



HAL
open science

Axioms for a theory of signature bases

Pierre Lairez

► **To cite this version:**

| Pierre Lairez. Axioms for a theory of signature bases. 2022. hal-03830003v1

HAL Id: hal-03830003

<https://hal.science/hal-03830003v1>

Preprint submitted on 26 Oct 2022 (v1), last revised 8 Jan 2024 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Axioms for a theory of signature bases

Pierre Lairez ✉ 

Université Paris-Saclay, Inria, 91120 Palaiseau, France

Abstract Twenty years after the discovery of the F5 algorithm, Gröbner bases with signatures are still challenging to understand and to adapt to different settings. This contrasts with Buchberger’s algorithm, which we can bend in many directions keeping correctness and termination obvious. I propose an axiomatic approach to Gröbner bases with signatures with the purpose of uncoupling the theory and the algorithms, and giving general results applicable in many different settings (e.g. Gröbner for submodules, F4-style reduction, noncommutative rings, non-Noetherian settings, etc.).

2012 ACM Subject Classification Computing methodologies → Algebraic algorithms

Keywords and phrases Gröbner basis, F5 algorithm, signature basis

Funding This work has been supported by the European Research Council under the European Union’s Horizon Europe research and innovation programme, grant agreement 101040794 (10000 DIGITS); and by the ANR grant ANR-19-CE40-0018 (De Rerum Natura).

1 Introduction

Context Introduced by Faugère (2002) to compute Gröbner bases, the F5 algorithm proposes the concept of *signature* to avoid the redundant computations that arise in Buchberger’s algorithm (Buchberger, 1965, 1965/2006). Each polynomial handled by the algorithm is augmented with a signature designed to enforce a fundamental postulate, which we may state as “two elements with the same signature are substitutable”. We can find precursive ideas in the work of Gebauer and Möller (1986) and Möller et al. (1992) and signatures also share some ideas with Hilbert-driven algorithms (Traverso, 1996).

Today’s situation of signature algorithms is equivocal. F5’s relevance, from the pure aspect of performance, was demonstrated by a success on cryptographic challenges early on (Faugère & Joux, 2003). But none of the current best implementations for computing Gröbner bases uses signatures, be it Magma (Bosma et al., 1997), msolve (Berthomieu et al., 2021), Singular (Decker et al., 2022) or Maple. They prefer Buchberger’s algorithm, handling S-pairs as Gebauer and Möller (1988) do and using simultaneous reductions in the F4 style (Faugère, 1999) – see the report of Monagan and Pearce (2015) on this approach. The theoretical benefits of signature algorithms are diminished by a higher implementation complexity and a larger output (the signature bases computed by signature algorithms are more constrained than Gröbner bases). More than benchmarks, literature about signature algorithms is turned towards revealing the core ideas behind F5 and understanding what makes a signature algorithm terminate. Termination is a very peculiar aspect, not as transparent as termination of Buchberger’s algorithm. Nonetheless, thanks to decisive work by Hashemi and Ars (2010), Gao et al. (2010), Arri and Perry (2011), Eder and Perry (2011), and Gao et al. (2016) these goals have been reached – see the survey by Eder and Faugère (2017).

The point of studying signature algorithm may not be the quest of new world records for polynomial system solving, but rather the understanding of signatures themselves, and what we can extract from them. Signature bases convey an extra information compared to Gröbner bases, related to the syzygy module of the input generators. Many ideal-theoretic

operations – intersection, quotient, saturation, Ext modules (Stillman, 1990) – are related to syzygy modules, and signature algorithms seem to give an efficient access to them (Gao et al., 2010; Sun & Wang, 2011; Faugère, 2001; Eder et al., 2022). Porting these ideas to more general settings is a strong motivation to engage in the study of signatures. Yet, in my view, the lack of flexibility of the theory of signature algorithms hinders further development, both practical and theoretical. For example, modern implementations for computing Gröbner bases make it clear that simultaneous reductions in the F4 style are key towards high performance. Yet, there is no satisfactory description of a signature algorithm with F4-style reduction (Albrecht and Perry (2010) do not prove termination and Eder and Faugère (2017, §13) are superficial).

Contribution I propose a set of axioms that specifies a context in which signature algorithms are applicable. They fit many known settings – such as solvable algebras (Sun et al., 2012) and free algebras (Hofstadler & Verron, 2022) – and some previously unknown settings – such as the module case or localizations of Weyl algebras with an elimination order, see Section 6.5. Note however that the case where the coefficient ring is not a field is out of scope, as well as local orderings. Many of the ideas of the most recent frameworks for signature algorithms (Eder & Perry, 2011; Gao et al., 2016) work smoothly in this axiomatic setting, so many statements will of course be familiar, yet with a wider applicability.

Working with axioms makes some useful ideas emerge. At least two of them are worth attention. Firstly, the concept of *prebasis* is introduced to uncouple the theory from the input. Previous work all start by fixing the input and developing the theory with respect to this input (e.g. Eder & Faugère, 2017, Definition 2.2). This is problematic when trying to define what a signature basis is, independently of an input. Gröbner bases are defined by the equality $\langle \text{lm } G \rangle = \text{lm } \langle G \rangle$, or the confluence of some rewriting system, we do not need to say a Gröbner basis *of something* (e.g. Becker & Weispfenning, 1993, Definition 5.37). It is a very desirable definition which should have an equivalent in the signature setting. The main obstacle here is the definition of signatures independently of an input. This raise an interesting question: given a set of signatures – basically anything well-ordered on which act monomials –, what are the admissible inputs? This leads to the concept of prebasis, that is a set of sigpoly pairs which satisfy the fundamental postulates of signatures (elements with equal signatures are substitutable). I prove that if a set of sigpoly pairs is a Gröbner basis in the module representation, then it is a prebasis (Theorem 11).

Secondly, I introduce *sigtrees* to uncouple the termination criterion from the algorithms themselves. Sigtrees make a very general termination criterion. For Buchberger’s algorithm, termination follows from a general principle, Dixon’s Lemma, not from *ad hoc* arguments. The concept of sigtrees is a tentative to provide such a general argument. As a direct application, it proves the termination of signature algorithms with out-of-order signature handling and the F5 reductant selection strategy (among all possible reductants, choose the most recent one). It was unproved even in the classical polynomial setting. We may blend in an F4-style reduction, the termination argument remains the same.

Lastly, as a didactic contribution, I try to emphasize an elemental feature of signature bases (or rather, for that matter, rewrite bases), putting a clear distinction with Gröbner bases. To check that a given set G is a Gröbner basis of an ideal I , it is enough to check that (1) G generates I , and (2) the S-pairs reduce to zero. Typically (1) will hold by design if G has been constructed from a generating set of I by adding new elements obtained by allowed reduction steps. The condition (2) is more difficult to check and requires arithmetic operations in the base field. This is typically a costly operation. To check that a given set G with signatures is

a rewrite basis of an ideal I , it is enough to check that (1') G is a *prebasis* of I , and (2') the leading monomials and the signatures satisfy some combinatorial property (Corollary 20). The concept of prebasis is introduced in Section 3.2, but for the moment, it is enough to say that (1') will hold by design if G is obtained by allowed reduction steps from an initial prebasis of I . The important part is the nature of (2'): it requires no arithmetic operations to be checked, only operations on monomials. Algorithms for signature bases are all about exploiting this combinatorial structure. This reminds of *staggered linear bases* introduced by Gebauer and Möller (1986) to compute Gröbner bases, they feature a similar combinatorial structure – and the link with signatures have recently been investigated by Hashemi and Javanbakht (2021).

Plan In Section 2, we define the algebraic structure in which we consider signature bases, *monomial modules*, that are vector spaces with a “leading monomial” map and an action of a monoid with some compatibility rules. We also introduce the rewriting system defined by the top reduction. In Section 3, we define signatures, signature bases and prebases. In Section 4, we define rewrite bases and state a combinatorial criterion for a set to be a rewrite basis. In Section 5, we introduce Noetherian hypotheses, termination arguments and review algorithm *templates*. In Section 6, we illustrate the axioms by several different settings in which they apply.

Acknowledgment I am grateful to Hadrien Brochet and Frédéric Chyzak for a very careful reading and useful comments.

2 Gröbner bases

Before going to signatures, we lay down some definitions. The main ones are the definitions of a *monomial space* – a vector space with a concept of leading monomial, see 2.2 – and a *monomial module* – a monomial space endowed with a linear action of a (non necessarily commutative) monoid, compatible with leading monomials, see Section 2.3.

In monomial spaces, we develop a (short) theory of *top reduction modulo tail equivalence*, using the terminology of rewriting systems, see Section 2.1 and 2.2. Using rewriting systems to describe the theory of Gröbner bases in polynomial rings is done in several textbooks (e.g. Becker & Weispfenning, 1993; Winkler, 1996; Kreuzer & Robbiano, 2000; Mora, 2005): in a few words, we say that a polynomial f can be reduced by a polynomial g , if we can cancel out one of the terms of f by subtracting a multiple of the leading monomial of g . The context of signatures puts the emphasis on *top reduction* – the reduction of the leading monomial – as opposed to *tail reduction*. The practice of Gröbner bases computation also shows that tail reduction steps are optional, they are irrelevant as far as termination and correctness is concerned. Lastly, tail reduction does not enjoy nice properties. For example, if g reduces f then mg reduces mf for any monomial m , in a polynomial setting. But this implication breaks if m is a polynomial rather than a monomial, or if f and g lie in a Weyl algebra, unless the reduction is a top reduction. All of this hints at replacing tail reduction by a more flexible *tail equivalence* and replace the customary reduction by the top reduction modulo tail equivalence. This fits the abstract setting of “reduction modulo equivalence” developed by Huet (1980).

2.1 Rewriting systems

Let X be a set and $\overset{\cdot}{\rightarrow}$ a binary relation on X . “ $x \overset{\cdot}{\rightarrow} y$ ” reads “ x reduces to y ”. Following Huet (1980), we define the following binary relations:¹

- $x \overset{n}{\rightarrow} y$, for $n > 0$, if there is some $z \in X$ such that $x \overset{\cdot}{\rightarrow} z$ and $z \overset{n-1}{\rightarrow} y$;
- $x \rightarrow y$ if $x \overset{n}{\rightarrow} y$ for some $n \geq 0$, this is the reflexive transitive closure of $\overset{\cdot}{\rightarrow}$;
- $x \uparrow y$ if there is some $z \in X$ such that $z \rightarrow x$ and $z \rightarrow y$;
- $x \downarrow y$ if there is some $z \in X$ such that $x \rightarrow z$ and $y \rightarrow z$.

The relation $\overset{\cdot}{\rightarrow}$ is *Noetherian* if there is no infinite sequence $x_0 \overset{\cdot}{\rightarrow} x_1 \overset{\cdot}{\rightarrow} \dots$. An element $x \in X$ is $\overset{\cdot}{\rightarrow}$ -*reduced* if there is no $y \in X$ such that $x \overset{\cdot}{\rightarrow} y$. If $x \rightarrow y$ and y is $\overset{\cdot}{\rightarrow}$ -reduced, then y is a *normal form* of x . If $\overset{\cdot}{\rightarrow}$ is Noetherian, then every element has at least one normal form. The relation $\overset{\cdot}{\rightarrow}$ is *confluent* if $x \uparrow y$ implies $x \downarrow y$ for any $x, y \in X$. If $\overset{\cdot}{\rightarrow}$ is confluent, then any $x \in X$ has at most one normal form.

Moreover, given an equivalence relation \sim on X , we define:

- $x \check{\rightarrow} y$ if there are $z, z' \in X$ such that $z \sim z'$, $z \rightarrow x$ and $z' \rightarrow y$;
- $x \downarrow y$ if there are $z, z' \in X$ such that $z \sim z'$, $x \rightarrow z$ and $y \rightarrow z'$;

The relation $\overset{\cdot}{\rightarrow}$ is *confluent modulo* \sim if for any $x, y \in X$, if $x \check{\rightarrow} y$ then $x \downarrow y$.

2.2 Top reduction

Let \mathcal{M} be a set with a well-order \leq and a minimal element denoted 0. Let M be a linear space over a field K endowed with a surjective map $\text{lm} : M \rightarrow \mathcal{M}$ (where lm stands for *leading monomial* in analogy with the usual polynomial setting) such that

$$\text{L1 } \forall x \in M, \text{lm } x = 0 \Leftrightarrow x = 0;$$

$$\text{L2 } \forall x, y \in M, \text{lm } x = \text{lm } y \neq 0 \Leftrightarrow \exists \lambda \in K^\times, \text{lm}(x - \lambda y) < \text{lm } x.$$

A linear space with such a map lm is called a *monomial space*. The *set of monomials* of M is the corresponding set \mathcal{M} .² It is useful to introduce the relation \equiv_{lt} defined on M by $x \equiv_{\text{lt}} y$ if $x = y = 0$ or $\text{lm}(x - y) < \text{lm } x$, to be understood as “ x and y have the same leading term”. We first state useful relations between sums and leading monomials that follow directly from the axioms.

► **Lemma 1.** For any $x, y, z \in M$,

- $\forall \lambda \in K^\times, \text{lm}(\lambda x) = \text{lm } x$;
- $\text{lm}(x + y) \leq \max(\text{lm } x, \text{lm } y)$;
- $x \equiv_{\text{lt}} y \Rightarrow \text{lm } x = \text{lm } y$;
- \equiv_{lt} is an equivalence relation.

A subset $E \subseteq M$ defines a relation $\overset{\cdot}{\rightarrow}_E$, called *top reduction*, on M defined by

$$x \overset{\cdot}{\rightarrow}_E y \Leftrightarrow \text{lm } y < \text{lm } x \text{ and } \exists \lambda \in K^\times, \exists e \in E, y = x - \lambda e.$$

¹ Actually Huet denotes \rightarrow the one-step reduction, which I denote $\overset{\cdot}{\rightarrow}$, and \rightarrow^* the multistep reduction, which I denote \rightarrow .

² We could define $\mathcal{M} \setminus \{0\}$ to be a basis of M and then define the leading monomial in the usual way. But for a single determination of the map lm , there would be many ways to choose such a basis. The axiomatic definition above highlights the irrelevance of nonleading terms. Nonetheless, the easiest way to describe the map lm on concrete example is indeed to exhibit a well-ordered basis of M .

In other words, $x \xrightarrow{E} y$ if y is the result of cancelling the leading monomial of x using a reducer in E . In this situation, we always have $\lambda e \equiv_{\text{lt}} x$. Moreover, we introduce a relation \perp_E , called *tail equivalence*, defined by

$$x \perp_E y \Leftrightarrow \exists \lambda \in K^\times, \exists e \in E, y = x - \lambda e \text{ and } \text{lm } e < \text{lm } x.$$

In this situation, we always have $x \equiv_{\text{lt}} y$. The reflexive transitive closure of \perp_E is denoted \smile_E . The confluence relations $\check{\uparrow}_E$ and $\check{\downarrow}_E$ are defined using \smile_E . The tail equivalence is not a reduction since it is symmetric, we cannot tell which side of an equivalence $x \smile_E y$ is more reduced.

The following statement is a variant, in the setting of monomials spaces, of the Buchberger's well known criterion for polynomial ideals. A subset $E \subseteq M$ is a *pivot basis* if it satisfies the equivalent properties of Theorem 2. The concept of pivot basis is similar to that of a row echelon form of a matrix. For $E \subseteq M$, let $\langle E \rangle$ denote the K -linear subspace generated by E .

► **Theorem 2** (Buchberger's criterion for monomial spaces). *Let E be a subset of M . The following assertions are equivalent:*

(Characterization by leading monomials)

B1 $\forall x \in \langle E \rangle, x \neq 0 \Rightarrow \exists e \in E, \text{lm } e = \text{lm } x$.

(Characterization by rewriting)

B2 $\forall x \in \langle E \rangle, x \rightarrow_E 0$;

(Characterizations by confluence properties)

B3 $\forall x, y \in M, x - y \in \langle E \rangle \Rightarrow x \check{\downarrow}_E y$;

B4 \rightarrow_E is confluent modulo \smile_E , that is $\forall x, y \in M, x \check{\uparrow}_E y \Rightarrow x \check{\downarrow}_E y$;

(Characterizations by S-pairs)

B5 $\forall e, f \in E, \forall \lambda \in K^\times, e \equiv_{\text{lt}} \lambda f \Rightarrow e - \lambda f \rightarrow_E 0$;

B6 $\forall e, f \in E, \forall \lambda \in K^\times, e \equiv_{\text{lt}} \lambda f \Rightarrow e - \lambda f \in \langle g \in E \mid \text{lm } g < \text{lm } e \rangle$;

Proof that B1 implies B2. Let $x \in \langle E \rangle$ be a nonzero element and let y be a \rightarrow -normal form of x . In particular $y \in \langle E \rangle$. By hypothesis, either $y = 0$, or $\text{lm } y = \text{lm } e$ for some $e \in E$. The latter would contradict the irreducibility of y , so $y = 0$ and $x \rightarrow 0$.

Proof that B2 implies B3. Let $x, y \in M$ such that $x - y \in \langle E \rangle$. There is some $e \in E$ and $\lambda \in K^\times$ such that $x - y \xrightarrow{E} x - y - \lambda e \rightarrow 0$ (unless $x = y$ but this case is trivial). In particular $\lambda e \equiv_{\text{lt}} x - y$.

If $\text{lm } x > \text{lm } y$, then $x \xrightarrow{E} x - \lambda e$ and $x - \lambda e - y \in \langle E \rangle$ and by induction on $\max(\text{lm } x, \text{lm } y)$, we may assume that $x - \lambda e \check{\downarrow}_E y$ and therefore $x \check{\downarrow}_E y$. The case $\text{lm } y > \text{lm } x$ is similar. If $\text{lm } x = \text{lm } y$, there is some $\mu \in K^\times$ such that $\text{lm}(x - \mu y) < \text{lm } x$. There are again two cases. If $\mu = 1$, that is $x \equiv_{\text{lt}} y$, then the sequence of top-reduction $x - y \xrightarrow{E} u_1 \xrightarrow{E} u_2 \xrightarrow{E} \dots \xrightarrow{E} u_n \xrightarrow{E} 0$ gives a sequence of tail equivalence $y \smile y + u_n \smile y + u_{n-1} \smile \dots \smile y + (x - y) = x$. In particular, $x \check{\downarrow}_E y$. If $\mu \neq 1$, then $\text{lm } x = \text{lm } y = \text{lm } e$, so there are reductions $x \xrightarrow{E} x - \kappa e$ and $y \xrightarrow{E} y - \nu e$, for some $\kappa, \nu \in K^\times$. By induction on $\max(\text{lm } x, \text{lm } y)$, we may assume that $x - \kappa e \check{\downarrow}_E y - \nu e$, which implies $x \check{\downarrow}_E y$.

Proof that B3 implies B4. Let $x, y \in M$ such that $x \check{\uparrow}_E y$. Both \xrightarrow{E} and \perp_E preserve equality modulo $\langle E \rangle$, so $y - x \in \langle E \rangle$, therefore $x \check{\downarrow}_E y$, by hypothesis.

Proof that B4 implies B5. If $e \equiv_{\text{lt}} \lambda f$, then $e \xrightarrow{E} e - \lambda f$. Moreover $e \xrightarrow{E} e - e = 0$. The confluence hypothesis implies that $e - \lambda f \rightarrow z$ and $0 \rightarrow z'$ for some $z, z' \in M$ such that $z \smile z'$. But 0 is reduced and only \smile -equivalent to itself. So $z = 0$ and $e - \lambda f \rightarrow 0$.

Proof that B5 implies B6. The rewriting $e - \lambda f \xrightarrow{E} 0$ implies, by definition of \rightarrow_E , this implies that $e - \lambda f \in \langle g \in E \mid \text{lm } g \leq \text{lm}(e - \lambda f) \rangle$. Since $\text{lm}(e - \lambda f) < \text{lm } e$, this gives the claim.

Proof that B6 implies B1. Let $x \in \langle E \rangle$ and let $m \in \mathcal{M}$ minimal such that $x \in \langle e \in E \mid \text{lm } e \leq m \rangle$. We can write $x = \lambda_1 e_1 + \cdots + \lambda_r e_r$ with $e_i \in E$, $\lambda_i \in K$ and $\text{lm } e_i \leq m$. By minimality of m , and up to reordering the indices, we may assume that $\text{lm } e_1 = m$.

Assume for contradiction that $\text{lm } x < m$. For each i , there is some $\mu_i \in K$ such that $e_i - \mu_i e_1 \in \langle e \in E \mid \text{lm } e < m \rangle$: if $\text{lm } e_i < m$, we choose $\mu_i = 0$, and if $\text{lm } e_i = m$, we choose μ_i such that $e_i \equiv_{\text{lt}} \mu_i e_1$ (using L2) and then B6 ensures that $e_i - \mu_i e_1 \in \langle e \in E \mid \text{lm } e < m \rangle$. The equality

$$x - \sum_{i=1}^r \lambda_i (e_i - \mu_i e_1) = \left(\sum_{i=1}^r \lambda_i \mu_i \right) e_1$$

implies that $\lambda_1 \mu_1 + \cdots + \lambda_r \mu_r = 0$, otherwise, the leading monomial of the right-hand side would be m , greater than the leading monomial of the left-hand side. Therefore

$$x = \sum_{i=1}^r \lambda_i (e_i - \mu_i e_1).$$

Thus, $x \in \langle e \in E \mid \text{lm } e < m \rangle$, contradicting the minimality of m . Therefore $\text{lm } x = m = \text{lm } e_1$. ◀

The following minor lemma, on increasing unions of pivot bases, will be used in Sections 3.1 and 4.1.

► **Lemma 3.** *Let I be a totally ordered set and let $(E_i)_{i \in I}$ be a family of subsets of M . If $E_i \subseteq E_j$ for any $i, j \in I$ with $i < j$, and if each E_i is a pivot basis, then $\cup_{i \in I} E_i$ is a pivot basis.*

Proof. We check the criterion B5. Let $e, f \in \cup_i E_i$ and $\lambda \in K^\times$ such that $e \equiv_{\text{lt}} \lambda f$. By definition, e is in some E_j while f is in some E_k , so both e and f are in $E_{\max(j,k)}$. Since $E_{\max(j,k)}$ is a pivot basis, $e - \lambda f \rightarrow 0$ with respect to $E_{\max(j,k)}$. *A fortiori*, it rewrites to 0 with respect to $\cup_i E_i$, which contains $E_{\max(j,k)}$. ◀

2.3 Monomial modules

A *monoid* is a set A with an associative composition law $A \times A \rightarrow A$ (denoted multiplicatively) which admits an identity element denoted 1_A . A *monomial module* over a monoid A is a monomial space M with a linear action of A on M (denoted also multiplicatively) with some compatibility with the map lm and the order \leq . In details, we require that:

$$\mathbf{M1} \quad \forall f \in M, 1_A f = f;$$

$$\mathbf{M2} \quad \forall a, b \in A, \forall f \in M, (ab)f = a(bf);$$

$$\mathbf{M3} \quad \forall a \in A, \forall f, g \in M, \forall \lambda \in K, a(f + \lambda g) = af + \lambda(ag);$$

$$\mathbf{M4} \quad \forall a \in A, \forall f, g \in M, \text{lm } f = \text{lm } g \Rightarrow \text{lm}(af) = \text{lm}(ag);$$

$$\mathbf{M5} \quad \forall a \in A, \forall f, g \in M, \text{lm } f < \text{lm } g \Rightarrow \text{lm}(af) < \text{lm}(ag);$$

Assumption M5 implies also the following:

$$\mathbf{M6} \quad \forall a \in A, \forall f \in M, \text{lm}(af) \geq \text{lm } f.$$

Indeed, if $\text{lm}(af) < \text{lm } f$, then $\text{lm}(a^{k+1}f) < \text{lm}(a^k f)$ for any $k \geq 0$, which would contradict the well-orderedness of \mathcal{M} . Note also that M is torsionfree: if $g \neq 0$, then $ag \neq 0$ for all $a \in A$, as a consequence of M5.

Assumption M4 and the surjectivity of lm makes it possible to define an action of A on the set of monomials \mathcal{M} by $a \text{lm } f \doteq \text{lm}(af)$. A *divisor* of $m \in \mathcal{M}$ is an element $n \in \mathcal{M}$ such that $an = m$ for some $a \in A$.

We can always embed A in a K -algebra R such that the composition law in A is the restriction of the multiplication in R and such that A generates R , so that M is a left module over R . However, the assumptions M4 and M5 may not extend to R , or even to the nonzero elements of R .

2.4 Gröbner bases

In the context above, a subset $G \subseteq M$ is a *Gröbner basis* if AG , that is $\{af \mid a \in A, f \in G\}$, is a pivot basis. It is naturally a key concept, see (Cox et al., 2015) for an introduction to the topic. The purpose of signatures is not to extend or generalize it, but rather to give a way to compute Gröbner bases.

► Remark 4 (Singletons). Let f be a non zero element of M . Is $\{f\}$ a Gröbner basis? It will be the case in many practical settings but it is not a consequence of the axioms above. Unfolding the definitions, we see that it is implied by the following assumption:

$$\mathbf{M7} \quad \forall f \in M, \forall g \in \langle af \mid a \in A \rangle \setminus \{0\}, \exists a \in A, \text{lm } g = \text{lm}(af).$$

This holds, for example, when A is the multiplicative monoid of a K -algebra and if M is a module over A , with the obvious compatibility conditions.

To check that G is a Gröbner basis, we may apply Criterion B5. This requires, in general, infinitely many checks, many of them are redundant due to the action of A . This leads to the well known Buchberger's criterion. To state it, we define a *set of critical pairs* of a set $G \subseteq M$, as any set $P \subseteq \langle AG \rangle$, such that for any $f, g \in G$, $a, b \in A$ and $\lambda \in K^\times$ such that $\text{lm}(af - \lambda bg) < \text{lm}(af)$, there are some $c \in A$ and $h \in P$ such that $af - \lambda bg = ch$. In the polynomial case, we may count on the existence of finite sets of critical pairs. We will not elaborate further in this direction.

► **Theorem 5** (Buchberger's criterion). *Let $G \subseteq M$ and let P be a set of critical pairs for G . Then G is a Gröbner basis if and only if $h \rightarrow_G 0$ for any $h \in P$.*

Proof. This follows directly from Criterion B5 and Lemma 6 below. ◀

► **Lemma 6.** *For any $G \subseteq M$, $f \in G$ and $a \in A$, if $f \rightarrow_{AG} 0$, then $af \rightarrow_{AG} 0$.*

Proof. We proceed by induction on n such that $f \xrightarrow{n} 0$. The base case $n = 0$ is trivial. Assuming $n > 0$, let $\lambda \in K$, $b \in A$ and $g \in G$ such that $f \xrightarrow{1} f - \lambda bg \xrightarrow{n-1} 0$. By induction hypothesis, $af - \lambda abg \rightarrow 0$ and it remains to check that $af \xrightarrow{1} af - \lambda bag$. This follows from M5 as follows: $\text{lm}(af - \lambda abg) = \text{lm}(a(f - \lambda bg)) < \text{lm}(af)$. ◀

3 Signatures

3.1 Signature bases

From now on, we fix a monoid A and two monomial modules over A , denoted M and S , with respective sets of monomials denoted \mathcal{M} and \mathcal{S} . A signature is an element of S . We are interested in computing Gröbner bases in M while S is the module of signatures.

In addition to the axioms above for the monomial module S , we also require that

$$\mathbf{S1} \quad \forall a \in A, \forall \sigma \in S, \forall m \in \mathcal{M}, a\sigma = \sigma \text{ and } \sigma \neq 0 \Rightarrow am = m.$$

$$\mathbf{S2} \quad \forall a, b \in A, \forall \sigma \in S, \forall m \in \mathcal{M}, a\sigma \leq b\sigma \text{ and } \sigma \neq 0 \Rightarrow am \leq bm.$$

Naturally S2 implies S1, but we state them separately because S2 will only be useful later in Section 5 (and specifically in Lemma 28) when we will study algorithms for computing signature bases and termination issues. This hypothesis is called *compatibility* by Gao et al. (2016) and others.

A *sigpair* is an element of $M \times \mathcal{S}$. The first element of a sigpair f is denoted f^{\natural} , it is the *polynomial part* of f (eventhough f may not be a polynomial, strictly speaking). The second element of a sigpair f is denoted $\text{sig } f$, it is the *signature* of f .

For a sigpair f and some $a \in A$, we define the sigpair $af = (af^{\natural}, a \text{sig } f)$. For $\lambda \in K^{\times}$, we define $\lambda f = (\lambda f^{\natural}, \text{sig } f)$. A *sigset* is a set of sigpairs. For any sigset G , let AG denote the sigset $AG = \{af \mid a \in A, f \in G\}$. For $\sigma \in \mathcal{S}$, let

$$AG^{\sigma} \doteq \{af^{\natural} \mid a \in A \text{ and } a \text{sig } f = \sigma\}, \quad AG^{\leq \sigma} \doteq \cup_{\tau \leq \sigma} AG^{\tau} \text{ and } AG^{< \sigma} \doteq \cup_{\tau < \sigma} AG^{\tau}.$$

They are subsets of M , not sigsets. Each set $AG^{< \sigma}$ defines a reduction rule $\xrightarrow{1}_{AG^{< \sigma}}$, that we denote $\xrightarrow{\sigma}_G$, the *regular reduction* in signature σ . On $M \times \mathcal{S}$, we define $f \xrightarrow{1}_G g$ if $\text{sig } f = \text{sig } g$ and $f^{\natural} \xrightarrow{\text{sig } f}_G g^{\natural}$. This is the *regular reduction* of sigpairs. The tail equivalence relations \sim_G^{σ} and \sim_G are defined similarly using $\sim_{AG^{< \sigma}}$.

The following statements are direct consequences of the axioms for monomial modules.

► **Lemma 7.** *Let G be a sigset, let $\sigma \in \mathcal{S}$ and $a \in A$. Then*

- for any $\tau \leq \sigma$, $AG^{\leq \tau} \subseteq AG^{\leq \sigma}$;
- for any $f \in AG^{\leq \sigma}$, $af \in AG^{\leq a\sigma}$;
- for any $f \in AG^{< \sigma}$, $af \in AG^{< a\sigma}$.

Signature-based algorithms for Gröbner bases actually compute something more constrained than Gröbner bases. A *signature basis* is a sigset such that for any $\sigma \in \mathcal{S}$ the set $AG^{\leq \sigma}$ is a pivot basis. In view of Theorem 2, a sigset G is a signature basis if and only if regular reduction \rightarrow_G is confluent modulo tail equivalence \sim_G . This concept is a refinement of the concept of Gröbner basis, and forgetting the signatures in a signature basis gives a Gröbner basis.

► **Lemma 8.** *If G is a signature basis, then $G^{\natural} = \{f^{\natural} \mid f \in G\}$ is a Gröbner basis.*

Proof. The set AG^{\natural} is the union of all $AG^{< \sigma}$, with $\sigma \in \mathcal{S}$. By construction, $AG^{< \sigma} \subseteq AG^{< \tau}$ if $\sigma \leq \tau$. So Lemma 3 applies and shows that AG^{\natural} is a pivot basis. ◀

3.2 Prebases

A sigset is a *prebasis* if

P1 $AG^0 \subseteq \{0_M\}$;

P2 $\forall \sigma \in \mathcal{S}, \forall f, g \in AG^{\sigma}, \exists \lambda \in K^{\times}, f - \lambda g \in \langle AG^{< \sigma} \rangle$.

Equivalently, P2 means that any $f \in AG^{\sigma}$ generates the quotient space $\langle AG^{\leq \sigma} \rangle / \langle AG^{< \sigma} \rangle$ as a K -linear space. The concept of prebasis embodies the postulate that “two elements with the same signature are substitutable”. A prebasis is an admissible input for signature algorithms.

► **Example 9.** A trivial choice for the set of signatures is $\mathcal{S} = \mathcal{M}$. Let G be a sigset such that $\text{sig } f = \text{lm } f^{\natural}$ for any $f \in G$. Then G is a prebasis if and only if G^{\natural} is a Gröbner basis. Indeed, in this case, $\langle AG^{< \sigma} \rangle = \langle ag^{\natural} \mid a \in A, g \in G, \text{lm}(ag) < \sigma \rangle$. So the condition for being a prebasis is exactly Criterion B6 (together with L2) for AG^{\natural} to be a pivot basis, that is for G^{\natural} to be a Gröbner basis.

There is a way to add elements to a prebasis which yields another prebasis. To this purpose, we say that a sigset H is a *sigsafe extension* of a sigset G if $G \subseteq H$ and for any $h \in H$, there is some $f \in AG^{\text{sig}h}$ and some $\lambda \in K^\times$ such that $h \equiv \lambda f \pmod{\langle AG^{\text{sig}h} \rangle}$. The problem of computing signature bases is more formally stated as “given a sigset G , compute a signature basis that is a sigsafe extension of G ”. (Compare with the problem of computing Gröbner bases: “given a set G , compute a Gröbner basis that generates the same ideal/module as G ”). Sigsafe extensions preserve many properties.

► **Lemma 10.** *Let G be a prebasis and let H be a sigsafe extension of G . Then:*

- $\forall \sigma \in S, \langle AG^{<\sigma} \rangle = \langle AH^{<\sigma} \rangle$ and $\langle AG^{\leq\sigma} \rangle = \langle AH^{\leq\sigma} \rangle$;
- H is a prebasis;
- if H' is a sigsafe extension of H , it is also a sigsafe extension of G .

We skip the proof, which is a simple application of Lemma 7.

Generalizing Example 9, we may construct prebasis in M from a Gröbner basis in S .

► **Theorem 11.** *Let $\phi : S \rightarrow M$ be a K -linear map commuting with the action of A . If $H \subseteq S$ is a Gröbner basis, then $\{(\phi(h), \text{lm } h) \mid h \in H\}$ is a prebasis.*

Proof. Let $G = \{(\phi(h), \text{lm } h) \mid h \in H\}$. We first check P1. Let $f \in AG^0$. By definition, there is some $h \in H$ and $a \in A$ such that $f = \phi(ah)$ and $\text{lm}(ah) = 0$. By L1, this implies $ah = 0$, so $f = 0$.

As for P2, let $\sigma \in S, f, g \in G$ and $a, b \in A$ such that $\sigma = a \text{ sig } f = b \text{ sig } g$. By definition, there are some $h, k \in H$ such that $\text{lm } h = \text{sig } f, \text{lm } k = \text{sig } g, f = \phi(h)$ and $g = \phi(k)$. In particular $\sigma = \text{lm}(ah) = \text{lm}(bk)$. By L2, there is some $\lambda \in K$ such that $ah \equiv_t \lambda bk$. By Criterion B6 applied to the pivot basis AH , we have $ah - \lambda bk = \sum_i m_i \mu_i c_i l_i$ for some $\mu_i \in K, c_i \in A$ and $l_i \in H$ such that $\text{lm}(c_i l_i) < \sigma$. In particular, $af - \lambda bg = \sum_i \mu_i c_i \phi(l_i)$ and $c_i \phi(l_i) \in AG^{<\sigma}$. ◀

► Remark 12 (Constructing prebases “for free”). As a special case of this result, we recover the following classical construction. Given $g_1, \dots, g_r \in M$, we want to find a signature set S and signatures $\sigma_1, \dots, \sigma_r$ such that $\{(g_i, \sigma_i)\}_{1 \leq i \leq r}$ is a prebasis. Then the idea, is to compute a sigsafe extension of this prebasis that is a signature basis to obtain in the end a Gröbner basis of the submodule generated by g_1, \dots, g_r , by removing signatures, Lemma 8.

Let r be a positive integer. Let $S = (M_{>} \times \{1, \dots, r\}) \cup \{0\}$, where $M_{>} = \{m \in M \mid m > 0\}$. There are two natural well-orderings on S , called *position-over-term* (POT) and *term-over-position* (TOP):

POT $(m, i) \leq_S (n, j)$ if $i < j$ or $i = j$ and $m \leq_M n$;

TOP $(m, i) \leq_S (n, j)$ if $m <_M n$ or $m = n$ and $i \leq j$.

(And 0 is always the minimal element.) Let $S = M^r$. For $m \in M$, let $m \otimes e_i$ denote the element $(0, \dots, 0, m, 0, \dots, 0)$ of S , where the nonzero component is at the i th position. If B is a basis of M such that $\text{lm}|_B$ is injective, then $\{b \otimes e_i \mid b \in B, 1 \leq i \leq r\}$ is a basis of S and we define $\text{lm}(b \otimes e_i) = (\text{lm } b, i)$, which extends uniquely to a map $\text{lm} : S \rightarrow S$ that turns S into a monomial space.

The monoid A acts on S by $a(h_1, \dots, h_r) = (ah_1, \dots, ah_r)$ and this turns S into a monomial module. Moreover, S1 and S2 are satisfied, so S is a suitable signature module, with either the POT or the TOP ordering. Let $\phi : S \rightarrow M$ defined by $\phi(h_1, \dots, h_r) = h_1 + \dots + h_r$.

Let $g_1, \dots, g_r \in M$ such that each singleton set $\{g_i\}$ is a Gröbner basis – recall that it is not automatic, see Remark 4. Let $H = \{g_i \otimes e_i \mid 1 \leq i \leq r\}$. Then it is easy to check that H is a Gröbner basis. By Theorem 11, this implies that

$$G = \{(g_i, (\text{lm } g_i, i)) \mid 1 \leq i \leq r\}$$

is a prebasis such that $G^{\text{h}} = \{g_1, \dots, g_r\}$.

► Remark 13 (Sigsafe extensions and module representation). Let $\phi : S \rightarrow M$ be a K -linear map commuting with the action of A . Consider a subset $H \subseteq S$, $G = \{(\phi(h), \text{lm } h) \mid h \in H\}$ and G' a sigsafe extension of G . Then there is a set $H' \subseteq S$ containing H such that $G' = \{(\phi(h), \text{lm } h) \mid h \in H'\}$.

4 Rewrite bases

We have defined Gröbner bases and introduced signatures. The definition of signature bases is a natural refinement of Gröbner bases in the context of signatures. Then, the definition of prebases reflects what we expect from signatures, and it is related to Gröbner bases. However, we can obtain prebases for free, that is craft a signature module and choose signatures that will turn a finite subset of M into a prebasis.

In this section, we introduce rewrite bases. The definition is purely combinatorial, it only depends on leading monomials and signatures. We will see that a prebasis that satisfies these combinatorial conditions is a signature basis. Being a signature basis is a matter of subtle arithmetic conditions. (One cannot change the coefficients of a signature basis and hope that it remains a signature basis. The same is true for Gröbner bases of course, and also prebases.) Somehow, we can split these conditions into the prebasis property and the combinatorial properties of rewrite bases. The concept was first introduced by Eder and Rouné (2013). It is simplified here by removing the need for what Eder and Rouné call a “rewrite order”. (So my definition of rewrite basis is actually different than theirs.)

4.1 Definition and basic results

For $\sigma \in \mathcal{S}$, a sigset G is a *rewrite basis at σ* if either $AG^\sigma = \emptyset$, or there is some \rightarrow_G -reduced element $f \in AG$ with $\text{sig } f = \sigma$. A sigset G is a *rewrite basis* if it is a prebasis and a rewrite basis at σ for any $\sigma \in \mathcal{S}$.

Checking if G is a rewrite basis at σ involves only manipulations in A , \mathcal{M} and \mathcal{S} , but no operations in the base field K . Moreover, if G is not a rewrite basis at some σ , then it is easy to compute a sigsafe extension of G which is a rewrite basis at σ : simply pick some $f \in AG$ with $\text{sig } f = \sigma$, compute a \rightarrow_G -normal form, and insert the result into G . This leads to an algorithm schema for computing (Pseudo-algorithm 1).

■ **Pseudo-algorithm 1** Algorithm schema for computing rewrite bases

input A full prebasis G

output A sigsafe extension H of G that is a rewrite basis

```

1 while  $G$  is not a rewrite basis do
2   pick  $\sigma \in \mathcal{S}$  such that  $G$  is not a rewrite basis at  $\sigma$ 
3   pick  $f \in AG$  with  $\text{sig } f = \sigma$             $-f$  is called the reductant
4    $g \leftarrow$  any  $\rightarrow_G$ -normal form of  $f$ 
5    $G \leftarrow G \cup \{g\}$                         $-G$  is now a rewrite basis at  $\sigma$ 
6 return  $G$ 

```

There are two significant difficulties to turn this schema into an actual algorithm. Firstly, how to check that G is a rewrite basis? And how to pick a signature at which G is not a rewrite basis? These questions are addressed in Section 4.3. Secondly, how to ensure termination? This

is addressed, in Section 5, under Noetherian hypotheses and under some restrictions on the choice of σ on line 2, or the choice of f on line 3.

Before studying rewrite bases in more details, we first introduce a classification of signatures. Let G be a prebasis. For any $\sigma \in \mathcal{S}$, either $AG^\sigma = \emptyset$, this is a trivial case, or any element of AG^σ generates the quotient $\langle AG^{\leq \sigma} \rangle / \langle AG^{< \sigma} \rangle$. In the latter case, either every element of AG^σ is in $\langle AG^{< \sigma} \rangle$, if the quotient is zero-dimensional, or no element of AG^σ is in $\langle AG^{< \sigma} \rangle$, if the quotient is one-dimensional. This leaves the following categories. A signature $\sigma \in \mathcal{S}$ is:

- an *empty signature* if $AG^\sigma = \emptyset$;
- a *nonempty signature* if $AG^\sigma \neq \emptyset$;
- a *regular signature* if it is nonempty and $AG^\sigma \cap \langle AG^{< \sigma} \rangle = \emptyset$;
- a *syzygy signature* if it is nonempty and $AG^\sigma \subseteq \langle AG^{< \sigma} \rangle$.

A nonempty signature is either regular or syzygy, as long as G is a prebasis. This classification is relative to the sigset G , but we check easily that it remains unchanged under sigsafe extensions.

► **Proposition 14.** *Let G be a prebasis and let $\sigma \in \mathcal{S}$ such that G is a rewrite basis at any signature $\tau < \sigma$. For any $f, g \in AG^\sigma + \langle AG^{< \sigma} \rangle$, there is some $\lambda \in K^\times$ such that $f \downarrow_G^\sigma \lambda g$.*

Proof. By Lemma 16 below, the set $AG^{< \sigma}$ is a pivot basis. Let $f, g \in AG^\sigma + \langle AG^{< \sigma} \rangle$. P2 implies that there is some $\lambda \in K^\times$ such that $f - \lambda g \in \langle AG^{< \sigma} \rangle$. Criterion B3 implies that $f \downarrow_G^\sigma \lambda g$. ◀

► **Lemma 15.** *Let G be a prebasis and let $\sigma \in \mathcal{S}$ such that $AG^{< \sigma}$ is a pivot basis. If G is a rewrite basis at σ or if σ is not a regular signature, then $AG^{\leq \sigma}$ is a pivot basis.*

Proof. If σ is not regular, then $\langle AG^{\leq \sigma} \rangle = \langle AG^{< \sigma} \rangle$ so $AG^{\leq \sigma}$ is a pivot basis because it contains the pivot basis $AG^{< \sigma}$.

So we may assume that σ is regular and that G is a rewrite basis at σ and prove that $AG^{\leq \sigma}$ is a pivot basis using Criterion B2. Let $f \in \langle AG^{\leq \sigma} \rangle$ and let $g \in AG^\sigma$ be a \rightarrow_G^σ -reduced element (which exists because G is a rewrite basis at σ). We write $f = c_1 f_1 + \dots + c_r f_r + h$ with $f_i \in AG^\sigma$, $c_i \in K$ and $h \in \langle AG^{< \sigma} \rangle$. For each f_i , there is a $\lambda_i \in K^\times$ such that $f_i - \lambda_i g \in \langle AG^{< \sigma} \rangle$, because G is a prebasis. So $f - \mu g \in \langle AG^{< \sigma} \rangle$, with $\mu = \sum_i c_i \lambda_i$, and Criterion B3 implies that $f \downarrow_G^\sigma \mu g$. But g is \rightarrow_G^σ -reduced, so $f \rightarrow_G^\sigma h \smile_G^\sigma \lambda g$ for some $h \in \langle AG^{\leq \sigma} \rangle$. Next, $h \rightarrow_{\{g\}} h - \lambda g$, because $h \equiv_{\text{lt}} \lambda g$. And then $h - \lambda g \rightarrow_G^\sigma \nu g$, for some $\nu \in K$. But ν must be zero because $\text{lm}(h - \lambda g) < \text{lm } g$. All in all,

$$f \rightarrow_G^\sigma h \rightarrow_{\{g\}} h - \lambda g \rightarrow_G^\sigma 0.$$

Therefore $f \rightarrow_{AG^{\leq \sigma}} 0$ and $AG^{\leq \sigma}$ is a pivot basis, by Criterion B2. ◀

► **Lemma 16.** *Let G be a prebasis and let $\sigma \in \mathcal{S}$.*

1. *If G is a rewrite basis at any regular signature $\tau < \sigma$ then $AG^{< \sigma}$ is a pivot basis.*
2. *If G is a rewrite basis at any regular signature $\tau \leq \sigma$ then $AG^{\leq \sigma}$ is a pivot basis.*

Proof. Both assertions are proved simultaneously by induction on σ .

Assume that G is a rewrite basis at any regular signature $\tau < \sigma$. Then the second point (applied in signatures $< \sigma$) implies that $AG^{< \tau}$ is a pivot basis for any $\tau < \sigma$. Lemma 3 (applied as in Example 9), implies that $AG^{< \sigma}$, which is nothing but $\cup_{\tau < \sigma} AG^{\leq \tau}$, is a pivot basis.

Assume that G is a rewrite basis at any regular signature $\tau \leq \sigma$. Then the first point implies that $AG^{< \sigma}$ is a pivot basis. Lemma 15 then implies that $AG^{\leq \sigma}$ is a pivot basis. ◀

4.2 Relation between signature bases and rewrite bases

It follows immediatly from Lemma 16 that rewrite bases are signature bases. Conversely, the only property that a signature basis misses to be a rewrite basis, is an explicit marking of syzygy signatures by sigpairs with polynomial parts equal to zero.

► **Proposition 17.** *Let $G \subseteq M$ be a prebasis. G is a signature basis if and only if G is a rewrite basis at any regular signature.*

Proof. Assume first that G is a signature basis. Let $\sigma \in S$ be a regular signature and let us prove that G is a rewrite basis at σ . Because σ is regular there is some $f \in AG$ with $\text{sig } f = \sigma$. Let v be a \rightarrow_G -normal form of f , with respect to G .

The signature σ is regular, so $AG^\sigma \cap \langle AG^{<\sigma} \rangle = \emptyset$. In particular f^{\natural} is not in $\langle AG^{<\sigma} \rangle$, and thus v^{\natural} is not zero. Because $AG^{<\sigma}$ is a pivot basis, v^{\natural} is reducible with respect to $AG^{<\sigma}$. So there is some $g \in AG$ such that $\text{sig } g \leq \sigma$ and $\text{lm } g = \text{lm } v$. But v is \rightarrow_G -reduced so $\text{sig } g = \sigma$. Moreover $\text{lm } v = \text{lm } g$, so g is also \rightarrow_G -reduced. So G is a rewrite basis at σ .

The converse follows from the follows directly from Lemma 16. ◀

One step further, we state this equivalence theorem between signature bases with explicit marking of syzygy signatures and rewrite bases. The proof is an easy exercise. This statement does not hold for the original definition of rewrite bases by Eder and Roune (2013, §3.2). This is the main motivation for the simplified definition.

► **Theorem 18.** *Let $G \subseteq M$ be a prebasis. G is a rewrite basis if and only if the following hold:*

- G is a signature basis;
- for any syzygy signature σ , there is a τ dividing σ such that $0 \in G^\tau$.

4.3 A criterion for rewrite bases

There is a criterion (that we will call *Faugère's criterion*) to check that a prebasis is a rewrite basis. It plays the same role as Buchberger's criterion plays for Gröbner bases: reducing a definition that involves infinitely many monomials or signatures to finitely many computations. However, the analogy between the two criterion is rather thin. For one, Faugère's criterion is not derived from Buchberger's one and I could not find either a derivation of Buchberger's criterion from Faugère's one. Moreover, Faugère's criterion only involves combinatorial operations (on leading monomials and signatures) while Buchberger's criterion involves arithmetic operations through the reductions of S-pairs. When applying Faugère's criterion, the arithmetic side (that is how the coefficients are relevant) is hidden in the prebasis hypothesis.

The slogan of signature-based algorithms for Gröbner bases is “process at most one S-pair per signature”, an algorithmic point of view on the idea that “two elements with the same signature are substitutable”. Going one step further, we may ask at which signature we *need* to process a S-pair. In what follows, the concept of S-pair, inherited from Buchberger's algorithm, fades in favor of a study of the signatures themselves. This approach is somewhat closer to the concept of J-pairs proposed by Gao et al. (2016).

Our goal here, given a prebasis G , is to define a set of signatures $\Sigma(G)$ such that it is enough to check that G is a rewrite basis at any signature in $\Sigma(G)$ to prove that G is a rewrite basis. Naturally we want $\Sigma(G)$ to be as small as possible. And as soon as we will have introduced Noetherian hypotheses, we will want $\Sigma(G)$ to be finite and computable in a combinatorial way (that is without arithmetic operations in the base field).

For a sigset G and a sigpair f , the *critical set* of f modulo G , denoted $\Sigma(f, G)$, is the set of all $\sigma \in S$ such that:

- C1** $\exists a \in A, a \text{ sig } f = \sigma$ and af is not \rightarrow_G -reduced;
C2 $\forall b, c \in A, (cb \text{ sig } f = \sigma \text{ and } b \text{ sig } f < \sigma) \Rightarrow bf$ is \rightarrow_G -reduced.

In other words, the condition C1 defines a subset of S that is closed under the action of A , whereas the condition C2 retains only some minimal elements for divisibility. The *critical set* of G , is the set of signatures

$$\Sigma(G) \doteq \bigcup_{f \in G} \Sigma(f, G).$$

There is a resemblance with the notion of critical pairs in Buchberger's criterion (see Section 2.4) but also an important difference: critical pairs are elements of M , while the critical set $\Sigma(f, G)$ only contains signatures, it is a combinatorial content. Note that $\Sigma(f, G)$ is included in the union $\cup_{g \in G} \Sigma(f, \{g\})$ and $\Sigma(f, \{g\})$ may be thought as the set of signatures of the possible S-pairs between f and g . In the classical polynomial setting, $\#\Sigma(f, \{g\}) = 1$. In the general case, $\Sigma(f, \{g\})$ can contain zero, one, finitely many or infinitely many elements, see Section 6 for examples.

► **Theorem 19.** *Let G be a prebasis. If G is a rewrite basis at σ at any signature $\tau < \sigma$, then G is a rewrite basis at σ or $\sigma \in \Sigma(G)$.*

Proof. Assume that G is not a rewrite basis at σ and let us prove that $\sigma \in \Sigma(G)$. We may assume that $AG^\sigma \neq \emptyset$, otherwise G is a rewrite basis at σ . Let $a \in A$ and $f \in G$ such that $a \text{ sig } f = \sigma$. We choose f so that $\text{lm}(af)$ is smallest. By hypothesis, af is not \rightarrow_G -reduced (otherwise G is a rewrite basis at σ). Let $b \in A$ such that:

- bf is not G -reduced;
- $\exists c \in A, cb \text{ sig } f = \sigma$.

(These conditions are not empty, we may choose $b = a$ and $c = 1_A$.) Let $\tau = b \text{ sig } f$. Among all possible choice of b , we minimize τ . We check easily that this implies $\tau \in \Sigma(f, G)$.

If $\tau = \sigma$, we are done: $\sigma \in \Sigma(f, G)$. For contradiction, assume that $\tau < \sigma$. In particular, G is a rewrite basis at τ . So there is some \rightarrow_G^τ -reduced $g \in AG^\tau$. By Proposition 14, there is $\lambda \in K^\times$ such that $g \downarrow_G^\tau \lambda bf$. Since g is \rightarrow_G^τ -reduced and bf is not, this implies that $\text{lm } g < \text{lm}(bf)$, and, by M5, that $\text{lm}(cg) < \text{lm}(cbf)$. Moreover, $cb \text{ sig } f = a \text{ sig } f$, so S1 implies that $\text{lm}(cbf) = \text{lm}(af)$, and therefore $\text{lm}(cg) < \text{lm}(af)$. This contradicts the minimality of $\text{lm}(af)$. ◀

This statement will be often used in the following form.

► **Corollary 20.** *Let G be a prebasis. If G is a rewrite basis at any $\sigma \in \Sigma(G)$, then G is a rewrite basis.*

4.4 Minimal elements in rewrite bases

We first introduce a binary relation on the set of sigpairs. We say that g *dominates* f , and denote it $g \sqsubseteq f$, if one of the following holds:

- D1** $\exists a \in A, a \text{ sig } g = \text{sig } f$ and $a \text{ lm } g \leq \text{lm } f$;
D2 $\exists a \in A, a \text{ sig } g < \text{sig } f$ and $a \text{ lm } g = \text{lm } f$.

The sigpairs f and g are \sqsubseteq -equivalent if $f \sqsubseteq g$ and $g \sqsubseteq f$ (which actually means that $\text{lm } f = \text{lm } g$ and $\text{sig } f = \text{sig } g$, see Lemma 21 below.) We say that $g \sqsubset f$ if $g \sqsubseteq f$ and f is not equivalent to g . A sigpair f is *dominant* in a sigset G if $f \in G$ and for any $g \in G$ such that $g \sqsubseteq f$, we also have $f \sqsubseteq g$. Note that the domination relation may not be transitive, although both D1 and D2, considered separately, define a transitive relation. Note also that D1 is the covering relation defined by Gao et al. (2016, p. 454).

The elements of a sigset that are strictly dominated are useless in a rewrite basis. It is important to understand why. The condition D2 means that ag can be used to top-reduce f , so f will never help any sigset containing also g to be a rewrite basis. The interpretation of the condition D1 splits into two cases. Firstly, when $a \text{lm } g = \text{lm } f$, then f will not help because ag can serve just as well in any situation where f would serve. When $a \text{lm } g < \text{lm } f$, Proposition 14 proves that f will never be reduced in a rewrite basis containing g .

► **Lemma 21.** *Sigpairs f and g are \sqsubseteq -equivalent if and only if $\text{lm } f = \text{lm } g$ and $\text{sig } f = \text{sig } g$.*

Proof. If $f \sqsubseteq_2 g$ (meaning that f and g satisfy D2), then the reverse domination $f \sqsubseteq g$ implies that $a\sigma < \sigma$ for some $a \in A$, in contradiction with M6. Similarly, $g \sqsubseteq_2 f$ leads to a contradiction. So we may assume that $f \sqsubseteq_1 g$ and $g \sqsubseteq_1 f$, that is, there are $a, b \in A$ such that, on the one hand, $a \text{sig } g = \text{sig } f$ and $a \text{lm } g \leq \text{lm } f$, and on the other hand, $b \text{sig } f = \text{sig } g$ and $b \text{lm } f \leq \text{lm } g$. By M6, this implies that $\text{sig } f = \text{sig } g$ and $\text{lm } f = \text{lm } g$. ◀

► **Theorem 22.** *Let G be a prebasis and H be a sigsafe extension such that every element of H is dominated by an element of G . Let σ be a signature such that H and G are rewrite bases at any signature $\tau < \sigma$. Then H is a rewrite basis at σ if and only if G is a rewrite basis at σ .*

Proof. A sigsafe extension of a rewrite basis is a rewrite basis, so one inclusion is clear. Conversely, assume that H is a rewrite basis at σ . Because H is a rewrite basis at σ , there are $b \in A$ and $f \in H$ such that $b \text{sig } f = \sigma$ and bf is \rightarrow_H -reduced (and thus \rightarrow_G -reduced too). By hypothesis, there is some $g \in G$ such that $g \sqsubseteq f$. Since bf is \rightarrow_G -reduced, D2 cannot hold, so D1 does: there is some $a \in A$ such that $a \text{sig } g = \text{sig } f$ and $a \text{lm } g \leq \text{lm } f$.

Since H is a sigsafe extension of G , $f \in AG^\sigma + \langle AG^{<\sigma} \rangle$ (maybe after a scalar multiplication), by definition. By Proposition 14, there is some $\lambda \in K^\times$ such that $bf \downarrow_G \lambda bag$. Since bf is \rightarrow_G -reduced, this implies $\text{lm}(bf) \leq \text{lm}(bag)$. Combining with the previous inequality, we obtain that $\text{lm}(bag) = \text{lm}(bf)$. So bag , which has same leading monomial and signature as bf , is \rightarrow_G -reduced and thus G is a rewrite basis at σ . ◀

Combining with Corollary 20, we obtain the following corollary which may be used to reduce the number of signatures to consider when computing a rewrite basis.

► **Corollary 23.** *Let G be a prebasis and H be a sigsafe extension such that every element of H is dominated by an element of G . If H is a rewrite basis at any σ for any $\sigma \in \Sigma(G)$, then G and H are rewrite bases.*

4.5 Syzygies

When a rewrite basis comes from a Gröbner basis in the signature module through a map $\phi : S \rightarrow M$ (see Section 3.2), the syzygy signatures have an interpretation in terms of the kernel of ϕ . This is an important feature of rewrite bases that can be exploited to compute efficiently colon ideals and saturations (Gao et al., 2010; Eder et al., 2022).

► **Proposition 24.** *Let $\phi : S \rightarrow M$ be a linear map commuting with the action of A , let $H \subseteq S$ be a Gröbner basis, let $G = \{(\phi(h), \text{lm } h) \mid h \in H\}$, and $K = \{h \in H \mid \phi(h) = 0\}$. If G is a rewrite basis, then K is a Gröbner basis and $\langle AK \rangle = \ker \phi \cap \langle AH \rangle$.*

Proof. It is clear that $\langle AK \rangle \subseteq \ker \phi \cap \langle AH \rangle$. Let $h \in \ker \phi \cap \langle AH \rangle$, let $\sigma = \text{lm } h$ and let us prove that there is some $k \in AK$ such that $\text{lm } h = \text{lm } k$. This will prove at the same time that K is a Gröbner basis, using Criterion B1, and that $\langle AK \rangle = \ker \phi \cap \langle AH \rangle$.

Because H is a Gröbner basis, we can decompose h as $\lambda p + q$, with $p \in AH^\sigma$, $q \in \langle AH^{<\sigma} \rangle$ and $\lambda \in K^\times$ (using the first reduction step of the reduction given by Criterion B2). In particular $\lambda^{-1}\phi(h) \in AG^\sigma + \langle AG^{<\sigma} \rangle$. Since G is a rewrite basis at σ , there is some $a \in A$ and $g \in G$ such that ag is \rightarrow_G -reduced and $a \text{ sig } g = \sigma$. By Proposition 16, we have $\phi(h) \downarrow_G^\sigma \mu ag^{\text{h}}$ for some $\mu \in K^\times$. But $\phi(h) = 0$ and ag^{h} is \rightarrow_G^σ -reduced, so $ag^{\text{h}} = 0$ and therefore $g^{\text{h}} = 0$. By definition of H , $g = (\phi(k), \text{lm } k)$ for some $k \in H$. And since $g^{\text{h}} = 0$, we have $k \in K$. In particular, $\text{lm } h = \sigma = \text{lm}(ak)$. ◀

5 Algorithm templates

In all this section we assume that M and S are Noetherian monomial modules, which we define in Section 5.1. This will imply the finiteness of the critical set $\Sigma(G)$ of finite sigsets G as well as the existence of finite sigsafe extensions that are rewrite bases, for any sigsets.

As it will become clear, there is not a single algorithm for computing rewrite bases. There are many possible variants, some major, such as F5 selection strategy or F4-style reduction, and some minor. There are also many possible ways to combine them. More than to prescribe some algorithms, the goal of this section is to highlight design principles.

Section 5.1 introduces the Noetherian hypotheses. Section 5.2 studies an algorithm where signatures are processed *in order*, that is when a signature is always processed after any smaller signatures. This is a natural setting, yielding simple proofs of termination, but it does not fit all situations. Section 5.3 studies the idea of minimizing the leading monomial of the reductant, in the style of Arri and Perry (2011) and Sun and Wang (2011). Again, it leads to rather simple proofs of termination, but it leaves aside other reductant selection strategy, such as the original F5 strategy.

To study algorithms where the signatures may be processed *out of order* and the reductant selected (almost) freely, Section 5.4 introduces *sigtrees*. It is a tree whose nodes are the elements of the rewrite basis being computed, and g is a child of f if was obtained from a reduction a multiple of f . Under mild hypotheses, sigtrees are finite (Proposition 34) and this gives a very useful termination criterion (Theorem 35). This criterion is put into practice in Section 5.5, to study the F5 selection strategy with out-of-order signature processing, in Section 5.6, to study the most general selection strategy, according to the sigtree criterion, and in Section 5.7 to study simultaneous reduction in the F4 style.

5.1 Noetherian monomial modules

A partial order \preceq on a set X is a *well partial order* (or *wpo*) if for any sequence $(x_i)_{i \geq 0}$ in X , there are some $i < j$ such that $x_i \preceq x_j$. A subset T of a partially ordered set X is *closed* if $a \preceq b$ and $a \in T$ imply $b \in T$. Wpos have several equivalent characterizations.

► **Lemma 25** (Higman (1952, Theorem 2.1)). *Let X be a set with a partial order \preceq . The following assertions are equivalent:*

- N1** any sequence $T_0 \subseteq T_1 \subseteq \dots$ of closed subsets of X stabilizes;
N2 for any sequence $(x_i)_{i \geq 0}$ in S , there are some $i < j$ such that $x_i \preceq x_j$;
N3 for any sequence $(x_i)_{i \geq 0}$ in S , there is a subsequence $(x_j)_{j \geq 0}$ such that $x_j \preceq x_{j+1}$ for any $j \geq 0$;
N4 for any closed set $T \subseteq X$, there is finite set B such that $T = \{x \in X \mid \exists b \in B, b \preceq x\}$.

A monomial set \mathcal{M} of a monomial module M is partially ordered by divisibility, an order that we will denote \preceq , not to be confused with the total order \leq . Namely, $m \preceq n$ if there is some $a \in A$ such that $am = n$. Nonetheless, if $a \preceq b$ then $a \leq b$, by M6. The monomial space M is *Noetherian* if \preceq is a wpo. In case M is a module over some algebra generated by A (with the obvious compatibility hypotheses), it is possible that M is Noetherian as a module but not as a monomial module, see Section 6.5 for an example. However, if M is Noetherian as a monomial space, it is Noetherian as a module.

So we now assume that the monomial spaces M and S (the signature module) are Noetherian. The first interesting consequence is the finiteness of the critical set $\Sigma(G)$ for a given finite sigset G .

► **Lemma 26.** *Let G be a finite sigset. If S is Noetherian then $\Sigma(G)$ is finite.*

Proof. Let $f \in G$. By definition, $\Sigma(f, G)$ is the set of \preceq -minimal elements of some closed subset of S . By Criterion N4, it is finite. ◀

The termination arguments will not follow from the Noetherianity of M or S alone, but in conjunction. More precisely, in $\mathcal{M} \times S$ we define $(m, \sigma) \preceq (n, \tau)$ if $m \preceq n$ and $\sigma \preceq \tau$. In other words, $(m, \sigma) \preceq (n, \tau)$ if there $a, b \in A$ such that $am = n$ and $b\sigma = \tau$. Let me insist that a and b may not be equal.

► **Lemma 27.** *If M and S are Noetherian monomial modules, then \preceq is a wpo on $\mathcal{M} \times S$.*

Proof. Let $((m_i, \sigma_i))_{i \geq 0}$ be an infinite sequence in $\mathcal{M} \times S$. By Criterion N3, we may assume, up to extracting a subsequence, that $m_i \preceq m_{i+1}$. Similarly, we may assume, up to extracting a subsubsequence, that $\sigma_i \preceq \sigma_{i+1}$. So \preceq on $\mathcal{M} \times S$ satisfies Criterion N3. ◀

► **Lemma 28.** *For any sigpairs f and g , if $(\text{sig } f, \text{lm } f) \preceq (\text{sig } g, \text{lm } g)$ then $f \sqsubseteq g$.*

Proof. Let $a, b \in A$ such that $a \text{ sig } g = \text{sig } f$ and $b \text{ lm } g = \text{lm } f$. If $b \text{ sig } g < \text{sig } f$, then D2 holds (with b in place of a). Otherwise, if $a \text{ sig } g = \text{sig } f \leq b \text{ sig } g$, then S2 implies that $a \text{ lm } g \leq b \text{ lm } g = \text{lm } f$ (so D1 holds), unless $\text{sig } g = 0$. In this last case, we have $\text{sig } f = b \text{ sig } g = 0$ and $b \text{ lm } g = \text{lm } f$, so D1 also holds. ◀

The following statement will underlie all the termination proofs. It is an analogue of Dixon's Lemma for sigpairs. However, we will see that this statement may not apply directly. The relation \preceq on $\mathcal{M} \times S$, the domination relation \sqsubseteq and Lemma 28 appeared first in the work of Arri and Perry (2011, 2017) and they have been used several times since then (Eder & Perry, 2011; Roune & Stillman, 2012; Gao et al., 2016).

► **Proposition 29** (Dixon's Lemma for sigpairs). *For any infinite sequence $(f_i)_{i \geq 0}$ of sigpairs, there are indices $i < j$ such that $f_i \sqsubseteq f_j$.*

Proof. It is a direct corollary of Lemma 27, Criterion N2 and Lemma 28. ◀

5.2 Processing signatures in order

By Theorem 19, we can compute the smallest signature at which a given prebasis G is not a rewrite basis: it must be an element of the critical set $\Sigma(G)$, which is finite by Lemma 26. This signature has many good properties induced by Proposition 14, and in particular we deduce the following one.

► **Proposition 30.** *Let G be a prebasis and let σ such that G is a rewrite basis at any $\tau < \sigma$. Let $f \in AG$ with $\text{sig } f = \sigma$ and let h be any \rightarrow_G -normal form of f . Then $g \not\sqsubseteq h$ for any $g \in G$.*

Proof. By contradiction, assume that there is some $g \in G$ such that $g \sqsubseteq h$. Domination condition D2 is ruled out because h is \rightarrow_G -reduced. Therefore D1 holds: there is some $a \in A$ such that $a \text{ sig } g = \sigma$ and $\text{lm}(ag) \leq \text{lm } h$.

By minimality of σ and Proposition 14, $f \downarrow_G \lambda ag$ for some $\lambda \in K^\times$. By confluence, we also have $h \downarrow_G \lambda ag$. Since h is \rightarrow_G -reduced, this implies that $\text{lm } h \leq \text{lm}(ag)$. Combining the condition D1, we obtain that $\text{lm}(ag) = \text{lm } h$ and therefore that ag is \rightarrow_G -reduced. So G is a rewrite basis at σ , in contradiction with the hypothesis. ◀

This leads to Algorithm 2. There is no restriction whatsoever on the choice of the reductant on line 9, they all reduce to the same sigpair, up to scaling and tail equivalence \simeq_G (Proposition 14).

■ **Algorithm 2** Computation of a rewrite basis handling signatures in increasing order

input A finite prebasis G

output A finite sigsafe extension of G which is a rewrite basis

```

7  while  $G$  is not a rewrite basis at all  $\sigma \in \Sigma(G)$  do
8       $\sigma \leftarrow \min \{ \sigma \in \Sigma(G) \mid G \text{ is not a rewrite basis at } \sigma \}$ 
9      pick any  $f \in AG$  with  $\text{sig } f = \sigma$ 
10      $g \leftarrow$  any  $\rightarrow_G$ -normal form of  $f$ 
11      $G \leftarrow G \cup \{g\}$ 
12 return  $G$ 

```

► **Theorem 31.** *Algorithm 2 is correct and terminates.*

Proof. Correction follows from Theorem 19. For contradiction, assume that the algorithm does not terminate for some input. Let g_1, g_2, \dots be the sigpairs that are inserted to G on line 11 on each iteration. By Proposition 29, there are some indice $i < j$ such that $g_i \sqsubseteq g_j$. However, this contradicts Proposition 30. ◀

This algorithm is close in essence to the original F5 algorithm (Faugère, 2002) and more generally to the RB algorithm of Eder and Perry (2011). The notion of critical set and the notation $\Sigma(G)$ greatly simplify the presentation of the algorithm, but it hides combinatorial computations. For example, how to update $\Sigma(G)$ after inserting a new element? How to find the next signature to handle? How to check the halting condition? These questions are addressed in Section 5.6.

5.3 Minimizing the leading monomial of the reductant

Processing S-pairs in increasing order seems to be a natural option but it is also important to understand what happens when signatures are processed in any order. There may be

various reasons to do so: parallel computing, simultaneous reduction in the F4 style (Faugère, 1999). Recently, Eder et al. (2022) used signature algorithms to compute the saturations, this involves enlarging the the input ideal on the fly. It can be interpreted as an algorithm processing signatures out of order.

In a time where the termination of F5 was still unsettled, Arri and Perry (2011) – mind the erratum (Arri & Perry, 2017) – introduced the idea of choosing carefully the sigpair to be reduced, I call it the *reductant*, at a given signature to ensure termination. This is based on the following observation.

► **Proposition 32.** *Let G be a prebasis and let σ be a signature at which G is not a rewrite basis. Let $f \in AG$ with $\text{sig } f = \sigma$ and $\text{lm } f$ minimal. Let h be a \rightarrow_G -normal form of f . Then $g \not\sqsubseteq h$ for any $g \in G$.*

Proof. By contradiction, assume that there is some $g \in G$ such that $g \sqsubseteq h$. Condition D2 is ruled out because h is \rightarrow_G -reduced. Therefore D1 holds: there is some $a \in A$ such that $a \text{ sig } g = \sigma$ and $\text{lm}(ag) \leq \text{lm } h$. Besides, G is not a rewrite basis at σ , it follows that f is not G -reduced, and thus $\text{lm } h < \text{lm } f$ since $f \rightarrow_G h$. It follows that $\text{lm}(ag) < \text{lm } f$, which contradicts the minimality of f . ◀

Although Arri and Perry still requires to process signatures by increasing order, Proposition 32 opens the way for out-of-order signature handling, as Sun and Wang (2013) and Gao et al. (2016) did. The formulation that I propose (Algorithm 3) is mostly equivalent to that of the latter. The choice of signature on line 14 is unconstrained, but the choice of the reductant is imposed.

■ **Algorithm 3** Computation of a rewrite basis with out-of-order signature processing, minimizing the leading monomial of the reductant

input A finite prebasis G

output A finite sigsafe extension of G which is a rewrite basis

```

13 while  $G$  is not a rewrite basis at all  $\sigma \in \Sigma(G)$  do
14     pick any  $\sigma \in \mathcal{S}$  such that  $G$  is not a rewrite basis at  $\sigma$ 
15     pick  $f \in AG$  with  $\text{sig } f = \sigma$  and  $\text{lm } f$  minimal
16      $g \leftarrow$  any  $\rightarrow_G$ -normal form of  $f$ 
17      $G \leftarrow G \cup \{g\}$ 
18 return  $G$ 

```

► **Theorem 33.** *Algorithm 3 is correct and terminates.*

Proof. Identical to the proof of Theorem 31, but using Proposition 32 instead of Proposition 30. ◀

5.4 Well-formed sigtrees

A *tree* is a set \mathcal{T} , finite or infinite, of finite sequences of nonnegative integers such that for any $(k_1, \dots, k_r) \in \mathcal{T}$ (with $r \geq 1$), the prefix subsequence (k_1, \dots, k_{r-1}) is also in \mathcal{T} . The elements of \mathcal{T} are called *nodes*. The *children* of a given node $n \in \mathcal{T}$ are the sequences in \mathcal{T} that extend n by exactly one integer. The *root node* is the empty sequence, denoted ε , it is the only node that is not the child of another node. The set $\mathcal{T} \setminus \{\varepsilon\}$ of nonroot nodes is denoted \mathcal{T}^+ . The nodes are

partially ordered: $n \leq_{\mathcal{T}} m$ if n is a prefix of m . For any node $n \in \mathcal{T}$, the set $\{m \in \mathcal{T} \mid m \leq_{\mathcal{T}} n\}$ is the set of *ancestors* of n .

A *sigtree* is a tree \mathcal{T} together with a rank function $\text{rk} : \mathcal{T} \rightarrow \mathbb{N}$ and a label function $\lambda : \mathcal{T}^+ \rightarrow M \times S$ (recall that $M \times S$ is the set of sigpairs). A sigtree is a natural way to represent the process of computing a rewrite basis. Indeed, Algorithms 3 or 2, as well as Pseudo-algorithm 1, produce naturally a sigtree as follows. The elements of G are the labels of a sigtree, which at the beginning of the algorithm, is made only of the root node and one leaf for each element in the input. Then, each time some $f \in AG$ is picked, reduced, and inserted into G , we can write $f = a\lambda(n)$, for some nonroot node n of the sigtree, and we insert in the sigtree a new child of n containing the new element. The rank function reflects the *birthdate* of a node.

A *well-formed sigtree* is a sigtree \mathcal{T} such that:

- T1** $\forall n \in \mathcal{T}^+, \forall m$ child of $n, \exists a \in A, a \text{ sig } \lambda(n) = \text{sig } \lambda(m)$ and $a \text{ lm } \lambda(n) > \text{lm } \lambda(m)$,
“a child is more reduced than its parent”;
- T2** $\forall n \in \mathcal{T}^+, \lambda(n)$ is \rightarrow -reduced with respect to the sigset $\{\lambda(p) \mid \varepsilon <_{\mathcal{T}} p <_{\mathcal{T}} n\}$,
“a child is reduced modulo its ancestors”;
- T3** $\forall n \in \mathcal{T}, \forall p, q$ children of $n, \text{rk}(p) < \text{rk}(q) \Rightarrow \text{sig } \lambda(p)$ does not divide $\text{sig } \lambda(q)$,
“the signature of a node does not divide that of younger sibling nodes”.

Typically, T1 and T2 are satisfied *by design* if $\lambda(m)$ is obtained by reducing $a\lambda(n)$ modulo a sigset containing at least the labels of the ancestors of m , and assuming that $a\lambda(n)$ is indeed reducible to account for the strict inequality in T1. T3 is the real constraint. In the context above, T3 puts a constraint on the choice of the reductant. It means that whenever we want to reduce $a\lambda(n)$, we must first check that we have not previously computed a reduction $b\lambda(n) \rightarrow \lambda(m)$ for some child m of n and for some b such that $\exists c \in A, cb = a$. In which case we can reduce $c\lambda(m)$ instead of $a\lambda(n)$.

► **Proposition 34.** *Let \mathcal{T} be a well-formed sigtree. If $\{n \in \mathcal{T} \mid \text{rk}(n) = k\}$ is finite for any $k \in \mathbb{N}$, then \mathcal{T} is finite.*

Proof. Let \mathcal{T} be a well-formed sigtree. By König’s lemma, it is enough to prove that \mathcal{T} has no infinite branch and that every node has at most finitely many children.

If there is an infinite branch, then there is a sequence of nodes $(n_i)_{i \geq 0}$ such that n_{i+1} is a child of n_i . By Proposition 29, there are some indices $i < j$ such that $\lambda(n_i) \sqsubseteq \lambda(n_j)$. Condition D2 would contradict T2 so Condition D1 holds: there is some $b \in A$ such that $b \text{ sig } \lambda(n_i) = \text{sig } \lambda(n_j)$ and $b \text{ lm } \lambda(n_i) \leq \text{lm } \lambda(n_j)$. By T1 (applied all along the path from n_i to n_j), there is some $a \in A$ such that $a \text{ sig } \lambda(n_i) = \text{sig } \lambda(n_j)$ and $a \text{ lm } \lambda(n_i) > \text{lm } \lambda(n_j)$. But S1 implies that $a \text{ lm } \lambda(n_i) = b \text{ lm } \lambda(n_i)$, leading to a contradiction.

If a node has infinitely many children, the hypothesis ensures that we can extract an infinite sequence of children with increasing ranks. By Noetherianity of S , the signature of one child would divide the signature of another with higher rank. This contradicts T3. ◀

We will most often consider a variant of this statement. An extension of sigtree \mathcal{T} (with label function λ and rank function rk) is a sigtree \mathcal{T}' (with label function λ and rank function rk') such that $\mathcal{T} \subseteq \mathcal{T}'$, $\lambda'|_{\mathcal{T}^+} = \lambda$, and $\text{rk}'|_{\mathcal{T}} = \text{rk}$.

► **Theorem 35.** *Let $(\mathcal{T}_i)_{i \geq 0}$ be a sequence of well-formed sigtrees such that \mathcal{T}_{i+1} is an extension of \mathcal{T}_i , for any $i \geq 0$. If for any $r \geq 0$, the number of nodes of rank r in \mathcal{T}_i is finite and bounded as $i \rightarrow \infty$, then there is some $i \geq 0$ such that $\mathcal{T}_i = \mathcal{T}_{i+1}$.*

Proof. Let $\mathcal{T}_\infty = \cup_{i \geq 0} \mathcal{T}_i$, with the label function λ_∞ uniquely defined by $\lambda_\infty|_{\mathcal{T}_i} = \lambda_i$, and the rank function defined similarly. Since each \mathcal{T}_i is well-formed, we check easily that \mathcal{T}_∞ satisfies T1, T2, and T3. So \mathcal{T}_∞ is a well-formed sigtree.

For a given $r \in \mathbb{N}$, the number of nodes of rank r in \mathcal{T}_∞ is the limit of the number of nodes of rank r in \mathcal{T}_i , which is finite, by hypothesis. Therefore Proposition 34 applies and shows that \mathcal{T}_∞ is finite. So there is some $i \geq 0$ such that $\mathcal{T}_i = \mathcal{T}_{i+1} = \mathcal{T}_\infty$. ◀

5.5 The F5 reductant selection strategy

In the original presentation of F5 (Faugère, 2002), Faugère proposes to choose a reductant af where, among all possible choices, f is the “most recent”. This leads to Algorithm 4. This selection strategy leads naturally to a well-formed sigtree. So we can prove that the corresponding algorithm terminates, even if signatures are handled out of order.

■ **Algorithm 4** Computation of a rewrite basis with out-of-order signature processing and F5 selection strategy of the reductant

input A finite prebasis G

output A finite sigsafe extension of G which is a rewrite basis

```

19  $R \leftarrow$  empty dictionary mapping sigpairs to integers
20  $r \leftarrow 1$ 
21 for  $g \in G$  do  $R[g] \leftarrow 0$ 
22 while  $G$  is not a rewrite basis at all  $\sigma \in \Sigma(G)$  do
23   pick any  $\sigma \in \mathcal{S}$  such that  $G$  is not a rewrite basis at  $\sigma$ 
24   pick some  $a \in A$  and  $f \in G$  such that  $a \text{ sig } f = \sigma$  and  $R[f]$  maximal
25    $g \leftarrow$  any  $\rightarrow_G$ -normal form of  $af$ 
26    $G \leftarrow G \cup \{g\}$ 
27    $R[g] \leftarrow r$ 
28    $r \leftarrow r + 1$ 
29 return  $G$ 

```

► **Theorem 36.** *Algorithm 4 is correct and terminates.*

Proof. Correctness follows from Theorem 19. For termination, consider the sigtree induced by the algorithm: each sigpair g inserted into G on line 26 is the label of a node whose parent is the node labeled with f , where f is the sigpair picked on line 24. The rank of a node is given by R .

At each iteration of the algorithm, this sigtree is well-formed. T1 and T2 follow by construction. To check T3, we observe that the rank of a node is always greater than the rank of its parent. So, on line 24, if the node corresponding to f has already a child whose signature divides σ , this child has a higher rank than that of f , which contradicts the maximality of $R[f]$.

The number of node of a given rank is at most one, so Theorem 35 applies and shows that the algorithm terminates. ◀

5.6 Explicit management of the critical set

The presentation of Algorithms 2, 3 and 4 takes advantage of the notation $\Sigma(G)$ to abstract the handling of set of signatures to be handled from concrete questions that theory may ignore but

not practical implementations. There is a lot of room to design a proper handling of signatures, I simply show some possible variants.

5.6.1 Base algorithm In this section, we assume that we know how to operate on \mathcal{M} and sig (that is compare, test divisibility, etc.) and we assume that we have a procedure to compute the critical set $\Sigma(f, \{g\})$ of a pair of sigpairs f and g (simply denoted $\Sigma(f, g)$). Without more information on A , \mathcal{M} and S we cannot go further down into the details. In the polynomial setting, the set $\Sigma(f, g)$ may contain zero or one element and its computation amounts to a few operations on monomials, see Section 6.1,

There are many ways to proceed and Algorithm 5 is one of them. In this algorithm, the set Q contains signatures, and, at the beginning of each iteration of the “while” loop, we have the following invariant:

$$\forall \sigma \in \Sigma(G), \sigma \in Q \text{ or } G \text{ is a rewrite basis at } \sigma. \quad (1)$$

Indeed, when an element g is inserted in G , we remove sig g from Q and insert all the elements in the sets $\Sigma(g, h) \cup \Sigma(h, g)$, for $h \in G$. Since g is \rightarrow_G reduced, $G \cup \{g\}$ is a rewrite basis at σ and the inclusion

$$\Sigma(G \cup \{g\}) \subseteq \Sigma(G) \cup \bigcup_{h \in G} (\Sigma(g, h) \cup \Sigma(h, g))$$

proves that Invariant (1) is preserved. With Invariant (1) in hand, it is clear that when Algorithm 5 terminates, it returns a rewrite basis, by Theorem 19.

Termination is ensured *by design*: the sigtree induced by the computation is explicitly constructed, through the lists T , L , and R , and the selection procedure of reductants makes it sure that the sigtree is wellformed. Each iteration of the “while” loop either removes an element of Q or increase the size of the sigtree. The latter cannot happen infinitely many times, in view of Theorem 35, so Q is eventually empty and the algorithm terminates.

5.6.2 F5 variant We can specialize the reductant selection strategy to match the one of F5, exposed in Section 5.5. In this variant, it is not necessary to maintain the sigtree explicitly, we may ignore the list T .

5.6.3 A variant with signature pruning In the set Q , we may remove any element that is divided by a different element of Q . Instead of Invariant (1), we maintain the following one:

$$\forall \sigma \in \Sigma(G), (\exists \tau \in Q, \tau \text{ divides } \sigma) \text{ or } G \text{ is a rewrite basis at } \sigma. \quad (2)$$

This leads to Algorithm 7. Checking correctness is an easy exercise.

5.7 Simultaneous reduction

As another variation of Algorithm 8, we may handle several signature at a time, in the F4 style (Faugère, 1999; Albrecht & Perry, 2010; Eder & Faugère, 2017, §13). This has two aspects. The first one: the sigset G that is used to compute the reductions is only updated once in a while, not each time a new element is discovered. The new elements are inserted in a sigset N and after a bunch of signatures is handled, the elements of N are inserted in G . On line 96, the reductant g is reduced with respect to G (and as usual, multiples of elements of G can be used

■ **Algorithm 5** Computation of a rewrite basis, with explicit construction of a well-formed sigtree and explicit handling of the critical set

```

30 « initialize signature queue and sigtree » → line 43
31 while Q is not empty do
32    $\sigma \leftarrow$  some element of Q
33    $Q \leftarrow Q \setminus \{\sigma\}$ 
34   « select a reductant f in signature  $\sigma$  with corresponding node k » → line 63
35   if f is  $\rightarrow_G$ -reducible then
36      $g \leftarrow$  a  $\rightarrow_G$ -normal form of f
37     « insert a node with label g, parent k and rank r » → line 55
38     « update the queue with the new relation g » → line 59
39      $G \leftarrow G \cup \{g\}; r \leftarrow r + 1$ 
40 return G
41
42 Chunks
43 « initialize signature queue and sigtree »  $\equiv$ 
44    $Q \leftarrow \emptyset$                                 —signature queue
45    $T \leftarrow$  empty list                          —maps a node to its children
46    $L \leftarrow$  empty list                          —maps a node to its label
47    $R \leftarrow$  empty list                          —maps a node to its rank
48    $T[0] \leftarrow \emptyset; R[0] \leftarrow 0; L[0] \leftarrow \text{NULL}$  —the root node has no label
49    $n \leftarrow 1; r \leftarrow 1$                   —node and rank counters
50    $k \leftarrow 0$ 
51   for  $g \in G$  do                                  —create root nodes
52     « insert a node with label g, parent k and rank r » → line 55
53     « update the queue with the new relation g » → line 59
54
55 « insert a node with label g, parent k and rank r »  $\equiv$ 
56    $L[n] \leftarrow g; T[k] \leftarrow T[k] \cup \{n\}; R[n] \leftarrow r; T[n] \leftarrow \emptyset$ 
57    $n \leftarrow n + 1$ 
58
59 « update the queue with the new relation g »  $\equiv$ 
60   for  $h \in G$  do
61      $Q \leftarrow Q \cup \Sigma(g, h) \cup \Sigma(h, g)$ 
62
63 « select a reductant f in signature  $\sigma$  with corresponding node k »  $\equiv$ 
64    $k \leftarrow 0$                                 —start the search from the root node
65   for  $c \in T[k]$  do
66     if sig  $L[c]$  divides  $\sigma$  then
67        $k \leftarrow c$                             —go down the tree
68     goto 65                                       —continue the search from the new position
69   pick  $a \in A$  such that a sig  $L[k] = \sigma$ 
70    $f \leftarrow aL[k]$ 

```

■ **Algorithm 6** Variant of Algorithm 5 with the F5 strategy for the reductant selection

```

71 Similar to Algorithm 5, except for the following chunk
72 « select a reductant  $f$  in signature  $\sigma$  with corresponding node  $k$  »  $\equiv$ 
73    $k \leftarrow 0$ 
74   for  $1 \leq j < n$  do
75     if  $\text{sig } L[j]$  divides  $\sigma$  and  $R[j] > R[k]$  then  $k \leftarrow j$ 
76     pick  $a \in A$  such that  $a \text{ sig } L[k] = \sigma$ 
77      $f \leftarrow aL[k]$ 

```

■ **Algorithm 7** Variant of Algorithm 5 with signature pruning

```

78 Similar to Algorithm 5, except for the following chunk
79 « update the queue with the new relation  $g$  »  $\equiv$ 
80   for  $h \in G$  do
81      $Q \leftarrow Q \cup \Sigma(g, h) \cup \Sigma(h, g)$ 
82   for  $\sigma \in Q$  do
83     if  $\exists \tau \in Q \setminus \{\sigma\}, \tau$  divides  $\sigma$  then
84        $Q \leftarrow Q \setminus \{\sigma\}$ 

```

in reduction steps) and also with respect to N (but multiples cannot be used in reduction steps). In other words, the polynomial part g^h is reduced modulo the set $AG^{<\sigma} \cup N^{<\sigma}$.

The second aspect of simultaneous reduction is not actually shown in the pseudo-code, but it is made possible by the first point: it is possible to perform all the reductions in the “for” loop by to reduction to linear algebra. This second aspect is crucial for high-performance computations but it is a transparent transformation of the algorithm: it does not change what is computed.

► **Theorem 37.** *Algorithm 8 is correct and terminates.*

Proof. Termination is clear because the algorithm produces a well-formed sigtree, and at each iteration, either Q diminishes or the sigtree grows. Correctness follows from Invariant (1) which also holds for this algorithm, with a slightly different argument than the one in Section 5.6.1.

When an element g is inserted into G , if g is \rightarrow_G -reduced, then $G \cup \{g\}$ is a rewrite basis at σ so we may remove σ from Q without breaking the invariant. However, due to the nature of simultaneous reduction, it may happen that we insert an element that is not \rightarrow_G -reduced. In this case, then there is some $h \in G$ which reduces g and we check easily that $\Sigma(g, h) = \{\text{sig } g\}$. So in this case, $\text{sig } g$ is not actually removed from Q and the invariant is preserved. ◀

5.8 Counter example for termination

Proving termination with Theorem 35 still poses some constraint on the choice of the reductant, even though they are compatible with classical strategies. We cannot lift them in the axiomatic setting presented here.

Consider the polynomial algebra $R = K[w, x, y, z]$ with the lexical monomial order with $w < x < y < z$. Let $M = R, A = R \setminus \{0\}$ and $S = R^4$, with a position-over-term order: the signatures are pairs (m, i) made of a monomial $m \in \mathcal{M}$ and an index $i \in \{1, \dots, 4\}$, they are compared first with respect to the index, then to the monomial. For brevity, a pair (m, i) is denoted $m\mathbf{i}$, or just \mathbf{i} if $m = 1$. Consider the input set

$$G = G_0 = \{(x - w, \mathbf{1}), (z^2, \mathbf{2}), (yz, \mathbf{3}), (z - y, \mathbf{4})\}$$

■ **Algorithm 8** Computation of a rewrite basis, with simultaneous reduction. Pseudocode chunks are defined in Algorithm 5.

```

85 « initialize signature queue and sigtree » → line 43
86 while Q is not empty do
87   S ← some nonempty subset of Q
88   Q ← Q \ S
89   F ← ∅                                     –set of reductants and corresponding nodes
90   for σ ∈ S do                               –selection of reductants
91     « select a reductant f in signature σ with corresponding node k » → line 63
92     if f is →G-reducible then
93       F ← F ∪ {(f, k)}                       –we keep the information of the parent
94     N ← ∅                                     –set of newly computed relations
95     for (f, k) ∈ F by increasing order of sig f do –reduction of reductants
96       gh ← a →-normal form of fh w.r.t. AG<σ ∪ {hh | h ∈ N}
97       g ← (gh, σ)
98       « insert a node with label g, parent k and rank r » → line 55
99       « update the queue with the new relation g » → line 59
100      N ← N ∪ {g}                             –insertion of g in G is delayed
101    G ← G ∪ N
102    r ← r + 1
103 return G

```

Consider a run of Algorithm 3 but without constraint on the choice of the reductant: at the i th iteration, we choose a signature $\sigma \in \Sigma(G_{i-1})$ at which G_{i-1} is not a rewrite basis, choose a reductant $af \in AG_{i-1}$ with signature σ , reduce it modulo G_{i-1} and insert it into G_{i-1} to form G_i .

Table 1 shows an infinite run where the signature σ , the base reductant f and the multiplier a are all poorly chosen (Run 1). The same table also shows a run (Run 2) where the signature σ and the multiplier a are poorly chosen (as poorly as I could) but f is chosen following the F5 strategy, see Section 5.5.

6 Settings

This section describes different monomial spaces coming from different settings in computer algebra. Some are noncommutative or non-Noetherian.

6.1 Polynomial ring

Let $M = K[x_1, \dots, x_n]$ be the polynomial ring in n variables over K , which we endow with a monomial order, so the function lm is well defined. Let $A = \{x_1^{i_1} \cdots x_n^{i_n} \mid i_1, \dots, i_n \in \mathbb{N}\}$. The axioms for monomial orders ensure that M is a monomial module over A . It is Noetherian. Moreover, it satisfies the extra property M7, so construction of prebases is easy, see Remark 12.

For sigpairs f and g , the critical set $\Sigma(f, g)$ has zero or one element. There is the trivial case where $f^h = 0$ or $g^h = 0$. In this case, every multiple of f is $\rightarrow_{\{g\}}$ -reduced, so $\Sigma(f, g) = \emptyset$. When f^h and g^h are both nonzero, there are monomials $a, b \in A$ such that $a \text{ lm } f = b \text{ lm } g = \text{lcm}(\text{lm } f, \text{lm } g)$. Then there are two cases, if $a \text{ sig } f \leq b \text{ sig } g$, then $\Sigma(f, \{g\}) = \emptyset$; on the contrary, if $b \text{ sig } g < a \text{ sig } f$, then $\Sigma(f, g) = \{a \text{ sig } f\}$.

Note that we could also choose the monoid $A = K[x_1, \dots, x_n]$.

	σ	f^{\natural}	$\text{sig } f$	a	$\text{TopReduce}(af, AG^\sigma)$
run 1	$z4$	$z - y$	4	$z - y$	$z^2 - 2yz + y^2 \rightarrow y^2$
	$xz4$	$z - y$	4	$x(z - y^2)$	$xz^2 - xy^2z - xyz + xy^3 \rightarrow xy^3 \rightarrow wy^3$
	x^2z4	$z - y$	4	$x^2(z - y^3)$	$x^2z^2 - x^2y^3z - x^2yz + x^2y^4 \rightarrow x^2y^4 \rightarrow w^2y^4$
	\vdots				
	x^kz4	$z - y$	4	$x^k(z - y^{k+1})$	$\dots \rightarrow w^k y^{k+2}$
run 2	$z4$	$z + y$	4	$z - y^{10}$	$z^2 - zy^{10} - zy + y^{11} \rightarrow y^{11}$
	$xz4$	y^{11}	$z4$	x	$xy^{11} \rightarrow wy^{11} \rightarrow 0$
	z^24	y^{11}	$z4$	$z + y^{20}$	$zy^{20} + y^{31} \rightarrow y^{31} \rightarrow 0$
	$z3$	yz	3	$z + y^{30}$	$z^2y + zy^{31} \rightarrow 0$
	$x3$	xyz	3	x	$xyz \rightarrow wyz \rightarrow 0$
	$x2$	xz^2	2	x	$xz^2 \rightarrow wz^2 \rightarrow 0$

■ **Table 1** Two different tentative to compute a rewrite basis from the input sigset G_0 , see Section 5.8.

6.2 Modules over polynomial rings

Let r be a positive integer and let $M = K[x_1, \dots, x_n]^r$, which we endow with a term ordering – typically position-over-term, term-over-position, or Schreyer’s order (Kreuzer & Robbiano, 2000, §1.4). The monoid A is the same as before. M is a Noetherian monomial module and satisfies the extra condition M7.

The computation of $\Sigma(f, g)$ is slightly different. In the case where f^{\natural} and g^{\natural} are both nonzero, it may happen that no multiple of $\text{lm } f$ and $\text{lm } g$ coincide. Indeed, nonzero monomials in M have an *index* in $\{1, \dots, r\}$ which is unchanged under multiplication. Therefore, if $f^{\natural} = 0$ or $g^{\natural} = 0$, or $\text{lm } f$ and $\text{lm } g$ have different indices, then $\Sigma(f, g) = \emptyset$. Otherwise, there are monomial $a, b \in A$ such that $a \text{lm } f = b \text{lm } g$ (and $a \text{lm } f$ minimal). Depending on the comparison of $a \text{sig } f$ and $b \text{sig } g$, $\Sigma(f, g)$ is either \emptyset or $\{a \text{sig } f\}$, as in the polynomial case.

6.3 Monoid algebras

Let A be a submonoid of $\{x_1^{i_1} \cdots x_n^{i_n} \mid i_1, \dots, i_n \in \mathbb{N}\}$ and let $M = K[A]$ be the ring of polynomials whose monomials are contained in A . It is clear that M is a Noetherian monomial module over A . This case includes the “semigroup algebras” studied by Bender et al. (2019). It also includes some algebras that are interesting in singularity theory such that $K[x^2, xy, y^2]$, that are polynomial ring with finitely many monomials removed (in this case x, y , and xy).

The critical set $\Sigma(f, g)$ can contain more than one element. Assume, for example, that $M = K[x^2, xy, y^2]$ – that is $A = \{x^i y^j \mid i + j \geq 2\}$ – and that $f^{\natural} = x^2$ and $g^{\natural} = xy$. The set of all $a \in A$ such that $\text{lm}(af^{\natural})$ is divided by $\text{lm}(g^{\natural})$ is generated by xy and y^2 . It is not generated by y because y is not in A . Assuming that $xy \text{sig } f > x^2 \text{sig } g$ and $y^2 \text{sig } f > xy \text{sig } g$, we have

$$\Sigma(f, g) = \{xy \text{sig } f, y^2 \text{sig } f\}.$$

6.4 Weyl algebras

Let $M = K\langle x_1, \dots, x_n, \partial_1, \dots, \partial_n \rangle$ be the Weyl algebra on n variables. It is noncommutative. We may define it as the subalgebra of $\text{End}_K(K[X_1, \dots, X_n])$ where x_i is the multiplication by X_i and ∂_i is the differentiation with respect to X_i . Concretely, $x_i x_j = x_j x_i$, $\partial_i \partial_j = \partial_j \partial_i$, $\partial_i x_j = x_j \partial_i$

(if $i \neq j$) and $\partial_i x_i = x_i \partial_i + 1$. A basis of M is given by the monomials $x_1^{i_1} \cdots x_n^{i_n} \partial_1^{j_1} \cdots \partial_n^{j_n}$ and we can consider the same monomial orderings as we would do for a commutative polynomial ring in $2n$ variables.

For the monoid A , we cannot choose the set of monomials because it is not closed under multiplication. We choose instead A to be the submonoid of M generated by x_1, \dots, x_n and $\partial_1, \dots, \partial_n$. We could also choose $A = M$. This turns M into a Noetherian monomial module with the extra property M7, so we can construct signature modules with Remark 12. We could also choose A to be the monoid of nonzero elements of R . In spite of the noncommutativity, things behave similarly to the polynomial case. It is in fact *quasicommutative*: for any $a, b \in M$, $\text{lm}(ab) = \text{lm}(ba)$.

6.5 Localization of Weyl algebra

Let $f \in K[X_1, \dots, X_n]$ be a non constant polynomial. We consider M to be the Weyl algebra as above, but with an extra variable T which we think of as $1/f$. So we have the following commutation rules: $x_i T = T x_i$ and $\partial_i T = T \partial_i - \frac{\partial f}{\partial x_i} T^2$. The monomials $x_1^{i_1} \cdots x_n^{i_n} \partial_1^{j_1} \cdots \partial_n^{j_n} T^k$ form a basis of M . We choose A to be the submonoid of M generated by T, x_1, \dots, x_n and $\partial_1, \dots, \partial_n$. If we choose a monomial ordering that considers first the total degree with respect to $\partial_1, \dots, \partial_n$, we are in a quasicommutative setting, M is a Noetherian monomial module and behaves similarly to the polynomial case.

However, for computing saturations, we may want to consider an elimination order for T , so we compare first the degree with respect to T . In particular, we obtain that the leading monomial of $T \partial_i$ is $\text{lm}\left(\frac{\partial f}{\partial x_i}\right) T^2$, it is not $\partial_i T$. The action of A on M is well defined by the formula $a \text{lm}(f) = \text{lm}(af)$, but it is *not* given by adding the exponents. Yet, M is a monomial module. It is not Noetherian, even though M is Noetherian as a ring, or even as a module over the Weyl algebra. It does not satisfy the extra condition M7.

6.6 Differential algebras

Let $M = K[t, x_0, x_1, x_2, \dots]$, a polynomial ring in infinitely many variables with a derivation defined by $t' = 1$ and $x_i' = x_{i+1}$. Let $W = M\langle \partial \rangle$ be the subalgebra of $\text{End}_K M$ where M acts by multiplication and ∂ be the derivation, similarly to the Weyl algebra case. This turns M into a left W -module and *differential ideals* are defined to be the submodules of M . We choose on M a lexicographic ordering with $t < x_0 < x_1 \dots$

We choose A to be the monoid generated by $\partial, t, x_0, x_1 \dots$. This turns M into a monomial module. It is quasicommutative (that is $\text{lm}(abm) = \text{lm}(bam)$ for any $a, b \in W$ and $m \in M$) but not Noetherian. However, it satisfies the extra condition M7 and the critical sets $\Sigma(f, G)$ are finite. This example extends to several independent variables and several function variables.

6.7 Free algebras

Let M be the free algebra generated by n variables x_1, \dots, x_n . A basis of M is given by the set of words in x_1, \dots, x_n . A monomial order may be given, for example, by comparing the degree first and then the lexicographic order. We choose A to be the monoid of words, which acts naturally on M by left multiplication. In this way we may deal with left ideals in M . To deal with two-sided ideals of M , we choose A to be the set of pairs of words, with the composition $(a, b)(a', b') = (aa', b'b)$ and the action on M given by $(a, b)m = amb$.

In both cases, M is a monomial module with the extra condition M7, but not Noetherian. In the one-sided case, the critical sets are finite, but they are not in the two-sided case. However, in the two-sided case, the critical sets contain only finitely many nonsyzygy signatures (Hofstadler & Verron, 2022).

References

- Albrecht, M., & Perry, J. (2010). *F4/5*. arxiv:1006.4933
- Arri, A., & Perry, J. (2011). The F5 criterion revised. *J. Symb. Comput.*, *46*(9), 1017–1029. doi:10/cd5td7
- Arri, A., & Perry, J. (2017). Corrigendum to “The F5 criterion revised”. *J. Symb. Comput.*, *82*, 164–165. doi:10/gp8639
- Becker, T., & Weispfenning, V. (1993). *Gröbner Bases: A Computational Approach to Commutative Algebra*. Springer-Verlag. doi:10/cfhwn9
- Bender, M. R., Faugère, J.-C., & Tsigaridas, E. (2019). Gröbner basis over semigroup algebras: Algorithms and applications for sparse polynomial systems. *Proc. ISSAC 2019*, 42–49. doi:10/gnt5z9
- Berthomieu, J., Eder, C., & Safey El Din, M. (2021). Msolve: A library for solving polynomial systems. *Proc. ISSAC 2021*, 51–58. doi:10/gk8549
- Bosma, W., Cannon, J., & Playoust, C. (1997). The Magma algebra system I: The user language. *J. Symb. Comput.*, *24*(3-4), 235–265. doi:10/ckdngx
- Buchberger, B. (1965). *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. Innsbruck University.
- Buchberger, B. (2006). An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal (M. P. Abramson, Trans.). *J. Symb. Comput.*, *41*(3), 475–511. doi:10/dz9kz6. (Original work published 1965)
- Cox, D. A., Little, J., & O’Shea, D. (2015). *Ideals, Varieties, and Algorithms* (4th ed.). Springer. doi:10/hzv6
- Decker, W., Greuel, G.-M., Pfister, G., & Schönemann, H. (2022). Singular 4-3-0 — A computer algebra system for polynomial computations.
- Eder, C., & Faugère, J.-C. (2017). A survey on signature-based algorithms for computing Gröbner bases. *J. Symb. Comput.*, *80*(3), 719–784. doi:10/ggck7f
- Eder, C., Lairez, P., Mohr, R., & Safey El Din, M. (2022). A signature-based algorithm for computing the nondegenerate locus of a polynomial system. arxiv:2202.13784
- Eder, C., & Perry, J. (2011). Signature-based algorithms to compute Gröbner bases. *Proc. ISSAC 2011*, 99–106. doi:10/dmwqmp
- Eder, C., & Roune, B. H. (2013). Signature rewriting in Gröbner basis computation. *Proc. ISSAC 2013*, 331–338. doi:10/ggkppx
- Faugère, J.-C. (1999). A new efficient algorithm for computing gröbner bases (F_4). *J. Pure Appl. Algebra*, *139*(1-3), 61–88. doi:10/bpq5dx
- Faugère, J.-C. (2001). Finding all the solutions of Cyclic 9 using Gröbner basis techniques. *Comput. Math.*, 1–12. doi:10/d9297x
- Faugère, J.-C. (2002). A new efficient algorithm for computing gröbner bases without reduction to zero (F_5). *Proc. ISSAC 2002*, 75–83. doi:10/bd4nnq

- Faugère, J.-C., & Joux, A. (2003). Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Gröbner bases. *CRYPTO 2003*, 44–60. doi:10/fpfzgc
- Gao, S., Guan, Y., & Volny, F. (2010). A new incremental algorithm for computing Groebner bases. *Proc. ISSAC 2010*, 13–19. doi:10/cwg6rj
- Gao, S., Volny, F., & Wang, M. (2016). A new framework for computing Gröbner bases. *Math. Comp.*, 85(297), 449–465. doi:10/f7t889
- Gebauer, R., & Möller, H. M. (1986). Buchberger’s algorithm and staggered linear bases. *Symp. Symb. Algebr. Comput.*, 218–221. doi:10/cb24fn
- Gebauer, R., & Möller, H. M. (1988). On an installation of Buchberger’s algorithm. *J. Symb. Comput.*, 6(2), 275–286. doi:10/bfjdwc
- Hashemi, A., & Ars, G. (2010). Extended F5 criteria. *J. Symb. Comput.*, 45(12), 1330–1340. doi:10/bmfh29
- Hashemi, A., & Javanbakht, M. (2021). On the construction of staggered linear bases. *J. Algebra Appl.*, 20(8), 2150132. doi:10/gqwrpn
- Higman, G. (1952). Ordering by divisibility in abstract algebras. *Proc. Lond. Math. Soc.*, 2(1), 326–336. doi:10/fmt8nh
- Hofstadler, C., & Verron, T. (2022). Signature Gröbner bases, bases of syzygies and cofactor reconstruction in the free algebra. *J. Symb. Comput.*, 113, 211–241. doi:10/gp85ss
- Huet, G. (1980). Confluent reductions: Abstract properties and applications to term rewriting systems. *J. ACM*, 27(4), 797–821. doi:10/fj7n4g
- Kreuzer, M., & Robbiano, L. (2000). *Computational commutative algebra 1*. Springer. doi:10/ffxbqr
- Möller, H. M., Mora, T., & Traverso, C. (1992). Gröbner bases computation using syzygies. *Proc. ISSAC 1992*, 320–328. doi:10/cgb2ts
- Monagan, M., & Pearce, R. (2015). A compact parallel implementation of F4. *Proc. PASCO 2015*, 95–100. doi:10/ggpbnk
- Mora, T. (2005). *Solving polynomial equation systems* (Vol. 2). Cambridge University Press. doi:10/jdwm
- Roune, B. H., & Stillman, M. (2012). Practical Gröbner basis computation. *Proc. ISSAC 2012*, 203–210. doi:10/ggkppd
- Stillman, M. (1990). Methods for computing in algebraic geometry and commutative algebra. *Acta Appl. Math.*, 21(1), 77–103. doi:10/dsx7mz
- Sun, Y., & Wang, D. (2011). Solving detachability problem for the polynomial ring by signature-based groebner basis algorithms. arxiv:1108.1301
- Sun, Y., & Wang, D. (2013). A new proof for the correctness of the F5 algorithm. *Sci. China Math.*, 56(4), 745–756. doi:10/gp867m
- Sun, Y., Wang, D., Ma, X., & Zhang, Y. (2012). A signature-based algorithm for computing Gröbner bases in solvable polynomial algebras. *Proc. ISSAC 2012*, 351–358. doi:10/ghkxm
- Traverso, C. (1996). Hilbert functions and the Buchberger algorithm. *J. Symb. Comput.*, 22(4), 355–376. doi:10/b3x2ct
- Winkler, F. (1996). *Polynomial Algorithms in Computer Algebra*. Springer-Verlag. doi:10/bkh6hq