'' \cdot g)) \succeq \mathrm{LM}(f_i)\,\mathrm{LM}(\gamma_i \cdot g_i).$$

Hence,

$$\textsc{spol}(\sigma' \cdot f, \tau' \cdot g) = \delta \sum_{i=1}^{l} f_i(\gamma_i \cdot g_i) = \sum_{i=1}^{l} (\delta \cdot f_i)((\delta + \gamma_i) \cdot g_i)$$

is a Gröbner representation of $\textsc{spol}(\sigma' \cdot f, \tau' \cdot g)$ since

$$\begin{aligned} \textsc{lm}(\textsc{spol}(\sigma' \cdot f, \tau' \cdot g)) &= \delta \,\textsc{lm}(\textsc{spol}(\sigma'' \cdot f, \tau'' \cdot g)) \succeq \\ &\succeq \delta \,\textsc{lm}(f_i)\,\textsc{lm}(\gamma_i \cdot g_i) = \textsc{lm}(\delta \cdot f_i)\,\textsc{lm}((\delta + \gamma_i) \cdot g_i). \end{aligned}$$

□

The Σ-criterion (Theorem 3.48) leads us to a procedure for computing a Σ-basis. We call it a procedure and not an algorithm because, in general, we can not ensure it eventually stops.

Procedure(Σ-Basis)

Input: A finite Σ-generating set H of a Σ-ideal $I \subset P$, and a Σ-ordering $\succ$.

Output: G a Σ-basis of I

```
1  G := {g ∈ H | g ≠ 0};
2  G' := G;
3  r := 0;
4  repeat
5  |  G ← G';
6  |  for σ, τ ∈ Σ, f, g ∈ G such that gcd(σ, τ) = 0,
   |   gcd(σ · LM(f), τ · LM(g)) ≠ 1 do
7  |  |  r ← Σ-Reduce'(SPOL(σ · f, τ · g),≻,Σ · G);
8  |  |  if r ≠ 0 then
9  |  |  |  G' ← G' ∪ {r};
   |  |  └
   |  └
10 until G = G';
11 return G;
```

Remark 3.49. We observe a few things about this procedure:

- The algorithm "Σ-Reduce' " used in line 7 is the variant of the algorithm Σ-reduce which returns only the remainder r.
- If the procedure stops after a finite number of steps, then G is a Σ-basis of I. This happens because if $G = G'$, then, for every $f, g \in G$ and $\sigma, \tau \in \Sigma$ such that $\gcd(\sigma\,\textsc{lm}(f), \tau\,\textsc{lm}(g)) \neq 1$ and $\gcd(\sigma, \tau) = 0$:

  $$0 = \Sigma\text{-Reduce'}(\textsc{spol}(\sigma f, \tau g), \succ, \Sigma \cdot G).$$

  Hence, for such $f, g \in G$ and $\sigma, \tau \in \Sigma$, $\textsc{spol}(\sigma f, \tau g)$ has a Gröbner representation with respect to $\Sigma \cdot G$; thus, by the Σ-criterion, G is a Σ-basis of I.
- We can not ensure that the procedure stops in general, since there exist infinitely generated monomial Σ-ideals as we saw in Example 3.41.
- When computing the S-polynomials, the case $f = g$ may be considered whenever $\sigma \neq \tau$.

- It may seem that line 5 can not be performed because there could be infinite pairs $\sigma, \delta \in \Sigma$ such that $\gcd(\sigma, \delta)$ and $\gcd(\sigma f, \tau g) \neq 1$. However, the possibilities for such $\sigma, \tau \in \Sigma$ are finite. We provide a proof of this fact in the next section in Remark 3.65.

We observe that the main problem for computing a finite Σ-basis of a finitely generated ideal I, is that the monomial Σ-ideal $\langle \text{LM}(I) \rangle$ may not be finitely generated, so clearly, it is impossible to compute a finite generating set of a Σ-ideal if such a set does not exist. The next part of this section introduces an approach to address this problem.

3.3. **Order Function and Truncation.** In this part, we present a notion of order on P with the intention to provide an algorithm to compute finite Σ-bases on a subalgebra of P, where the number of variables is bounded and the Noetherianity is guaranteed.

Denote by $\overline{\mathbb{N}} := \mathbb{N} \cup \{-\infty\}$.

Definition 3.50. We define $(R, +, \cdot)$ as a commutative semi-ring if both operations are associative, commutative, have an identity element, and if the product "$\cdot$" distributes over the sum "$+$".

Definition 3.51. We extend the maximum and the addition in $\mathbb{N}$ to $\overline{\mathbb{N}}$ as usual, where $(-\infty) + a = a + (-\infty) = -\infty$, $(-\infty) + (-\infty) = -\infty$, and $\max\{-\infty, a\} = \max\{a, -\infty\} = a$ and $\max\{-\infty, -\infty\} = -\infty$ for any $a \in \mathbb{N}$.

Since $+$ and max are associative and commutative in $\overline{\mathbb{N}}$, and $+$ distributes with respect to max, $(\overline{\mathbb{N}}, \max, +)$ is a commutative semi-ring.

Definition 3.52. Let $\sigma = \sum_{i=1}^{r} \alpha_i \sigma_i \in \Sigma$ with $\alpha_i \in \mathbb{N}$. Then, we define

$$\deg(\sigma) := \sum_{i=1}^{r} \alpha_i \in \mathbb{N}.$$

Let $\delta \in \Sigma$, and $m \in M$. Making an abuse of notation, we write $\delta \in \text{supp}(m)$ if there exists an $x \in X$ such that $x(\delta) \in \text{supp}(m)$.

The next definition is of particular importance because it introduces the notion of order on P that we use to bound the set of variables of P. To give this definition, we require the following map:

$$\begin{aligned} \text{ord}\colon M &\longrightarrow \overline{\mathbb{N}} \\ m &\longmapsto \begin{cases} -\infty & \text{if } m = 1 \\ \max\limits_{\delta \in \text{supp}(m)} \{\deg(\delta)\} & \text{if } m \neq 1. \end{cases} \end{aligned}$$

Definition 3.53. Let $p \in P$, we define the order of p as follows:

- If $p \neq 0$, we write $p = \sum_{i=1}^{l} c_i m_i \in P$ with $m_i \in M$ and $c_i \in K$ different than zero, where all monomials m_i of the sum are different. Then,

$$\text{ord}(p) := \max_{1 \leq i \leq l} \{\text{ord}(m_i)\}.$$

- If $p = 0$, $\text{ord}(p) := -\infty$.

Moreover, we call the map $\text{ord}\colon P \to \overline{\mathbb{N}}$ the order function of P.

If $p \neq 0$, there is a unique way to write $p = \sum_{i=1}^{l} c_i m_i \in P$ with $m_i \in M$, $c_i \in K$ different than zero, and where all monomials m_i are different (up to reordering the summands), hence, the $\mathrm{ord}(p)$ is well defined.

The following proposition is useful for computations involving the order function of P.

Proposition 3.54. *Let $m, m' \in M$, and $\sigma \in \Sigma$, then, the map* $\mathrm{ord}\colon M \to \overline{\mathbb{N}}$ *satisfies that*

(1) $\mathrm{ord}(m\,m') = \max\{\mathrm{ord}(m), \mathrm{ord}(m')\}$
(2) $\mathrm{ord}(\sigma \cdot m) = \deg(\sigma) + \mathrm{ord}(m)$
(3) $\mathrm{ord}(\mathrm{lcm}(m, m')) = \mathrm{ord}(m\,m')$.

Proof. (1) Note that $\mathrm{supp}(m) \cup \mathrm{supp}(m') = \mathrm{supp}(m\,m')$, therefore

$$\mathrm{ord}(m\,m') = \max_{\delta \in \mathrm{supp}(m\,m')}\{\deg(\delta)\} = \max_{\delta \in \mathrm{supp}(m) \cup \mathrm{supp}(m')}\{\deg(\delta)\} =$$

$$= \max\left\{\max_{\delta \in \mathrm{supp}(m)}\{\deg(\delta)\}, \max_{\delta \in \mathrm{supp}(m')}\{\deg(\delta)\}\right\} = \max\{\mathrm{ord}(m), \mathrm{ord}(m')\}$$

(2) First, we observe that $\deg(\sigma + \delta) = \deg(\sigma) + \deg(\delta)$ for any $\sigma, \delta \in \Sigma$, by definition of deg. Let now $\sigma \in \Sigma$, and $m \in M$. If $m = 1 \in K$, then

$$\deg(\sigma 1) = \deg(1) = -\infty = \deg(\sigma) - \infty = \deg(\sigma) + \mathrm{ord}(1).$$

If $m \neq 1$, we have that

$$\mathrm{ord}(\sigma \cdot m) = \max_{\sigma + \delta \in \mathrm{supp}(\sigma m)}\{\deg(\sigma + \delta)\} = \max_{\delta \in \mathrm{supp}(m)}\{\deg(\sigma) + \deg(\delta)\} =$$
$$= \deg(\sigma) + \max_{\delta \in \mathrm{supp}(m)}\{\deg(\delta)\} = \deg(\sigma) + \mathrm{ord}(m).$$

(3) Let $m, m' \in M$. Note that $\mathrm{supp}(m\,m') = \mathrm{supp}(\mathrm{lcm}(m, m'))$. Hence,

$$\mathrm{ord}(\mathrm{lcm}(m, m')) = \max_{\delta \in \mathrm{supp}(\mathrm{lcm}(m, m'))}\{\deg(\delta)\} =$$
$$= \max_{\delta \in \mathrm{supp}(m\,m')}\{\deg(\delta)\} = \mathrm{ord}(m\,m').$$

□

In what follows, we give the necessary notions to present a truncated version of the Procedure Σ-basis with respect to the order function. This algorithm will compute finite Σ-bases of a Σ-ideal inside a given sub K-algebra of P, where this sub K-algebra consists of all polynomials in P whose order is bounded by a given positive integer.

Let us first define the notion of a Σ-basis of a Σ-ideal inside a K-algebra of P.

Definition 3.55. Let I be a Σ-ideal of P, and $P' = K[Y]$ (with $Y \subset X(\Sigma)$) be a free K-subalgebra of P. The intersection $I \cap P'$ is an ideal inside the K-subalgebra P'.

- Then, $G \subset I \cap P'$ Σ-generates $I \cap P'$ in P' if $\langle G\rangle_\Sigma \cap P' = I \cap P'$.
- We define G to be a Σ-basis of $I \cap P'$ in P' if $\langle \mathrm{LM}(G)\rangle_\Sigma \cap P' = \langle \mathrm{LM}(I)\rangle \cap P'$.

Definition 3.56. Let $p = \sum_{i=1}^{l} c_i m_i \in P$ where $c_i \in K$, $c_i \neq 0$ and $m_i \in M$. We define $\operatorname{supp}(p) = \bigcup_{i=1}^{l} \operatorname{supp}(m_i)$.

Notation 3.57. We denote by $\Sigma_{(d)} := \{\sigma \in \Sigma \mid \deg(\sigma) \leq d\}$.

Definition 3.58. Let $d \in \overline{\mathbb{N}}$ and H a subset of P. We denote by

$$H_{(d)} := \{f \in H \mid \operatorname{ord}(f) \leq d\},$$

Definition 3.59. Let $Y_{(d)} := \{x(\sigma) \in X(\Sigma) \mid \operatorname{ord}(\sigma) \leq d\}$. We define the sub-$K$-algebra $P_{(d)} := K[Y_{(d)}] \subset P$.

Note that if I is a Σ-ideal and $\langle H \rangle_\Sigma = I$, then $H_{(d)}$ Σ-generates $I \cap P_{(d)}$ in $P_{(d)}$. That is, $\langle H_{(d)} \rangle_\Sigma \cap P_{(d)} = I \cap P_{(d)}$.

Since $P_{(d)}$ has finitely many variables, it is Noetherian due to Hilbert's basis Theorem ([4], Chapter 2, Theorem 4). Thus, we follow the same approach as in Section 1 to compute Σ-bases in $P_{(d)}$. To introduce this computation, we need a Σ-ordering compatible with the order function.

Definition 3.60. Let $\succ$ be a Σ-ordering on P. Then, $\succ$ is compatible with the order function if $\operatorname{ord}(m) < \operatorname{ord}(n) \implies m \prec n$.

In the next example, we present a Σ-ordering compatible with the order function.

Example 3.61. Let $>_{\text{deglex}}$ be the degree lexicographical ordering on Q and $\triangleright_{\text{degrlex}}$ be the reverse lexicographical ordering on $K[x_1, \ldots, x_n]$. We consider the block ordering $\succ_b$ on P induced by $>$ and $\triangleright$. Let us prove that $\succ_b$ is compatible with the order function. Suppose $\operatorname{ord}(m) > \operatorname{ord}(m')$. We write

$$m = m_1 \ldots m_k \qquad m' = m'_1 \ldots m'_k$$

where each $m_i, m'_i \in P(\delta_i)$ for some $\delta_i \in \Sigma$, in such a way that $\delta_1 > \ldots > \delta_n$. Then, $\max_{\delta \in \operatorname{supp}(m)}\{\deg(\delta)\} > \max_{\tau \in \operatorname{supp}(m')}\{\deg(\tau)\}$, so, taking $\delta' = \max_{\delta \in \operatorname{supp}(m)}\{\deg(\delta)\}$, $\forall \tau \in \operatorname{supp}(m')$, $\delta >_{\text{deglex}} \tau$. Therefore, $m_1, m'_1 \in P(\delta')$ and $m'_1 = 1$, thus, $m_1 \triangleright_{\text{degrlex}} m'_1$ and so $m \succ_b m'$.

In general, if $>$ and $\triangleright$ are monomial orderings on $K[\sigma_1, \ldots, \sigma_r]$ and $K[x_1, \ldots, x_n]$, such that $>$ satisfies that $\deg(\delta) > \deg(\tau) \implies \delta > \tau$ $\forall \sigma, \tau \in \Sigma$, then any block Σ-ordering obtained by $>$ and $\triangleright$ is compatible with the order function. Moreover, every such order $>$ on Q satisfies that $\forall \sigma_0 \in \Sigma$ $\#\{\sigma \in \Sigma \mid \sigma < \sigma_0\} < \infty$, therefore, every block ordering compatible with the order function can be represented by a matrix as in Proposition 3.33.

We now give a different version of the Σ-criterion (Theorem 3.48), which allow us to present an algorithm for computing Σ-bases in the subalgebra $P_{(d)}$, where $d \in \overline{\mathbb{N}}$.

Theorem 3.62 (Truncated Σ-criterion)**.** *Let $\succ$ be a Σ-ordering compatible with the order function on P, I be a finitely Σ-generated Σ-ideal by $G \subset I$, and $d \in \overline{\mathbb{N}}$. Then, $G_{(d)}$ is a Σ-basis of $I \cap P_{(d)}$ in $P_{(d)}$ if and only if for all non-zero $f, g \in G_{(d)}$ and for any $\sigma, \tau \in \Sigma$ such that $\gcd(\sigma, \tau) = 0$, $\gcd(\sigma \cdot \textsc{lm}(f), \tau \cdot \textsc{lm}(g)) \neq 1$ and $\operatorname{ord}(\textsc{spol}(\sigma \cdot f, \tau \cdot g)) \leq d$, $\textsc{spol}(\sigma \cdot f, \tau \cdot g)$ has a Gröbner representation with respect to $\Sigma_{(d)} \cdot G_{(d)}$.*

Proof. $\Longrightarrow$) Suppose $G_{(d)}$ is a Σ-basis of $I \cap P_{(d)}$ in $P_{(d)}$. Let $f, g \in G_{(d)}$, then any S-polynomial $\text{SPOL}(\sigma \cdot f, \tau \cdot g)$ with order smaller that d, must be in $I \cap P_{(d)}$. By computing Σ-reduce($\text{SPOL}(\sigma \cdot f, \tau \cdot g$,$\succ$,$\Sigma \cdot G$) we obtain a Gröbner representation of $\text{SPOL}(\sigma \cdot f, \tau \cdot g)$ with respect to $\Sigma \cdot G$. Moreover, since $\text{ord}(\text{SPOL}(\sigma \cdot f, \tau \cdot g)) \leq d$, it is actually a Gröbner representation with respect to $\Sigma_{(d)} \cdot G_{(d)}$.

$\Longleftarrow$) First note that $G_{(d)}$ Σ-generates $I \cap P_{(d)}$ in $P_{(d)}$. Proceeding as in the proof given of the left implication of the Σ-criterion, but restricting it to polynomials in the K-subalgebra $P_{(d)}$, we end up proving that

$$\langle \text{LM}(G_{(d)}) \rangle_\Sigma \cap P_{(d)} = \langle \text{LM}(I \cap P_{(d)}) \rangle \cap P_{(d)}.$$

Let us proof the following equality

$$\langle \text{LM}(I \cap P_{(d)}) \rangle \cap P_{(d)} = \langle \text{LM}(I) \rangle \cap P_{(d)}.$$

Since the inclusion from left to right is trivial, we prove the other inclusion. Let $m \in \langle \text{LM}(I) \rangle \cap P_{(d)}$, then there exists an $f \in I$ such that $m \mid \text{LM}(f) \in P_{(d)}$. Since $\succ$ is compatible with the order function, $f \in P_{(d)}$, hence, $m \in \langle \text{LM}(I \cap P_{(d)}) \rangle$. □

Inspired by this result, we give the following algorithm.

Algorithm 4: Truncated Σ-Basis

Input: $d \in \overline{\mathbb{N}}$, H a Σ-generating set of a Σ-ideal $I \subset P$, such that $H_{(d)}$ is finite, and a Σ-ordering compatible with the order function $\succ$.

Output: $G_{(d)}$ a finite Σ-basis of $I_{(d)}$ in $P_{(d)}$

```
1  G_(d) := {g ∈ H_(d) | g ≠ 0} ;
2  G'_(d) := G_(d);
3  r:=0;
4  repeat
5  |  G_(d) ← G'_(d);
6  |  for f, g ∈ G_(d) σ, τ ∈ Σ such that gcd(σ, τ) = 0,
   |   gcd(σ · LM(f), τ · LM(g)) ≠ 1, and ord(SPOL(σ · f, τ · g)) ≤ d do
7  |  |  r ← Σ-reduce'(SPOL(σ · f, τ · g)),≻,Σ_(d) · G_(d));
8  |  |  if r ≠ 0 then
9  |  |  |  G'_(d) := G'_(d) ∪ {r};
10 until G_(d) = G'_(d);
11 return G_(d);
```

Remark 3.63.

- The algorithm eventually stops because of the Noetherianity of $P_{(d)}$: suppose that the algorithm does not stop. Notice that all the computations are done in $P_{(d)}$, and let us denote by $G_{(d),i}$ the set "$G_{(d)}$" the i-th iteration inside the **repeat-until**. If the algorithm does not stop, for any $i \geq 0$, there exist some $r_i \in I$ such that $\text{LM}(r_i) \notin \langle \text{LM}(G_{(d),i}) \rangle_\Sigma \cap P_{(d)}$ and $\text{LM}(r_i) \in \langle \text{LM}(G_{(d),i+1}) \rangle_\Sigma \cap P_{(d)}$. Hence, we have a strictly increasing

chain

$$\langle \mathrm{LM}(G_{(d),0})\rangle_\Sigma \cap P_{(d)} \subsetneq \langle \mathrm{LM}(G_{(d),1})\rangle_\Sigma \cap P_{(d)} \subsetneq \langle \mathrm{LM}(G_{(d),2})\rangle_\Sigma \cap P_{(d)} \subsetneq \dots$$

of ideals in $P_{(d)}$, which contradicts the Noetherianity of $P_{(d)}$.

- When the algorithm stops, every S-polynomial $\mathrm{SPOL}(\sigma \cdot f, \tau \cdot g)$ with $f, g \in G_{(d)}$, $\sigma, \tau \in \Sigma$ and such that $\gcd(\sigma, \tau) = 0$, $\gcd(\sigma \cdot \mathrm{LM}(f), \tau \cdot \mathrm{LM}(g)) \neq 1$, and $\mathrm{ord}(\mathrm{SPOL}(\sigma \cdot f, \tau \cdot g)) \leq d$ has a Gröbner representation with respect to $\Sigma_{(d)} \cdot G_{(d)}$. Therefore, by the Truncated Σ-criterion (Theorem 3.48), $G_{(d)}$ is a Σ-basis of $I \cap P_{(d)}$.

Until now, we were unable to solve the membership problem for finitely generated Σ-ideals. However, the preceding steps provide a method to resolve it: let H be a finite set that Σ-generates I.

(1) Set $d = \mathrm{ord}(p)$, and consider H_d, which is finite because H is finite.
(2) Compute a Σ-basis $G_{(d)}$ of $I \cap P_{(d)}$ in $P_{(d)}$ using the "Truncated Σ-Basis" algorithm.
(3) Compute Σ-reduce'$(p, \succ, \Sigma \cdot G_{(d)})$. Since

$$\langle \mathrm{LM}(G_{(d)})\rangle_\Sigma \cap P_{(d)} = \langle \mathrm{LM}(I)\rangle \cap P_{(d)},$$

the remainder is zero if and only if $p \in I$.

We now present a refined version of the Σ-criterion for Σ-orderings compatible with the order function, with the intention to give a final algorithm that computes a finite Σ-basis of a finitely generated Σ-ideal $I \subset P$ when $\langle \mathrm{LM}(I)\rangle_\Sigma$ is finitely generated.

Theorem 3.64 (Finite Σ-Criterion)**.** *Let $\succ$ be a Σ-ordering on P compatible with the order function, $I \subset P$ a Σ-ideal Σ-generated by $G \subseteq I$, and $d = \max\{\mathrm{ord}(\mathrm{LM}(g)) \mid g \in G, g \neq 0\}$. Then, G is a Σ-basis of I if and only if for every non-zero $f, g \in G$ and $\sigma, \tau \in \Sigma$ such that $\gcd(\sigma, \tau) = 0$ and $\gcd(\sigma \cdot \mathrm{LM}(f), \tau \cdot \mathrm{LM}(g)) \neq 1$ we have that $\mathrm{SPOL}(\sigma \cdot f, \tau \cdot g)$ has a Gröbner representation with respect to $\Sigma_{(2d)} \cdot G$.*

Proof. Thanks to the Σ-Criterion, we know that the statement holds for a Gröbner representation with respect to $\Sigma \cdot G$. Let us prove that any Gröbner representation of an S-polynomial $\mathrm{SPOL}(\sigma \cdot f, \tau \cdot g)$ with respect to $\Sigma \cdot G$ and such that $\gcd(\sigma, \tau) = 0$, $\gcd(\sigma \cdot \mathrm{LM}(f), \tau \cdot \mathrm{LM}(g)) \neq 1$; is actually a Gröbner representation with respect to $\Sigma_{(2d)} \cdot G$. Let

$$\mathrm{SPOL}(\sigma \cdot f, \tau \cdot g) = \sum_\nu f_\nu(\nu \cdot g_\nu)$$

be a Gröbner representation with respect to $\Sigma \cdot G$, we prove that its order is smaller than $2d$. Since $\gcd(\sigma \cdot \mathrm{LM}(f), \tau \cdot \mathrm{LM}(g)) \neq 1$, there must exist some $x_i(\sigma + \gamma) = x_i(\tau + \delta) \in \mathrm{supp}(\sigma \cdot \mathrm{LM}(f)) \cap \mathrm{supp}(\tau \cdot \mathrm{LM}(g))$ so that $x_i(\gamma) \in \mathrm{supp}(\mathrm{LM}(f))$, $x_i(\delta) \in \mathrm{supp}(\mathrm{LM}(g))$. Hence, $\sigma + \gamma = \tau + \delta$, $\deg(\gamma) \leq \mathrm{ord}(\mathrm{LM}(f)) \leq d$ and $\deg(\delta) \leq \mathrm{ord}(\mathrm{LM}(g)) \leq d$.

Since $\gcd(\sigma, \tau) = 0$ and $\sigma + \gamma = \tau + \delta$, there exists $\sigma', \tau' \in \Sigma$ such that $\sigma + \sigma' = \delta$ and $\tau + \tau' = \gamma$, therefore, $\deg(\sigma) \leq d$ and $\deg(\tau) \leq d$. Now let $v = \mathrm{lcm}(\sigma \cdot \mathrm{LM}(f), \tau \cdot \mathrm{LM}(g))$, by the Proposition 3.54 we have that

$$\mathrm{ord}(v) = \max\{\deg(\sigma) + \mathrm{ord}(\mathrm{LM}(f)), \deg(\tau) + \mathrm{ord}(\mathrm{LM}(g))\} \leq 2\,d.$$

Finally, since $v \succ \mathrm{LM}(\mathrm{SPOL}(\sigma \cdot f, \tau \cdot g) \succeq \nu\, \mathrm{LM}(g_\nu)$ for every ν, and $\succ$ is compatible with the order function, $\deg(\nu) \leq \deg(\nu) + \mathrm{ord}(\mathrm{LM}(g_\nu)) < \mathrm{ord}(v) \leq 2\,d$. □

Remark 3.65. From this proof, we can see that if $\gcd(\sigma, \tau) = 0$ and $\gcd(\sigma \cdot f, \tau \cdot g) \neq 1$, then, $\deg(\sigma), \deg(\tau) \leq d$ where $d = \max\{\mathrm{ord}(\mathrm{LM}(g)) \mid g \in G, g \neq 0\}$. Hence, only finitely many $\sigma, \tau \in \Sigma$ satisfy that $\gcd(\sigma, \tau)$ and $\gcd(\sigma \cdot f, \tau \cdot g) \neq 1$, because $\sigma, \tau \in \Sigma_{(d)}$. So the line 6 in the Algorithm Σ-Basis can actually be performed. Note that this part of the proof does not use the hypothesis of $\succ$ being compatible with the order function.

Thanks to this result, we present the following algorithm.

Algorithm 5: Finite Σ-Basis

Input: $\succ$ a Σ ordering compatible with the order function of P, I a Σ-ideal finitely Σ-generated by H, such that $\langle \mathrm{LM}(I) \rangle$ is finitely Σ-generated

Output: A Σ-basis G of I.

```
1  G := {g ∈ H | g ≠ 0};
2  d' := −∞;
3  d := max{ord(LM(g) | g ∈ G};
4  while d'<d do
5      d' ← d;
6      G', G ←Truncated Σ-basis(d,G,≻) ;
7      for σ, τ ∈ Σ_(d), f, g ∈ G do
8          r ← Σ-reduce'(SPOL(σ · f, τ · g), ≻, Σ_(2d) · G);
9          if r ≠ 0 then
10             G' ← G' ∪ {r};
11     G ← G';
12     d ← max{ord(LM(g) | g ∈ G};
13 return G;
```

Proposition 3.66. *Algorithm Finite Σ-Basis stops after a finite number of steps, and it returns a finite Σ-basis.*

Proof. We first prove that if it stops, it returns a Σ-basis of I. Suppose that in some iteration we get $d = d'$, let r be some remainder in such iteration, then $r \in I \cap P_{(d)}$. But, since G is a Σ-basis of $I \cap P_{(d)}$ inside $P_{(d)}$, the reduction of r with respect to $\Sigma_{(2d)} \cdot G$ is zero, so $r = 0$. Therefore, all S-polynomials $\mathrm{SPOL}(\sigma \cdot f, \tau \cdot g)$ have a Gröbner representation with respect to $\Sigma_{(2d)} \cdot G$, by the finite Σ-criterion, G is a Σ-basis of I.

Let us prove it eventually stops. Let $\{\mathrm{LM}(f_1), \ldots, \mathrm{LM}(f_k)\}$ be a set of Σ-generators of $\langle \mathrm{LM}(I) \rangle$, $D = \max_{1 \leq i \leq k}\{\mathrm{ord}(\mathrm{LM}(f_i))\}$, and G be the Σ-basis of $I \cap P_{(d)}$ in $P_{(d)}$.

Then

$$\mathrm{LM}(f_1), \ldots, \mathrm{LM}(f_k) \in \langle \mathrm{LM}(I) \rangle \cap P_{(d)} = \langle \mathrm{LM}(G) \rangle_\Sigma \cap P_{(d)}.$$

for any $d \geq D$. Hence, inside the **while**, when we reach a d such that $d \geq D$, we have that

$$\langle \mathrm{LM}(I) \rangle = \langle \mathrm{LM}(f_1), \ldots, \mathrm{LM}(f_k) \rangle_\Sigma = \langle \mathrm{LM}(G) \rangle_\Sigma.$$

Therefore, G is a Σ-basis of I. By the Finite Σ-criterion, every S-polynomial in this iteration has a Gröbner representation with respect to $\Sigma_{(2d)} \cdot G$, hence the remainders are all zero and the algorithm stops. □

We know introduce an example of a computation of Σ-basis.

Example 3.67. Let us recover the setting in Example 3.40, where $\Sigma = \mathbb{N}$, $X = \{x\}$, $P = \mathbb{Q}[X(\mathbb{N})]$ and $\succ$ is a block ordering given by the degree orderings in $K[\sigma]$ and $K[x]$, which is compatible with the order function. We considered $I = \langle f_1, f_2 \rangle_\Sigma$, where $f_1 = x(2) + x(1)^3$ and $f_2 = x(1)^3 - x(1)^2$, and we observed that $\{f_1, f_2\}$ was not a Σ-basis of I. To compute a Σ-basis of I, we first aim to compute a Σ-basis of $I \cap P_{(2)}$ on $P_{(2)}$. If we follow the steps given in the algorithm "Truncated Σ-basis", the only S-polynomial that satisfies the conditions to be reduced is $p = (1) \cdot f_2 - x(2)^2 f_1 = x(1)^3 x(2)^2 - x(2)^2$. After reducing p with respect to $\Sigma_{(2)} \cdot \{f_1, f_2\}$, we get a non-zero remainder, namely, $2\,x(1)^2 =: f_3$, so, we add it to our basis $\{f_1, f_2, f_3\}$.

Next, we check if $\{f_1, f_2, f_3\}$ is a Σ-basis of $I \cap P_{(2)}$ in $P_{(2)}$. The only S-polynomials satisfying the conditions to be reduced are $-x(1)^2, x(1)^3 x(2)^2 - x(2)^2, x(1)^3 x(1)^2$, which return a zero remainder when reduced with respect to $\Sigma_{(2)} \cdot \{f_1, f_2, f_3\}$. Thus, $\{f_1, f_2, f_3\}$ is a Σ-basis of $I \cap P_{(2)}$ in $P_{(2)}$.

To finally check if $\{f_1, f_2, f_3\}$ is a Σ-basis of I, we reduce the rest of the S-polynomials. There is no S-polynomial that yields a non-zero remainder, therefore $\{f_1, f_2, f_3\}$ is a Σ-basis of I. Furthermore, since $\langle x(2), x(1)^3, x(1)^2 \rangle_\Sigma = \langle x(2), x(1)^2 \rangle_\Sigma$, the element f_2 is not necessary for the Σ-basis, and so we can reduce it to $\{f_1, f_3\}$.

Finally, it is worth giving a remark about the additional hypotheses we are using, namely, that $\langle \mathrm{LM}(I) \rangle$ is finitely generated. The main obstacle we have to compute a finite Σ-basis, is that there exist finitely Σ-generated Σ-ideals I whose ideal of leading monomials $\langle \mathrm{LM}(I) \rangle$ is not finitely Σ-generated (see Example 2.2 in [3]). For these Σ-ideals, there does not exist a finite Σ-basis, hence, the Finite Σ-Basis algorithm does not stop when given them as input. Therefore, the hypothesis is necessary.

4. An Illustrative Example

In this section, we apply the introduced theory to a concrete case in a similar way as done in Section 5 of [5]. Namely, we construct a finite difference approximation for a given P.D.E. and compute a Σ-basis of this approximation. We do this using the implementations of the algorithms given in the appendix; therefore, the code in this section does not work if the code in the appendix is not implemented.

We first consider a system of partial differential equations, in this case, we chose a system used to model an isentropic (constant entropy) gas in one dimension [11], where $\rho(t, x)$ is the density, $u(t, x)$ is the speed and $p(t, x)$ is

the pressure in time $t > 0$ and in position x:

$$\begin{cases} f_1 := \dfrac{\partial \rho}{\partial t} + \dfrac{\partial(\rho\, u)}{\partial x} = 0 \\ f_2 := \dfrac{\partial(\rho\, u)}{\partial t} + \dfrac{\partial(\rho\, u^2)}{\partial x} + \dfrac{\partial p}{\partial x} = 0 \\ f_3 := \dfrac{\partial p}{\partial t} + \dfrac{\partial p}{\partial x} u + \gamma \dfrac{\partial u}{\partial x} p = 0 \end{cases} \tag{4}$$

The first equation derives from the mass conservation, the second from the momentum conservation, and the third from the state equation $p = \rho^\gamma$. The parameter γ denotes the heat capacity ratio.

Then, define the difference algebra in relation to these equations as follows. We consider the fraction field $\mathbb{Q}(t, x)$, the abelian monoid $\mathbb{N}^2$, and the following action of $\mathbb{N}^2$ over $\mathbb{Q}$:

$$\begin{aligned} \mathbb{Q}(t,x) \times \mathbb{N}^2 &\longrightarrow \mathbb{Q}(t,x) \\ (f(t,x),(a,b)) &\longmapsto f(a,b) := f(t + a\,h, x + b\,h) \end{aligned}$$

where $h \in \mathbb{Q}$. In this way, $\mathbb{Q}(t, x)$ can be seen as an $\mathbb{N}^2$-field. Let us fix $V = \{u(t,x), \rho(t,x), p(t,x)\}$ $\mathbb{N}^2$-algebraically independent elements over $\mathbb{Q}(t, x)$ in some $\mathbb{N}^2$-field extension of $\mathbb{Q}(t, x)$. We consider the difference algebra $\tilde{P} = \mathbb{Q}(t,x)[V(\mathbb{N}^2)]$.

To give a finite difference approximation, we use forward differences to approximate the partial derivatives of the function $u(t, x)$:

$$\begin{aligned} \frac{\partial u}{\partial t} &\approx \frac{u(t+h,x) + u(t,x)}{h} = \frac{u(1,0) + u(0,0)}{h} \\ \frac{\partial u}{\partial x} &\approx \frac{u(t,x+h) + u(t,x)}{h} = \frac{u(0,1) + u(0,0)}{h} \end{aligned}$$

Similarly, approximating the partial derivatives for ρ and p and substituting them in (4), yields the following partial difference system, which approximates the solutions of (4):

$$\begin{aligned} \tilde{f}_1 :=& \rho(1,0) + \rho(0,0) + u(0,0)(\rho(0,1) + \rho(0,0)) + \rho(0,0)(u(0,1) + u(0,0)) = 0 \\ \tilde{f}_2 :=& (\rho(1,0) + \rho(0,0))u(0,0) + (u(1,0) + u(0,0))\rho(0,0) + p(0,1) + p(0,0) \\ &+ (\rho(0,1) + \rho(0,0))u(0,0)^2 + 2(\rho(0,1) + \rho(0,0))\rho(0,0)u(0,0) = 0 \\ \tilde{f}_3 :=& p(1,0) + p(0,0) + (p(0,1) + p(0,0))u(0,0) + \gamma\,(u(0,1) + u(0,0))p(0,0) = 0. \end{aligned}$$

The mesh steps h have been simplified in the previous expressions.

We consider the Σ-ideal $I = \langle \tilde{f}_1, \tilde{f}_2, \tilde{f}_3 \rangle_\Sigma$, and we want to compute a (hopefully finite) Σ-basis of I. For that, we need a finite Σ-ordering compatible with the order function. Let $\succ$ be the block ordering given by the monomial orderings $>_{\text{degrlex}}$ on $\mathbb{Q}[(1,0),(0,1)]$ and by $\triangleright_{\text{lex}}$ on $\mathbb{Q}[u, \rho, p]$. The following code implements this Σ-ordering.

```
n=3 # Number of variables
r=2 # Range of the free monoid
M1=np.array([[1,1],[0,-1]])  # Matrix of degrlex ordering.
```

```
print(M1)
M2=np.array([[1,0,0],[0,1,0],[0,0,1]]) # Matrix of lex ordering.
print(M2)
s_order=sigma_order(M1,M2,r,n) # We define the block ordering.
```

Moreover, to stablish all the previous setting and define the Σ-polynomials $\tilde{f}_1, \tilde{f}_2, \tilde{f}_3$ we execute the following code.

```
n=3 # Only if the previous lines have not been executed
r=2

# We define the variables u=x[0], rho=x[1], p=x[2].
x = [sp.symbols(f"x_{i}") for i in range(1, n + 1)]

# We define the parameter gamma
gamma = sp.symbols('gamma')

# We define the neutral elements and the generators of N^2.
s0=sigma((0,0),rang=r)
s1=sigma((1,0),rang=r)
s2=sigma((0,1),rang=r)

# We define the polynomials.
p=s1.act(x[1])+s0.act(x[0]) +
(s2.act(x[1])+s0.act(x[1]))*s0.act(x[0]) +
(s2.act(x[0]) + s0.act(x[0]))*s0.act(x[1])

q=(s1.act(x[1])+s0.act(x[1]))*s0.act(x[0]) +
(s1.act(x[0])+s0.act(x[0]))*s0.act(x[0]) +
(s2.act(x[1])+s0.act(x[1]))*s0.act(x[0])**2 +
2*(s2.act(x[0])+s0.act(x[0]))*s0.act(x[0])*
s0.act(x[1])+s2.act(x[2])+s0.act(x[2])

h= s1.act(x[2])+s0.act(x[2]) +
(s2.act(x[2])+s0.act(x[2]))*s0.act(x[0]) +
gamma*(s2.act(x[0])+s0.act(x[0]))*s0.act(x[2])
```

If we `print` the polynomial p we get:

```
x_1_0|0*(x_2_0|0 + x_2_0|1) + x_1_0|0 +
x_2_0|0*(x_1_0|0 + x_1_0|1) + x_2_1|0
```

We see that, for instance, the variable $u(0,1)$ appears as `x_1_0|1`, where the `x_1` part refers to the $\mathbb{N}^2$-indeterminate u (even though we called it `x[0]`, when printed, the subindex increments in one unit), and the `0|1` part refers to the shift, in this case $(0,1)$.

We define the sigma polynomials, indicating whether we have constants in them

```
f_1=sigma_pol(p,set()) # No constants
```

```
f_2=sigma_pol(q,set()) # No constants
f_3=sigma_pol(h,{gamma}) # gamma is the only constant
```

Finally, a Σ-basis is computed if, after the previous code, we execute

```
G=[f_1,f_2,f_3]
Finite_Sigma_Basis(G,s_order,r,n)
```

We observe that $\{f_1, f_2, f_3\}$ is already a finite Σ-basis of I. The computation of a Σ-basis in this context is useful to check the strong consistency of the finite difference approximation we are considering. For further details, we refer the reader to [1], where an algorithm for the verification of the strong consistency is given. This algorithm relies on a computation of a Σ-basis.

5. Appendix: Implementations

Here, we give a way to implement the algorithms in Section 3 in Python, for those Σ-polynomial algebras $P = K[X\Sigma)]$ such that the action of Σ on K is trivial.

We first implemented the action of the free abelian monoid Σ, the notion of Σ-ordering, and Σ-polynomial. This enables subsequent implementations for computing S-polynomials, leading terms, and Σ-reductions. Finally, we implemented the Truncated Σ-basis and the Finite Σ-basis algorithms. To give these implementations, we used the following libraries:

```
from collections import Counter
import sympy as sp
import itertools
import numpy as np
import re
```

We start by giving a class `sigma` which is the class of elements of the free abelian monoid Σ.

```
class sigma:
def __init__(self, vector: tuple, rang: int):
    self.fixed_dim = rang

    # Same length.
    if len(vector) != self.fixed_dim:
        raise ValueError("Vector length and rank do not coincide")

    # All entries are natural
    if any(not isinstance(x, int) or x < 0 for x in vector):
        raise ValueError("Entries must be positive integers")

    self.vector = tuple(vector)
def sum(self, other):

    nuevo_vector = tuple(a + b for a, b in zip(self.vector, other.vector))
    return sigma(nuevo_vector, self.fixed_dim)
def sig_order(self) -> int:
    return sum(self.vector)

def to_suffix(self) -> str:
    # Constructs a string of the object
    return "|".join(str(x) for x in self.vector)
```

```
@classmethod
def from_suffix(cls, suffix: str, r: int):
    #Constructs an object from the string
    vector = tuple(int(x) for x in suffix.split("|"))
    return cls(vector, r)
def action(self, sig_poly):
    actual_variables = [var for var in sig_poly.pol.free_symbols-sig_poly.const]
    substitutions = {}

    for var in actual_variables:
    # Converts the string of the variable in a vector [x,id_variable,sigma(string)]
        parts = var.name.split('_')
        base_name = parts[0]
        id_variable = parts[1]

        # If it had previous action, it adds, if not, it creates the zero vector.
        if len(parts) > 2:
            previous = sigma.from_suffix(parts[2], self.fixed_dim)
        else:
            zero_vector = tuple(0 for _ in range(self.fixed_dim))
            previous = sigma(zero_vector, self.fixed_dim)

        # Adds both sigmas
        new_elmnt = previous.sum(self)

        # Converts vector to string.
        new_name = f"{base_name}_{id_variable}_{new_elmnt.to_suffix()}"

        # Creates a vector with the new variables to replace the old ones.
        substitutions[var] = sp.symbols(new_name)

        # NOTE: sp.symbols declares symbolic variables with the symbol
        # "new_name". If there already exist a variables with such
        # name, it does nothing

    # It returns a new sigma_polynomial with the substituted variables.
    return sigma_pol(sig_poly.pol.subs(substitutions),sig_poly.const)

def act(self, poly):
# This method does the same as the previous one.
# But in the input we have a polynomial
# expression (sp.Expr) instead of a sigma_polynomial.

    actual_variables = [var for var in poly.free_symbols]
    substitutions = {}

    for var in actual_variables:
        parts = var.name.split('_')
        base_name = parts[0]
        id_variable = parts[1]

        if len(parts) > 2:
            previous = sigma.from_suffix(parts[2], self.fixed_dim)
        else:
            zero_vector = tuple(0 for _ in range(self.fixed_dim))
            previous = sigma(zero_vector, self.fixed_dim)
```

```
            new_elmnt = previous.sum(self)
            new_name = f"{base_name}_{id_variable}_{new_elmnt.to_suffix()}"
            substitutions[var] = sp.symbols(new_name)

        return poly.subs(substitutions)

    def equals (self,other):
        return np.array_equals(self.vector,other.vector)
    def __repr__(self):
        return f"{self.vector}"
```

To implement the action, the idea is to "read" the variables of the inserted "Σ-polynomial" (which is an object of a class defined later). When we "read" the variable, we see if it has been multiplied by some element of the monoid earlier or not. If it has, we add both elements and create a new variable (a new `sp.symbol`) indexed by their sum. If it has not, we do the same, but just add the neutral element of the monoid as if it were the previous. To actually do this, since the names of the variables are strings, we need two methods, one that encodes an element of the monoid as a string `to_suffix`, and another one that does the opposite, given a string, it creates the element of the monoid related to that string `from_suffix`. Namely, the element given by the tuple `[a,b,c]` is associated to the string `a|b|c`.

To read the variables of the Σ-polynomial, we use the method `.free_symbols` from the library sympy. This method reads every `sp.symbol` from our polynomial, and this is a problem, because there are `sp.symbols` which we do not want to be treated as variables, namely, constants in our Σ-polynomial (in the example of Section 4, for instance, we have a constant γ). This motivates the definition of a class of Σ-polynomials, whose attributes are a polynomial expression (type `sp.Expr`) and the set of `sy.symbols` which are considered as constants in that expression.

```
class sigma_pol:
    def __init__(self,p,Const):
        self.pol=p # Polynomial expresion
        self.const=Const # Set of constants
    def __repr__(self):
        return f"{self.pol}"
```

We also define a class of Σ-orderings, where the attributes are two matrices, corresponding to two monomial orderings, one on $K[\sigma_1, \ldots, \sigma_r], >$, and another one on $Q = K[x_1, \ldots, x_n], \triangleright$. These matrices need to satisfy certain properties concerning their rank and the number of columns.

```
    class sigma_order:
    def __init__(self,sigma_matrix,var_matrix,r,n):
        # Check if they have enough columns
        if ((sigma_matrix.shape[1]!=r) or (var_matrix.shape[1]!=n)):
            raise ValueError("Matrices must have r and n columns respectively")

        # Check if their rank is maximum
        if ((np.linalg.matrix_rank(sigma_matrix)<r) or (np.linalg.matrix_rank(var_matrix)<
            raise ValueError("The rank of the matrices must be r and n respectively")
```

```
        # Matrix whose order is on K[sigma_1,...,sigma_r]
        self.sigma_matrix=sigma_matrix

        # Matrix whose order is on K[x_1,...,x_n]
        self.var_matrix=var_matrix

    def __repr__(self):
        return f"{self.sigma_matrix}_{self.var_matrix}"
```

To compute the leading term of a Σ-polynomial, we give a method that, given two monomials, it returns the largest of them with respect to a Σ-block ordering.

```
def greater_mon(m1,m2,sig_ord,n:int):
    #If one is constant, it returns the other
    if not sp.simplify(m1).free_symbols:
        return m2
    if not sp.simplify(m2).free_symbols:
        return m1

    #If one divides the other, it returns the big one:
    if sp.rem(m2,m1)==0: #Si m1 divide a m2:
        return m2
    if sp.rem(m1,m2)==0: #Si m2 divide a m1:
        return m1

    # We create tuples that encode the exponents and the variables.
    exp1 = tuple(
    (var.name.split('_'), exp) for var, exp in m1.as_powers_dict().items()
    if isinstance(var, sp.Symbol))

    exp2 = tuple(
    (var.name.split('_'), exp) for var, exp in m2.as_powers_dict().items()
    if isinstance(var, sp.Symbol))

    # We obtain the biggest sigma within the variables.
    maxi1=exp1[0][0][2]
    maxi2=exp2[0][0][2]

    for i in range(1,len(exp1)):
        if greater_sig(maxi1,exp1[i][0][2],sig_ord)==exp1[i][0][2]:
            maxi1=exp1[i][0][2]
        if greater_sig(maxi2,exp2[i][0][2],sig_ord)==exp2[i][0][2]:
            maxi2=exp2[i][0][2]

    if maxi1==maxi2 : #If they are equal, we compare inside P(sigma)

    # We obtain the vectors of the exponents whose variables have maxi
        v1=submonmial_tuple(exp1,n,maxi1)
        v2=submonmial_tuple(exp2,n,maxi1)

    # If they are still equal, we apply this method again recursively,
    # but removing the variables that yield maxi1.
        if(v1==v2):
            m3=monomial_no_sigma(m1,maxi1)
            m4=monomial_no_sigma(m2,maxi2)
```

```
            if greater_mon(m3,m4,sig_ord,n)==m3: # Call recursively
                return m1
            else:
                return m2

    # If they are not equal, we compare both tuples.
        else:
            v_1=sig_ord.var_matrix@v1
            v_2=sig_ord.var_matrix@v2
            for i in range(len(v_1)):
                if v_1[i]>v_2[i]:
                    return m1
                if v_2[i]>v_1[i]:
                    return m2

    # If maxi1 and maxi2 are not equal, we take the maximum
    else:
        maxi=greater_sig(maxi1,maxi2,sig_ord)
        if maxi==maxi1:
            return m1
        if maxi==maxi2:
            return m2
```

To compute this algorithm, we have used the following methods

- This method returns the biggest of two strings, which will correspond to elements of the monoid.

```
def greater_sig(s1:str,s2:str,sig_ord):
    if s1==s2:
        return s1
    else:
        # Mutliplication: (matrix) · (column vector)
        v1=sig_ord.sigma_matrix@tuple(int(x) for x in s1.split('|'))
        v2=sig_ord.sigma_matrix@tuple(int(x) for x in s2.split('|'))

        for i in range(len(v1)):
            if v1[i]>v2[i]:
                return s1
            if v2[i]>v1[i]:
                return s2
```

- The input of this method is a tuple with variables and exponents, and it returns a subtuple with only those exponents and variables that yield some given $\sigma \in \Sigma$, with σ encoded as a string.

```
def submonmial_tuple(exp,n:int,sigma_str:str):
    tupla = [0] * n
    for i in range(len(exp)):
        if exp[i][0][2]==sigma_str:
            tupla[int(exp[i][0][1])-1]=exp[i][1]
    return tupla
```

- This method receives a monomial m and some $\sigma \in \Sigma$ encoded as a string, and returns the monomial obtained by removing from m those variables that yield σ.

```
def monomial_no_sigma(m,sigma_str:str):
    nuev_mon=[]
    tupla=tuple(
    (var, exp) for var, exp in m.as_powers_dict().items()
    if isinstance(var, sp.Symbol))
    for (var,exp) in tupla:
        if var.name.split('_')[2]!=sigma_str:
            nuev_mon.append(var**exp)
    return sp.Mul(*nuev_mon) if nuev_mon else sp.Integer(1)
```

We know can implement the algorithms for computing the leading monomial of a Σ-polynomial and the S-polynomial. The implementation of the leading term will return a 2-tuple, with the leading coefficient in the first coordinate and the leading monomial in the second.

```
def LT(p,sig_ord,n):
    # The polynomial can not be zero
    if p.pol==0:
        raise ValueError("We can not compute the leading term of 0")

    coef_mon=[]

    # We first create a tuple of terms of the polynomial
    terms=sp.Add.make_args(p.pol.expand())

    for term in terms:
        # We form a tuple of coefficients and monomials
        coef_mon.append(term.as_independent(*(p.pol.free_symbols
        -p.const), True))

    # We compare each monomial and take the biggest one.
    maxi=coef_mon[0][1]
    coef=coef_mon[0][0]
    for i in range(1,len(coef_mon)):
        if greater_mon(maxi,coef_mon[i][1],sig_ord,n)==coef_mon[i][1]:
            maxi=coef_mon[i][1]
            coef=coef_mon[i][0]

    # We return a tuple with the L.C. and the L.M.
    return [coef,maxi]
```

Notice that it was crucial to distinguish which elements of `sy.symbols` we considered as variables and which we considered as constants when building the tuple of coefficients and monomials.

The code for computing the S-polynomial is the following:

```
def Spol(p,q,sig_ord,n):
lt_p=LT(p,sig_ord,n)
lt_q=LT(q,sig_ord,n)
```

```
    # We compute the least common multiple.
    L=sp.lcm(lt_p[1],lt_q[1])

    return sigma_pol(((L/(lt_p[0]*lt_p[1]))*p.pol -
    (L/(lt_q[0]*lt_q[1]))*q.pol).expand(),p.const|q.const)
```

Note that the set of variables of the S-polynomial is the union of both set of variables of the inputs `p` and `q`.

Notice that to implement "Finite Σ-basis", it suffices to implement Σ-reduce when the input G is finite, so for simplicity, we present an implementation for the finite case. Moreover, we remark that Σ-reduce returns a set; however, our implementation returns a tuple $(g_1, f_1, \ldots, g_s, f_s, r)$ instead.

```
def Finite_Sigma_Reduce(G,p,sig_ord,n:int):
    # Set the remainder to zero and define the tuple with the representation
    G_rep=[]
    r=sigma_pol(0,p.const)

    while not sp.simplify(p.pol) ==0 :
        divided=False
        lt_p=LT(p,sig_ord,n)

        for g in G :
            lt_g=LT(g,sig_ord,n)

            # Check if LM(g) divides LM(p)
            if sp.rem(lt_p[1],lt_g[1])==0:
                divided=True

            # If g is not in the tuple, we add it.
                if g not in G_rep:
                    index=len(G_rep)
                    G_rep.append(g)
                    G_rep.append(sigma_pol(0,g.const))

                else:
                # We index  the position of g inside G_rep
                    index =
                    [i for i, g1 in enumerate(G_rep) if g1.pol == g.pol][0]

                # We add the quotient to f_i
                G_rep[index+1].pol =
                G_rep[index+1].pol+(lt_p[0]/lt_g[0])*(lt_p[1]/lt_g[1])

                # We reduce p
                p.pol =
                (p.pol-(lt_p[0]/lt_g[0])*(lt_p[1]/lt_g[1])*g.pol).expand()

                # The set of constants is the union of both of g and p
                p.const=p.const|g.const
                break
        if not divided :

        # If LM(g) does not ivide LM(p), we add LT(p) to the remainder
            r.pol=r.pol+lt_p[0]*lt_p[1]
```

```
            r.const=r.const|p.const
            p.pol=p.pol-lt_p[0]*lt_p[1]

    # We finally add the emainder to the list.
    G_rep.append(r)
    return G_rep
```

For this implementation, we used the following method, which returns a tuple with all possible tuples with positive integer entries such that the sum of their entries is smaller or equal to some $d \in \mathbb{N}$.

```
    def less_order_than(d: int,r: int):
    elements=[]
    for actual_order in range(d + 1):
        for c in itertools.combinations_with_replacement(range(r), actual_order):
            vector = [0] * r
                # We add 1 in the indicated positions
            for i in c:
                vector[i] += 1
            elements.append(tuple(vector))

    return elements
```

To implement the Algorithm Truncated Σ-basis, however, we will need a simpler implementation of the "Σ-reduce" that only returns the remainder:

```
def remainder(G,p,sig_ord,n:int):
    r=sigma_pol(0,p.const)
    while not sp.simplify(p.pol)==0 :
        divided=False
        lt_p=LT(p,sig_ord,n)
        for g in G :
            lt_g=LT(g,sig_ord,n)
            if sp.rem(lt_p[1],lt_g[1])==0:
                divided=True
                p.pol=(p.pol-(lt_p[0]*lt_p[1])/(lt_g[0]*lt_g[1])*g.pol)
                p.const=p.const|g.const
                break
        if not divided :
            r.pol=r.pol+lt_p[0]*lt_p[1]
            r.const=r.const|p.const
            p.pol=p.pol-lt_p[0]*lt_p[1]
    return r
```

Moreover, we following methods will also be useful to provide such implementations.

- Given a set of Σ-polynomials G, this method computes the set $\Sigma_d \cdot G$ for some $d \in \overline{\mathbb{N}}$.

  ```
  def dot_set(G,d:int,r):
      Sigma_d=less_order_than(d,r)
      G_dot_Sigma=[]
      for v in Sigma_d:
          s=sigma(v,r)
          for p in G :
              if s.action(p) not in G_dot_Sigma:
                  G_dot_Sigma.append(s.action(p))
  ```

```
        return G_dot_Sigma
```

- The following method computes the order of a Σ-polynomial. If the polynomial is zero, instead of returning $-\infty$, it returns -1.

```
def order(p):
    if sp.simplify(p.pol)==0:
        return -1

    variables = list(p.pol.free_symbols-p.const)
    if len(variables)==0:
        return 0

    v = variables[0].name.split('_')[2].split('|')
    maxi=sum(int(x)for x in v)

    for i in range(1,len(variables)):
    v = variables[i].name.split('_')[2].split('|'))

        if sum(int(x) for x in v >maxi:
            maxi=sum(int(x) for x in v)

    return maxi
```

- The following method will return true if the greatest common divisor of two elements of Σ is zero, and false in any other case.

```
def zero_gcd(s1,s2):
    v = []

    if len(s1.vector)!=len(s2.vector):
        raise ValueError
        ("Both must have the same lenght to compute the gcd")
    coprime=True

    for i in range(len(s1.vector)):
        if (s1.vector[i]!=0 and s2.vector[i]!=0):
            coprime=False
            break

    return coprime
```

We can now give an implementation of the Truncated Σ-Basis algorithm.

```
def Truncated_Sigma_Basis(G,d,sig_ord,r:int,n:int):
    #Construc G_d y G_d_prim
    G_d=[g for g in G if (order(g)<=d and sp.simplify(g.pol)!=0)]
    G_d_prim=[g for g in G if (order(g)<=d and sp.simplify(g.pol)!=0)]

    while True :
    # We update G_d with G_d_prim
        G_d.extend(G_d_prim[len(G_d):])
    # We compute Sigma_dot_G in each iteration
        Sigma_dot_G=dot_set(G_d,d,r)

        # Consider all pairs of sigmas with order
        # smaller than d with repetition
        for s1_vec,s2_vec in itertools.product(less_order_than(d,r),
```

```
        repeat=2):
            s1=sigma(s1_vec,r)
            s2=sigma(s2_vec,r)

            # Consider all pairs of polynomials in
            # G_d without repetition
            for f,g in itertools.combinations(G_d,2):

                # Compute the S-plynomial
                s=Spol(s1.action(f),s2.action(g),sig_ord,n)

                # The condition that must be satisfied for reducing.
                condition = (zero_gcd(s1,s2) and
                sp.gcd(s1.act(LT(f,sig_ord,n)[1]),
                s2.act(LT(g,sig_ord,n)[1]))!=1 and order(s)<=d)

                if condition:

                    # Compute the remainder
                    rem=remainder(Sigma_dot_G,s,sig_ord,n)
                    rem.pol=sp.simplify(rem.pol)

                    # If it is non-zero, we add it
                    if rem.pol!=0:
                        G_d_prim.append(rem)

        # We repeat this process until all remainders are zero.
        if(np.array_equal(G_d,G_d_prim)):
            break

    return G_d
```

Note that it actually returns a tuple with the truncated Σ-basis, not a set. Finally we can give an implementation of the Finite Σ-Basis algorithm.

```
def Finite_Sigma_Basis(G,sig_ord,r,n):

    # We check if G is empty
    if len(G)==0:
        return G

    # We take the non-zero elements of G, set d_prime=-1 and d.
    G=[g for g in G if sp.simplify(g.pol)!=0]
    d_prime=-1
    d=max(order(g) for g in G)
    while d_prime<d :
        d_prime=d

        # We compute the truncated Sigma basis
        G.extend(Truncated_Sigma_Basis(G,d,sig_ord,r,n)[len(G):])
        G_prime=G.copy()

        # We compute Sgiam_dot_G until 2d.
        Sigma_dot_G=dot_set(G,2*d,r)

        for s1_vec,s2_vec in
        itertools.product(less_order_than(d,r),repeat=2):
```

```
            s1=sigma(s1_vec,r)
            s2=sigma(s2_vec,r)

            for f,g in itertools.combinations(G,2):
                s=Spol(s1.action(f),s2.action(g),sig_ord,n)

                # We check only for the remaining S-polynomials: order(s)>d
                condition=(zero_gcd(s1,s2) and sp.gcd(s1.act(LT(f,sig_ord,n)
                [1]),s2.act(LT(g,sig_ord,n)[1]))!=1 and order(s)>d)

                # If the condition is passed and the
                # remainder is non-zero, we add it.
                if condition:
                    rem=remainder(Sigma_dot_G,s,sig_ord,n)
                    rem.pol=sp.simplify(rem.pol)
                    if rem.pol!=0:
                        G_prime.append(rem)

        # We print the basis we are getting so far
        print("Basis for d equal to")
        print(d)
        print(G_prime)

        # We update G and d
        G.extend(G_prime[len(G):])
        d=max(order(g) for g in G)
    return G
```

In general, computing a finite Gröbner basis is not an easy task, because the number of operations grows exponentially with respect to the number of variables in the polynomial ring. Since the number of variables of P is infinite, the computation of a finite Σ-basis is even harder. Consequently, these implementations aim to provide tools for computing basic examples of Σ-bases and building intuition, rather than optimizing performance on the computation. In Section 5, we gave an example where all this code is used in practice.

References

[1] Pierluigi Amodio et al. "On consistency of finite difference approximations to the Navier-Stokes equations". In: *International Workshop on Computer Algebra in Scientific Computing*. Springer. 2013, pp. 46–60.

[2] Michael F Atiyah and Ian Grant Macdonald. *Introduction to commutative algebra*. CRC press, 2018.

[3] Yu-Ao Chen and Xiao-Shan Gao. "Criteria for Finite Difference Gröbner Bases of Normal Binomial Difference Ideals". In: *Proceedings of the 2017 ACM International Symposium on Symbolic and Algebraic Computation*. 2017, pp. 93–100.

[4] David Cox et al. *Ideals, varieties, and algorithms*. Springer, 1997.

[5] Vladimir Gerdt and Roberto La Scala. "Noetherian quotients of the algebra of partial difference polynomials and Gröbner bases of symmetric ideals". In: *Journal of Algebra* 423 (2015), pp. 1233–1261.

[6] Vladimir P Gerdt, Yuri A Blinkov, Vladimir V Mozzhilkin, et al. "Gröbner bases and generation of difference schemes for partial differential equations". In: *SIGMA. Symmetry, Integrability and Geometry: Methods and Applications* 2 (2006), p. 051.

[7] Martin Kreuzer and Lorenzo Robbiano. *Computational Commutative Algebra 1*. Heidelberg: Springer-Verlag, 2000. ISBN: 978-3-540-67733-8.

[8] Alexander Levin. *Difference algebra*. Springer, 2008.

[9] Lorenzo Robbiano. "Term orderings on the polynomial ring". In: *European Conference on Computer Algebra*. Springer. 1985, pp. 513–517.

[10] Alexander V Smirnov and Vladimir A Smirnov. "Applying Gröbner bases to solve reduction problems for Feynman integrals". In: *Journal of High Energy Physics* 2006.01 (2006), pp. 001–001.

[11] Shyam Sundar Ghoshal and Animesh Jana. "On blow up of solutions of isentropic Euler system". In: *arXiv e-prints* (2023), arXiv–2308.