Axioms for a theory of signature bases

Pierre Lairez

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, giving general results applicable in many different settings (e.g. Gröbner for submodules, F4-style reduction, noncommutative rings, non-Noetherian settings, etc.), and extending the reach of signature algorithms.

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 somes 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). Moreover, the predictability of F5 in certain situations enables complexity analyses that are particularly relevant in cryptanalysis (Bardet et al., 2015). 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 in the polynomial case – 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 extra information compared to Gröbner bases, related to the syzygy module of the input generators. Many ideal-theoretic operations – intersection, quotient, saturation, Ext modules – are related to syzygy modules (Stillman, 1990), and signature algorithms seem to give an efficient access to them (Gao et al., 2010; Sun & Wang, 2011; Faugère, 2001; Eder et al., 2023). 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).

This work considers the setting of a module over an algebra over a field, with well-ordered monomials. This covers many interesting case but excludes some recent developments of signature algorithms which demonstrate the wide applicability of the concept: signatures in local rings (Lu et al., 2018), coefficients in Euclidean rings (Eder et al., 2017), principal ideal domains (Francis & Verron, 2020; Hofstadler & Verron, 2023), or Tate algebras (Caruso et al., 2020), and signatures in a tropical setting (Vaccon & Yokoyama, 2017; Vaccon et al., 2018).

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 differential algebras (§6.5). Very importantly, the axioms describe *modules over a ring*, rather than focusing on the special case of ideals. While many ideal theoretic constructions (such as ideal intersection) are best understood in terms of modules, they are not addressed in previous works. 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. First, the concept of *prebasis* is introduced to precisely describe admissible inputs for signature algorihms, or, in other words, to describe what it means for signatures to be consistent. Previous works all starts by fixing the input, crafting specific signatures, and developing the theory with respect to these specific signatures. This is overly restrictive. Moreover, this approach does not highlight the essential properties of signatures, ensuring correctness and termination, among the incidental properties of this specific construction. This is also problematic when trying to define what signatures and signature bases are. Gröbner bases are defined by the equality $\langle \mathrm{lm}\, G\rangle = \mathrm{lm}\, \langle G\rangle$, or the confluence of some rewriting system (e.g. Becker & Weispfenning, 1993, Definition 5.37), we do not need to say a Gröbner basis *of something*. 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 the specific construction that is usually performed for a given input. This raises an interesting question: given a set of signatures – basically anything well-ordered on which act monomials – what are the admissible inputs? Similarly, imagine running some signature basis algorithm from an input $f_1, \dots, f_m$, and stopping it midway. In this intermediate state, we have polynomials $g_1, \dots, g_r$ with signatures deriving from the original input by legal operations on polynomials and signatures. These signatures must be consistent in some sense. Can we characterize this consistency property without refering to the original input? This leads to the concept of prebasis, that is a set of

polynomials with signatures 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 24). A concrete application is the reuse of signatures from one computation to another, which is a way to avoid redundant computations.

Second, I introduce *sigtrees* to uncouple the termination criterion from the algorithms themselves. Sigtrees make a "one size fits all" termination criterion. For Buchberger's algorithm, termination follows from a general principle, Dickson's Lemma, not from *ad hoc* arguments. The concept of sigtrees is a tentative to provide such a general argument. It proves the termination of all known signature algorithms and also settles positively a conjecture in the classical polynomial setting: 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). 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 usual reduction steps. Checking (2) is more difficult and requires arithmetic operations in the base field. This is typically a costly operation. In constrast, 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 (Theorem 38). 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, prebases, and rewrite bases. We also state a combinatorial criterion for a set to be a rewrite basis. Section 4 gathers secondary properties of rewrite bases, such as a precise comparison with signature bases, that are not necessary for the next sections. Section 5 introduces Noetherian hypotheses, termination arguments and review algorithm *templates*. Section 6 illustrates 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. I thank the referees for thoughtful reports.

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 Section 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* (Section 2.2), using the terminology of rewriting systems (Section 2.1). 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 substracting a scalar 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, to perform them or not is only a matter of performance. 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 these 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 $\xrightarrow{1}$ a binary relation on X. "$x \xrightarrow{1} y$" reads "x reduces to y". Following Huet (1980), we define the following binary relations:[1]

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

The relation $\xrightarrow{1}$ is *Noetherian* if there is no infinite sequence $x_0 \xrightarrow{1} x_1 \xrightarrow{1} \cdots$. An element $x \in X$ is $\xrightarrow{1}$*-reduced* if there is no $y \in X$ such that $x \xrightarrow{1} y$. If $x \to y$ and y is $\xrightarrow{1}$-reduced, then y is a *normal form* of x. If $\xrightarrow{1}$ is Noetherian, then every element has at least one normal form. The relation $\xrightarrow{1}$ is *confluent* if $x \uparrow y$ implies $x \downarrow y$ for any $x, y \in X$. If $\xrightarrow{1}$ is confluent, then any $x \in X$ has at most one normal form.

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

- $x \mathrel{\tilde{\uparrow}} y$ if there are $z, z' \in X$ such that $z \backsim z'$, $z \to x$ and $z' \to y$;
- $x \mathrel{\tilde{\downarrow}} y$ if there are $z, z' \in X$ such that $z \backsim z'$, $x \to z$ and $y \to z'$;

The relation $\xrightarrow{1}$ is *confluent modulo* $\backsim$ if $x \mathrel{\tilde{\uparrow}} y$ implies $x \mathrel{\tilde{\downarrow}} y$, for any $x, y \in X$.

2.2 Top reduction

► **Definition 1** (Monomial space, leading monomial, $\equiv_{\mathrm{lt}}$)**.** *A* monomial space *over a field K is a K-linear space M with a basis $B \subset M$ endowed with a well-order relation $\leqslant$. The* leading monomial *of $f \in M$, denoted* $\operatorname{lm} f$*, is the $\leqslant$-maximal element of B with a nonzero coefficient in f,*

[1] Actually Huet denotes either $\to$ or $\xrightarrow{1}$ the one-step reduction, which I denote only $\xrightarrow{1}$, and $\xrightarrow{*}$ the multistep reduction, which I denote $\to$.

or 0 *if* $f = 0$*. The* set of leading monomials *of* M *is defined to be the well-ordered set* $B \cup \{0\}$ *where* 0 *is added as the smallest element.*

An equivalence relation $\equiv_{\mathrm{lt}}$ *is defined on* M *by* $x \equiv_{\mathrm{lt}} y$ *if* $x = y = 0$ *or* $\operatorname{lm}(x - y) < \operatorname{lm} x$*, to be understood as "*x *and* y *have the same leading term".*

The convention that $\operatorname{lm} 0 = 0$ is useful to simplify many statements: being able to write $\operatorname{lm} f$ without checking that $f \neq 0$ avoids a case distinction. From now on, we fix a field K and a monomial space M over K. The set of leading monomials of M is denoted $\mathcal{M}$.

▶ Remark 2 (Equivalent monomial spaces). Different choices of a basis B may lead to equivalent monomial spaces, in the following sense. Another well-ordered basis B' of M gives an equivalent monomial space if there is an increasing bijection $\iota : B \to B'$ such that $\operatorname{lm}' f = \iota(\operatorname{lm} f)$, where $\operatorname{lm}' f$ is the leading monomial of f relatively to B'. The theory is described only using lm, not the basis B, so it does not distinguish between equivalent monomial spaces. From an axiomatic point of view, we can check that the maps lm from M onto a well ordered set that come from a well-ordered basis, are exactly the maps satisfying

L1 $\forall x \in M, \operatorname{lm} x = 0 \Leftrightarrow x = 0$;
L2 $\forall x, y \in M, \operatorname{lm} x = \operatorname{lm} y \neq 0 \Leftrightarrow \exists \lambda \in K^\times, \operatorname{lm}(x - \lambda y) < \operatorname{lm} x$.

For a given $f \in M$, we do not define the terms of f, its monomial support, or the coefficient of a monomial in f, because these notions depend on a specific choice of B, which indicates that they are irrelevant in our setting.

▶ **Example 3.** For polynomial rings, the monomial basis is a very natural choice. In a noncommutative setting however, there may be several natural bases. For example, the Weyl algebra W_1 generated by two elements x and ∂ subject to the relation $\partial x = x\partial + 1$, the two natural bases are $B = \{x^n \partial^m\}$ and $B' = \{\partial^m x^n\}$ (with the same possible orderings as the polynomial case). These two bases give equivalent monomial spaces.

▶ **Definition 4** (Top reduction, $\to_E$)**.** *For any* $E \subseteq M$*, the* top reduction $\xrightarrow{1}_E$ *is defined on* M *by*

$$x \xrightarrow{1}_E y \Leftrightarrow \operatorname{lm} y < \operatorname{lm} x \text{ and } \exists \lambda \in K^\times, \exists e \in E, y = x - \lambda e.$$

In other words, $x \xrightarrow{1}_E y$ if y is the result of cancelling the leading monomial of x using a reducer in E. In this situation, we always have $\lambda e \equiv_{\mathrm{lt}} x$. Since $x \xrightarrow{1}_E y$ implies $\operatorname{lm} x < \operatorname{lm} y$, and the set of leading monomials is well-ordered, it is clear that $\xrightarrow{1}_E$ is Noetherian.

▶ **Definition 5** (Tail equivalence, $\smile_E$, $\overset{\vee}{\uparrow}_E$, $\overset{\vee}{\downarrow}_E$)**.** *For any subset* $E \subseteq M$*, we define a relation* $\overset{1}{\smile}_E$ *on* M*, called* tail equivalence*, defined by*

$$x \overset{1}{\smile}_E y \Leftrightarrow \exists \lambda \in K^\times, \exists e \in E, y = x - \lambda e \text{ and } \operatorname{lm} e < \operatorname{lm} x.$$

The reflexive transitive closure of $\overset{1}{\smile}_E$ *is denoted* $\smile_E$*. The confluence relations* $\overset{\vee}{\uparrow}_E$ *and* $\overset{\vee}{\downarrow}_E$ *are defined using* $\smile_E$*.*

Note that $x \smile_E y$ implies $x \equiv_{\mathrm{lt}} y$. The tail equivalence is not a reduction since it is symmetric, it is not defined 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 Buchberger's well known criterion for polynomial ideals. For $E \subseteq M$, let $\langle E \rangle$ denote the K-linear subspace generated by E.

▶ **Theorem 6** (Buchberger's criterion for monomial spaces)**.** *Let* E *be a subset of* M*. The following assertions are equivalent:*

(Characterization by leading monomials)

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

(Characterization by rewriting)

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

(Characterizations by confluence properties)

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

B4 $\to_E$ *is confluent modulo* $\smile_E$;

(Characterizations by S-pairs)

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

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

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

Proof that B2 implies B3. Let $x, y \in M$ such that $x - y \in \langle E\rangle$. By hypothesis, $x - y \to_E 0$. So there is some $e \in E$ and $\lambda \in K^\times$ such that $x - y \xrightarrow{1}_E x - y - \lambda e \to_E 0$ (unless $x = y$ but this case is trivial). In particular $\lambda e \equiv_{\mathrm{lt}} x - y$.

If $\operatorname{lm} x > \operatorname{lm} y$, then $x \xrightarrow{1}_E x - \lambda e$ and $x - \lambda e - y \in \langle E\rangle$ and by induction on $\max(\operatorname{lm} x, \operatorname{lm} y)$, we may assume that $x - \lambda e \mathrel{\breve{\downarrow}_E} y$ and therefore $x \mathrel{\breve{\downarrow}_E} y$. The case $\operatorname{lm} y > \operatorname{lm} x$ is similar. If $\operatorname{lm} x = \operatorname{lm} y$, there is some $\mu \in K^\times$ such that $\operatorname{lm}(x - \mu y) < \operatorname{lm} x$. There are again two cases. If $\mu = 1$, that is $x \equiv_{\mathrm{lt}} y$, then the sequence of top-reduction $x - y \xrightarrow{1} u_1 \xrightarrow{1}_E u_2 \xrightarrow{1}_E \cdots \xrightarrow{1}_E u_n \xrightarrow{1} 0$ gives a sequence of tail equivalence

$$x = y + (x - y) \smile_E y + u_1 \smile_E \cdots \smile_E y + u_n \smile_E y.$$

In particular, $x \mathrel{\breve{\downarrow}_E} y$. If $\mu \neq 1$, then $\operatorname{lm} x = \operatorname{lm} y = \operatorname{lm} e$, so there are reductions $x \xrightarrow{1}_E x - \kappa e$ and $y \xrightarrow{1}_E y - \nu e$, for some $\kappa, \nu \in K^\times$. By induction on $\max(\operatorname{lm} x, \operatorname{lm} y)$, we may assume that $x - \kappa e \mathrel{\breve{\downarrow}_E} y - \nu e$, which implies $x \mathrel{\breve{\downarrow}_E} y$.

Proof that B3 implies B4. Let $x, y \in M$ such that $x \mathrel{\breve{\uparrow}_E} y$. Both $\xrightarrow{1}_E$ and $\smile_E$ preserve equality modulo $\langle E\rangle$, so $x - y \in \langle E\rangle$, therefore $x \mathrel{\breve{\downarrow}_E} y$, by hypothesis.

Proof that B4 implies B5. If $e \equiv_{\mathrm{lt}} \lambda f$, then $e \xrightarrow{1}_E e - \lambda f$. Moreover $e \xrightarrow{1}_E e - e = 0$. The confluence hypothesis implies that $e - \lambda f \to_E z$ and $0 \to z'$ for some $z, z' \in M$ such that $z \smile_E z'$. But 0 is reduced and only $\smile_E$-equivalent to itself. So $z = 0$ and $e - \lambda f \to_E 0$.

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

Proof that B6 implies B1. Let $x \in \langle E\rangle$ and let $m \in \mathcal{M}$ minimal such that $x \in \langle e \in E \mid \operatorname{lm} e \leqslant m\rangle$. We can write $x = \lambda_1 e_1 + \cdots + \lambda_r e_r$ with $e_i \in E$, $\lambda_i \in K$ and $\operatorname{lm} e_i \leqslant m$. We also assume that the e_i are chosen in such a way that the number N of indices i such that $\operatorname{lm} e_i = m$ is minimal. By minimality of m, we have $N \geqslant 1$.

Assume for contradiction that $\operatorname{lm} x < m$. In particular $N \geqslant 2$ (otherwise, the leading monomials of the e_i cannot cancel to give $\operatorname{lm} x < m$). Up to reordering the indices, we may assume that $m = \operatorname{lm} e_1 = \operatorname{lm} e_2$. Then B6 ensures that $e_1 - \mu e_2 \in \langle e \in E \mid \operatorname{lm} e < m\rangle$ for some $\mu \in K^\times$. We can rewrite x as

$$x = \lambda_1(e_1 - \mu e_2) + (\lambda_2 + \lambda_1\mu)e_2 + \lambda_3 e_3 + \cdots + \lambda_r e_r,$$

in contradiction with the minimality of N. So $\operatorname{lm} x = m$. ◂

▶ **Definition 7** (Pivot basis)**.** *A subset $E \subseteq M$ is a* pivot basis *if it satisfies the equivalent properties of Theorem 6.*

The concept of pivot basis is similar to that of a row echelon form of a matrix. The following minor lemma, on increasing unions of pivot bases, will be used in Sections 3.1 and 3.3.

▶ **Lemma 8.** *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 $e \equiv_{\mathrm{lt}} \lambda f$. 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, $e - \lambda f \to 0$ 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

Recall that a monoid is a set A with an associative composition law $A \times A \to A$ (denoted multiplicatively) which admits an identity element denoted 1_A.

▶ **Definition 9** (Monomial module)**.** *A* monomial module *over a monoid A is a monomial space M with a linear action of A on M (denoted also multiplicatively) such that:*

M1 $\forall a \in A, \forall f, g \in M, \operatorname{lm} f = \operatorname{lm} g \Rightarrow \operatorname{lm}(af) = \operatorname{lm}(ag)$*;*
M2 $\forall a \in A, \forall f, g \in M, \operatorname{lm} f < \operatorname{lm} g \Rightarrow \operatorname{lm}(af) < \operatorname{lm}(ag)$*;*

Note that M2 implies also the following:

M3 $\forall a \in A, \forall f \in M, \operatorname{lm}(af) \geqslant \operatorname{lm} f$.

Indeed, if $\operatorname{lm}(af) < \operatorname{lm} f$, then $\operatorname{lm}(a^{k+1} f) < \operatorname{lm}(a^k f)$ for any $k \geqslant 0$, which would contradicts 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 M2.

▶ **Definition 10** (Action on the set of monomials, divisibility)**.** *The monoid A acts on the set of monomials $\mathcal{M}$ by $a \operatorname{lm} f \doteq \operatorname{lm}(af)$. A* divisor *of $m \in \mathcal{M}$ is an element $n \in \mathcal{M}$ such that $an = m$ for some $a \in A$.*

In the case where M is a module over a polynomial algebra $R = K[x_1, \dots, x_n]$, it is natural to choose $A = \{x_1^{i_1} \cdots x_n^{i_n} \mid i_1, \dots, i_n \geqslant 0\}$, although $A = R \setminus \{0\}$ is also a possible choice (with no major theoretical difference). When M is a module over a noncommutative ring, the monoid $A = R \setminus \{0\}$ is a natural choice. For example, in the Weyl algebra W_1 (see Example 3), the set of monomials $\{x^n \partial^m\}$ is not closed under multiplication (because $\partial x = x\partial + 1$). We can also choose A to be the monoid generated by x and ∂. Section 6 presents more examples.

2.4 Gröbner bases

▶ **Definition 11** (Gröbner basis)**.** *Let M be a monomial space over a monoid A. 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 replace the concept of Gröbner bases, but rather to give a way to compute them.

▶ Remark 12 (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. A counterexample in a free algebra in two variables is given by Green et al. (1998) (see also Section 6.6).

Unfolding the definitions, we see that for every singleton $\{f\}$ to be a Gröbner basis is necessary and sufficient that:

M4 $\forall f \in M, \forall g \in \langle af \mid a \in A\rangle \setminus \{0\}, \exists a \in A, \operatorname{lm} g = \operatorname{lm}(af)$.

This holds in most usual settings, and all settings presented in Section 6. For example, if $R = M$ is a polynomial ring, and $A \subset R$ the monoid of monomials, then for any $g \in \langle af \mid a \in A\rangle$, there is some $h \in R$ such that $g = hf$ and we have $\operatorname{lm} g = \operatorname{lm}(\operatorname{lm}(h)f)$.

3 Signatures

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 $\mathcal{S}$. We are interested in computing Gröbner bases in M while S is the module of signatures.

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

S1 $\forall a, b \in A, \forall \sigma \in \mathcal{S}, \forall m \in \mathcal{M}, a\sigma = b\sigma$ and $\sigma \neq 0 \Rightarrow am = bm$.
S2 $\forall a, b \in A, \forall \sigma \in \mathcal{S}, \forall m \in \mathcal{M}, a\sigma \leqslant b\sigma$ and $\sigma \neq 0 \Rightarrow am \leqslant bm$.

Naturally S2 implies S1, but we state them separately because S2 will only be useful later in Section 5 (and specifically in Lemma 50) when we will study algorithms for computing signature bases and termination issues. This hypothesis is called *compatibility* by Gao et al. (2016) and others.

In concrete situations, we will have a natural module map $\phi : S \to M$ (that is a K-linear map commuting with the action of A), but it is not a requirement for the theory. As Arri and Perry (2011), or Kambe (2023) more recently, we can define in this context a notion of signature for the elements of $\phi(S)$ by

$$\widetilde{\operatorname{sig}}(f) \doteq \min\{\operatorname{lm} p \in \mathcal{S} \mid p \in S \text{ and } \phi(p) = f\}.$$

This lead however to conceptual difficulties, because the signatures that appear in computations may not coincide with this definition, creating a gap between the theory and tha algorithms. In the axiomatic approach, we do not define what *is* the signature of elements in $\phi(S)$. Instead, we adjunct elements of M with signatures, and describe the required consistency properties.

3.1 Signature bases

▶ **Definition 13** (Sigpair, sigset, polynomial part, signature)**.** *A* sigpair *is an element of $M \times \mathcal{S}$. A* sigset *is a set of sigpairs. The first element of a sigpair f, denoted $f^\natural$, is the* polynomial part *of f (eventhough f may not be a polynomial, strictly speaking). The second element of a sigpair f, denoted* $\operatorname{sig} f$*, is the* signature *of f.*

▶ **Definition 14** ($AG^{<\sigma}$, $AG^{\leqslant\sigma}$, regular reduction)**.** *For a sigpair f and some $a \in A$, we define the sigpair $af = (af^\natural, a \operatorname{sig} f)$. For notational convenience, we also define a scalar multiplication $\lambda f = (\lambda f^\natural, \operatorname{sig} f)$, for $\lambda \in K^\times$. For any sigset G, let AG denote the sigset $AG = \{af \mid a \in A, f \in G\}$. For $\sigma \in \mathcal{S}$, let*

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

They are subsets of M, not sigsets. Each set $AG^{<\sigma}$ defines a reduction rule $\xrightarrow{1}_{AG^{<\sigma}}$ (Definition 4), that we denote $\xrightarrow{1}{}^{\sigma}_{G}$, the regular reduction *in signature σ. On $M \times \mathcal{S}$, we define $f \xrightarrow{1}_G g$ if $\operatorname{sig} f = \operatorname{sig} g$ and $f^\natural \xrightarrow{1}{}^{\operatorname{sig} f}_{G} g^\natural$. This is the* regular reduction *of sigpairs. The tail equivalence relations $\backsim^{\sigma}_{G}$ and $\backsim_G$ are defined similarly using $\backsim_{AG^{<\sigma}}$.*

The reduction relations $\xrightarrow{1}{}^{\sigma}_{G}$, for any $\sigma \in \mathcal{S}$, are Noetherian, as any top-reduction relation in a monomial space. So $\xrightarrow{1}_G$ is also Noetherian, and every sigpair has at least one normal form modulo regular reduction. The regular reduction $\to_G$ is the same as the *regular top $\mathfrak{s}$-reduction* defined by Eder and Faugère (2017). In contrast, we will not make use of *singular $\mathfrak{s}$-reductions* and *tail $\mathfrak{s}$-reductions*.

▶ **Example 15** (Univariate polynomials)**.** Let $M = S = K[x]$, $\mathcal{M} = \mathcal{S} = \{x^n \mid n \geqslant 0\} \cup \{0\}$, with the usual ordering. Let $A = \{x^n \mid n \geqslant 0\}$. Let G be the sigset $\{(x-1, x)\}$, made of a single sigpair with polynomial part $x - 1$ and signature x. For any $m \geqslant 0$, we check that

$$AG^{<x^m} = \left\{x^k(x-1) \,\middle|\, 0 \leqslant k < m-1\right\} \text{ and } AG^{\leqslant x^m} = \left\{x^k(x-1) \,\middle|\, 0 \leqslant k \leqslant m-1\right\}.$$

Both are pivot bases.

Consider the sigpairs $f_1 = (x^2, x)$ and $f_2 = (x^2, x^3)$. They have the same polynomial part x^2 but not the same signature. The sigpair f_1 is $\to_G$ reduced. Indeed, the only possible reduction to investigate is that of f_1 by xg (where g is the unique element of G), but $\operatorname{sig}(xg) = x^2$, which exceeds $\operatorname{sig}(f_1) = x$, forbidding the reduction. In contrast, we have reductions

$$f_2 \xrightarrow{1}_G (x, x^3) \xrightarrow{1}_G (1, x^3),$$

using xg and g successively. This exemplifies the additional constraints that signatures impose on reductions, compared to the usual setting without signatures.

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

▶ **Lemma 16.** *Let G be a sigset, let $\sigma \in \mathcal{S}$ and $a \in A$. Then*

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

▶ **Definition 17** (Signature basis)**.** *A* signature basis *is a sigset G such that for any $\sigma \in \mathcal{S}$ the set $AG^{\leqslant\sigma}$ is a pivot basis.*

Using Theorem 6, and a bit of signature manipulation to reduce from $AG^{\leqslant\sigma}$ to $AG^{<\sigma}$, we can prove a sigset G is a signature basis if and only if regular reduction $\to_G$ is confluent modulo tail equivalence $\backsim_G$. Signature bases are a refinement of Gröbner bases, in the sense that forgetting the signatures in a signature basis gives a Gröbner basis.

▶ **Lemma 18.** *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 $AG^{\leqslant\sigma}$, with $\sigma \in \mathcal{S}$. By construction, $AG^{\leqslant\sigma} \subseteq AG^{\leqslant\tau}$ if $\sigma \leqslant \tau$. So Lemma 8 applies and shows that $AG^\natural$ is a pivot basis. ◀

3.2 Prebases

► **Definition 19** (Prebasis)**.** *A sigset G 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 $\langle AG^{\leqslant\sigma}\rangle / \langle AG^{<\sigma}\rangle$ 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 20.** A trivial choice for the set of signatures is $\mathcal{S} = \mathcal{M}$. Let G be a sigset such that $\operatorname{sig} f = \operatorname{lm} f^\natural$ for any $f \in G$. Then G is a prebasis if and only if $G^\natural$ is a Gröbner basis. Indeed, in this case, $\langle AG^{<\sigma}\rangle = \langle ag^\natural \mid a \in A, g \in G, \operatorname{lm}(ag) < \sigma\rangle$. So the condition for being a prebasis is exactly Criterion B6 for $AG^\natural$ to be a pivot basis, that is for $G^\natural$ to be a Gröbner basis.

► **Example 21.** If $AG^0 = \varnothing$ and if AG^σ contains exactly zero or one element for any $\sigma \in \mathcal{S}$, then G is a prebasis. This follows directly from the definition.

In the course of computing a signature basis, or a rewrite basis, as we will see latter, we will add new elements to a prebasis G given as input. Naturally, the construction of new elements must respect both the polynomial part and the signature. In particular, we want to preserve the prebasis property. Typically, we construct new elements by regular reduction: for any $a \in A$ and $g \in G$, we allow the insertion of any sigpair h such that $ag \to_G h$. In view of using F4-style reductions (Section 5.7), we give a wider definition of allowed extensions of a sigset, that we call *sigsafe extensions*.

► **Definition 22** (Sigsafe extension)**.** *A sigset H is a* sigsafe extension *of a sigset G if $G \subseteq H$ and for any $h \in H$, there is some $f \in AG^{\operatorname{sig} h}$ and some $\lambda \in K^\times$ such that $h^\natural \equiv \lambda f^\natural \pmod{\langle AG^{<\operatorname{sig} h}\rangle}$.*

The problem of computing signature bases is more formally stated as "given a prebasis G, compute a signature basis that is a sigsafe extension of G". For computing a Gröbner basis of the submodule of M generated by a set $G \subseteq M$, we will follow the steps: first, choose a signature module S, second, we endow the elements of G with signatures to turn it into a prebasis; third, we compute a signature basis that is a sigsafe extension of G; four, we remove signatures to obtain a Gröbner basis (Lemma 18).

► **Lemma 23.** *Let G be a prebasis and let H be a sigsafe extension of G. Then:*

- *$\forall \sigma \in \mathcal{S}, \langle AG^{<\sigma}\rangle = \langle AH^{<\sigma}\rangle$ and $\langle AG^{\leqslant\sigma}\rangle = \langle AH^{\leqslant\sigma}\rangle$;*
- *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 16.

Generalizing Example 20, we may construct prebases in M from a Gröbner basis in S.

► **Theorem 24.** *Let $\phi : S \to M$ be a K-linear map commuting with the action of A. If $H \subseteq S$ is a Gröbner basis, then $\{(\phi(h), \operatorname{lm} h) \mid h \in H\} \subseteq M \times \mathcal{S}$ is a prebasis.*

Proof. Let $G = \{(\phi(h), \operatorname{lm} h) \mid h \in H\}$. 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 $\operatorname{lm}(ah) = 0$. This implies $ah = 0$, so $f = 0$.

As for P2, let $\sigma \in \mathcal{S}$, $f, g \in G$ and $a, b \in A$ such that $\sigma = a \operatorname{sig} f = b \operatorname{sig} g$. By definition, there are some $h, k \in H$ such that $\operatorname{lm} h = \operatorname{sig} f$, $\operatorname{lm} k = \operatorname{sig} g$, $f = \phi(h)$ and $g = \phi(k)$. In

particular $\sigma = \operatorname{lm}(ah) = \operatorname{lm}(bk)$, and there is some $\lambda \in K$ such that $ah \equiv_{\operatorname{lt}} \lambda bk$. H is a Gröbner basis, so AH is a pivot basis, so Criterion B6 applies and we have $ah - \lambda bk = \sum_i m\mu_i c_i l_i$ for some $\mu_i \in K$, $c_i \in A$ and $l_i \in H$ such that $\operatorname{lm}(c_i h_i) < \sigma$. In particular, $af - \lambda bg = \sum_i \mu_i c_i \phi(l_i)$ and $c_i\phi(l_i) \in AG^{<\sigma}$. ◂

► Remark 25 (Constructing prebases "for free"). As a special case of Theorem 24, we recover the following classical construction which underlies all previous work on signature algorithms. Given $g_1, \dots, g_r \in M$, we want to find a signature module $\mathcal{S}$ and signatures $\sigma_1, \dots, \sigma_r$ such that $\{(g_i, \sigma_i)\}_{1 \leqslant i \leqslant r}$ is a prebasis. This is the first step of the general strategy for computing Gröbner bases using signatures. The following construction applies when each of the singletons $\{g_i\}$ is a Gröbner basis (this is the common case, see Remark 12).

We choose the signature module $S = M^r \simeq M \otimes K^r$. If B_M is the distinguished basis of the monomial space M, we define $B_S = \{m \otimes e_i \mid m \in B_M, 1 \leqslant i \leqslant r\}$ as the distinguished basis of S, where $\{e_1, \dots, e_r\}$ denotes the canonical basis of K^r. Let $\mathcal{S} = B_S \cup \{0\}$ denote the set of leading monomials of S. There are two natural well-orders on $\mathcal{S}$, called *position-over-term* (POT) and *term-over-position* (TOP): 0 is always the minimal elements, and for the nonzero monomials,

POT $m \otimes e_i \leqslant_{\mathcal{S}} n \otimes e_j$ if $i < j$ or $i = j$ and $m \leqslant_{\mathcal{M}} n$;
TOP $m \otimes e_i \leqslant_{\mathcal{S}} n \otimes e_j$ if $m <_{\mathcal{M}} n$ or $m = n$ and $i \leqslant j$.

The monoid A acts on S by $a \cdot (f_1, \dots, f_r) = (af_1, \dots, af_r)$, turning S into a monomial module over A. Moreover, S1 and S2 are satisfied, so S is a suitable signature module, with either the POT or the TOP ordering. Let $\phi : S \to M$ defined by $\phi(f_1, \dots, f_r) = f_1 + \cdots + f_r$, which commutes with the action of A.

Since $\{g_i\}$ is a Gröbner basis in M, for any $1 \leqslant i \leqslant r$, it follows easily that $\{g_i \otimes e_i\}$ is a Gröbner basis in S. Moreover, the leading monomials of some $ag_i \otimes e_i$ and some other $bg_j \otimes e_j$ can never be equal, unless $i = j$. So it follows that the set $H = \{g_i \otimes e_i \mid 1 \leqslant i \leqslant r\}$ is a Gröbner basis in S. By Theorem 24, this implies that

$$G = \{(g_i, \operatorname{lm} g_i \otimes e_i) \mid 1 \leqslant i \leqslant r\}$$

is a prebasis. And, by construction, $G^\natural = \{g_1, \dots, g_r\}$.

This construction shows that, at least under the extra assumption M4 on M, we have a systematic way to construct prebases from arbitrary finite sets of M, enabling the general strategy of using signatures to compute Gröbner bases.

► **Example 26.** Consider the case where $M = \mathbb{Q}[x, y]$, where the monomial basis of M is given the degree reverse lexicographic order, with $x < y$, $A = \left\{x^i y^j \;\middle|\; i, j \geqslant 0\right\}$, and consider

$$g_1 = \underline{x^2y^2} - 1,\ g_2 = \underline{y^5} - x^2y, \text{ and } g_3 = \underline{x^5} - xy^2,$$

with leading monomial underlined. This is an example of Mora (1994). Following the recipe in Remark 25, we consider the signature module $S = M^3$, with the TOP ordering, and we endow the g_i with signatures $\operatorname{sig} g_i = \operatorname{lm} g_i \otimes e_i$, so we construct the following sigset:

$$G = \left\{\left(\underline{x^2y^2} - 1, x^2y^2 \otimes e_1\right), \left(\underline{y^5} - x^2y, y^5 \otimes e_2\right), \left(\underline{x^5} - xy^2, x^5 \otimes e_3\right)\right\}.$$

In this case, the fact that G is a prebasis follows from the trivial reason exposed in Example 21.

It is also common to choose the unshifted signature $\operatorname{sig} g_i = 1 \otimes e_i$. It is equally valid to choose $\operatorname{sig} g_i = m_i \otimes e_i$, for any non zero $m_i \in \mathcal{M}$, from the theoretical point of view. The choice $\operatorname{sig} g_i = \operatorname{lm} g_i \otimes e_i$ comes naturally because in the general setting of a monomial module over A, there is no "1", so we cannot write, in general, $\operatorname{sig} g_i = 1 \otimes e_i$, while we can always

write $\operatorname{sig} g_i = \operatorname{lm} g_i \otimes e_i$. Eder and Faugère (2017) only work in the polynomial case and fix $\operatorname{sig} g_i = 1 \otimes e_i$. However, we can change the term ordering on S to what they call *lt-pot*, or Schreyer's order (Eder & Faugère, 2017, Definition 2.5), and recover the behavior of the choice $\operatorname{sig} g_i = \operatorname{lm} g_i \otimes e_i$ with the TOP order on S. Eder and Faugère (2017, §14) suggest that this natural choice $\operatorname{sig} g_i = \operatorname{lm} g_i \otimes e_i$, is better for performance than the unshifted signatures. This is exemplified in Figures 2 and 4.

► **Example 27** (Sum of submodules)**.** Let G and H be two finite Gröbner bases in M. Consider the problem of computing a Gröbner basis J such that $\langle AG\rangle + \langle AH\rangle = \langle AJ\rangle$. We could use, as in Remark 25, the signature module $M^{\#G+\#H}$ to turn $G \cup H$ into a prebasis. However, this will lead to many useless computations (reductions to zero) because we did not take into account the fact that G and H are already Gröbner bases, so all the S-pairs between two elements of G (resp. H) already reduce to 0.

Instead, we consider the monomial signature module $S = M^2$ with the map $\phi : (u, v) \in S \to u + v \in M$. The set $B = \{(g, 0) \mid g \in G\} \cup \{(0, h) \mid h \in H\}$ is a Gröbner basis in S because the elements $(g, 0)$ cannot interact with the elements $(0, h)$. By Theorem 24, the sigset

$$\{(\phi(b), \operatorname{lm} b) \mid b \in B\} = \{(g, \operatorname{lm} g \otimes e_1) \mid g \in G\} \cup \{(h, \operatorname{lm} h \otimes e_2) \mid h \in H\}$$

is a prebasis. We can use it as a starting point to compute a Gröbner basis of the sum $\langle AG\rangle + \langle AH\rangle$. This saves some computations because the signatures encode that elements of G (resp. H) do not need to be reduced with each other.

3.3 Rewrite bases

We now introduce rewrite bases. The definition is purely combinatorial, depending only on leading monomials and signatures, in addition to the prebasis condition. We will see that a rewrite basis is a signature basis (Corollary 31). As for pivot bases, prebases, and Gröbner bases, 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.) Somehow, we can split these conditions into, on the one hand, the prebasis property, and on the other hand, the combinatorial properties of rewrite bases. The concept was first introduced by Eder and Roune (2013). It is simplified here by removing the need for what they call a "rewrite order". So my definition of rewrite basis is actually different from theirs, but relates more simply to signature bases (Theorem 40).

► **Definition 28** (Rewrite basis)**.** *For $\sigma \in \mathcal{S}$, a sigset G is a* rewrite basis at σ *if either $AG^\sigma = \emptyset$, or there is some $\to_G$-reduced element $f \in AG$ with $\operatorname{sig} f = \sigma$. A sigset G is a* rewrite basis *if it is a prebasis and a rewrite basis at σ for any $\sigma \in \mathcal{S}$.*

► **Example 29** (continued)**.** The sigset in Example 26 is a prebasis but not a rewrite basis. The smallest signature at which it is not a rewrite basis is $\sigma = x^2y^5 \otimes e_2$. Indeed,

$$AG^\sigma = \{x^2 g_2\} = \left\{\left(\underline{x^2y^5} - x^4 y, x^2y^5 \otimes e_2\right)\right\},$$

and there is a top reduction of $x^2 g_2$ by $y^3 g_1$. So AG^σ does not contain any $\to_G$-reduced element. Note that $x^2 g_2$ does not reduce $y^3 g_1$ because $\operatorname{sig}(x^2 g_2) > \operatorname{sig}(y^5 g_1)$, so G is a rewrite basis at $x^2y^5 \otimes e_1$. In constrast to the classical setting, the symmetry of critical pairs is broken by the signatures.

We also check that G is a rewrite basis at any signature $m \otimes e_1$, for $m \in \mathcal{M}$. These signatures are not empty when m is a multiple of $\operatorname{lm} g_1 = x^2y^2$, say $m = a \operatorname{lm} g_1$. There may be a possible reduction when $m = b \operatorname{lm} g_i$ (with $i = 2$ or 3), but in this case, we have $a \operatorname{sig} g_1 = a(\operatorname{lm} g_1 \otimes e_1) = m \otimes e_1$ and, similarly, $b \operatorname{sig} g_i = m \otimes e_i$. The definition of the TOP order, ensures that $a \operatorname{sig} g_1 < b \operatorname{sig} g_i$, so the reduction is not possible.

The defining property of rewrite bases implies that of signature bases. This is the first aspect of the definition. (See Section 4.1 for more details on the relation between rewrite bases and signature bases.)

▶ **Proposition 30.** *Let G be a prebasis and let $\sigma \in \mathcal{S}$ such that G is a rewrite basis at any signature $\tau \leqslant \sigma$. Then $AG^{\leqslant\sigma}$ is a pivot basis.*

Proof. For contradiction, assume that $AG^{\leqslant\sigma}$ is not a pivot basis, and let τ be the smallest signature such that $AG^{\leqslant\tau}$ is not a pivot basis. In particular, $AG^{<\tau}$ is a pivot basis: indeed, $AG^{<\tau}$ is the increasing union $\cup_{\rho<\tau} AG^{\leqslant\rho}$. so Lemma 8 applies.

The set AG^{τ} is nonempty, as otherwise $AG^{\leqslant\tau} = AG^{<\tau}$ and the latter is a pivot basis. Since G is a rewrite basis at τ, there is a $g \in AG^{\tau}$ which is $\to^{\tau}_{G}$-reduced. Since $AG^{\leqslant\tau}$ is not a pivot basis there is a $f \in \langle AG^{\leqslant\tau}\rangle$ such that $\operatorname{lm} f \neq \operatorname{lm} h$ for any $h \in AG^{\leqslant\tau}$. By the prebasis condition P2, there is a $\lambda \in K$ such that $f - \lambda g \in \langle AG^{<\tau}\rangle$. And because $AG^{<\tau}$ is a pivot basis, by hypothesis, Criterion B3 implies that $f \mathrel{\check{\downarrow}^{\tau}_{G}} \lambda g$. Since both f and g are $\to^{\tau}_{G}$-reduced, we have in fact a tail equivalence $f \mathrel{\smile^{\tau}_{G}} g$, and therefore $\operatorname{lm} f = \operatorname{lm} g$, which is a contradiction ◀

▶ **Corollary 31.** *If G is a rewrite basis, then G is a signature basis.*

Proof. It follows directly from the definitions and Proposition 30. ◀

▶ **Corollary 32.** *Let G be a prebasis and let $\sigma \in \mathcal{S}$ such that G is a rewrite basis at any signature $\tau < \sigma$. Then $AG^{<\sigma}$ is a pivot basis.*

Proof. For contradiction, assume that $AG^{<\sigma}$ is not a pivot basis. Since $AG^{<\sigma}$ is the increasing union $\cup_{\tau<\sigma} AG^{\leqslant\tau}$, there is at least one $\tau < \sigma$ such that $AG^{\leqslant\tau}$ is not a pivot basis, by Lemma 8. This contradicts Proposition 30. ◀

The following statement is an effective form of the prebasis condition, it states that when G is a rewrite basis, the regular reduction $\to_G$ is able to witness the prebasis condition: two elements with same signatures have equal $\to_G$-normal forms, up to scaling and tail equivalence.

▶ **Corollary 33.** *Let G be a prebasis and let $\sigma \in \mathcal{S}$ such that G is a rewrite basis at any $\tau < \sigma$. For any $f, g \in AG^{\sigma} + \langle AG^{<\sigma}\rangle$, there is some $\lambda \in K^{\times}$ such that $f \mathrel{\check{\downarrow}^{\sigma}_{G}} \lambda g$.*

Proof. Let $f, g \in AG^{\sigma} + \langle AG^{<\sigma}\rangle$. P2 implies that there is some $\lambda \in K^{\times}$ such that $f - \lambda g \in \langle AG^{<\sigma}\rangle$. By Corollary 32, $AG^{<\sigma}$ is a pivot basis, so Criterion B3 implies that $f \mathrel{\check{\downarrow}^{\sigma}_{G}} \lambda g$. ◀

The second aspect of the definition of rewrite bases is the algorithmic content. 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 $\operatorname{sig} f = \sigma$, compute a $\to_G$-normal form, and insert the result into G. This suggests an algorithm schema for computing rewrite bases (Pseudo-algorithm 1).

There are two significant difficulties to turn this schema into an actual algorithm. First, how to check that G is a rewrite basis? And how to pick a signature at which G is not a rewrite

Pseudo-algorithm 1 Algorithm schema for computing rewrite bases

input A 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 σ
3 pick $f \in AG$ with $\operatorname{sig} f = \sigma$ —f is called the *reductant*
4 $g \leftarrow$ any $\to_G$-normal form of f
5 $G \leftarrow G \cup \{g\}$ —G is now a rewrite basis at σ
6 **return** G

basis? These questions are addressed in Section 3.4. Second, 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.

3.4 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 criteria 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): the set of critical signatures that we introduce below is closely related to the set of signatures of J-pairs that need to be handled in the GVW algorithm.

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).

▶ **Definition 34** (Critical set)**.** *For a sigset G and a sigpair f, the* critical set *of f modulo G, denoted $\Sigma(f, G)$, is the set of all $\sigma \in \mathcal{S}$ such that:*

C1 $\exists a \in A, a \operatorname{sig} f = \sigma$ *and af is not $\to_G$-reduced;*
C2 $\forall b \in A, \big(b \operatorname{sig} f$ *is a proper divisor of* $\sigma \Rightarrow bf$ *is $\to_G$-reduced*$\big)$.

The critical set *of G, is the set of signatures*

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

In other words, the condition C1 defines a subset of $\mathcal{S}$ corresponding to the signatures at which a multiples af is not $\to_G$-reduced. This subset is closed under the action of A. Indeed, if there is a reduction $af \xrightarrow{1}_G h$, then for any $b \in A$, there is also a reduction $bfa \xrightarrow{1}_G bh$. Among all these signatures, the condition C2 retains only the minimal ones for divisibility. This will be important latter to ensure finiteness. The important property is the following.

▶ **Lemma 35.** *For any sigset G, any sigpair f, and any $a \in A$, if af is not $\to_G$-reduced, then there is some $\sigma \in \Sigma(f, G)$ which divides $a \operatorname{sig} f$.*

Proof. Let $a' \in A$ such that $a' \operatorname{sig} f$ divides $a \operatorname{sig} f$, $a'f$ is not $\to_G$-reduced, and $a' \operatorname{sig} f$ is minimal. Let $\sigma = a' \operatorname{sig} f$. We check that $\sigma \in \Sigma(f, G)$. Indeed C1 follows from the definition. For C2, let $b \in A$ such that $b \operatorname{sig} f$ is a proper divisor of σ. In particular, $b \operatorname{sig} f$ divides $a \operatorname{sig} f$ and $b \operatorname{sig} f < \sigma$. By minimality of σ, bf is $\to_G$-reduced. ◀

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, $\Sigma(f, \{g\})$ contains at most one element. In the general case, $\Sigma(f, \{g\})$ can contain zero, one, finitely many or infinitely many elements, see Section 6 for examples.

▶ **Example 36** (continued)**.** Consider the sigset G defined in Example 26 and developed in Example 29. We compute that

$$\Sigma(g_1, G) = \emptyset,\ \Sigma(g_2, G) = \{x^2y^5 \otimes e_2\},\ \Sigma(g_3, G) = \{x^5y^2 \otimes e_3\}.$$

Note that $\Sigma(g_3, \{g_1\}) = \{x^5y^5 \otimes e_3\}$, reflecting the reduction of $y^5 g_3$ by $x^5 g_1$, but this signature disappears in $\Sigma(g_3, G)$ because it is divided by $x^5y^2 \otimes e_3$, which comes from the reduction of $y^2 g_3$ by $x^3 g_1$.

▶ **Proposition 37.** *Let G be a prebasis and let $\sigma \in \mathcal{S}$. If G is a rewrite basis 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 $a \operatorname{sig} f = \sigma$. We choose f so that $\operatorname{lm}(af)$ is smallest. By hypothesis, af is not $\to_G$-reduced (otherwise G is a rewrite basis at σ). By Lemma 35, there is a signature $\tau \in \Sigma(f, G)$ which divides σ. Let $b, c \in A$ such that $b \operatorname{sig} f = \tau$ and $c\tau = \sigma$.

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 $\to_G^\tau$-reduced $g \in AG^\tau$. By Corollary 33, there is $\lambda \in K^\times$ such that $g \downarrow_G^\tau \lambda bf$. Since g is $\to_G^\tau$-reduced and bf is not, this implies that $\operatorname{lm} g < \operatorname{lm}(bf)$, and, by M2, that $\operatorname{lm}(cg) < \operatorname{lm}(cbf)$. Moreover, $cb \operatorname{sig} f = a \operatorname{sig} f$, so S1 implies that $\operatorname{lm}(cbf) = \operatorname{lm}(af)$, and therefore $\operatorname{lm}(cg) < \operatorname{lm}(af)$. This contradicts the minimality of $\operatorname{lm}(af)$. ◀

From Proposition 37, we easily deduce the following statement.

▶ **Theorem 38** (Faugère's criterion)**.** *Let G be a prebasis. If G is a rewrite basis at any $\sigma \in \Sigma(G)$, then G is a rewrite basis.*

4 Additional properties of rewrite bases

This section gathers some properties of rewrite bases that are not central, and not used in the next sections, but that connect to previous works.

4.1 Relation between signature bases and rewrite bases

Corollary 31 shows that rewrite bases are signature bases. With two competing definitions, it is worth studying more precisely the relation between them.

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 $\langle AG^{\leqslant\sigma}\rangle / \langle AG^{<\sigma}\rangle$. 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 $AG^\sigma \neq \emptyset$.

Moreover, a nonempty signature is:

- a *regular signature* if $AG^\sigma \cap \langle AG^{<\sigma}\rangle = \emptyset$;
- a *syzygy signature* if $AG^\sigma \subseteq \langle AG^{<\sigma}\rangle$.

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.

▶ **Example 39** (continued)**.** Consider again the prebasis G from Example 26. We check easily that:

- $1 \otimes e_1$ is an empty signature, because it is not multiple of any of the signatures in G.
- $\sigma = x^5y^5 \otimes e_3$ is a (nonempty) syzygy signature. Indeed, $AG^\sigma = \{y^5 g_3^\natural\}$ and $y^5 g_3 \to_G 0$, using the reducer $x^5 g_2$. So $y^5 g_3^\natural \in \langle AG^{<\sigma}\rangle$, and therefore $AG^\sigma \subseteq \langle AG^{<\sigma}\rangle$.
- $\sigma = x^2y^2 \otimes e_1$ is a (nonempty) regular signature. Indeed $AG^{<\sigma} = \emptyset$, so $\langle AG^{<\sigma}\rangle = 0$ while AG^σ contains the nonzero element $g_1^\natural$.

▶ **Proposition 40.** *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 $\sigma \in \mathcal{S}$ 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 $\operatorname{sig} f = \sigma$. Let v be a $\to_G$-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 $AG^{\leqslant\sigma}$ is a pivot basis, by definition of a signature basis, $v^\natural$ is reducible with respect to $AG^{\leqslant\sigma}$. So there is some $g \in AG$ such that $\operatorname{sig} g \leqslant \sigma$ and $\operatorname{lm} g = \operatorname{lm} v$. But v is $\to_G$-reduced so $\operatorname{sig} g = \sigma$. Moreover $\operatorname{lm} v = \operatorname{lm} g$, so g is also $\to_G$-reduced. So G is a rewrite basis at σ.

The converse follows from the same argument used in the proof of Corollary 31. ◀

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. The data of syzygy signatures is a by-product of all known signature-based algorithms. So actually, they compute rewrite bases, not only signature bases. The following statement establishes an equivalence which does not hold for the original definition of rewrite bases by Eder and Roune (2013, §3.2), only

the "rewrite basis ⇒ signature basis" implication holds for this definition.[2] This is the main motivation for the simplified definition.

▶ **Corollary 41.** *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 some $g \in G$ such that* $\operatorname{sig} g$ *divides σ and $g^\natural = 0$.*

4.2 Minimal elements in rewrite bases

We first introduce a binary relation on the set of sigpairs.

▶ **Definition 42** (Domination relation, $\sqsubseteq$)**.** *We say that g* dominates *f, and denote it $g \sqsubseteq f$, if one of the following holds:*

D1 $\exists a \in A, a \operatorname{sig} g = \operatorname{sig} f$ *and* $a \operatorname{lm} g \leqslant \operatorname{lm} f$*;*

D2 $\exists a \in A, a \operatorname{sig} g < \operatorname{sig} f$ *and* $a \operatorname{lm} g = \operatorname{lm} f \neq 0$.

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. First, when $a \operatorname{lm} g = \operatorname{lm} f$, then f will not help because ag can serve just as well in any situation where f would serve. When $a \operatorname{lm} g < \operatorname{lm} f$, Corollary 33 proves that f will never be reduced in a rewrite basis containing g.

▶ **Theorem 43.** *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 implication 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 $b \operatorname{sig} f = \sigma$ and bf is $\to_H$-reduced (and thus $\to_G$-reduced too). By hypothesis, there is some $g \in G$ such that $g \sqsubseteq f$. Since bf is $\to_G$-reduced, D2 cannot hold, so D1 does: there is some $a \in A$ such that $a \operatorname{sig} g = \operatorname{sig} f$ and $a \operatorname{lm} g \leqslant \operatorname{lm} f$.

Since H is a sigsafe extension of G, $f \in AG^\sigma + \langle AG^{<\sigma} \rangle$ (maybe after a scalar multiplication), by definition. By Corollary 33, there is some $\lambda \in K^\times$ such that $bf \downarrow_G \lambda bag$. Since bf is $\to_G$-reduced, this implies $\operatorname{lm}(bf) \leqslant \operatorname{lm}(bag)$. Combining with the previous inequality, we obtain that $\operatorname{lm}(bag) = \operatorname{lm}(bf)$. So bag, which has same leading monomial and signature as bf, is $\to_G$-reduced and thus G is a rewrite basis at σ. ◀

Combining with Theorem 38, we obtain the following corollary which may be used to reduce the number of signatures to consider when computing a rewrite basis. It allows, during the computation of a rewrite basis, to consider only the critical signatures relative to the dominant elements, while retaining the nondominant elements for computing the reductions.

[2] With the simplified definition, there must be at least one $\to_G$-reduced element per signature. With the original signature, one specific element must be $\to_G$-reduced.

► **Corollary 44.** *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 $\sigma \in \Sigma(G)$, then G and H are rewrite bases.*

4.3 Syzygies

When a rewrite basis comes from a Gröbner basis in the signature module through a map $\phi : S \to M$ (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., 2023).

► **Proposition 45.** *Let $\phi : S \to M$ be a linear map commuting with the action of A, let $H \subseteq S$ be a Gröbner basis, let $G = \{(\phi(h), \operatorname{lm} h) \mid h \in H\}$, and $J = \{h \in H \mid \phi(h) = 0\}$. If G is a rewrite basis, then J is a Gröbner basis and $\langle AJ\rangle = \ker\phi \cap \langle AH\rangle$.*

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

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 $\to_G$-reduced and $a \operatorname{sig} g = \sigma$. By Corollary 33, we have $\phi(h) \overset{\sigma}{\underset{G}{\downarrow}} \mu a g^\natural$ for some $\mu \in K^\times$. But $\phi(h) = 0$ and $ag^\natural$ is $\to_G^\sigma$-reduced, so $ag^\natural = 0$ and therefore $g^\natural = 0$. By definition of H, $g = (\phi(k), \operatorname{lm} k)$ for some $k \in H$. And since $g^\natural = 0$, we have $k \in J$. In particular, $\operatorname{lm} h = \sigma = \operatorname{lm}(ak)$. ◄

5 Algorithm templates

In all this section we assume that M and S are Noetherian monomial modules, which we define in Section 5.1. (Note that this is unrelated to the property that the regular reduction $\to_G$ is Noetherian.) 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 by a multiple of f. Under mild hypotheses, sigtrees are finite (Theorem 60), giving a very useful

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

▶ **Lemma 46** (Higman, 1952, Theorem 2.1)**.** *Let X be a set with a partial order $\trianglelefteq$. 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 $(x_i)_{i\geqslant 0}$ in $\mathcal{S}$, there are some $i < j$ such that $x_i \trianglelefteq x_j$ (i.e. $\trianglelefteq$ is a wpo);*
N3 *for any sequence $(x_i)_{i\geqslant 0}$ in $\mathcal{S}$, there is a subsequence $(x_j)_{j\geqslant 0}$ such that $x_j \trianglelefteq x_{j+1}$ for any $j \geqslant 0$;*
N4 *for any closed set $T \subseteq X$, there is finite set B such that $T = \{x \in X \mid \exists b \in B, b \trianglelefteq x\}$.*

A monomial set $\mathcal{M}$ of a monomial module M is partially ordered by divisibility, an order that we will denote $\trianglelefteq$, not to be confused with the total order $\leqslant$. Namely, $m \trianglelefteq n$ if there is some $a \in A$ such that $am = n$. Nonetheless, if $a \trianglelefteq b$ then $a \leqslant b$, by M3.

▶ **Definition 47** (Noetherian monomial space)**.** *A monomial space M is* Noetherian *if $\trianglelefteq$ is a wpo.*

From now on, we 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 48.** *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 $\trianglelefteq$-minimal elements of some closed subset of $\mathcal{S}$. 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 $\mathcal{M} \times \mathcal{S}$ we define $(m, \sigma) \trianglelefteq (n, \tau)$ if $m \trianglelefteq n$ and $\sigma \trianglelefteq \tau$. In other words, $(m, \sigma) \trianglelefteq (n, \tau)$ if there are $a, b \in A$ such that $am = n$ and $b\sigma = \tau$. Let us insist that a and b may not be equal.

▶ **Lemma 49.** *If M and S are Noetherian monomial modules, then $\trianglelefteq$ is a wpo on $\mathcal{M} \times \mathcal{S}$.*

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

The following statement relates $\trianglelefteq$ with the domination relation $\sqsubseteq$ (Definition 42).

▶ **Lemma 50.** *For any sigpairs f and g, if $(\operatorname{sig} g, \operatorname{lm} g) \trianglelefteq (\operatorname{sig} f, \operatorname{lm} f)$ then $g \sqsubseteq f$.*

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

The following statement will underlie all the termination proofs. It is an analogue of Dickson's Lemma for sigpairs. However, we will see that this statement may not apply directly. The relation $\trianglelefteq$ on $\mathcal{M} \times \mathcal{S}$, the domination relation $\sqsubseteq$ and Lemma 50 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 51** (Dickson's Lemma for sigpairs)**.** *For any infinite sequence $(f_i)_{i\geqslant 0}$ of sigpairs, there are indices $i < j$ such that $f_i \sqsubseteq f_j$.*

Proof. It is a direct corollary of Lemma 49, Criterion N2 and Lemma 50. ◀

5.2 Processing signatures in order

By Proposition 37, 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 48. This signature has many good properties induced by Corollary 33, and in particular we deduce the following one.

▶ **Proposition 52.** *Let G be a prebasis and let σ such that G is a rewrite basis at any $\tau < \sigma$. Let $f \in AG$ with $\operatorname{sig} f = \sigma$ and let h be any $\to_G$-normal form of f. Then either G is a rewrite basis at σ, or $g \not\sqsubseteq h$ for any $g \in G$.*

Proof. Assume that there is some $g \in G$ such that $g \sqsubseteq h$. Domination condition D2 is ruled out because h is $\to_G$-reduced. Therefore D1 holds: there is some $a \in A$ such that $a \operatorname{sig} g = \sigma$ and $\operatorname{lm}(ag) \leqslant \operatorname{lm} h$. By Corollary 33, $f \downarrow_G \lambda ag$ for some $\lambda \in K^\times$. By confluence, we also have $h \downarrow_G \lambda ag$. Since h is $\to_G$-reduced, this implies that $\operatorname{lm} h \leqslant \operatorname{lm}(ag)$. Combining with the condition D1, we obtain that $\operatorname{lm}(ag) = \operatorname{lm} h$ and therefore that ag is also $\to_G$-reduced. So G is a rewrite basis at σ. ◀

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 $\backsim_G$ (Corollary 33).

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 σ ∈ Σ(G) do
8      σ ← min {σ ∈ Σ(G) | G is not a rewrite basis at σ}
9      pick any f ∈ AG with sig f = σ
10     g ← any →_G-normal form of f
11     G ← G ∪ {g}
12 return G
```

▶ **Theorem 53.** *Algorithm 2 is correct and terminates.*

Proof. Correction follows from Theorem 38. For contradiction, assume that the algorithm does not terminate for some input. Let $g_1, g_2, \ldots$ be the sigpairs that are inserted to G on line 11 on each iteration. By Proposition 51, there are some indice $i < j$ such that $g_i \sqsubseteq g_j$. Proposition 37 implies that when g_j is picked, G is a rewrite basis at any signature $< \operatorname{sig} g_j$. So Proposition 52 implies that $g_i \not\sqsubseteq g_j$, which is a contradiction. ◀

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.

▶ **Example 54** (continued)**.** Let us apply Algorithm 2 to Mora's system (Examples 26, 29 and 36). Let G_{i+3} denote the value of G at the end of the ith iteration. (So that G_3 is the input, starting the counter at 3 because the input contains already 3 elements.) At the start of the algorithm, we have $\Sigma(G_3) = \left\{x^2y^5 \otimes e_2, x^5y^2 \otimes e_3\right\}$. The minimal element of $\Sigma(G_3)$ is $\sigma_4 = x^2y^5 \otimes e_2$ and G is not a rewrite basis at σ_4. We pick the reductant x^2g_2 (only possible choice) and using y^3g_1, we compute the reduction

$$x^2g_2 \to_G \underline{-x^4y} + y^3, \text{in signature } x^2y^5 \otimes e_2$$

We have a new basis element $g_4 = (\underline{-x^4y} + y^3, \sigma_4)$ to obtain G_4. The set $\Sigma(g_4, G_4)$ gives two new elements of $\Sigma(G_4)$:

$$\Sigma(G_4) = \Sigma(G_3) \cup \left\{x^2y^6 \otimes e_2, x^3y^5 \otimes e_2\right\},$$

which gives two new elements in $\Sigma(G_4)$. The minimal element of $\Sigma(G_4)$ at which G_4 is not a rewrite basis is $\sigma_5 = x^5y^2 \otimes e_3$. We pick the reductant y^2g_3 (only choice) and using x^3g_1, we compute the reduction

$$y^2g_3 \to_G \underline{-xy^4} + x^3, \text{in signature } x^5y^2 \otimes e_3.$$

We have a new basis element $g_5 = (\underline{-xy^4} + x^3, \sigma_5)$ to obtain G_5. We have two new elements in the critical set $\Sigma(G_5)$:

$$\Sigma(G_5) = \Sigma(G_4) \cup \left\{x^6y^2 \otimes e_2, x^5y^3 \otimes e_2\right\}.$$

The next signature is $\sigma_6 = x^2y^6 \otimes e_2$. We pick the reductant yg_4, and we have

$$yg_4 \to_{G_5} \underline{y^4} - x^2, \text{ in signature } x^2y^6 \otimes e_2,$$

leading to a new element g_6. Another possible choice of reductant is x^2yg_2, which would lead to the same g_6. From the computational point of view, it is clear that yg_4 is a better choice than x^2yg_2 because g_4 was already obtained by reducing x^2g_2, so there will be less work to reduce yg_4 than to reduce x^2yg_2. When the signatures are not process in order, the choice of the reductant have a theoretical importance discussed in the following sections.

The process goes on similarly. We can represent the output of Algorihm 2 in the form of trees (which are instances of the concept of *well-formed sigtree* introduced in Section 5.4). We say that $h \in G$ is the parent of $g \in G$ if g is obtained, on line 10, from the reduction of $f = ah$ for some $a \in A$. To display the tree, we show only the leading monomials of the sigpairs, the iteration at which they have been inserted, and an edge $h \to g$ is labelled by the element $a \in A$ defined above. For the input discussed above, we obtain Figure 1. The data displayed in this tree (leading monomials and signatures) is enough to certify that the output is a rewrite basis, using Theorem 38.

▶ **Example 55** (Katsura-6)**.** We consider the system Katsura-6 (from the famous benchmark family Katsura-n, available in Sagemath with the function `sage.rings.ideal.Katsura`),

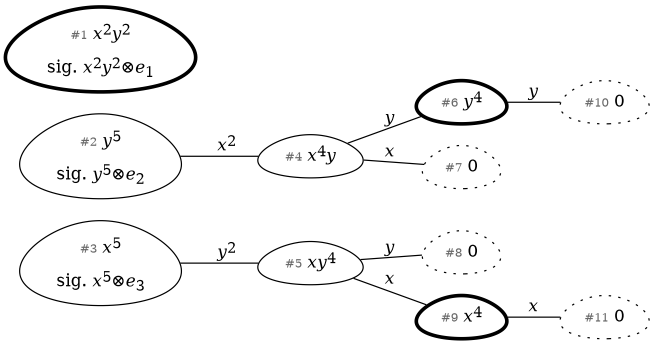

Figure 1 Graphical trace of Algorithm 2 applied to the system in Example 26. Root nodes, on the left, represent input polynomials. Bold nodes represent elements of the rewrite basis whose leading monomial is the leading monomial of some element of the reduced Gröbner basis of the input ideal. The signature of a node n can be obtained by multiplying the labels of the edges from n to the root node, and then multiplying by the signature of the corresponding root node. For example, the signature of the input node 3 is $x^5 \otimes e_3$, so the signature of the node 11 is $x^7y^2 \otimes e_3$.

given in $\mathbb{Q}[a, b, c, d, e, f]$ (with degree reverse lexicographic ordering) by the polynomials

$$g_1 = a + 2b + 2c + 2d + 2e + 2f - 1, \qquad g_2 = c^2 + 2bd + 2ae + 2bf - e,$$
$$g_3 = bc + ad + be + cf - \tfrac{1}{2}d, \qquad g_4 = b^2 + 2ac + 2bd + 2ce + 2df - c,$$
$$g_5 = ab + bc + cd + de + ef - \tfrac{1}{2}b, \qquad g_6 = a^2 + 2b^2 + 2c^2 + 2d^2 + 2e^2 + 2f^2 - a.$$

Figures 2, 3 and 4 shows the result of running Algorithm 2 on this input, with different signature orderings. At each iteration, when there are multiple possible choices, we pick the one that comes from the most recently inserted element of G, this is the F5 selection strategy, see Section 5.5. The computations are displayed in the form of sigtrees, as in Example 54.

5.3 Minimizing the leading monomial of the reductant

Processing critical signatures 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. (2023) used signature algorithms to compute saturation ideals, this involves enlarging 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 (Galkin, 2014) was still unsettled, Arri and Perry (2011, 2017) introduced the idea of choosing carefully the sigpair to be reduced, called the *reductant*, at a given signature to ensure termination. This is based on the following observation.

▶ **Proposition 56.** *Let G be a prebasis and let σ be a signature at which G is not a rewrite basis. Let $f \in AG$ with $\operatorname{sig} f = \sigma$ and $\operatorname{lm} f$ minimal. Let h be a $\to_G$-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 $\to_G$-reduced. Therefore D1 holds: there is some $a \in A$ such that $a \operatorname{sig} g = \sigma$ and $\operatorname{lm}(ag) \leqslant \operatorname{lm} h$. Besides, G is not a rewrite basis at σ, it follows that f is not $\to_G$-reduced, and thus $\operatorname{lm} h < \operatorname{lm} f$ since $f \to_G h$. It follows that $\operatorname{lm}(ag) < \operatorname{lm} f$, which contradicts the minimality of f. ◀

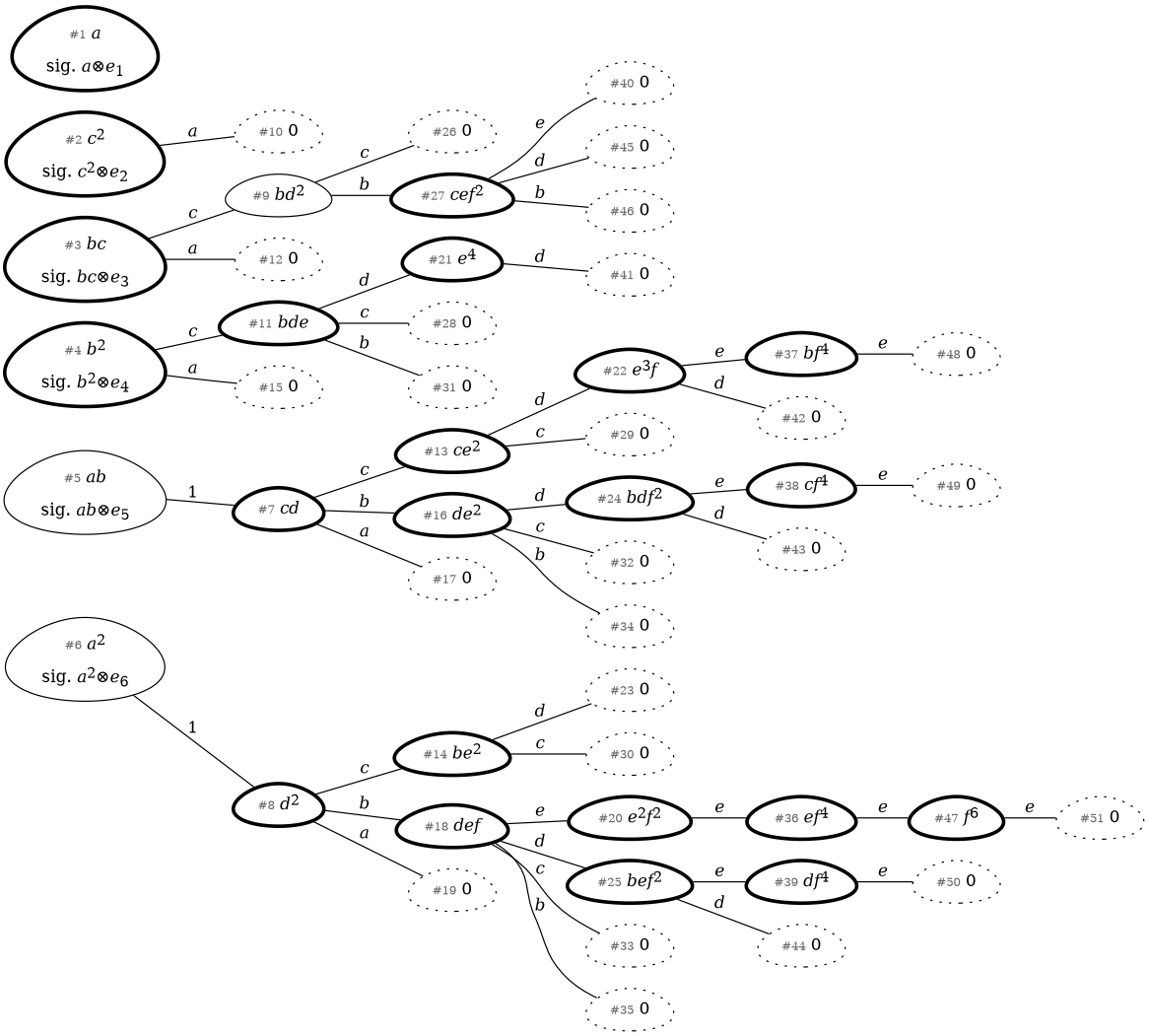

Figure 2 Trace of the computation of a rewrite basis for Katsura-6 (Example 55) with the TOP order on the signatures, and the F5 selection strategy of the reductant. The input polynomials are given the signatures $\operatorname{sig} g_i = \operatorname{lm} g_i \otimes e_i$.

Although Arri and Perry still requires to process signatures by increasing order, Proposition 56 opens the way for out-of-order signature handling, as Sun and Wang (2013) and Gao et al. (2016) did. The formulation that proposed here (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.

► **Theorem 57.** *Algorithm 3 is correct and terminates.*

Proof. Identical to the proof of Theorem 53, but using Proposition 56 instead of Proposition 52. ◄

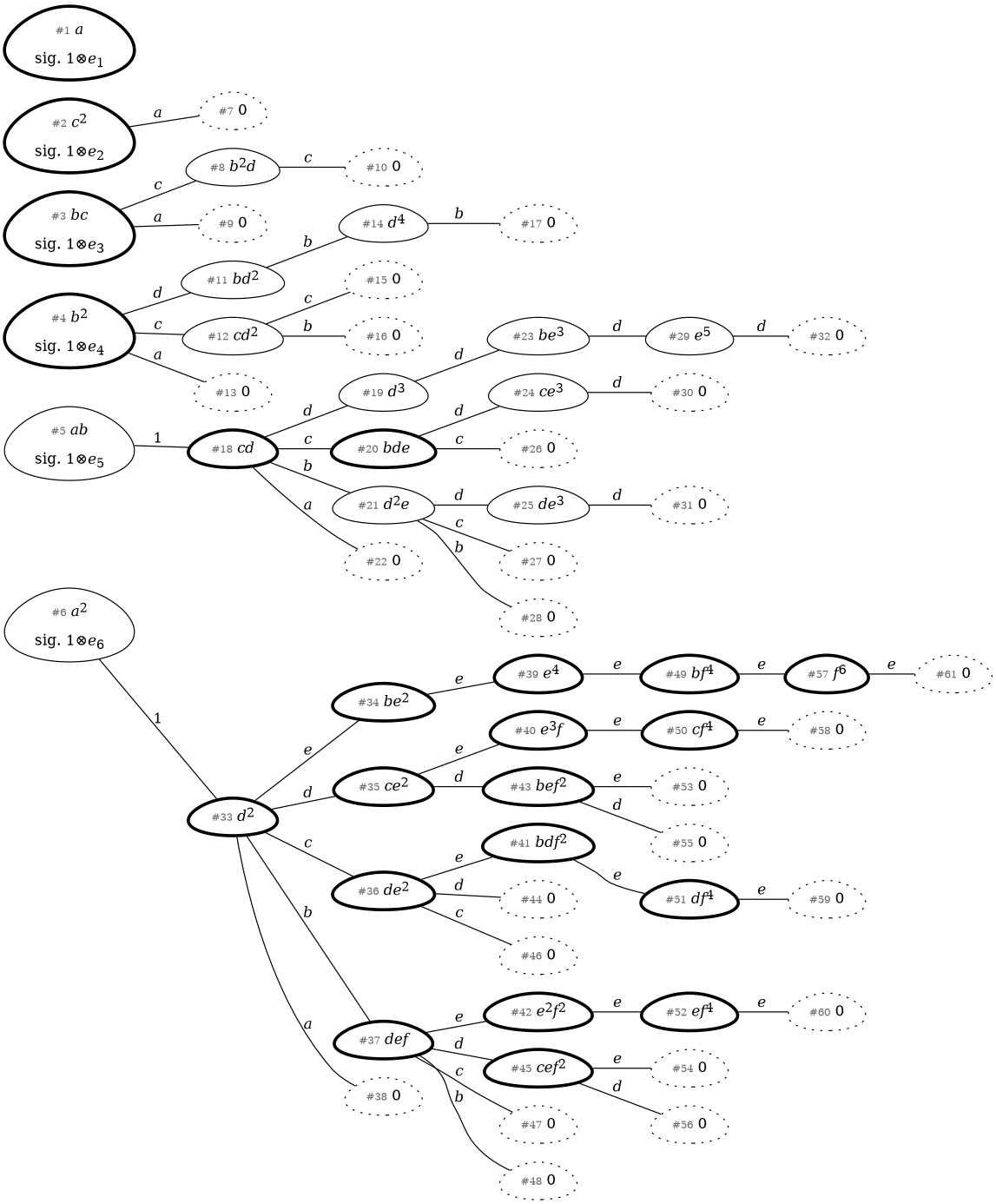

Figure 3 Trace of the computation of a rewrite basis for Katsura-6 (Example 55) with the POT order on the signatures, and the F5 selection strategy of the reductant. The input polynomials are given the signatures $\operatorname{sig} g_i = 1 \otimes e_i$.

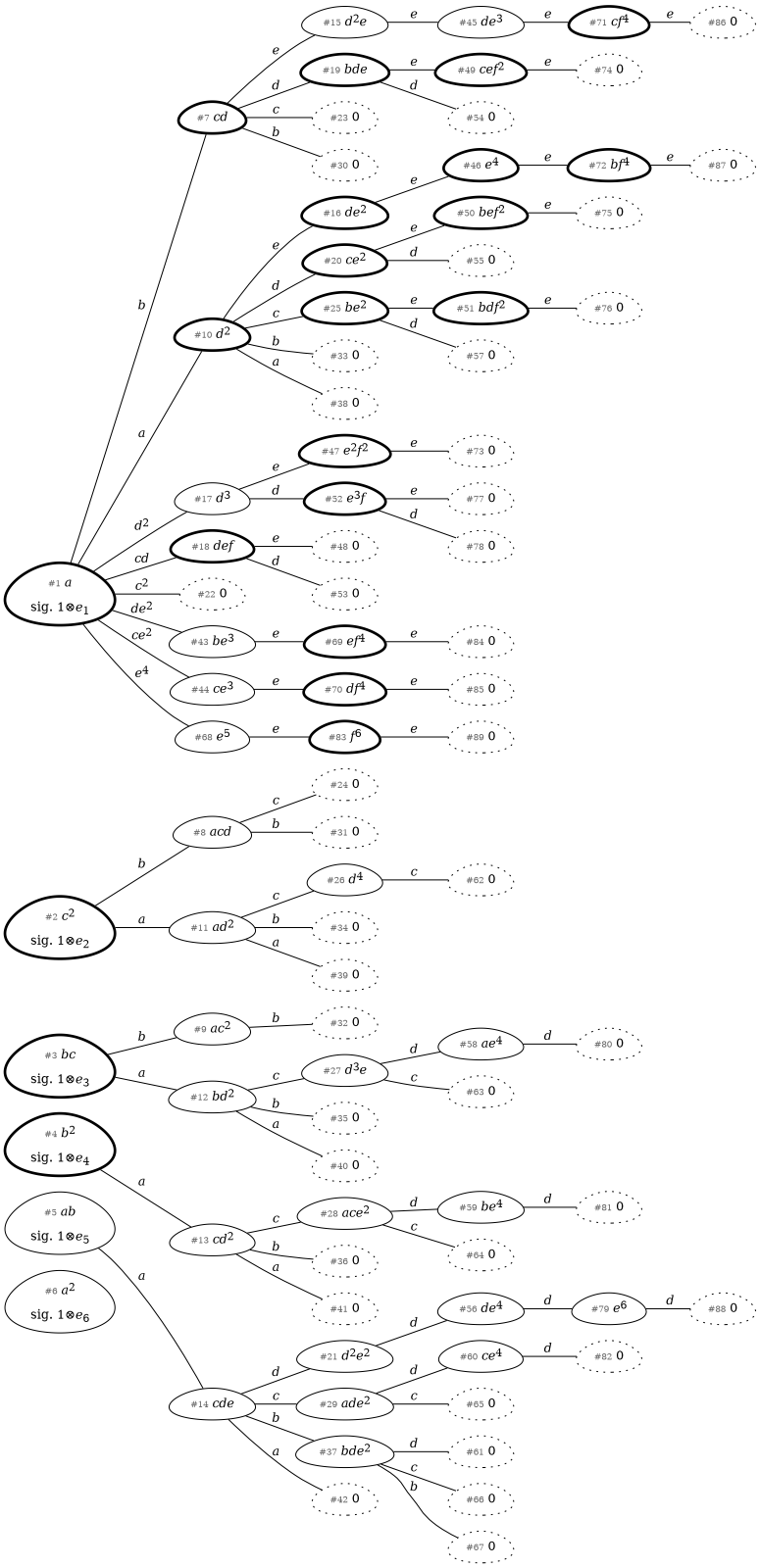

Figure 4 Trace of the computation of a rewrite basis for Katsura-6 (Example 55) with the TOP order on the signatures, and the F5 selection strategy of the reductant. The input polynomials are given the unshifted signatures $\operatorname{sig} g_i = 1 \otimes e_i$. In contrast to Figure 2, note that many elements are not necessary to form a Gröbner basis, eventhough they are necessary to form a rewrite basis. For example, the 79th element g_{79} with $\operatorname{lm} g_{79} = e^6$ and $\operatorname{sig} g_{79} = ad^3 \otimes e_4$ cannot be reduced by the earlier g_{46}, as suggested by the relation $e^2 \operatorname{lm} g_{46} = \operatorname{lm} g_{79}$, because $e^2 \operatorname{sig} g_{46} = ae^4 \otimes e_1$ is bigger than $\operatorname{sig} g_{79}$.

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

while G is not a rewrite basis at all $\sigma \in \Sigma(G)$ **do**
    pick any $\sigma \in \mathcal{S}$ such that G is not a rewrite basis at σ
    pick $f \in AG$ with $\operatorname{sig} f = \sigma$ and $\operatorname{lm} f$ minimal
    $g \leftarrow$ any $\to_G$-normal form of f
    $G \leftarrow G \cup \{g\}$
return G

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 $r \geqslant 1$), 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 *ancestors* of a node $n = (k_1, \dots, k_r)$ are the nodes $(k_1, \dots, k_j)$ for $0 \leqslant j < r$.

▶ **Definition 58** (sigtree)**.** *A* sigtree *is a tree $\mathcal{T}$ together with a rank function* $\operatorname{rk} : \mathcal{T} \to \mathbb{N}$ *and a label funtion $\lambda : \mathcal{T} \to M \times \mathcal{S}$ (recall that $M \times \mathcal{S}$ is the set of sigpairs).*

Sigtrees are a natural way to represent the process of computing a rewrite basis. Indeed, Algorithms 3 or 2, as well as Pseudo-algorithm 1, produce sigtrees as follows. The elements of G, the rewrite basis begin computed, are the labels of the sigtrees. There is one sigtree for each element of the input sigset. The root node of each sigtree is labelled with the corresponding input element. At the beginning of the algorithms, there are only the root nodes. Then, each time some $f \in AG$ is picked, reduced, and inserted into G, we can write $f = a\lambda(n)$, for some node n of the sigtree, and some $a \in A$, and we insert in the sigtree a new child node of n containing the new element. The rank function reflects the *birthdate* of a node. Figures 1, 2, 3 and 4 are examples of sigtrees obtained in this way.

▶ **Definition 59** (well-formed sigtree)**.** *A* well-formed sigtree *is a sigtree $\mathcal{T}$ such that:*

T1 $\forall n \in \mathcal{T}, \forall m$ *child of* $n, \exists a \in A, a \operatorname{sig} \lambda(n) = \operatorname{sig} \lambda(m)$ *and* $a \operatorname{lm} \lambda(n) > \operatorname{lm} \lambda(m)$,
"a child is more reduced than its parent"*;*

T2 $\forall n \in \mathcal{T}$, $\lambda(n)$ *is $\to$-reduced with respect to the sigset* $\{\lambda(p) \mid p$ *is an ancestor of* $n\}$,
"a child is reduced modulo its ancestors"*;*

T3 $\forall n \in \mathcal{T}, \forall p, q$ *children of* n, $\operatorname{rk}(p) < \operatorname{rk}(q) \Rightarrow \operatorname{sig} \lambda(p)$ *does not divide* $\operatorname{sig} \lambda(q)$,
"the signature of a node does not divide that of younger sibling nodes".

T4 $\forall k \in \mathbb{N}, \{n \in \mathcal{T} \mid \operatorname{rk}(n) = k\}$ *is finite.*

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. T4 will follow from an appropriate bookkeeping. T3 is the real constraint. In the context above, T3 puts a contraint 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 $b\lambda(n) \to \lambda(m)$ for some child m of n and for some b such that $\exists c \in A, cb\lambda(n) = a\lambda(n)$. In which case we can reduce $c\lambda(m)$ instead of $a\lambda(n)$.

▶ **Theorem 60.** *A well-formed sigtree 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 $(n_i)_{i\geqslant 0}$ such that n_{i+1} is a child of n_i. By Proposition 51, 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 $b \operatorname{sig} \lambda(n_i) = \operatorname{sig} \lambda(n_j)$ and $b \operatorname{lm} \lambda(n_i) \leqslant \operatorname{lm} \lambda(n_j)$. By T1 (applied all along the path from n_i to n_j), there is some $a \in A$ such that $a \operatorname{sig} \lambda(n_i) = \operatorname{sig} \lambda(n_j)$ and $a \operatorname{lm} \lambda(n_i) > \operatorname{lm} \lambda(n_j)$. Since $a \operatorname{sig} \lambda(n_i) = b \operatorname{sig} \lambda(n_i)$, S1 implies that $a \operatorname{lm} \lambda(n_i) = b \operatorname{lm} \lambda(n_i)$, leading to a contradiction.

If a node has infinitely many children, T4 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. ◀

5.5 The F5 reductant selection strategy

In the original presentation of F5, Faugère (2002) 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 σ
24 pick some $a \in A$ and $f \in G$ such that $a \operatorname{sig} f = \sigma$ and $R[f]$ maximal
25 $g \leftarrow$ any $\to_G$-normal form of af
26 $G \leftarrow G \cup \{g\}$
27 $R[g] \leftarrow r$
28 $r \leftarrow r + 1$
29 **return** G

▶ **Theorem 61.** *Algorithm 4 is correct and terminates.*

Proof. Correctness follows from Theorem 38. For termination, consider the sigtrees (one for each input element) 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. If the algorithm does not terminate, at least one of the sigtrees is infinite. Therefore, to prove termination, it is enough to check that the sigtrees are finite.

These sigtrees are 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 nodes of a given rank

is at most one, this gives T4. Theorem 60 applies and shows that the sigtrees are finite, so 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 $\mathcal{S}$ (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 $\mathcal{S}$ 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:

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

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 $\to_G$ 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) and Theorem 38 in hand, it is clear that Algorithm 5 returns a rewrite basis when it terminates.

Termination is ensured *by design* by contructing well-formed sigtrees. The algorithm maintains two lists *children* and L. The L list contains the labels: $L[i]$ is the label of the ith node in the sigtree. The *children* list encodes the tree structure: *children*$[i]$ is the set of chidren of the node i. The set *children*$[0]$ contains the root nodes. The rank of the ith node is defined to be i. The selection procedure of the reductant makes it sure that the sigtree is well formed. 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 60, 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 explicitely, we may ignore the *children* list. (To really match with F5 algorithm, the reductant is chosen to be zero if possible, even if it does not correspond to the most recent possible reductant.)

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:

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

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

```
« initialize signature queue and sigtree » ⟶ line 43
while Q is not empty do
    σ ← some element of Q
    Q ← Q \ {σ}
    « select a reductant f in signature σ with corresponding node k » ⟶ line 64
    if f is →_G-reducible then
        g ← a →_G-normal form of f
        « insert a node with label g and parent k » ⟶ line 54
        « update the queue with the new relation g » ⟶ line 60
        G ← G ∪ {g}
return G

Chunks
« initialize signature queue and sigtree » ≡
    Q ← ∅                                   —signature queue
    children ← empty list                   —maps a node to its children
    L ← empty list                          —maps a node to its label
    children[0] ← ∅                         —the set of root nodes
    n ← 1                                   —node counter
    k ← 0                                   —index of the root node
    for g ∈ G do                            —create nodes for input elements
        « insert a node with label g and parent k » ⟶ line 54
        « update the queue with the new relation g » ⟶ line 60

« insert a node with label g and parent k » ≡
    L[n] ← g ;
    children[k] ← children[k] ∪ {n} ;
    children[n] ← ∅
    n ← n + 1

« update the queue with the new relation g » ≡
    for h ∈ G do
        Q ← Q ∪ Σ(g, h) ∪ Σ(h, g)

« select a reductant f in signature σ with corresponding node k » ≡
    k ← 0                                   —start the search from the root node
    for c ∈ children[k] do                  —the order of iteration does not matter
        if sig L[c] divides σ then
            k ← c                           —go down the tree
            goto 66                         —continue the search from the new position
    pick a ∈ A such that a sig L[k] = σ
    f ← aL[k]
```

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

```
72 Similar to Algorithm 5, except for the following chunk
73 « select a reductant f in signature σ with corresponding node k » ≡
74     k ← 0
75     for 1 ⩽ j < n do
76         if sig L[j] divides σ then k ← j
77         if lm L[j] = 0 then break          —stop the search if σ is a syzygy signature
78     pick a ∈ A such that a sig L[k] = σ
79     f ← aL[k]
```

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

Algorithm 7 Variant of Algorithm 5 with signature pruning

```
80 Similar to Algorithm 5, except for the following chunk
81 « update the queue with the new relation g » ≡
82     for h ∈ G do
83         Q ← Q ∪ Σ(g, h) ∪ Σ(h, g)
84     for σ ∈ Q do
85         if ∃τ ∈ Q \ {σ}, τ divides σ then
86             Q ← Q \ {σ}
```

5.7 Simultaneous reduction

As another variation of Algorithm 5, we may handle several signatures at a time, in the F4 style (Faugère, 1999; Albrecht & Perry, 2010; Eder & Faugère, 2017, §13). Concretely, the sigset G that is used to compute the reductions not updated each time a new element is discovered. The new elements are inserted in a sigset N and after a bunch of signatures is handled (how many is to be determined by the implementation), the elements of N are inserted in G. On line 98, the reductant g is reduced with respect to G (and as usual, multiples of elements of G can be used in reduction steps) and also with respect to N (but multiples cannot be used in reduction steps). In other words, the polynomial part $f^\natural$ is reduced modulo the set $AG^{<\sigma} \cup N$.

The reason to delay insertion into G is the principle of simultenous reduction. If we have to perform the reductions of sigpairs $f_1, \dots, f_r$ with respect to the same sigset G, it is possible to formulate the problem in terms of a matrix whose rows represent the f_i and all possibly useful reducers from AG in a reduction chain starting from any of the f_i. Once this matrix is computed (this is the *symbolic preprocessing* step), it can be used to compute the reductions efficiently. This matrix aspect is crucial for high-performance computations but it is a transparent transformation of the algorithm: it does not change what is computed, compared to the naive reductions of the f_i. For a more detailed introduction to the F4 strategy, see Cox et al. (2015, Chapter 10, §3).

▶ **Theorem 62.** *Algorithm 8 is correct and terminates.*

Proof. Termination is clear because the algorithm produces a well-formed sigtree (where the rank of a node is the number of the iteration at which it was produced), 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. Indeed,

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

```
« initialize signature queue and sigtree » ⟶ line 43
while Q is not empty do
    S ← some nonempty subset of Q                    —select several signatures at a time
    Q ← Q \ S
    F ← ∅                                            —set of reductants and corresponding nodes
    for σ ∈ S do                                     —selection of reductants
        « select a reductant f in signature σ with corresponding node k » ⟶ line 64
        if f is →_G-reducible then
            F ← F ∪ {(f, k)}                         —we keep the information of the parent
    N ← ∅                                            —set of newly computed relations
    for (f, k) ∈ F by increasing order of sig f do   —reduction of reductants
        g♮ ← a →-normal form of f♮ w.r.t. AG^{<σ} ∪ {h♮ | h ∈ N}
        g ← (g♮, σ)
        « insert a node with label g and parent k » ⟶ line 54
        « update the queue with the new relation g » ⟶ line 60
        N ← N ∪ {g}                                  —insertion of g in G is delayed
    G ← G ∪ N
    return G
```

when an element g is inserted into G, if g is $\to_G$-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 $\to_G$-reduced. In this case, then there is some $h \in G$ which reduces g and we check easily that $\Sigma(g, h) = \{\operatorname{sig} g\}$. So in this case, $\operatorname{sig} g$ is not actually removed from Q and the invariant is preserved. ◂

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 M4, so construction of prebases is easy, see Remark 25.

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

6.2 Modules over polynomial rings

Let r be a positive integer and let $M = K[x_1, \ldots, x_n]^r$, 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 M4.

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 $\operatorname{lm} f$ and $\operatorname{lm} g$ coincide. Indeed, nonzero monomials in $\mathcal{M}$ have an *index* in $\{1, \ldots, r\}$ which is unchanged under multiplication. Therefore, if $f^\natural = 0$ or $g^\natural = 0$, or $\operatorname{lm} f$ and $\operatorname{lm} g$ have different indices, then $\Sigma(f, g) = \varnothing$. Otherwise, there are monomial $a, b \in A$ such that $a \operatorname{lm} f = b \operatorname{lm} g$ (and $a \operatorname{lm} f$ minimal). Depending on the comparison of $a \operatorname{sig} f$ and $b \operatorname{sig} g$, $\Sigma(f, g)$ is either $\varnothing$ or $\{a \operatorname{sig} f\}$, as in the polynomial case.

6.3 Monoid algebras

Let A be a submonoid of $\{x_1^{i_1} \cdots x_n^{i_n} \mid i_1, \ldots, i_n \in \mathbb{N}\}$ 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 finitelty 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 $A = \left\{x^i y^j \,\middle|\, i + j \geqslant 2\right\}$ – and that $f^\natural = x^2$ and $g^\natural = xy$. The set of all $a \in A$ such that $\operatorname{lm}(af^\natural)$ is divided by $\operatorname{lm}(g^\natural)$ is generated by xy and y^2. It is not generated by y because y is not in A. Assuming that $xy \operatorname{sig} f > x^2 \operatorname{sig} g$ and $y^2 \operatorname{sig} f > xy \operatorname{sig} g$, we have

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

6.4 Weyl algebras

Let $M = K\langle x_1, \ldots, x_n, \partial_1, \ldots, \partial_n\rangle$ be the Weyl algebra on n variables. It is noncommutative. We may define it as the subalgebra of $\operatorname{End}_K(K[X_1, \ldots, X_n])$ 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 $x_1, \ldots, x_n$ and $\partial_1, \ldots, \partial_n$. We could also choose $A = M$. This turns M into a Noetherian monomial module with the extra property M4, so we can construct signature modules with Remark 25. We could also choose A to be the monoid of nonzero elements of R. Things behave similarly to the polynomial case, due to *quasicommutativity*: for any $a, b \in M$, $\operatorname{lm}(ab) = \operatorname{lm}(ba)$.

6.5 Differential algebras

Let $M = K[t, x_0, x_1, x_2, \ldots]$ be 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 $\operatorname{End}_K M$ 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 $t < x_0 < x_1 ...$

We choose A to be the monoid generated by $\partial, t, x_0, x_1 ...$ This turns M into a monomial module. It is quasicommutative (that is $\mathrm{lm}(abm) = \mathrm{lm}(bam)$ for any $a, b \in W$ and $m \in M$) but not Noetherian. However, it satisfies the extra condition M4 and the critical sets $\Sigma(f, G)$ are finite. This example extends to several independent variables and several function variables.

6.6 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 monoid 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. This turn M into a monomial space with extra condition M4. If $n > 1$, it is not Noetherian, but the critical sets are finite.

To deal with two-sided ideals of M, we need to consider not only left multiplications but also right multiplications. We introduce the monoid $A' = A \times A^{\mathrm{op}}$ of pairs of words with with the composition $(a, b)(a', b') = (aa', b'b)$ and the action on M given by $(a, b)m = amb$. This turn M into another mononomial space. When $n > 1$, it is not Noetherian and does not satisfy extra condition M4. For example, as shown by Green et al. (1998), if $x_1 > x_2$ then $x_1x_1 - x_1x_2$ generates a two-sided ideal without a finite Gröbner basis. Moreoever, the critical sets may be infinite, eventhough they may 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. https://doi.org/10/cd5td7

Arri, A., & Perry, J. (2017). Corrigendum to “The F5 criterion revised”. *J. Symb. Comput.*, *82*, 164–165. https://doi.org/10/gp8639

Bardet, M., Faugère, J.-C., & Salvy, B. (2015). On the complexity of the F5 Gröbner basis algorithm. *J. Symb. Comput.*, *70*, 49–70. https://doi.org/10/gntfcb

Becker, T., & Weispfenning, V. (1993). *Gröbner Bases: A Computational Approach to Commutative Algebra*. Springer-Verlag. https://doi.org/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. https://doi.org/10/gnt5z9

Berthomieu, J., Eder, C., & Safey El Din, M. (2021). Msolve: A library for solving polynomial systems. *Proc. ISSAC 2021*, 51–58. https://doi.org/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. https://doi.org/10/ckdngx

Buchberger, B. (1965). *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*.

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. https://doi.org/10/dz9kz6 (Original work published 1965)

Caruso, X., Vaccon, T., & Verron, T. (2020). Signature-based algorithms for Gröbner bases over Tate algebras. *Proc. ISSAC 2020*, 70–77. https://doi.org/10/gp99pq

Cox, D. A., Little, J., & O'Shea, D. (2015). *Ideals, Varieties, and Algorithms* (4th ed.). Springer. https://doi.org/10/hzv6

Decker, W., Greuel, G.-M., Pfister, G., & Schönemann, H. (2022). Singular 4-3-0 — A computer algebra system for polynomial computations. http://www.singular.uni-signaturebasiskl.de

Eder, C., & Faugère, J.-C. (2017). A survey on signature-based algorithms for computing Gröbner bases. *J. Symb. Comput.*, *80*(3), 719–784. https://doi.org/10/ggck7f

Eder, C., Lairez, P., Mohr, R., & Safey El Din, M. (2023). A signature-based algorithm for computing the nondegenerate locus of a polynomial system. *J. Symb. Comput.*, *119*, 1–21. https://doi.org/10/jxn2

Eder, C., & Perry, J. (2011). Signature-based algorithms to compute Gröbner bases. *Proc. ISSAC 2011*, 99–106. https://doi.org/10/dmwqmp

Eder, C., Pfister, G., & Popescu, A. (2017). On signature-based Gröbner bases over Euclidean rings. *Proc. ISSAC 2017*, 141–148. https://doi.org/10/gsbxs2

Eder, C., & Roune, B. H. (2013). Signature rewriting in Gröbner basis computation. *Proc. ISSAC 2013*, 331–338. https://doi.org/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. https://doi.org/10/bpq5dx

Faugère, J.-C. (2001). Finding all the solutions of Cyclic 9 using Gröbner basis techniques. *Comput. Math.*, 1–12. https://doi.org/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. https://doi.org/10/bd4nnq

Faugère, J.-C., & Joux, A. (2003). Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Gröbner bases. *CRYPTO 2003*, 44–60. https://doi.org/10/fpfzgc

Francis, M., & Verron, T. (2020). A signature-based algorithm for computing Gröbner bases over principal ideal domains. *Math.Comput.Sci.*, *14*(2), 515–530. https://doi.org/10/gsbxs4

Galkin, V. V. (2014). Termination of the F5 algorithm. *Program. Comput. Softw.*, *40*(2), 47–57. https://doi.org/10/ghjx58

Gao, S., Guan, Y., & Volny, F. (2010). A new incremental algorithm for computing Groebner bases. *Proc. ISSAC 2010*, 13–19. https://doi.org/10/cwg6rj

Gao, S., Volny, F., & Wang, M. (2016). A new framework for computing Gröbner bases. *Math. Comp.*, *85*(297), 449–465. https://doi.org/10/f7t889

Gebauer, R., & Möller, H. M. (1986). Buchberger's algorithm and staggered linear bases. *Symp. Symb. Algebr. Comput.*, 218–221. https://doi.org/10/cb24fn

Gebauer, R., & Möller, H. M. (1988). On an installation of Buchberger's algorithm. *J. Symb. Comput.*, *6*(2), 275–286. https://doi.org/10/bfjdwc

Green, E. D., Mora, T., & Ufnarovski, V. (1998). The non-commutative Gröbner freaks. In M. Bronstein, V. Weispfenning, & J. Grabmeier (Eds.), *Symb. Rewriting Tech.* (pp. 93–104). Birkhäuser. https://doi.org/10/dfbt7g

Hashemi, A., & Ars, G. (2010). Extended F5 criteria. *J. Symb. Comput.*, *45*(12), 1330–1340. https://doi.org/10/bmfh29

Hashemi, A., & Javanbakht, M. (2021). On the construction of staggered linear bases. *J. Algebra Appl.*, *20*(8), 2150132. https://doi.org/10/gqwrpn

Higman, G. (1952). Ordering by divisibility in abstract algebras. *Proc. Lond. Math. Soc.*, *2*(1), 326–336. https://doi.org/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. https://doi.org/10/gp85ss

Hofstadler, C., & Verron, T. (2023). Signature Gröbner bases in free algebras over rings. *Proc. ISSAC 2023*, 298–306. https://doi.org/10/gskrd2

Huet, G. (1980). Confluent reductions: Abstract properties and applications to term rewriting systems. *J. ACM*, *27*(4), 797–821. https://doi.org/10/fj7n4g

Kambe, Y. (2023). *Analysis of computing Gr\"obner bases and Gr\"obner degenerations via theory of signatures*. arXiv: 2305.13639 [math]. Retrieved 2023, from http://arxiv.org/abs/2305.13639

Kreuzer, M., & Robbiano, L. (2000). *Computational commutative algebra* (Vol. 1). Springer. https://doi.org/10/ffxbqr

Lu, D., Wang, D., Xiao, F., & Zhou, J. (2018). Extending the GVW algorithm to local ring. *Proc. ISSAC 2018*, 271–278. https://doi.org/10/gsbxr9

Möller, H. M., Mora, T., & Traverso, C. (1992). Gröbner bases computation using syzygies. *Proc. ISSAC 1992*, 320–328. https://doi.org/10/cgb2ts

Monagan, M., & Pearce, R. (2015). A compact parallel implementation of F4. *Proc. PASCO 2015*, 95–100. https://doi.org/10/ggpbmk

Mora, T. (1994). An introduction to commutative and noncommutative Gröbner bases. *Theor. Comput. Sci.*, *134*(1), 131–173. https://doi.org/10/dvwxsv

Mora, T. (2005). *Solving polynomial equation systems* (Vol. 2). Cambridge University Press. https://doi.org/10/jdwm

Roune, B. H., & Stillman, M. (2012). Practical Gröbner basis computation. *Proc. ISSAC 2012*, 203–210. https://doi.org/10/ggkpqd

Stillman, M. (1990). Methods for computing in algebraic geometry and commutative algebra. *Acta Appl. Math.*, *21*(1), 77–103. https://doi.org/10/dsx7mz

Sun, Y., & Wang, D. (2011). *Solving detachability problem for the polynomial ring by signature-based Gröbner 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. https://doi.org/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. https://doi.org/10/ghtkmx

Traverso, C. (1996). Hilbert functions and the Buchberger algorithm. *J. Symb. Comput.*, *22*(4), 355–376. https://doi.org/10/b3x2ct

Vaccon, T., Verron, T., & Yokoyama, K. (2018). On affine tropical F5 algorithms. *Proc. ISSAC 2018*, 383–390. https://doi.org/10/d3mr

Vaccon, T., & Yokoyama, K. (2017). A tropical F5 algorithm. *Proc. ISSAC 2017*, 429–436. https://doi.org/10/gsbxs8

Winkler, F. (1996). *Polynomial algorithms in computer algebra*. Springer-Verlag. https://doi.org/10/bkh6hq