JOHANNES KEPLER
UNIVERSITÄT LINZ
Netzwerk für Forschung, Lehre und Praxis

Numerical Computation of Gröbner Bases

Dissertation

zur Erlangung des akademischen Grades

Doktor der technischen Wissenschaften

Angefertigt am *Institut für Symbolisches Rechnen*

Betreuung:

Prof. Franz Winkler
Dr. Hans Stetter

Eingereicht von:

Aleksey Kondratyev

Linz, Oktober 2003

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Dissertation selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe

Aleksey Kondratyev
Hagenberg, 1. Oktober 2003

Abstract

The method of Gröbner bases is a custom tool in modern computer algebra to solve systems of polynomial equations. The topic of this thesis is the theory and computation of Gröbner bases in a numerical environment. Particular emphasis is given to an application of Gröbner bases for solving a zero-dimensional polynomial system by the eigenvector method. Here, the multiplicative structure of the associated quotient algebra of the ideal of the system is exploited to translate the problem of finding the solutions into a matrix eigenvector problem; such problems are well-studied in numerical analysis and therefore attractive for a numerical approach.

We present a completely developed variation of the theory of Gröbner bases with a modified notion of a term order and some basic translation laws which establish a connection with the customary Gröbner bases. An implementation of the respectively modified Buchberger algorithm for constructing Gröbner bases adopted for a numerical environment and executed in floating-point arithmetic has also been developed. Our variation of the theory permits the construction of an algorithm which incorporates an analogue to complete pivoting (as in Gaussian elimination of a linear system), which is beyond the frame of classical Gröbner bases where the "direction" of elimination is rigidly controlled by a chosen term order. This more flexible pivoting mechanism stabilizes both the course and the result of the computation in the sense proposed by Hans Stetter. Our numerical implementation also employs a rearrangement of the usual Buchberger algorithm which is attractive from a numerical point of view; it has initially been introduced by J. C. Faugere for the customary Gröbner bases and exact arithmetic. A collection of numerical examples performed with our implementation is provided.

Zusammenfassung

Die Methode der Gröbnerbasen ist das Standardhilfsmittel in der modernen Computeralgebra, um Systeme von Polynomgleichungen zu lösen. Das Thema dieser Doktorarbeit ist die Theorie und Berechnung von Gröbnerbasen in einer numerischen Umgebung.

Besonders hervorgehoben wird eine Anwendung von Gröbnerbasen, die das Lösen von 0-dimensionalen Polynomsystemen mit der sogenannten Eigenvektor-Methode ermöglicht. In diesem Verfahren wird die multiplikative Struktur der assoziierten Quotientenalgebra von dem Ideal des Systems ausgenutzt, um das Problem zum Finden von Lösungen auf das Eigenvektor Problem zu übersetzen; solche Probleme sind wohl studiert in der numerischen Analysis und folglich attraktiv für einen numerischen Ansatz.

Wir präsentieren eine vollständig entwickelte Variation von der Gröbnerbasentheorie mit einem modifizierten Begriff von Termordnung und einigen fundamentalen Übersetzungsregeln, welche die Verbindung zu den herkömmlichen Gröbnerbasen herstellen. Eine Implementierung von dem entsprechend modifizierten Buchberger Algorithmus zur Konstruktion von Gröbernbasen ist entwickelt worden, welche für die numerische Umgebung angepasst ist und in numerischer Gleikomma Arithmetik ausgeführt wird. Unsere Variation von der Theorie erlaubt die Konstruktion von einem Algorithmus, welcher ein Analogon zum vollständigen Pivoting (wie in der Gauß-Elimination von einem linearen System) integriert. Diese Methode geht über den klassischen Rahmen der Gröbnerbasen hinaus, wo die "Richtung" der Elimination von der Termordnung strikt kontrolliert wird. Dieser flexible Pivot-Mechanismus stabilisiert sowohl den Pfad als auch das Ergebnis der Berechnung im Sinne von Hans Stetters Definitionen. Unsere numerische Implementierung involviert auch eine Modifizierung von dem gewöhnlichen Buchberger Algorithmus, welcher geeigneter ist vom numerischen Standpunkt; er wurde ursprünglich von J. C. Faugere für die gewöhnlichen Gröbnerbasen und exakter Arithmetik vorgestellt. Eine Sammlung von numerischen Beispielen illustriert unsere Implementierung.

Contents

Chapter 1

Introduction

The topic of this thesis is the machinery of Gröbner bases in a numerical environment with a particular emphasis given to the solving of a zero-dimensional polynomial system by the eigenvector method where the problem of finding the solutions is translated into a matrix eigenvector problem. This is possible by exploiting the multiplicative structure of the associated quotient algebra of the ideal of the system. Matrix eigenvector problems are well-studied in numerical analysis and therefore attractive in a numerical setting. The eigenvector method has been known from the 19-th century. Relatively recently it has been taken up in [AS88] and further investigated in [Moe93], [Ste93], [MS], [Ste96b] and various other publications.

When we consider polynomial systems with numerically specified coefficients with a limited meaningful accuracy, the method of Gröbner bases is unfortunately not a very adopted tool. The computation of a Gröbner basis by Buchberger algorithm is essentially an elimination process like Gaussian elimination in a linear system; but in place of the variables in a linear system we have terms, the power products in variables, and this Buchberger type elimination generally requires linear combinations of shifts with terms, the so-called S-polynomials.

For the machinery of Gröbner bases, the term order is the key ingredient as it introduces a well-defined "direction" of elimination of the terms in the polynomial system in place of the linear order of the variables in Gaussian elimination. But it is well-known and understood in the numerical analysis that a numerically stable Gaussian elimination must take into account the magnitude of the coefficients employed in the elimination. This mechanism is called pivoting. In the computation of a Gröbner basis, the place of pivots is taken by the coefficients at the leading terms which are used for S-polynomials and reduction. Once a term order is specified, the leading term and hence the leading coefficient of any polynomial are uniquely defined.

Hence the elimination algorithm leaves no freedom for an alternate selection of pivots which could take into account the magnitude of the coefficients. This limitation makes the customary Gröbner basis algorithm numerically unstable, which is aggravated by its potential computational complexity. Thus, it can not generally be executed in floating-point arithmetic, which would be highly desirable in many practical situtations.

But there is another, mathematically more striking drawback, concerning not the course but the result of the computation: It has been pointed out in [Ste97] that the totally reduced Gröbner basis of a polynomial system may depend discontinuously on the system: A slight change in the coefficients of the system may change the *structure* of the Gröbner basis and the associated normal set. As the coefficients of the system approach a discontinuous situation, some coefficients of the Gröbner basis polynomials diverge to infinity. Hence the Gröbner basis may be a very ill-conditioned representation of the ideal of the system even if the system is perfectly well-conditioned itself. In this situation, the Gröbner basis is naturally unsuitable for whatever subsequent numerical computation.[1]

The potential occurrence of such artificial discontinuities introduced in Gröbner bases by the strict adherence to a term order makes it hard to design and develop a general-purpose black-box floating-point software for solving polynomial systems with coefficients of limited accuracy via Gröbner bases. The analysis of the origin of these discontinuities and a concept for a *stabilized Gröbner basis algorithm* which permits a safe use of floating-point arithmetic by locally overriding the demands of a term order in favor of numerical stability have been developed by Hans Stetter [Ste97]. This is the foundation for the research presented in this thesis: We have developed and implemented a variant of such a stabilized Gröbner basis algorithm.

To base such an algorithm on a firm theoretical basis, it has been necessary to develop and analyze a variation of the theory of Gröbner bases which provides a possibility to override the demands of a term order if necessary to maintain the numerical stability. The result of such a stabilized Gröbner basis algorithm applied to a polynomial system is generally not a Gröbner basis in the usual computer algebra sense but a so-called *extended Gröbner basis* (Definition 4.1 in [Ste97]), a slightly perturbed genuine Gröbner basis of a system in close vicinity of the input system. This is an appropriate representation of the ideal of the system which avoids the artificial representation discontinuities possibly introduced by a genuine Gröbner basis.

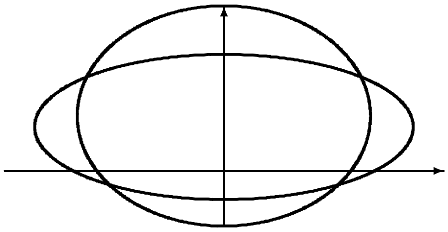
[1] Such a Gröbner basis jeopardizes the reliability and accuracy of virtually any computed result due to a high sensitivity to the parameters and, in particular, to a small uncertainty in the coefficients of the polynomial system generally known with limited meaningful accuracy in a numerical setting.

In the next sections we present a realistic demonstrative example of an artificial discontinuity, the analysis of the origin of such discontinuities and the principal idea of the stabilized Gröbner basis algorithm according to Hans Stetter, an overview of the main ideas, results and the structure of this thesis; we also list some questions for future research.

1.1 Illustration of an artificial discontinuity

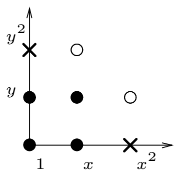
The following simple but nontrivial example has recently become a "classical" illustration of a situation when a Gröbner basis introduces artificial discontinuities due to the choice of a term order. Similar examples have previously been used by H. Stetter, W. Windsteiger and others. We adduce the example in the "purest possible" and, at the same time, particularly simple form presented in [Mou99].

Consider the following system of polynomial equations in two variables x and y over the field $\mathbb{C}$ of complex numbers



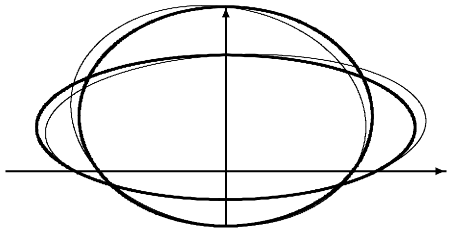
$$\begin{cases} p_1 := ax^2 + by^2 + \ell_1(x, y) = 0 \\ p_2 := cx^2 + dy^2 + \ell_2(x, y) = 0 \end{cases} \tag{1}$$

with $a, b, c, d \in \mathbb{C}$ such that $ad - bc \neq 0$ and linear forms ℓ_1, ℓ_2. Fix the total degree lexicographical term order induced by $y < x$. The totally reduced Gröbner basis of (1) consists of two polynomials with the leading terms x^2, y^2 and the associated normal set is



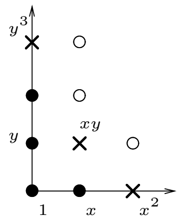
$$N = \{1, y, x, xy\}.$$

Now consider a perturbation



$$\begin{cases} p_1 + \varepsilon_1 xy = 0 \\ p_2 + \varepsilon_2 xy = 0 \end{cases} \tag{2}$$

of (1) with small parameters $\varepsilon_1, \varepsilon_2 \in \mathbb{C}$. Geometrically the zero sets of (1) and (2) are very similar. However, the totally reduced Gröbner basis of (2) consists of three polynomials with the leading terms xy, x^2, y^3 and the associated normal set is



$$\widetilde{N} = \{1, y, x, y^2\}.$$

Furthermore some terms of the totally reduced normalized Gröbner basis of (2) have large coefficients as the computation involves divisions by coefficients which are proportional to negative powers of ε_1, ε_2.

1.2 Origin of artificial discontinuities

Naturally, the origin of phenomena like in the preceding example, has to be understood in order to use Gröbner bases safely in a numerical context. The following is a simplified brief summary of section 2 in [Ste97] where the analysis of the origin as well as a possible remedy have been established in a fairly general context.

Consider a multivariate zero-dimensional polynomial system

$$F = \{f_1, \ldots, f_n\} \subseteq \mathbb{C}[X]$$

over the field $\mathbb{C}$ of complex numbers in the indeterminates $X = \{x_1, \ldots, x_n\}$. Denote the terms of $\mathbb{C}[X]$ by $\mathbb{T} = X^*$. The problem of finding solutions of F is equivalent to finding the points of the variety of the ideal $I = \langle F \rangle$ generated by F in $\mathbb{C}[X]$ which consists of finitely many isolated points $Z = \{z_1, \ldots, z_m\}$.

It is well-known that in this case the quotient algebra $Q{=}\mathbb{C}[X]/I$ is a finite dimensional $\mathbb{C}$-vector space and the dimension $\dim_{\mathbb{C}}(Q)$ coincides with the number of zeros Z counting their multiplicities in the well-known sense [MMM95], [MS]. The number $m{=}\dim_{\mathbb{C}}(Q)$ is a characteristic property of the ideal I which remains invariant under a small change in the coefficients of F in the complete intersection case which is assumed.

Let G be the totally reduced Gröbner basis of F with respect to some fixed term order $<$. The normal set N associated with G is the set of all terms $\mathbb{T}$ not divisible by the leading term of any polynomial of G. It is well-known that N is a closed set in the usual sense that $t' \,|\, t$ implies $t'{\in}N$ for any $t{\in}N$ and $t'{\in}\mathbb{T}$. It is well-known and crucial for the subsequent analysis that N represents a $\mathbb{C}$-linear space basis of the quotient Q which means that, for any polynomial $f{\in}\mathbb{C}[X]$, there exists the unique representation

$$f \equiv \sum_{t\in N} \alpha_t(f){\cdot}t \mod I \quad \text{with} \quad \alpha_t(f){\in}\mathbb{C} \quad \text{for all } t{\in}N. \tag{†}$$

A possible discontinuity of G on small changes in F is actually related to the discontinuities of the Gram matrix for the multivariate interpolation problem on the zero set Z with the interpolation space spanned by the normal set N uniquely determined by the ideal I and the term order $<$. Assume all solutions Z of F are well-conditioned, tolerant to small changes in the coefficients of F. Assume for simplicity that all the m solutions of F are distinct and sufficiently separated.[2] In this case the normal set consists of m terms $t_1, \ldots, t_m$. Consider the matrix

$$N(Z) = \Big[t_i(z_j) \;\Big|\; 1{\leq}i,j{\leq}m\Big] = \begin{bmatrix} t_1(z_1) & \cdots & t_m(z_1) \\ \vdots & & \vdots \\ t_1(z_m) & \cdots & t_m(z_m) \end{bmatrix}$$

of evaluations of the normal set elements t_i at the solutions z_j, $1{\leq}i,j{\leq}m$. It is easy to see that

$$f(Z){\equiv}0 \quad \text{if and only if} \quad f{\in}I \tag{1}$$

and

$$\det\big(N(Z)\big) \neq 0 \tag{2}$$

[2]The general case with multiple zeros is not conceptually different but less intuitive and technically more complicated. See [Ste97] and [MS] for details. The case of *near-confluent* zeros is naturally considered as multiple zeros under a small perturbation, see [Ste96a]. Technically, the situation of a multivariate polynomial system with near-confluent zeros is quite complicated both for analysis and implementation.

by Hilbert's Nullstellensatz, the assumption that all solutions of F are distinct and the fact that the normal set N represents a $\mathbb{C}$-linear space basis of the quotient Q. But then, according to (1), for any $f\in\mathbb{C}[X]$, the uniquely defined, according to (†), coefficients $\alpha_{t_1}(f),\ldots,\alpha_{t_m}(f)$ can be seen as the uniquely defined, according to (2), solutions of the linear system

$$\begin{bmatrix} f(z_1) \\ \vdots \\ f(z_m) \end{bmatrix} = N(Z)\cdot\begin{bmatrix} \alpha_{t_1}(f) \\ \vdots \\ \alpha_{t_m}(f) \end{bmatrix} \quad \text{in} \quad \alpha_{t_1}(f),\ldots,\alpha_{t_m}(f). \tag{3}$$

It is well-known that the set of leading terms of the totally reduced Gröbner basis G is precisely the so-called *corner set* $C(N)$ of N, i.e. the set of all terms $t\in\mathbb{T}$ such that $t\notin N$ and $x\,|\,t$ implies $t/x\in N$ for any $x\in X$; $C(N)$ is uniquely defined by N. Hence, by the well-known uniqueness of the totally reduced and normalized Gröbner basis, the Gröbner basis G is uniquely (up to scaling) determined from N and Z by the collection of the systems (3) for $f{=}t\in C(N)$:

$$\begin{bmatrix} t(z_1) \\ \vdots \\ t(z_m) \end{bmatrix} = N(Z)\cdot\begin{bmatrix} \alpha_{t_1}(t) \\ \vdots \\ \alpha_{t_m}(t) \end{bmatrix} \tag{4}$$

and

$$G = \Big\{t-\sum_{i=1}^{m}\alpha_{t_i}(t)\cdot t_i \;\big|\; t\in C(N)\Big\}.$$

To view G as the solution of the multivariate polynomial interpolation problem (4) is the key to understand the possible ill-conditioning of the representation G for the ideal I of F, a high sensitivity of G to small changes in the coefficients of F. Indeed, when we assume that the original polynomial system F is well-conditioned w.r.t. the determination of its zeros, the conditioning of G is essentially the conditioning of the linear systems (4) which is essentially determined by the condition number

$$\kappa := \kappa\big(N(Z)\big) = \parallel N(Z) \parallel \cdot \parallel N(Z)^{-1} \parallel,$$

as it is well-known from numerical linear algebra. The condition number κ becomes large if the row vectors $\big((t_1(z),\ldots,t_m(z)\big)$, $z\in Z$, of $N(Z)$ are *almost linearly dependent*; strict linear dependence can not occur due to (2).
The polynomial equation

$$\det\big(N(Z)\big) = 0 \quad \text{in} \quad z_1,\ldots,z_m \tag{5}$$

defines an $(n{\times}m{-}1)$-dimensional *singularity manifold* $S(N)$ of N in the affine space $(\mathbb{C}^n)^m$. It is clear that $\det\bigl(N(Z)\bigr)$ is a continuous function of Z; hence, generally speaking, the condition number κ is small if the zero set Z of F is sufficiently far away from $S(N)$ while it tends to infinity for Z in the vicinity of $S(N)$. In the latter case G is an extremely ill-conditioned representation of the ideal I of F and hence not suitable for numerical computation.

The preceding analysis has shown that the problem of potential ill-conditioning of G arises when the zeros Z of F are close to the singularity manifold $S(N)$. For a given position of the zeros of F, the crucial quantity of κ depends solely on the *choice* of N but not on the polynomials F! However N is *traditionally* not "chosen" but uniquely defined as the normal set of a Gröbner basis of F for a chosen term order. Thus, for one and the same polynomial system F, the Gröbner basis G may provide a perfectly stable or extremely ill-conditioned representation of the ideal of F depending on the chosen term order. Unfortunately the singularity manifold $S(N)$ is never empty for $n, m{>}1$ as it is clear from (5) since the Gram matrix $N(Z)$ always has singularities, contrary to the univariate Vandermonde matrix.

The situation when Z is near the singularity manifold $S(N)$ is to be analyzed more carefully. It is precisely the "numerically catastrophic" situation which calls for a remedy. Consider a system $\widetilde{F}$ in a close neighborhood of F i.e. $\widetilde{F}$ is F with slightly different coefficients (a precise metric and a neighborhood definition are irrelevant for the present analysis). Denote the ideal generated by $\widetilde{F}$ in $\mathbb{C}[X]$ by $\widetilde{I}{=}\langle\widetilde{F}\,\rangle$, the zeros of $\widetilde{F}$ by $\widetilde{Z}$, $\widetilde{G}$ is the totally reduced Gröbner basis of $\widetilde{F}$ w.r.t. $<$, $\widetilde{N}$ is the normal set associated with $\widetilde{G}$. Note that $\widetilde{F}$ has m solutions and all the solutions of $\widetilde{F}$ are distinct, sufficiently separated and well-conditioned due to the respective assumptions on the solutions of F. Hence all the proceeding analysis is applicable as well to $\widetilde{F}$. If the zero set Z of F is close to the singularity manifold $S(N)$, so that $\kappa\bigl(N(Z)\bigr)$ is very large, then it is quite clear that we can choose $\widetilde{F}$ such that $\widetilde{Z}{\in}S(N)$. In this case N is *linearly dependent* modulo $\widetilde{I}$, cf. (1), (2). Hence N can not be the normal set associated with $\widetilde{G}$ and therefore $\widetilde{N}$ differs from N. Note once again that N is *traditionally* uniquely defined as the normal set associated with G and thus, while $\widetilde{N}$ represents a basis in the quotient $Q{=}\mathbb{C}[X]/\langle F\rangle$ if the considered neighborhood of F is sufficiently small and Z is not generally close to the singularity manifold $S(\widetilde{N})$, cf. (5), this does not help at all staying in the traditional frame of Gröbner bases. However it points a way to a remedy: If we use $\widetilde{N}$ for all systems in the neighborhood of F (including F!) then we will have a stable situation as long as Z is sufficiently far from $S(\widetilde{N})$.

Naturally, if Z is near the intersection of the singularity manifolds of two or more normal sets then we may have to look further in order to find a suitable $\widetilde{N}$. Intuitively speaking the singularity manifolds are generally related to some *degeneracy* in the position of zeros. According to the above discussion, in a situation with some near-degeneracy of the zeros, a numerically reliable *stabilized* modification of the Gröbner basis algorithm should lead to the normal set appropriate for the *most degenerate nearby situation.*

1.3 Stabilized computation of Gröbner bases

The principal idea for a stabilized Gröbner basis computation formulated at the end of the previous section suggests a general recipe how such algorithm can be designed (cf. section 4 in [Ste97]): Assume we totally reduce but do not normalize the intermediate polynomials during the Gröbner basis computation. Assume further that, through S-polynomial formation or reduction, a polynomial has been created with a *very small leading coefficient* and some other coefficients with a modulus of order 1. If this polynomial is combined, either in S-polynomial formation or reduction, with another polynomial whose matching coefficient is of order 1, it must be *multiplied by a very large factor* and the resulting polynomial will have very large coefficients.

It is thus the appearance of leading terms with relatively very small coefficients which signals attention. Such leading terms *must not be used for S-polynomial formation and reduction.* Indeed, a small perturbation of the original system may cause this coefficients to vanish at this point and hence *discretely* change the further course of computation. According to the considerations in the previous section, it is crucial in this case to proceed along the path associated with the *vanishing* coefficient. Hence such leading terms with relatively very small coefficients must formally be deprived of their function as leading terms; this function is taken over by the next term in the term order which has a coefficient of order 1. The "relegated" terms are collected in a separate "$O(\varepsilon)$"-part of the polynomial. Thus the Gröbner basis computation moves along the same path which it would have taken if the zero set of the system had been exactly on the singularity manifold $S(N)$ of the normal set N associated with a genuine Gröbner basis of the original system and the $O(\varepsilon)$-terms would not have been present. We arrive at a set of polynomials which have the same formal leading terms as the elements of a genuine Gröbner basis of a slightly perturbed original system with the zero set strictly on the singularity manifold $S(N)$ and hence with a different associated normal set.

The totally inter-reduced result of a stabilized Gröbner basis algorithm is generally not a Gröbner basis in the usual computer algebra sense but a so-called *extended Gröbner basis*, a slightly perturbed genuine Gröbner basis of a slightly perturbed original system:

Definition 1.3.1 (cf. Definition 4.1 **in [Ste97]).** Let $F \subseteq \mathbb{C}[X]$ be a multivariate polynomial system over the field $\mathbb{C}$ of complex numbers in the indeterminates X. An extended Gröbner basis $\widehat{G} \subseteq \mathbb{C}[X]$ of F w.r.t. a specified term order $<$ on $\mathbb{T}=X^*$ is

$$\widehat{G} = \Big\{ g + \sum_{t \in \widetilde{N}} \beta_t(g) \cdot t \;\big|\; g \in \widetilde{G} \Big\} \quad \text{with} \quad \beta_t(g) \in \mathbb{C} \quad \text{for} \quad t \in \widetilde{N},\ g \in \widetilde{G},$$

where $\widetilde{G}$ is the genuine totally reduced Gröbner basis of a slightly perturbed F, $\widetilde{N}$ is the normal set associated with $\widetilde{G}$ and all coefficients $\beta_t(g)$, $t \in \widetilde{N}$, $g \in \widetilde{G}$, are sufficiently small.

Despite the fact that an extended Gröbner basis $\widehat{G}$ of a polynomial system $F \subseteq \mathbb{C}[X]$ w.r.t. a specified term order $<$ is generally not a Gröbner basis of F w.r.t. $<$ in the usual computer algebra sense, it *looks like* a Gröbner basis and it is quite straightforward to establish the counterpart of a *normal form* modulo $\widehat{G}$ uniquely defined for any polynomial in $\mathbb{C}[X]$ (cf. Theorems 4.1 and 4.2 in [Ste97]). Therefore it can be used in a wide range of well-known methods like in the eigenvector method for finding the solutions of F provided that $\widehat{G}$ is a basis of the ideal $\langle F \rangle$ in $\mathbb{C}[X]$

1.4 Our main contributions

The principal idea of a stabilized Gröbner basis computation sketched in the preceding section works well for sufficiently simple cases and generates the expected results. However, it has not been clear so far how it can be turned into a general stabilized Gröbner basis algorithm which can be executed in floating-point arithmetic without serious harm for the result of computation. The theoretical, technical and algorithmic details of a possible approach to establish such an algorithm as well as its implementation are the content and the main results of this thesis.

We have developed a variation of the theory of Gröbner bases which provides the *computer algebra foundation* for extended Gröbner bases and a systematic way to compute such bases in general. As a matter of fact our variation is again a purely symbolic theory without any regard towards numerical considerations. However, the theory provides convenient *"placeholders"* which are utilized in our implementation to achieve a numerically stable variation of the Gröbner basis algorithm.
In more detail the main results of the thesis can be summarized as follows.

- Variation of the theory of Gröbner bases.
  - An extension of the term order concept.
  - The analogue of Buchberger's S-polynomial criterion.
  - The analogue of Buchberger algorithm for computing Gröbner bases.
  - An optimized Gröbner basis algorithm which uses Buchberger's first and second criteria for detecting redundant critical elements adopted for our setup.
  - Basic relations between the Gröbner bases in the usual sense and our Gröbner bases.
- Implementation of the stabilized Gröbner basis algorithm executed in floating-point arithmetic.
  - The utilization of the variation of the theory of Gröbner bases to establish the following two versions of the algorithm.
    - **Partially stabilized Gröbner basis algorithm:** Computes a genuine Gröbner basis but with a stabilized reduction process.
    - **Stabilized Gröbner basis algorithm:** Computes a possibly extended Gröbner basis, with a strongly stabilized reduction process.
  - A thorough reconsideration of the ideas of a rearranged Buchberger algorithm introduced by J. C. Faugere in his algorithm F_4 for customary Gröbner bases and exact arithmetic. The results of this reconsideration are implemented in our algorithm which differs from the original F_4 in the following important aspects.
    - An essentially recursive version of symbolic reduction.
    - Utilization of complete pivoting in the linear algebra part of reduction.
  - Highly sophisticated and carefully tuned selection strategies for critical elements and reduction.

1.5 Open questions for future research

Naturally the application of Gröbner bases to polynomial systems solving in a numerical context raises a great number of questions and related problems.

We restrict the following summary to the questions which "cry" for being analyzed and understood in conjunction with our work but have remained open due to the natural time and space restrictions.

- Relation between extended Gröbner bases and our variation of Gröbner bases from the *analytical* point of view.
- Verification and stability analysis of the implemented algorithm in the context of numerical polynomial algebra.
- Analysis of conditioning in the eigenvector method, investigation of the case of multiple zeros.
- Solving polynomial systems of positive dimension.

1.6 Structure of the thesis

- Chapter 2 contains a completely developed variation of the theory of Gröbner bases which provides a foundation for our implementation.
  - We begin with basic notions mainly to fix the notation (section 2.1 on page 16) and also provide a brief overview of the customary theory of Gröbner bases (section 2.2 on page 18).
  - Then, in section 2.3 on page 25, we introduce a so-called algebra of extended polynomials $\mathcal{P}(e)$, transfer the basic notions and techniques essential in the context of Gröbner bases to $\mathcal{P}(e)$ and establish the theory of Gröbner bases in $\mathcal{P}(e)$ by appropriate modification of the notion of a term order, normal forms, the reduction algorithm, the critical elements and the completion procedure.
  - Section 2.4 on page 89 describes the instance $\mathcal{P}_e$ of $\mathcal{P}(e)$ utilized in our implementation for the *partially stabilized Gröbner basis algorithm* for computing genuine Gröbner bases with a stabilized reduction process.
  - The main result of section 2.5 on page 113 is the translation principle of Gröbner bases in $\mathcal{P}_e$ which allows to readily transform a Gröbner basis in $\mathcal{P}_e$ into a customary Gröbner basis under appropriate circumstances. The translation principle establishes a relation between the Gröbner bases in $\mathcal{P}_e$ and the customary Gröbner bases from the algebraic point of view. Section 2.5 also contains a partially developed theory of Gröbner bases in a ring over a principle ideal ring used as an intermediate tool to establish the translation principle.

  - Section 2.6 on page 167 deals with ${}_e\mathcal{P}$, the instance of an algebra $\mathcal{P}(e)$ utilized in our implementation for the *stabilized Gröbner basis algorithm* for computing extended Gröbner bases with a strongly stabilized reduction algorithm, the main goal of our work. The bridge between the Gröbner bases in ${}_e\mathcal{P}$ and the customary Gröbner bases is provided by the respective version of the translation principle called *the symbolic perturbation of Gröbner bases in* ${}_e\mathcal{P}$ to readily transform a Gröbner basis in ${}_e\mathcal{P}$ into a customary Gröbner basis under appropriate circumstances.

- Chapter 3 describes the stabilized Gröbner basis algorithm based on the theory developed chapter 2.

  - At first we give a semi-formal overview of the algorithm and of the ideas how the theory in chapter 2 is turned into the foundation for our algorithm.
  - Section 3.1 on page 205 describes the preprocessing step of the algorithm where the input polynomial data is translated to the algebra $\mathcal{P}(e)$ for subsequent computation based on the theory of Gröbner bases in $\mathcal{P}(e)$ developed in chapter 2.
  - Section 3.2 describes a layout of the algorithm. It is intended to provide an overview and give the basic idea how the parts of the algorithm work together.
  - Section 3.3 provides a detailed description of how the sets of critical elements are managed in the algorithm.
  - Section 3.4 on page 223 describes the reduction algorithm utilized in our implementation. It is a very specific algorithm where the reduction is split into two phases called the symbolic reduction and the row echelon elimination. This version of reduction originated from ideas developed and implemented by J. C. Faugere in his F_4 algorithm.

- Chapter 4 presents computational experiments performed with our implementation.

  - In section 4.1 on page 271 we provide a brief description of the eigenvector method for solving zero-dimensional systems of polynomial equations.
  - Section 4.2 on page 274 describes the instantiation of the eigenvector method used as a backend of our implementation for the subsequent examples.

– Section 4.3 on page 283 contains a collection of numerical examples computed with our implementation. For each example, we provide a detailed analysis, with particular emphases for the different examples.

1.7 Acknowledgements

First of all I like to thank my supervisor Hans Stetter, without whose enthusiasm and deep insights into the numerical aspects of polynomial computer algebra, I would never have achieved all the novel results in my thesis.

I am especially grateful to Bernard Mourrain for his helpful suggestions, comments and for his remarkable idea towards an alternative to normal form algorithms which gave me a deep insight into the research topic.

Furthermore I would like to express my gratitude to my second advisor Franz Winkler for his dedicated support and fruitful discussions.

Moreover, I wish to thank the Computer Algebra group at RISC, especially Ralf Hemmecke, for all the critical comments and suggestions which have improved the thesis.

This work has been partially funded by the grant P 13266−MAT and the SFB grant F1304 of the Austrian "Fonds zur Förderung der wissenschaftlichen Forschung" (FWF).

Chapter 2

The theory of Gröbner bases in the algebra of extended polynomials

2.1 Notation and conventions

This section presents some well-known basic notions mainly to fix the notation and the language. Let R be an arbitrary commutative ring with unity i.e. R is an abelian additive group with the multiplicative monoid structure and the usual distributivity law. Denote $R^*{=}R\backslash\{0\}$.

Definition 2.1.1. Let $a{\in}R^*$, $b{\in}R$. Then $a\,|\,b$ – a **divides** b (in R) if and only if there exists an element denoted $\frac{b}{a}{\in}R$ such that $b = \frac{b}{a}{\cdot}a$. By convention, $a\,|\,0$.

Definition 2.1.2. The **divisibility relation** is the set of all pairs $(a, b){\in}R^{\times 2}$ such that $a\,|\,b$.

The divisibility relation has the following properties:

$1\,|\,a$ and $b\,|\,b$ for any $a{\in}R$ and $b{\in}R^*$,

transitivity: $a\,|\,b$ and $b\,|\,c$ follows $a\,|\,c$ for any $a, b{\in}R^*$ and $c{\in}R$,

$a\,|\,b$ follows $a{\cdot}c\,|\,b{\cdot}c$ for any $a, c{\in}R^*$ such that $a{\cdot}c{\neq}0$ and $b{\in}R$,

$a\,|\,b$ and $a\,|\,c$ follows $a\,|\,(b{\cdot}p{+}c{\cdot}q)$ for any $a{\in}R^*$ and $b, c, p, q{\in}R$.

Definition 2.1.3. An element $a{\in}R$ is called a **unit** (in R) if it has a multiplicative inverse in R i.e. there exists an element $b{\in}R$ such that $ab = 1$.

The (multiplicatively closed) set of all units of R is denoted $\mathrm{U}(R)$. Given a subring $A \lhd R$, the (multiplicatively closed) set $A \cap \mathrm{U}(R)$ of all elements of A which have the inverse in R is denoted $\mathrm{U}(A)_R$. Note that $\mathrm{U}(A) \subseteq \mathrm{U}(A)_R$ and the inclusion can be strict like in a classical example of integers embedded into the rational field.

Definition 2.1.4. An element $a \in R^*$ is called a **zero-divisor** (in R) if and only if there exists an element $b \in R^*$ such that $a \cdot b = 0$.

It is well-known (see e.g. [BW93] lemma 1.18 on page 21) that a zero divisor can not be a unit unless $R = \{0\}$.

Definition 2.1.5. A subset $I \subseteq R$ is called an **ideal** if and only if I is an additive subgroup and stable under multiplication with an arbitrary element of R:

$$0 \in I, \;\; a+b \in I, \;\; a \cdot c \in I \;\; \text{for any} \;\; a, b \in I \;\; \text{and} \;\; c \in R.$$

Definition 2.1.6. An ideal **generated by** $G \subseteq R \setminus \{0\}$ is the smallest ideal (denoted $\langle G \rangle_R$) in R which contains G. It is well-known that

$$\langle G \rangle_R = \left\{ \sum_{i=1}^{k} p_i \cdot g_i \;\middle|\; k \in \mathbb{N}_o \text{ and } p_i \in R, \;\; g_i \in G \text{ for each } 1 \leq i \leq k \right\}.$$

Definition 2.1.7.

- An ideal in R which can be generated by a single element is called a **principal ideal**.
- A ring R is a **principal ideal ring** if and only if every ideal in R is principal.

Definition 2.1.8. A ring R is called is a **domain** if and only if it does not have zero-divisors i.e. $a \cdot b = 0$ follows $a = 0$ or $b = 0$ for any $a, b \in R$.

Definition 2.1.9. A principal ideal ring is called a **principal ideal domain** if and only if it is a domain.

Definition 2.1.10. Let Q be a ring contained in R. Then R is called a Q-algebra. The important example if a Q-algebra is a polynomial ring $Q[X]$ over Q in finitely many indeterminates (variables, symbols) X algebraically independent over Q. Throughout this text the phrases "polynomial ring" and "polynomial algebra" are used interchangeably.

2.2 The classical theory of Gröbner bases

This section provides a brief reference material about the "classical" Gröbner bases theory for polynomial rings over a filed. Let $\Bbbk$ be a field, $\Bbbk^*{=}\Bbbk\backslash\{0\}$, $X = \{x_1, \ldots, x_n\}$ is a finite set of variables algebraically independent over $\Bbbk$, $\mathbb{T}$ is the set of terms in X and $\mathcal{P}{=}\Bbbk[X]$ is the algebra of polynomials (finite $\Bbbk$-linear combinations of terms) in X over $\Bbbk$. By construction of $\Bbbk[X]$, $\mathbb{T}$ is a $\Bbbk$-linear vector space basis of $\Bbbk[X]$ and hence every polynomial $f{\in}\mathcal{P}$ can be written in the unique way as a finite linear combination of terms $\mathbb{T}$ with non-zero coefficients $\Bbbk^*$, the finite set of terms involved in the latter unique representation is usually called the support of f denoted $\operatorname{supp}(f)$. For any set $F{\subseteq}\mathcal{P}$ the support of F is a commonly used shortcut for $\operatorname{supp}(F) = \bigcup\limits_{f \in F} \operatorname{supp}(f)$. For any term t the degree of t is $\deg(t){=}\sum\limits_{i=1}^{n} d_i$ where $d_1, d_2 \ldots, d_n{\in}\mathbb{N}_o$ are uniquely defined such that $t{=}x_1^{d_1}{\cdot}x_2^{d_2}{\cdots}x_n^{d_n}$. For any polynomial f the (total) degree of f is $\deg(f){=}\max\bigl(\deg(t) \mid t{\in} \operatorname{supp}(f)\bigr)$ if $f{\neq}0$ and undefined for the 0 polynomial.

The Gröbner bases theory is founded on the following four base ideas.

1. The term order as a special total order on terms.
2. The reduction algorithm.
3. The notion of a Gröbner basis as a special generating set of a polynomial ideal.
4. The completion procedure based on the notion of an S-polynomial.

2.2.1 The term order

Definition 2.2.1. A total (linear) order $<$ on the set of terms $\mathbb{T}$ is called a **term order** (also referred as an **admissible order**) if and only if it satisfies the following two additional properties.

1. *Comparability with multiplication of terms:* $t_1{<}t_2$ follows $t{\cdot}t_1{<}t{\cdot}t_2$ for any $t_1, t_2, t{\in}\mathbb{T}$.
2. *Global order property:* $1{\leq}t$ for any $t{\in}\mathbb{T}$.

Below are some examples of the most frequently used (in the literature and the computer algebra systems) term orders. See [MK00] (section 1.4), [Win96] (page 180) for more examples or [Rob85], [Rob86] for a complete characterization of term orders.

Example 2.2.1. The commonly used term orders are defined as follows for arbitrary terms $t_1{=}x_1^{i_1}{\cdot}x_2^{i_2}{\cdots}x_n^{i_n}$ and $t_2{=}x_1^{j_1}{\cdot}x_2^{j_2}{\cdots}x_n^{j_n}$.

The lexicographic order:

$$t_1 \underset{lex}{<} t_2 \text{ if and only if there exists } 1{\le}k{\le}n \text{ such that} \quad i_\ell{=}j_\ell \text{ for all } 1{\le}\ell{<}k \text{ and } i_k{<}j_k.$$

The total degree order:

$$t_1 \underset{tdeg}{<} t_2 \text{ if and only if} \quad \begin{array}{ll} \deg(t_1){<}\deg(t_2) & \text{or} \\ \deg(t_1){=}\deg(t_2) & \text{and } \; t_1 \underset{lex}{<} t_2. \end{array}$$

The inverse lexicographic order:

$$t_1 \underset{ilex}{<} t_2 \text{ if and only if there exists } 1{\le}k{\le}n \text{ such that} \quad i_\ell{=}j_\ell \text{ for all } k{<}\ell{\le}n \text{ and } i_k{<}j_k.$$

The inverse total degree order:

$$t_1 \underset{ideg}{<} t_2 \text{ if and only if} \quad \begin{array}{ll} \deg(t_1){<}\deg(t_2) & \text{or} \\ \deg(t_1){=}\deg(t_2) & \text{and } \; t_1 \underset{ilex}{<} t_2. \end{array}$$

The reverse total degree order:

$$t_1 \underset{rdeg}{<} t_2 \text{ if and only if} \quad \begin{array}{ll} \deg(t_1){<}\deg(t_2) & \text{or} \\ \deg(t_1){=}\deg(t_2) & \text{and } \; t_1 \underset{rlex}{<} t_2 \end{array}$$

where $\underset{rlex}{<}$ is the reverse lexicographical order which is a total order on $\mathbb{T}$ defined as follows:

$$t_1 \underset{rlex}{<} t_2 \text{ if and only if there exists } 1{\le}k{\le}n \text{ such that} \quad i_\ell{=}j_\ell \text{ for all } k{<}\ell{\le}n \text{ and } i_k{>}j_k.$$

Note that $\underset{rlex}{<}$ is not a term order because 1 is not the smallest term w.r.t. $\underset{rlex}{<}$, in fact it is easy to see that $t \underset{rlex}{<} 1$ for any $1{\neq}t{\in}\mathbb{T}$.

The treatment of the special properties of particular term orders is far beyond the scope of this Gröbner bases theory reference section. Therefore the symbol $<$ denotes an arbitrary fixed term order throughout the rest of the section.

Definition 2.2.2. The **leading term**, the **leading coefficient** and the **leading monomial** of a polynomial $f{\in}\mathcal{P}$ denoted $\ell\mathrm{t}(f)$, $\ell\mathrm{c}(f)$ and $\ell\mathrm{m}(f)$ are defined as the $<$-maximal element of the support of f, the coefficient at the leading term of f and the product $\ell\mathrm{c}(f){\cdot}\ell\mathrm{t}(f)$ respectively if $f{\neq}0$ and undefined for the 0 polynomial. For an arbitrary set $F{\subseteq}\mathcal{P}$ the leading terms of F denoted $\ell\mathrm{t}(F)$ is a conventional shortcut for the set $\{\ell\mathrm{t}(f) \mid f{\in}F, f{\neq}0\}$.

As a matter of fact a term order is always a well-ordering of terms. There are several well-known equivalent conditions for an arbitrary total order on a set to be a well-ordering of the set. The Gröbner bases theory mainly uses the following form of the well-ordering statement.

Lemma 2.2.1. Any strictly descending sequence of terms is finite.

Proof. The usual proof is based on the Hilbert basis theorem in conjunction with the property

$$\text{if } t_1|t_2 \text{ then } t_1{\leq}t_2 \text{ for any } t_1, t_2{\in}\mathbb{T}$$

which is an easy consequence of both additional axioms for a total order on terms to be a term order. The details can be found in [AL94] (theorem 1.4.6 on page 21), [BW93] (theorem 5.5 on page 190), [Buc98] (proposition on page 13), [MK00] (proposition 1.4.18 on page 55), [Win96] (lemma 8.2.3 on page 181). □

2.2.2 The reduction algorithm

The reduction algorithm generalizes two famous algorithms: the univariate polynomials division and the gaussian elimination of multivariate linear polynomials.

Definition 2.2.3. A polynomial $f{\in}\mathcal{P}$ is called **top-reducible modulo** a set $F{\subseteq}\mathcal{P}{\setminus}\{0\}$ if and only if $f{\neq}0$ and there exists $h{\in}F$ such that $\ell\mathrm{t}(h)|\ell\mathrm{t}(f)$; otherwise f is called **top-minimal** modulo F.

The following reduction algorithm `RED` takes arbitrary $f{\in}\mathcal{P}$, $F{\subseteq}\mathcal{P}{\setminus}\{0\}$ and sequentially transforms f to an equivalent modulo $\langle F\rangle_{\mathcal{P}}$ polynomial until a top-minimal modulo F polynomial is reached.

Algorithm 2.2.1 The reduction algorithm `RED` in $\mathcal{P}$

Require: $f{\in}\mathcal{P}$, $F{\subseteq}\mathcal{P}\backslash\{0\}$.
Ensure: $g{\in}\mathcal{P}$ is a top-minimal modulo F polynomial such that $f{-}g{\in}\langle F\rangle_{\mathcal{P}}$.
  $g := f$
  while $g{\neq}0$ and $\exists h{\in}F$ such that $\ell\mathrm{t}(h)|\ell\mathrm{t}(g)$ **do**
    choose $h{\in}F$ such that $\ell\mathrm{t}(h)|\ell\mathrm{t}(g)$
    $g := g - \frac{\ell\mathrm{m}(g)}{\ell\mathrm{m}(h)}{\cdot}h$
  end while

Lemma 2.2.2. For any $f{\in}\mathcal{P}$ and $F{\subseteq}\mathcal{P}\backslash\{0\}$ the algorithm `RED` computes a polynomial g such that g is top-minimal modulo F and $f{-}g{\in}\langle F\rangle_{\mathcal{P}}$.

Proof. The termination argument is given by lemma 2.2.1 with the strictly descending sequence of the leading terms of non-zero polynomial values sequentially taken by g in the flow of the while-loop execution. The condition $f{-}g{\in}\langle F\rangle_{\mathcal{P}}$ at the end of the algorithm is stipulated by the while-loop prerequisite $g{=}f$ together with the observation that every iteration of the while-loop body transforms g to an equivalent modulo $\langle F\rangle_{\mathcal{P}}$ polynomial. The details can be found in [AL94] (theorem 1.5.9 on page 30), [BW93] (proposition 5.22 on page 198), [Buc98] (proposition on page 15), [Buc83] (lemma 6.1 on page 189), [MK00] (theorem 1.6.4 on page 71). ☐

2.2.3 The Gröbner basis

Definition 2.2.4. A set $G{\subseteq}\mathcal{P}\backslash\{0\}$ is a **Gröbner basis** of an ideal $I{\lhd}\mathcal{P}$ if and only if $\big\langle\ell\mathrm{t}(I)\big\rangle_{\mathcal{P}}{=}\big\langle\ell\mathrm{t}(G)\big\rangle_{\mathcal{P}}$. By convention, a set G is a **Gröbner basis** (in its own) if and only if G is Gröbner basis of the ideal $\langle G\rangle_{\mathcal{P}}$ which it generates.

The word "basis" in the notion of a Gröbner basis is stipulated by the following fact.

Lemma 2.2.3. If $G{\subseteq}\mathcal{P}\backslash\{0\}$ is a **Gröbner basis** of an ideal $I{\lhd}\mathcal{P}$ then $I{=}\langle G\rangle_{\mathcal{P}}$.

Proof. See [AL94] (corollary 1.6.4 on page 33), [BW93] (the remark after theorem 5.41 on page 208), [Buc82] (the proof of the theorem on page 31). ☐

Note that according to a common practice in the theory of rings and, in particular, in polynomial algebra, an arbitrary generating set of an ideal is called a basis unlike the convention in linear algebra where a basis is always a minimal generating set of a vector space.

It is clear that given a polynomial ideal there exist a lot of Gröbner bases of the ideal even if the term order is fixed. For example the whole ideal without the zero element is a Gröbner basis of itself in view of definition 2.2.4 of a Gröbner basis. The importance of the theory of Gröbner bases as a tool for constructive commutative algebra is substantiated by the fact that for any ideal in $\mathcal{P}$ there exists a finite Gröbner basis. This remarkable fact results in a commonly accepted convention which we also adhere that a Gröbner basis in $\mathcal{P}$ is always a finite set. The details can be found in [AL94] (corollary 1.6.5 on page 34), [BW93] (theorem 5.41 on page 208), [Buc82] (the theorem on page 31), [MK00] (proposition 2.4.3 on page 112).

The following theorem states the mostly used equivalent characterizations of a Gröbner basis. The list of statements is very far from being complete but should be sufficient to grasp the flavor of the Gröbner bases theory.

Theorem 2.2.1. Let $G \subseteq \mathcal{P} \backslash \{0\}$ and $I \lhd \mathcal{P}$. The following conditions are equivalent:

a) G is a Gröbner basis of I;

b) for any $f \in I \backslash \{0\}$ there exists $g \in G$ such that $\ell\mathrm{t}(g) | \ell\mathrm{t}(f)$;

c) for any $f \in \mathcal{P}$

$f \in I$ if and only if $\mathtt{RED}(f, G) = 0$ - f top-reduces to 0 modulo G by the reduction algorithm (despite the non-deterministic nature of RED);

d) for any $f \in \mathcal{P}$

$f \in I$ if and only if there exist polynomials $\{q_h \mid h \in G \in\} \subseteq \mathcal{P}$ such that $f = \sum_{h \in G} q_h \cdot h$ and $\ell\mathrm{t}(q_h) \cdot \ell\mathrm{t}(h) \leq \ell\mathrm{t}(f)$ for each $h \in G$ such that $q_h \neq 0$.

Proof. See [AL94] (theorem 1.6.2 on page 32), [BW93] (proposition 5.38 on page 207), [MK00] (theorem 2.4.1 on page 111), [Win96] (theorem 8.3.4 on page 187). □

A Gröbner basis can be considered as a far generalization of a triangular linear space basis (this analogy becomes particularly transparent in view of the lexicographical term order, see e.g. [CLO97] "The Elimination Theorem" on page 113 and "The Extension Theorem" on page 115). As a matter of fact a Gröbner basis is a convenient in certain respects (see [Buc98] for a collection of examples) representation of a system of polynomial equations and, in particular, a couple of methods including our approach for solving systems of polynomial equations incorporate the transformation of an arbitrary polynomial system to a Gröbner basis as a preprocessing step (see [CLO98] theorem

4.5 on page 54 and the subsequent discussion, [MS], [Ste93], [Ste96b]). On the other hand, the theory of Gröbner bases would be hardly more but an auxiliary tool in polynomial algebra if there was no Buchberger's algorithm for computing Gröbner bases as a clever combination of the reduction algorithm with the completion procedure based on the notion of an S-polynomial.

2.2.4 The completion procedure

Definition 2.2.5. An S**-polynomial** of $f, g \in \mathcal{P} \setminus \{0\}$ is the polynomial

$$\mathcal{S}\,(f,g) = \frac{\mathrm{lcm}\big(\ell\mathrm{t}(f), \ell\mathrm{t}(g)\big)}{\ell\mathrm{m}(f)} \cdot f - \frac{\mathrm{lcm}\big(\ell\mathrm{t}(f), \ell\mathrm{t}(g)\big)}{\ell\mathrm{m}(g)} \cdot g$$

where $\mathrm{lcm}\big(\ell\mathrm{t}(f), \ell\mathrm{t}(g)\big)$ is the *least common multiple* of the leading terms $\ell\mathrm{t}(f) = x_1^{i_1} \cdots x_n^{i_n}$ and $\ell\mathrm{t}(g) = x_1^{j_1} \cdots x_n^{j_n}$ defined as the term

$$\mathrm{lcm}\big(\ell\mathrm{t}(f), \ell\mathrm{t}(g)\big) = x_1^{\max(i_1, j_1)} \cdots x_n^{\max(i_n, j_n)}.$$

The notion of an S-polynomial is the key to one more characterization of a Gröbner basis which is the foundation of the algorithm for computing Gröbner bases.

Theorem 2.2.2 (Buchberger). A set $G \subseteq \mathcal{P} \setminus \{0\}$ is a Gröbner basis (of the ideal $\langle G \rangle_{\mathcal{P}}$) if and only if the S-polynomial of any pair of different elements of G top-reduces to 0 modulo G:

$$\mathtt{RED}\big(\mathcal{S}\,(f,g), G\big) = 0 \quad \text{for any} \quad f, g \in \mathcal{P} \quad \text{such that} \quad f \neq g.$$

Proof. See [AL94] (theorem 1.7.4 on page 40), [BW93] (theorem 5.48 on page 211), [Buc98] (the "main theorem of Gröbner basis theory" on page 19), [MK00] (corollary 2.5.3 on page 122), [Win96] (theorem 8.3.1 on page 183). □

The algorithm `GBasis` rests on theorem 2.2.2. It takes an arbitrary finite set of non-zero polynomials F and computes a Gröbner basis G of the ideal $\langle F \rangle_{\mathcal{P}}$ starting from $G{=}F$ by successive enlargement of the current basis with the result of top-reduction of the $S-$polynomial $\mathcal{S}\,(f,g)$ for each pair of different polynomials f and g from the current basis unless the S-polynomial is top-reduced to 0.

Lemma 2.2.4. For any finite set of non-zero polynomials F the algorithm `GBasis` computes a Gröbner basis $G \subseteq \mathcal{P} \setminus \{0\}$ of the ideal $\langle F \rangle_{\mathcal{P}}$

Algorithm 2.2.2 The algorithm `GBasis` for computing Gröbner bases in the algebra $\mathcal{P}$ (cf. [Buc83] algorithm 6.2 on page 192)

Require: $F \subseteq \mathcal{P} \setminus \{0\}$ - a set of non-zero polynomials.
Ensure: $G \subseteq \mathcal{P} \setminus \{0\}$ - a Gröbner basis of $\langle F \rangle_{\mathcal{P}(e)}$.

```
G := F
C := {(f, g) | f, g∈F, f≠g}
while C≠∅ do
  choose (f, g)∈C
  r := RED(S(f, g))
  if r≠0 then
    G := G ∪ {r}
    C := C ∪ {(h, r) | h∈G}
  end if
end while
```

Proof. See [AL94] (theorem 1.7.8 on page 42), [BW93] (theorem 5.53 on page 213), [Buc98] (the "correctness of the Gröbner basis algorithm" proposition on page 22), [Buc83] (page 192), [MK00] (theorem 2.5.5 on page 123). ☐

For the performance of virtually any software which uses the output Gröbner basis of any implementation based on the idea of the algorithm `GBasis` as well as due the common sense reasons it is desirable to have as few polynomials in the Gröbner basis as possible which leads to the notion of a minimal Gröbner basis as a minimal by inclusion of sets Gröbner basis of the same ideal.

Definition 2.2.6. A Gröbner basis $G \subseteq \mathcal{P} \setminus \{0\}$ is called a **minimal Gröbner basis** if and only if it is the minimal by inclusion of sets Gröbner basis of the ideal $\langle G \rangle_{\mathcal{P}}$.

The following lemma formulates the rule to detect redundant polynomials in an arbitrary Gröbner basis.

Lemma 2.2.5. If $G \subseteq \mathcal{P} \setminus \{0\}$ is Gröbner basis (of the ideal $\langle G \rangle_{\mathcal{P}}$) and $g \in G$ is such that $\ell\mathrm{t}(h) | \ell\mathrm{t}(g)$ for some $h \in G \setminus \{g\}$ then $G \setminus \{g\}$ is a Gröbner basis of the ideal $\langle G \rangle_{\mathcal{P}}$.

Proof. See [AL94] (lemma 1.8.2 on page 47), [Win96] (theorem 8.3.5 on page 187). ☐

The algorithm `MinGBasis` uses the statement of lemma 2.2.5 as a rule for successive elimination of redundant polynomials in a Gröbner basis. It can be utilized as a postprocessing step in the algorithm `GBasis`.

Algorithm 2.2.3 The algorithm `MinGBasis` for computing minimal Gröbner bases

Require: $G \subseteq \mathcal{P} \setminus \{0\}$ - a Gröbner basis
Ensure: G - a minimal Gröbner basis of the same ideal
  while $\exists g \in G$ such that $\ell\mathrm{t}(h) | \ell\mathrm{t}(g)$ for some $h \in G \setminus \{g\}$ **do**
    $G := G \setminus \{g\}$
  end while

Lemma 2.2.6. For each Gröbner basis $G \subseteq \mathcal{P} \setminus \{0\}$ the algorithm `MinGBasis` computes a minimal Gröbner basis of the ideal $\langle G \rangle_{\mathcal{P}}$.

Proof. See [AL94] (corollary 1.8.3 on page 47), [BW93] (proposition 5.56 on page 216). □

If we allow the reduction of an arbitrary monomial in the support of a polynomial (instead of the leading term only) then we can also consider a totally reduced Gröbner basis which is a minimal Gröbner basis $G \subseteq \Bbbk[X] \setminus \{0\}$ such that

$$\big(\mathrm{supp}\,(g) \setminus \ell\mathrm{t}(g)\big) \cap \langle \ell\mathrm{t}(G) \rangle_{\Bbbk[X]} = \emptyset \quad \text{for any} \quad g \in G.$$

The notion of a totally reduced Gröbner basis has also a theoretic value due to the fact that a totally reduced Gröbner basis $G \subseteq \mathcal{P} \setminus \{0\}$ is uniquely defined (up to multiplication of the basis polynomials with non-zero scalars) by the ideal $\langle G \rangle_{\mathcal{P}}$ and the term order $<$. The details can be found in [AL94] (proposition 1.8.4 on page 47), [BW93] (theorem 5.43 on page 209), [Win96] (theorem 8.3.6 on page 188).

2.3 The ring of extended polynomials

Let $\Bbbk$ be a field, $\Bbbk^* = \Bbbk \setminus \{0\}$, $X = \{x_1, \ldots, x_n\}$ is a finite set of variables and $\mathcal{P} = \Bbbk[X]$ is the algebra of polynomials in X over $\Bbbk$. Denote the terms in X by $\mathbb{T}$ and fix an arbitrary global term order $\underset{\mathbb{T}}{<}$ on $\mathbb{T}$. Fix a positive integer $\mathcal{B}$ and let e be a new variable with the relation $e^{\mathcal{B}} = 0$. By abuse of the language, $\Bbbk[X, e]$ – the $\Bbbk-$algebra generated by X and e is called the algebra of extended polynomials and $\Bbbk[e] \subseteq \Bbbk[X, e]$ is the algebra of univariate polynomials in e. Denote

- $\mathcal{P}(e) = \Bbbk[X, e]$ – the algebra of extended polynomials in $X \cup \{e\}$,
- $\mathbb{T}(e) = \left\{ t \cdot e^d \mid t \in \mathbb{T}, d \in \mathbb{N}_o \right\} \subseteq \mathcal{P}(e)$ – the multiplicatively closed set of extended terms (or simply terms when it is not ambiguous),

- $\mathbb{T}(e)^* = \mathbb{T}(e)\backslash\{0\}$ – the set of non-zero extended terms,
- $\mathbb{M}(e) = \{\alpha{\cdot}s|\, \alpha{\in}\Bbbk^*, s{\in}\mathbb{T}(e)^*\}$ – the set of extended monomials,
- $\mathfrak{C}(m) \in\Bbbk^*, \mathfrak{T}(m) \in\mathbb{T}(e)^*$ – the coefficient and the term of an extended monomial $m{\in}\mathbb{M}(e)$ such that $m{=}\, \mathfrak{C}(m) \cdot \mathfrak{T}(m)$ (uniquely defined by the construction of $\mathcal{P}(e)$).
- $\mathbb{M}(e)_{\not{e}}{\subseteq}\mathbb{M}(e)$ – the set of $e-$free monomials (also referred as stable monomials) i.e. the monomials in main variables: $\deg_e(s){=}0$ for all $s{\in}\mathbb{M}(e)_{\not{e}}$.
- $(e)^* = \left\{e^d \mid 0{\le}d{<}\mathcal{B}\right\}$ – the set of all non-vanishing degrees of e.

Let y be a new variable. Denote

- $\mathcal{P}(y){=}\Bbbk[X, y]$ – the algebra of polynomials in $X{\cup}\{y\}$,
- $\mathbb{T}(y) \subseteq \mathcal{P}(y)$ – the terms in $X \cup \{y\}$,
- $\mathbb{M}(y) = \{\alpha{\cdot}u|\, \alpha{\in}\Bbbk^*, u{\in}\mathbb{T}(y)\}$ – the monomials of $\mathcal{P}(y)$.

The word **extended** in the definition of $\mathcal{P}(e)$ emphasizes the particular sense of the new variable "e" relative to the "ground" variables X. The following theorem summarizes the basic properties of the algebra $\mathcal{P}(e)$ and serves as the foundation for all the subsequent discussion about extended polynomials.

Theorem 2.3.1.

a) For any $t{\in}\mathbb{T}$, $d{\in}\mathbb{N}_o$ the extended term $t{\cdot}e^d{\in}\mathbb{T}(e)^*$ if and only if $d{<}\mathcal{B}$.

b) $T(e)^* = \left\{t{\cdot}e^d \mid t \in \mathbb{T}, 0{\le}d{<}\mathcal{B}\right\}$.

c) The non-zero extended terms $\mathbb{T}(e)^*$ are linear independent over $\Bbbk$.

d) For each $s{\in}\mathbb{T}(e)^*$ there exist unique $t{\in}\mathbb{T}$ and $0{\le}d{<}\mathcal{B}$ such that $s{=}t{\cdot}e^d$.

e) Let $f{\in}\mathcal{P}(e)$. Then there exists the unique representation

$$f = \sum_{s\in S} \alpha_s{\cdot}s$$

with S - a finite subset of $\mathbb{T}(e)^*$ and $\alpha_s{\in}\Bbbk^*$ for all $s{\in}S$.

f) The algebra $\mathcal{P}(e)$ is Nötherian.

g) A polynomial $u \in \Bbbk[e] \subseteq \Bbbk[X, e]$ is a unit (multiplicatively invertible element) in $\mathcal{P}(e)$ if and only if u has a non-zero constant part:

$$\mathrm{U}(\Bbbk[e])_{\mathcal{P}(e)} = \left\{ \sum_{0 \leq i < \mathcal{B}} \alpha_i \cdot e^i \,\middle|\, \alpha_i \in \Bbbk, \alpha_o \neq 0 \right\}.^{1}$$

Proof. By the universal property of polynomial rings ([BW93] lemma 2.15 on page 72), there exists the unique homomorphism

$$\phi : \mathcal{P}(y) \longrightarrow \mathcal{P}(e)$$

such that $\phi|_{\Bbbk}$ is the identity map, $\phi(y) = e$ and $\phi(x_i) = x_i$ for $0 \leq i \leq n$. By construction of $\mathcal{P}(e)$, ϕ is surjective. Furthermore, it is easy to see (e.g. from the fact that $\mathcal{P}(e) \cong \mathcal{P} \otimes \Bbbk[e]$) that $\ker(\phi) = \langle y^{\mathcal{B}} \rangle_{\mathcal{P}(y)}$ and hence

$$\mathcal{P}(e) \cong {}^{\mathcal{P}(y)}\!/_{\langle y^{\mathcal{B}} \rangle}. \tag{$*$}$$

The particular and simple representation of the ideal $\ker(\phi)$ as generated by the single term $y^{\mathcal{B}}$ results in easy rules to decide the ideal membership question for it. Given a term $u \in \mathbb{T}(y)$ (cf. proposition 1.1 in [GM94])

$$u \in \ker(\phi) \quad \Leftrightarrow \quad y^{\mathcal{B}} | u \quad \Leftrightarrow \quad \deg_y(u) \geq \mathcal{B}. \tag{1}$$

Furthermore, by the characteristic property of monomial ideals in $\mathcal{P}(y)$ (see e.g. [AL94] lemma 1.6.4 on page 33, [GM94] proposition 1.2), given an arbitrary polynomial $h \in \mathcal{P}(y)$

$$h \in \ker(\phi) \quad \Leftrightarrow \quad u \in \ker(\phi) \text{ for each } \; u \in \operatorname{supp}(h) \tag{2}$$

and, by rule (1),

$$h \in \ker(\phi) \quad \Leftrightarrow \quad \deg_y(u) \geq \mathcal{B} \;\text{ for each }\; u \in \operatorname{supp}(h)\,. \tag{3}$$

a) Given $t \in \mathbb{T}$ and $d \in \mathbb{N}_o$ consider a pre-image $t \cdot y^{\mathcal{B}}$ of the term $t \cdot e^d$ under the action of ϕ. Then

$$t \cdot e^d = 0 = \phi(t \cdot y^d) \;\Leftrightarrow\; t \cdot y^d \in \ker(\phi) \overset{(1)}{\Leftrightarrow} \deg_y\left(t \cdot y^d\right) = \deg_y\left(y^d\right) = d \geq \mathcal{B}.$$

b) By construction of extended terms $T(e)$ and part *a)*.

[1] Which also means that the embedding of $\Bbbk[e]$ into $\mathcal{P}(e)$ does not introduce any new units in $\Bbbk[e]$.

c) The linear independence of the set of non-zero extended terms $\mathbb{T}(e)^*$ is the linear independence of all non-empty finite subsets in $\mathbb{T}(e)^*$. Assume there is a finite set $\emptyset{\neq}S{\subseteq}\mathbb{T}(e)^*$ and coefficients $\alpha_s{\in}\Bbbk^*$ one for each $s{\in}S$ such that the polynomial

$$\mathcal{P}(e) \ni f = \sum_{s\in S} \alpha_s{\cdot}s = 0.$$

Let

$$t_s \in \mathbb{T} \quad \text{and (cf. part } b)) \quad 0 \leq d_s < \mathcal{B} \tag{!}$$

be such that $t_s{\cdot}e^{d_s} = s$ for each $s{\in}S$ provided by the construction of extended terms. Consider a pre-image

$$\mathcal{P}(y) \ni h = \sum_{s\in S} \alpha_s{\cdot}t_s{\cdot}y^{d_s} \in \ker(\phi)$$

of f under the action of ϕ. Then, by (3) and part *a*),

$$h \in \ker(\phi) \quad \Leftrightarrow \quad d_s \geq \mathcal{B} \;\; \text{for each} \;\; s \in S$$

which contradicts (!) since S is not empty by construction.

d) Take $s{\in}\mathbb{T}(e)^*$. Assume there exist $t_1, t_2{\in}\mathbb{T}$ and $0{\leq}d_1, d_2{<}\mathcal{B}$ such that

$$t_1{\cdot}e^{d_1} = s = t_2{\cdot}e^{d_2}.$$

Consider pre-images $u_i{=}t_i{\cdot}y^{d_i}{\in}\mathbb{T}(y)$ of $t_i{\cdot}e^{d_i}$ under ϕ with $i{=}1,2$. Then $u_2{-}u_1{\in}\ker(\phi)$. If $t_1{\neq}t_2$ or $d_1{\neq}d_2$ then $u_1{\neq}u_2$ and, by (2), $u_1, u_2{\in}\ker(\phi)$ which contradicts the choice of $s{=}\phi(u_1){=}\phi(u_2)$ as a non-zero extended term.

e) By a straightforward combination of parts *c*) and *d*) in view of the fact that, by construction of the algebra $\mathcal{P}(e)$ of extended polynomials, the non-zero extended terms $\mathbb{T}(e)^*$ generate $\mathcal{P}(e)$ as a $\Bbbk-$linear vector space. For the special case of the zero polynomial set $\operatorname{supp}(0)=\emptyset$ (which is compliant with the common convention that the value of the empty sum is zero).

f) It is well-known that a quotient of a Nötherian ring is Nötherian (see e.g. [Lan92] proposition 1.5 on page 415). On the other hand, the algebra $\mathcal{P}(y)$ is Nötherian by the Hilbert basis theorem ([Lan92] corollary 4.2 on page 187). Then, by $(*)$, $\mathcal{P}(e)$ is Nötherian as a quotient of $\mathcal{P}(y)$.

g) It is quite clear (e.g. by part *e*) that any element $u \in \Bbbk[e]$ has the unique representation

$$u = \sum_{0 \le i < \mathcal{B}} \alpha_i \cdot e^i \in \mathcal{P}(e) \quad \text{with} \quad \alpha_i \in \Bbbk.$$

We claim that u is a unit in $\mathcal{P}(e)$ if and only if $\alpha_o \neq 0$. Indeed, take a pre-image of u in $\Bbbk[y]$ under the action of ϕ:

$$h = \sum_{0 \le i < \mathcal{B}} \alpha_i \cdot y^i \in \Bbbk[y], \; \phi(h) = u.$$

- Let $\alpha_o \neq 0$. Then, by the elementary properties of the univariate polynomials gcd ([Win96] corollary from theorem 3.1.2 on page 53, [KGL92] theorem 2.6 on page 44 or any elementary algebra decent textbook), there exist polynomials $A, B \in \Bbbk[y]$ such that

$$\mathcal{P}(y) \ni 1 = \gcd(h, y^{\mathcal{B}}) = h \cdot A + y^{\mathcal{B}} \cdot B, \; \deg(A) < \mathcal{B}$$

  and hence

$$\mathcal{P}(e) \ni 1 = \phi(1) = \phi(h \cdot A) + \phi(y^{\mathcal{B}} \cdot B) = u \cdot \phi(A).$$

  Therefore u has the inverse in $\mathcal{P}(e)$: $\frac{1}{u} = \phi(A) \in \Bbbk[e] \subseteq \mathcal{P}(e)$.
- Let $\alpha_o = 0$. Then there is no inverse of u because it is either zero or a zero-divisor. Indeed, assume $u \neq 0$ and take $0 < k < \mathcal{B}$ such that $\alpha_k \neq 0$ and $\alpha_i = 0$ for all $i < k$ (such k exists because $u \neq 0$).
  Then
$$0 = \phi(h \cdot y^{\mathcal{B}-k}) = u \cdot e^{\mathcal{B}-k},$$
  where $e^{\mathcal{B}-k} \neq 0$ provided that $k < \mathcal{B}$.

□

Definition 2.3.1. Let $s = t \cdot e^d \in \mathbb{T}(e)^*$ with $t \in \mathbb{T}$ and $0 \le d < \mathcal{B}$.

- The **main part** of s is $\mathrm{T}(s) = t$.
- The $e-$**degree** of s is $\deg_e(s) = d$.

By part *d*) of theorem 2.3.1 the main part and the $e-$degree are uniquely defined for each non-zero extended term.

Lemma 2.3.1. Let $a, b \in \mathbb{T}_e{}^*$. Then *a)* $a{\cdot}b \neq 0$ if and only if $\deg_e(a){+}\deg_e(b){<}\mathcal{B}$;

b) if $a{\cdot}b \neq 0$ then

$$\begin{aligned}\deg_e(a{\cdot}b) &= \deg_e(a) + \deg_e(b),\\ \mathrm{T}(a{\cdot}b) &= \mathrm{T}(a){\cdot}\mathrm{T}(b).\end{aligned}$$

Proof. Straightforward according to the way the ring multiplication is defined on $\mathbb{T}(e){\subseteq}\mathcal{P}(e)$, part *b)* of theorem 2.3.1 together with the uniqueness of the $e-$degree and the main part for each non-zero extended term. □

The following important particular case of the latter lemma will be frequently used in the derivation of the completion procedure in $\mathcal{P}(e)$.

Corollary 2.3.1. Let $t{\in}\mathbb{T}$ and $s{\in}\mathbb{T}(e)^*$. Then $t{\cdot}s{\neq}0$, $\deg_e(t{\cdot}s){=}\deg_e(s)$ and $\mathrm{T}(t{\cdot}s){=}t{\cdot}\mathrm{T}(s)$.

Proof. If $t{\in}\mathbb{T}$ then $\deg_e(t){=}0$ and $\mathrm{T}(t){=}t$. The rest is by lemma 2.3.1. □

Definition 2.3.2 (divisibility in $\mathcal{P}(e)$). Let $s_1, s_2{\in}\mathbb{T}(e)^*$. Then s_1 **divides** s_2 written $s_1|s_2$ if and only if there exists an element $s{\in}\mathbb{T}(e)$ denoted $s{=}\frac{s_2}{s_1}$ such that $s_1{\cdot}s{=}s_2$.

Lemma 2.3.2. Let $s_1, s_2{\in}\mathbb{T}(e)^*$. Denote $t_1{=}\mathrm{T}(s_1)$, $t_2{=}\mathrm{T}(s_2)$, $d_1{=}\deg_e(s_1)$ and $d_2{=}\deg_e(s_2)$. Then $s_1|s_2$ if and only if $t_1|t_2$ and $d_1{\leq}d_2$.

Proof.

$\Longrightarrow$ If $s_1|s_2$ then, by definition 2.3.2 of divisibility in $\mathcal{P}(e)$ there exists $s{\in}\mathbb{T}(e)^*$ such that $s_1{\cdot}s{=}s_2$. Denote $t{=}\mathrm{T}(s)$ and $d{=}\deg_e(s)$. Then, by part *b)* of lemma 2.3.1,

$$\begin{gathered} t_1{\cdot}t = t_2 \;\text{ follows }\; t_1|t_2,\\ d_1{+}d = d_2 \;\text{ follows }\; d_1{\leq}d_2.\end{gathered}$$

$\Longleftarrow$ Let $t_1|t_2$ and $d_1{\leq}d_2$. Then $0{\leq}d_2{-}d_1{<}\mathcal{B}$, hence the term $s{=}\frac{t_2}{t_1}{\cdot}e^{d_2-d_1}$ is defined and, by part *a)* of lemma 2.3.1, $s{\in}\mathbb{T}(e)^*$. Furthermore

$$s_1{\cdot}s = t_1{\cdot}\frac{t_2}{t_1}{\cdot}e^{d_2-d_1}{\cdot}e^{d_1} = t_2{\cdot}e^{d_2} = s_2 \;\text{ follows }\; s_1|s_2.$$

□

Definition 2.3.3. Given $f{\in}\mathcal{P}(e)$ the **support** of f denoted $\mathrm{supp}\,(f)$ is the finite subset of $\mathbb{T}(e)$ such that f is the linear combination of the terms from $\mathrm{supp}\,(f)$ with non-zero coefficients uniquely defined according to part *f)* of theorem 2.3.1.

Notation. Let

$$f = \sum_{s\in\operatorname{supp}(f)} \alpha_s \cdot s \in \mathcal{P}(e) \tag{$*$}$$

with the uniquely defined coefficients $\alpha_s \in \mathbb{k}^*$ for $s \in \operatorname{supp}(f)$. To access the separate coefficients and monomials of f we borrow the notation from the polynomial case provided that the representation $(*)$ is uniquely defined.

- Take a term $s \in \operatorname{supp}(f)$. Then
  - the coefficient of (in) f at s is
  $$\mathfrak{C}_f(s) = \alpha_s;$$
  - the monomial of (in) f at s is
  $$\mathfrak{M}_f(s) = \alpha_s \cdot s.$$
- Also, for convenience, denote
  - the coefficients of (in) f is the set
  $$\mathfrak{C}(f) = \{\mathfrak{C}_f(s) \,|\, s \in \operatorname{supp}(f)\};$$
  - the monomials of (in) f is the set
  $$\mathfrak{M}(f) = \{\mathfrak{M}_f(s) \,|\, s \in \operatorname{supp}(f)\}.$$

Sometimes the fact that given a polynomial $f \in \mathcal{P}(e)$, $\mathfrak{M}_f(v)$ and $\mathfrak{C}_f(v)$ are undefined for a term $v \notin \operatorname{supp}(f)$ appears to be too cumbersome. We define the formal monomial and the formal coefficient to manage such cases.

Definition 2.3.4. Given a polynomial $f \in \mathcal{P}(e)$ and a term $v \in \mathbb{T}(e)$.

- The formal monomial of f at v is
$$[\mathfrak{M}_f](v) = \begin{cases} \mathfrak{M}_f(v) & v \in \operatorname{supp}(f), \\ 0 & \text{otherwise.} \end{cases}$$
- The formal coefficient of f at v is
$$[\mathfrak{C}_f](v) = \begin{cases} \mathfrak{C}_f(v) & v \in \operatorname{supp}(f), \\ 0 & \text{otherwise.} \end{cases}$$

Notation. Let $F{\subseteq}\mathcal{P}(e)$. By a slight abuse of the language and notation the support of F is

$$\operatorname{supp}(F) = \bigcup_{f\in F} \operatorname{supp}(f)$$

the set of terms "involved" in the support of the polynomials of F.

Definition 2.3.5. Let $S_1, S_2 \subseteq \mathbb{T}(e)$. The **free product** of S_1 and S_2 is the set

$$S_1 * S_2 = \{s_1{\cdot}s_2 \mid s_1{\in}S_1, s_2{\in}S_2\} \setminus \{0\}.$$

Given $s{\in}\mathbb{T}(e)^*$ we allow a sloppy notation $s{*}S_2$ for $\{s\}{*}S_2$. Observe a trivial but convenient for notation inclusion $\{\frac{w}{s} \mid w{\in}s{*}S_2\}{\subseteq}S_2$.

Lemma 2.3.3. Let $\alpha{\in}\mathbb{k}^*$, $v{\in}\mathbb{T}(e)^*$ and $f, g \in \mathcal{P}(e)$. Then

$$\begin{aligned}
&a) && \operatorname{supp}(\alpha{\cdot}f) = \operatorname{supp}(f)\,,\\
&b) && \operatorname{supp}(f) \bigtriangleup \operatorname{supp}(g) \subseteq \operatorname{supp}(f+g) \subseteq \operatorname{supp}(f) \cup \operatorname{supp}(g)\,,\\
&c) && \operatorname{supp}(fg) \subseteq \operatorname{supp}(f) * \operatorname{supp}(g)\,,\\
&d) && \operatorname{supp}(v{\cdot}g) = v * \operatorname{supp}(g)\,,\\
&e) && \operatorname{supp}(\alpha{\cdot}v{\cdot}g) = v * \operatorname{supp}(g)\,.
\end{aligned}$$

Proof. All non-trivial parts follow the same pattern of argumentation based on the uniqueness of the support for each extended polynomial in $\mathcal{P}(e)$. The uniqueness is used in the sense that given a representation of an extended polynomial as a linear combination of finitely many non-zero extended terms with non-zero coefficients this set of terms is the uniquely defined support which, in particular, does not depend on the way how the representation has been attained. The desired representation is achieved by a straightforward polynomial expressions manipulation provided that $\mathcal{P}(e)$ is a ring.

Let $\{\alpha_s | s{\in} \operatorname{supp}(f)\}, \{\beta_s | s{\in} \operatorname{supp}(g)\}{\subseteq}\mathbb{k}^*$ be the (uniquely defined) coefficients such that

$$f = \sum_{s\in\operatorname{supp}(f)} \alpha_s{\cdot}s, \qquad g = \sum_{s\in\operatorname{supp}(g)} \beta_s{\cdot}s.$$

a)

$$\alpha{\cdot}f = \alpha \cdot \left(\sum_{s\in\operatorname{supp}(f)} \alpha_s{\cdot}s \right) = \sum_{s\in\operatorname{supp}(f)} (\alpha\alpha_s){\cdot}s$$

where the coefficient $\alpha{\cdot}\alpha_s{\neq}0$ for each $s{\in} \operatorname{supp}(f)$ because $\mathbb{k}$ is a field. Hence $\operatorname{supp}(\alpha{\cdot}f) = \operatorname{supp}(f)$.

b) For each $s{\in}\operatorname{supp}(f)\cup\operatorname{supp}(g)$ define

$$\Bbbk \ni \gamma_s = \begin{cases} \alpha_s & \text{if } s \in \operatorname{supp}(f) \setminus \operatorname{supp}(g)\,, \\ \alpha_s+\beta_s & \text{if } s \in \operatorname{supp}(f) \cap \operatorname{supp}(g)\,, \\ \beta_s & \text{if } s \in \operatorname{supp}(g) \setminus \operatorname{supp}(f)\,. \end{cases} \tag{$*$}$$

Let

$$N=\Big\{ s \in \operatorname{supp}(f)\cup\operatorname{supp}(g) \;\Big|\; \gamma_s \neq 0 \Big\}.$$

Then

$$f+g=\sum_{s\in\operatorname{supp}(f)}\alpha_s{\cdot}s \;+\sum_{s\in\operatorname{supp}(g)}\beta_s{\cdot}s = \sum_{s\in\operatorname{supp}(f)\cup\operatorname{supp}(g)}\gamma_s{\cdot}s \;=\sum_{s\in N}\gamma_s{\cdot}s$$

where $\gamma_s{\neq}0$ for each $s{\in}N$ by construction of N and hence, by definition of the support, $\operatorname{supp}(f+g){=}N{\subseteq}\operatorname{supp}(f)\cup\operatorname{supp}(g)$. On the other hand, $\operatorname{supp}(f)\bigtriangleup\operatorname{supp}(g)\subseteq N{=}\operatorname{supp}(f+g)$ because, by ($*$), $\gamma_s{\neq}0$ for $s{\in}\operatorname{supp}(f)\bigtriangleup\operatorname{supp}(g)$.

c) For each $s\in\operatorname{supp}(f)*\operatorname{supp}(g)$ define

$$\Bbbk \ni \gamma_s = \sum_{\substack{s_1\in\operatorname{supp}(f), s_2\in\operatorname{supp}(g)\\ s_1 s_2 = s}} \alpha_{s_1}\beta_{s_2}.$$

Let

$$N=\Big\{ s \in \operatorname{supp}(f)*\operatorname{supp}(g) \;\Big|\; \gamma_s \neq 0 \Big\}.$$

Then

$$fg=\Big(\sum_{s\in\operatorname{supp}(f)}\alpha_s{\cdot}s\Big)\Big(\sum_{s\in\operatorname{supp}(g)}\beta_s{\cdot}s\Big) = \sum_{s\in\operatorname{supp}(f)*\operatorname{supp}(g)}\gamma_s{\cdot}s \;=\sum_{s\in N}\gamma_s{\cdot}s$$

where $\gamma_s{\neq}0$ for each $s{\in}N$ by construction of N and hence, by definition of the support, $\operatorname{supp}(fg)=N\subseteq\operatorname{supp}(f)*\operatorname{supp}(g)$.

d) By

$$v{\cdot}g = v\cdot\left(\sum_{s\in\operatorname{supp}(g)}\beta_s{\cdot}s\right) = \sum_{s\in\operatorname{supp}(g)}\beta_s{\cdot}(v{\cdot}s) = \sum_{\substack{s\in\operatorname{supp}(g)\\ vs\neq 0}}\beta_s{\cdot}(vs)$$

and the definition of the free product on the sets of non-zero terms it follows that

$$\operatorname{supp}(v{\cdot}g)=\{vs \mid s\in\operatorname{supp}(g)\,, vs\neq 0\} = \\ \{vs \mid s\in\operatorname{supp}(g)\}\setminus\{0\} = v{*}\operatorname{supp}(g)\,.$$

e) It is an easy combination of parts a) and d). Indeed

$$\operatorname{supp}(\alpha\cdot v\cdot g) \overset{a)}{=} \operatorname{supp}(v\cdot g) \overset{d)}{=} v*\operatorname{supp}(g)\,.$$

$\square$

As a matter of fact the proof of the latter lemma sets up more statements than formulated in the lemma. The following corollary fills up the missing formulations.

Corollary 2.3.2 (from the proof of lemma 2.3.3). Let $\alpha\in\Bbbk^*$, $v\in\mathbb{T}(e)^*$ and $f, g \in \mathcal{P}(e)$. Then

a) $\quad \mathfrak{C}_{\alpha\cdot f}(s) = \alpha\cdot\, \mathfrak{C}_f(s)$ for each $s\in\operatorname{supp}(f)$,

b) $\quad [\mathfrak{C}_{f+g}](s) = [\mathfrak{C}_f](s) + [\mathfrak{C}_g](s)$ for each $s\in\operatorname{supp}(f)\cup\operatorname{supp}(g)$,

c) $\quad [\mathfrak{C}_{fg}](s) = \sum\limits_{\substack{s_1\in\operatorname{supp}(f), s_2\in\operatorname{supp}(g)\\ s_1 s_2 = s}} \mathfrak{C}_f(s_1)\,\mathfrak{C}_g(s_2)$; for each $s\in\operatorname{supp}(f)*\operatorname{supp}(g)$,

d) $\quad \mathfrak{C}_{v\cdot g}(s) = \mathfrak{C}_g\left(\dfrac{s}{v}\right)$ for each $s \in \operatorname{supp}(v\cdot g)$,

e) $\quad \mathfrak{C}_{\alpha\cdot v\cdot g}(s) = \alpha\cdot\, \mathfrak{C}_g\left(\dfrac{s}{v}\right)$ for each $s \in \operatorname{supp}(\alpha\cdot v\cdot g)$.

Proof. Each part is easy to verify looking at the polynomial representation attained in the proof of the corresponding part of the latter lemma. $\square$

2.3.1 The monomial ideals in $\mathcal{P}(e)$

The fact that the non-zero extended terms $\mathbb{T}(e)^*$ are linear independent is also crucial in a different respect. It is the key to transfer the techniques conventionally used in construction of the classical theory of Gröbner bases to the case of the algebra of extended polynomials $\mathcal{P}(e)$. In particular it is straightforward to transfer the notion and the characteristic property of monomial ideals to $\mathcal{P}(e)$.

Definition 2.3.6. An ideal $I\triangleleft\mathcal{P}(e)$ is called a **monomial ideal** if it can be generated by a set of terms $S\subseteq\mathbb{T}(e)^*$. The set of terms S is called the **monomial basis** of I.

The following property of monomial ideals is important for the construction of Gröbner bases theory as well as for computational purposes since it gives a computationally efficient rule to check the monomial ideal membership in a special case of a term.

Lemma 2.3.4. Let $I{=}\langle S\rangle_{\mathcal{P}(e)}$ be the monomial ideal with a monomial basis $S{\subseteq}\mathbb{T}(e)^*$ and $v{\in}\mathbb{T}(e)^*$. Then $v{\in}I$ if and only if there is $s{\in}S$ such that $s|v$.

Proof. The "if" direction is trivial. Conversely, let $\{g_s{\in}\mathcal{P}(e)|\, s{\in}S\}$ be polynomials such that

$$v = \sum_{s\in S} g_s{\cdot}s. \qquad (*)$$

It is clear that s divides any term of $\operatorname{supp}(g_s) * s$ for each $s{\in}S$. Hence each term of the set

$$N = \bigcup_{s\in S} \operatorname{supp}(g_s) * s$$

is divisible by a term from S. To complete the proof note that, by the linear independence of $\mathbb{T}(e)^*$,

$$v \in \operatorname{supp}\left(\sum_{s\in S} g_s{\cdot}s\right) \subseteq N.$$

□

The following lemma describes the characteristic property of monomial ideals which is, in fact, interchangeable with the definition and usually more convenient for theoretical reasoning. The main reason the proof of the lemma, which is completely identical to the one in the polynomial case, is not omitted is to demonstrate the importance of the linear independence of terms for the characterization of monomial ideals.

Lemma 2.3.5. Given an ideal $I{\lhd}\mathcal{P}(e)$ the following conditions are equivalent.

a) The ideal I is monomial.

b) Let $f{\in}\mathcal{P}(e)$. Then $f{\in}I$ if and only if $\operatorname{supp}(f){\subseteq}I$.

Proof.

$a \to b$) The "if" direction is trivial. Conversely, let $S{\subseteq}\mathbb{T}(e)^*$ be a monomial basis of the ideal: $I{=}\langle S\rangle_{\mathcal{P}(e)}$, provided by part a). Assume $f{\in}I$. Then there exist polynomials $\{g_s{\in}\mathcal{P}(e)|\, s{\in}S\}$ such that

$$f = \sum_{s\in S} g_s{\cdot}s. \qquad (*)$$

Denote

$$N{=}\bigcup_{s\in S} \operatorname{supp}(g_s) * s.$$

It is clear that $N{\subseteq}I$ (by $S{\subseteq}I$). By the linear independence of non-zero extended terms, the equation $(*)$ and the uniqueness of $\operatorname{supp}(f)$,

$$\operatorname{supp}(f) = \operatorname{supp}\left(\sum_{s\in S} g_s{\cdot}s\right) \subseteq N.$$

Hence $\operatorname{supp}(f) \subseteq I$.

$b \to a)$ Let $G{\subseteq}I\backslash\{0\}$ be a basis of the ideal: $I{=}\langle G\rangle_{\mathcal{P}(e)}$. Then, by part $b)$,

$$\operatorname{supp}(G) \subseteq I \quad \text{and therefore} \quad \langle\operatorname{supp}(G)\rangle_{\mathcal{P}(e)} \subseteq I. \tag{$*$}$$

Also note the trivial fact that

$$\langle h\rangle_{\mathcal{P}(e)} \subseteq \big\langle \operatorname{supp}(h)\big\rangle_{\mathcal{P}(e)} \quad \text{for any} \quad h \in \mathcal{P}(e). \tag{$**$}$$

Hence

$$I = \langle G\rangle_{\mathcal{P}(e)} = \left\langle \bigcup_{g\in G} g \right\rangle_{\mathcal{P}(e)} = \sum_{g\in G} \langle g\rangle_{\mathcal{P}(e)} \overset{(**)}{\subseteq}$$

$$\sum_{g\in G} \big\langle \operatorname{supp}(g)\big\rangle_{\mathcal{P}(e)} = \left\langle \bigcup_{g\in G} \operatorname{supp}(g) \right\rangle_{\mathcal{P}(e)} = \big\langle \operatorname{supp}(G)\big\rangle_{\mathcal{P}(e)} \overset{(*)}{\subseteq} I$$

which means that $I{=}\big\langle \operatorname{supp}(G)\big\rangle_{\mathcal{P}(e)}$ and therefore the set of terms $\operatorname{supp}(G)$ is a monomial basis of I.

□

A slight refinement of the proof of the latter lemma gives a thoughtful fact that a monomial ideal can be generated by finitely many terms.

Corollary 2.3.3 (from the proof of lemma 2.3.5). Let $S{\subseteq}\mathbb{T}(e)^*$ be a monomial basis of the ideal $I{=}\langle S\rangle_{\mathcal{P}(e)}$. Then there is a finite subset $S_o{\subseteq}S$ such that $I = \langle S_o\rangle_{\mathcal{P}(e)}$.

Proof. According to part $f)$ of theorem 2.3.1 the ring $\mathcal{P}(e)$ is Nötherian. Let $G{\subseteq}I$ be a finite generating set of the ideal: $I = \langle G\rangle_{\mathcal{P}(e)}$. Then, by the argument in the proof $b \to a)$ of lemma 2.3.5, $\operatorname{supp}(G)$ is a monomial basis of I and, in particular, $\operatorname{supp}(G) \subseteq I$. For each $v{\in}\operatorname{supp}(G)$ choose $s_v{\in}S$ such that $s_v|s$ provided by lemma 2.3.4. Denote $S_o = \{s_v | v{\in}\operatorname{supp}(G)\}{\subseteq}S$. It is clear that S_o generates the same ideal as $\operatorname{supp}(G)$. Then

$$\langle S\rangle_{\mathcal{P}(e)} = \big\langle \operatorname{supp}(G)\big\rangle_{\mathcal{P}(e)} = \langle S_o\rangle_{\mathcal{P}(e)}$$

and therefore S_o is the desired finite monomial basis of I contained in S. □

2.3.2 The extended term order and leading terms

Definition 2.3.7. A total order $<$ on $\mathbb{T}(e)$ with "0" - the smallest term is called **quasi-monotonous** (under multiplication of terms) if $a_1{<}a_2$, $b_1{\leq}b_2$ and $a_2b_2{\neq}0$ follow $a_1b_1{<}a_2b_2$ for all $a_1, b_1{\in}\mathbb{T}(e)$, $a_2, b_2{\in}\mathbb{T}(e)^*$.

Lemma 2.3.6. Let $<$ be a quasi-monotonous order on $\mathbb{T}(e)$, $a_1{\in}\mathbb{T}(e)$ and $a_2, s{\in}\mathbb{T}(e)^*$ be such that $a_1{<}a_2$. Then $a_2s{\neq}0$ follows $a_1s{<}a_2s$.

Proof. Trivial. □

To establish a kind of a weak inverse of the latter lemma, important for the reduction part of the Gröbner bases theory in $\mathcal{P}(e)$, we would need one more notion reflected in the following fact.

Lemma 2.3.7. Let $a_1, a_2, b{\in}\mathbb{T}(e)^*$ be such that $a_1b{\neq}0$. Then $a_1b{=}a_2b$ if and only if $a_1{=}a_2$.

Proof. The "if" direction is trivial. Conversely, let $0{\neq}a_1b{=}a_2b$. Then, by lemma 2.3.1,

$$\begin{aligned} \deg_e(a_1) + \deg_e(b) = \deg_e(a_1b) &= \deg_e(a_2b) = \deg_e(a_2) + \deg_e(b) \quad \text{and} \\ \mathrm{T}(a_1){\cdot}\mathrm{T}(b) = \mathrm{T}(a_1b) &= \mathrm{T}(a_2b) = \mathrm{T}(a_2){\cdot}\mathrm{T}(b). \end{aligned}$$

Hence, $\deg_e(a_1){=}\deg_e(a_2)$ and, by the cancellation law in $\mathbb{T}$, $\mathrm{T}(a_1){=}\mathrm{T}(a_2)$. Therefore $a_1{=}a_2$ by part d) of theorem 2.3.1. □

Definition 2.3.8. The property stated in lemma 2.3.7 is called the **quasi-cancellation law** in $\mathbb{T}(e)$.

Lemma 2.3.8. Let $<$ be a quasi-monotonous order on $\mathbb{T}(e)$, $a_1{\in}\mathbb{T}(e)$ and $a_2, b{\in}\mathbb{T}(e)^*$ be such that $a_2b{\neq}0$. Then $a_1b{<}a_2b$ if and only if $a_1{<}a_2$.

Proof. The "if" direction is trivial by the quasi-monotonicity of $<$. Conversely, if $a_2{<}a_1$ then $ba_2{<}ba_1$ by the quasi-monotonicity. The equality case is excluded by the quasi-cancellation law. □

Definition 2.3.9. An **extended term order** on $\mathbb{T}(e)$ is a quasi-monotonous well-order on $\mathbb{T}(e)$.

Note that 0 is the smallest term w.r.t. any extended term order by definition 2.3.7 of a quasi-monotonous order. Throughout the rest of this section the symbol $<$ denotes a fixed extended term order. The **leading term**, the **leading coefficient** and the **leading monomial** w.r.t. $<$ are defined in a usual way on the non-zero elements of $\mathcal{P}(e)$ and kept undefined for the zero polynomial.

Definition 2.3.10. Let $f{\in}\mathcal{P}(e)\backslash\{0\}$.

- The **leading term** $\ell\mathrm{t}(f)$ of f w.r.t. $<$ is the $<-$maximal element of the support of f:
$$\ell\mathrm{t}(f) = \max_{<}\Big(\mathrm{supp}\,(f)\Big).$$
- The **leading coefficient** $\ell\mathrm{c}(f)$ of f w.r.t. $<$ is the coefficient in f at the leading term of f:
$$\ell\mathrm{c}(f) = \mathfrak{C}_f(\ell\mathrm{t}(f))\,.$$
- The **leading monomial** $\ell\mathrm{m}(f)$ of f w.r.t. $<$ is the monomial in f with the term $\ell\mathrm{t}(f)$:
$$\ell\mathrm{m}(f) = \mathfrak{M}_f(\ell\mathrm{t}(f))\,\Big[= \ell\mathrm{c}(f){\cdot}\ell\mathrm{t}(f)\Big].$$

Notation. Let $F{\subseteq}\mathcal{P}(e)$. The sets of **leading terms**, **leading monomials** and **leading coefficients** of non-zero elements of F are denoted

$$\begin{aligned}\ell\mathrm{t}(F) &= \{\ell\mathrm{t}(f) \mid f \in F\backslash\{0\}\}\,,\\ \ell\mathrm{m}(F) &= \{\ell\mathrm{m}(f) \mid f \in F\backslash\{0\}\}\,,\\ \ell\mathrm{c}(F) &= \{\ell\mathrm{c}(f) \mid f \in F\backslash\{0\}\}\,.\end{aligned}$$

The behavior of the leading term function under multiplication with a non-zero coefficient and addition in $\mathcal{P}(e)$ is exactly the same as in the classical case.

Lemma 2.3.9. Let $f{\in}\mathcal{P}(e)\backslash\{0\}$ and $\alpha{\in}\Bbbk^*$. Then $\alpha{\cdot}f{\neq}0$ and $\ell\mathrm{t}(\alpha{\cdot}f) = \ell\mathrm{t}(f)$.

Proof. Note that $\mathrm{supp}\,(f){\neq}\emptyset$ because $f{\neq}0$. Then, by part $a)$ of lemma 2.3.3,

$$\mathrm{supp}\,(\alpha{\cdot}f) = \mathrm{supp}\,(f) \neq \emptyset.$$

Then $\alpha{\cdot}f{\neq}0$ and hence $\ell\mathrm{t}(\alpha{\cdot}f)$ is defined. Furthermore

$$\ell\mathrm{t}(\alpha{\cdot}f) = \max\Big(\mathrm{supp}\,(\alpha{\cdot}f)\Big) = \max\Big(\mathrm{supp}\,(f)\Big) = \ell\mathrm{t}(f).$$

□

Corollary 2.3.4. If $f{\in}\mathcal{P}(e)\backslash\{0\}$ and $\alpha{\in}\Bbbk^*$ then$\alpha{\cdot}f{\neq}0$ and $\ell\mathrm{c}(\alpha{\cdot}f){=}\alpha{\cdot}\ell\mathrm{c}(f)$.

Proof. By lemma 2.3.9 and part *a*) of corollary 2.3.2,

$$\alpha{\cdot}f \overset{(L2.3.9)}{\neq} 0 \text{ and}$$

$$\ell\mathrm{c}(\alpha{\cdot}f) = \mathfrak{C}_{\alpha\cdot f}\Big(\ell\mathrm{t}(\alpha{\cdot}f)\Big) \overset{(L2.3.9)}{=} \mathfrak{C}_{\alpha\cdot f}\Big(\ell\mathrm{t}(f)\Big) \overset{(C2.3.2)}{=} \alpha{\cdot}\, \mathfrak{C}_f\Big(\ell\mathrm{t}(f)\Big) = \alpha{\cdot}\ell\mathrm{t}(f). \square$$

Lemma 2.3.10. Let $f, g{\in}\mathcal{P}(e)\backslash\{0\}$. The following holds:

a) if $f{+}g{\neq}0$ then $\ell\mathrm{t}(f{+}g){\leq}\max\Big(\ell\mathrm{t}(f), \ell\mathrm{t}(g)\Big)$,

b) if $\ell\mathrm{t}(f){\neq}\ell\mathrm{t}(g)$ or $\ell\mathrm{c}(f){+}\ell\mathrm{c}(g){\neq}0$ then

$$f{+}g \neq 0 \quad \text{and} \quad \ell\mathrm{t}(f{+}g) = \max\Big(\ell\mathrm{t}(f), \ell\mathrm{t}(g)\Big).$$

c) if $\ell\mathrm{t}(f){=}\ell\mathrm{t}(g)$, $\ell\mathrm{c}(f){+}\ell\mathrm{c}(g){=}0$ and $f{+}g{\neq}0$ then

$$\ell\mathrm{t}(f{+}g) < \max\Big(\ell\mathrm{t}(f), \ell\mathrm{t}(g)\Big).$$

Proof. Recall that, by part *b*) of lemma 2.3.3

$$\mathrm{supp}\,(f) \bigtriangleup \mathrm{supp}\,(g) \subseteq \mathrm{supp}\,(f{+}g) \subseteq \mathrm{supp}\,(f) \cup \mathrm{supp}\,(g) \qquad (*)$$

a) By $(*)$

$$\ell\mathrm{t}(f{+}g) = \max\Big(\mathrm{supp}\,(f{+}g)\Big) \overset{(*)}{\leq} \max\Big(\mathrm{supp}\,(f) \cup \mathrm{supp}\,(g)\Big) =$$
$$\max\Big(\max\big(\mathrm{supp}\,(f)\big), \max\big(\mathrm{supp}\,(g)\big)\Big) = \max\Big(\ell\mathrm{t}(f), \ell\mathrm{t}(g)\Big).$$

b) Consider two cases.

1. Let $\ell\mathrm{t}(f){\neq}\ell\mathrm{t}(g)$. W.l.o.g. assume $\ell\mathrm{t}(f){<}\ell\mathrm{t}(g)$. Then, by transitivity of $<$,

$$\ell\mathrm{t}(g) \in \mathrm{supp}\,(g) \setminus \mathrm{supp}\,(f) \subseteq \mathrm{supp}\,(f) \bigtriangleup \mathrm{supp}\,(g) \qquad (1)$$

which follows

$$\emptyset \neq \mathrm{supp}\,(f) \bigtriangleup \mathrm{supp}\,(g) \overset{(*)}{\subseteq} \mathrm{supp}\,(f{+}g) \neq \emptyset$$

and hence, by the uniqueness of support, $f{+}g \neq 0$. Then

$$\boxed{\ell\mathrm{t}(g)} \overset{(1)}{\leq} \max\Big(\mathrm{supp}\,(f) \bigtriangleup \mathrm{supp}\,(g)\Big) \overset{(*)}{\leq}$$
$$\max\Big(\mathrm{supp}\,(f{+}g)\Big) = \boxed{\ell\mathrm{t}(f{+}g)} \overset{a)}{\leq} \boxed{\ell\mathrm{t}(g)}.$$

2. Let $\ell\mathrm{t}(f){=}\ell\mathrm{t}(g)$ and $\ell\mathrm{c}(f){+}\ell\mathrm{c}(g){\neq}0$. Denote $w{=}\ell\mathrm{t}(f)$. Then, by part $b)$ of corollary 2.3.2,

$$[\mathfrak{C}_{f+g}](w) = [\mathfrak{C}_f](w) + [\mathfrak{C}_g](w) = \mathfrak{C}_f\Big(\ell\mathrm{t}(f)\Big) + \mathfrak{C}_g\Big(\ell\mathrm{t}(g)\Big) = \ell\mathrm{c}(f) + \ell\mathrm{c}(g) \neq 0$$

which follows

$$w \in \mathrm{supp}\,(f{+}g) \tag{2}$$

and hence $f{+}g{\neq}0$. Then

$$\boxed{w} \overset{(2)}{\leq} \max\Big(\mathrm{supp}\,(f{+}g)\Big) = \boxed{\ell\mathrm{t}(f{+}g)} \overset{a)}{\leq} \max\Big(\ell\mathrm{t}(f), \ell\mathrm{t}(g)\Big) = \boxed{w}\,.$$

c) Let $\ell\mathrm{t}(f){=}\ell\mathrm{t}(g)$, $\ell\mathrm{c}(f){+}\ell\mathrm{c}(g){=}0$ and $f{+}g{\neq}0$. Denote $w{=}\ell\mathrm{t}(f)$. Then, by part $b)$ of corollary 2.3.2,

$$[\mathfrak{C}_{f+g}](w) = [\mathfrak{C}_f](w) + [\mathfrak{C}_g](w) = \mathfrak{C}_f\Big(\ell\mathrm{t}(f)\Big) + \mathfrak{C}_g\Big(\ell\mathrm{t}(g)\Big) = \ell\mathrm{c}(f) + \ell\mathrm{c}(g) = 0$$

which follows

$$w \notin \mathrm{supp}\,(f{+}g) \quad \text{and hence} \quad w{\neq}\ell\mathrm{t}(f{+}g).$$

On the other hand, by part $a)$

$$\ell\mathrm{t}(f{+}g) \leq w.$$

Therefore

$$\ell\mathrm{t}(f{+}g) < w = \ell\mathrm{t}(f) = \max\Big(\ell\mathrm{t}(f), \ell\mathrm{t}(g)\Big). \qquad \square$$

Corollary 2.3.5. If $f, g{\in}\mathcal{P}(e)\backslash\{0\}$, $\ell\mathrm{t}(f){=}\ell\mathrm{t}(g)$ and $\ell\mathrm{c}(f){+}\ell\mathrm{c}(g){\neq}0$ then $f{+}g{\neq}0$ and $\ell\mathrm{c}(f{+}g){=}\ell\mathrm{c}(f) + \ell\mathrm{c}(g)$.

Proof. By part $b)$ of lemma 2.3.10 and part $b)$ of corollary 2.3.2,

$$\begin{aligned} &f{+}g \overset{(L2.3.10)}{\neq} 0 \;\text{ and} \\ &\ell\mathrm{c}(f{+}g) = \mathfrak{C}_{f+g}\Big(\ell\mathrm{t}(f{+}g)\Big) \\ &\quad\overset{(C2.3.2)}{=} [\mathfrak{C}_f]\Big(\ell\mathrm{t}(f{+}g)\Big) + [\mathfrak{C}_g]\Big(\ell\mathrm{t}(f{+}g)\Big) \\ &\quad\overset{(L2.3.10)}{=} \mathfrak{C}_f\Big(\ell\mathrm{t}(f)\Big) + \mathfrak{C}_g\Big(\ell\mathrm{t}(g)\Big) = \ell\mathrm{c}(f){+}\ell\mathrm{c}(g). \end{aligned} \qquad \square$$

Unlike the classical case, the leading term function is not, in general, stable under multiplication of extended polynomials. Fortunately there are conditions which are not burdensome for the theory of Gröbner bases in $\mathcal{P}(e)$ to assure the stability of the leading term under multiplication.

Lemma 2.3.11. Let $f,g{\in}\mathcal{P}(e)\backslash\{0\}$, $s{\in}\mathbb{T}(e)^*$. If $\ell\mathrm{t}(f){\cdot}\ell\mathrm{t}(g){\neq}0$ then $f{\cdot}g \neq 0$ and $\ell\mathrm{t}(f{\cdot}g) = \ell\mathrm{t}(f){\cdot}\ell\mathrm{t}(g)$.

Proof.
Let $\ell\mathrm{t}(f)\ell\mathrm{t}(g){\neq}0$. Then

$$f{\cdot}g = \underbrace{\ell\mathrm{m}(f){\cdot}\ell\mathrm{m}(g)}_{\substack{\| \\ \overbrace{\underbrace{\ell\mathrm{t}(f)\ell\mathrm{t}(g)}_{\substack{\nparallel \\ 0}}\cdot\,\underbrace{\ell\mathrm{c}(f)\ell\mathrm{c}(g)}_{\substack{\nparallel\ \Bbbk \text{ is a field} \\ 0}}}\neq 0}} + \overbrace{\sum_{\substack{v\in\mathrm{supp}(f),\, w\in\mathrm{supp}(g) \\ v\neq\ell\mathrm{t}(f)\,\vee\, w\neq\ell\mathrm{t}(g)}} \mathfrak{M}_f(v)\cdot\mathfrak{M}_g(w)}^{\substack{h \\ \|}}, \qquad (*)$$

by the linear independence of extended terms it is sufficient to show that $u{<}\ell\mathrm{t}(f){\cdot}\ell\mathrm{t}(g)$ for all $u{\in}\,\mathrm{supp}\,(h)$. It is quite clear that the latter condition is fulfilled if $v{\cdot}w\underset{\mathbb{T}}{<}\ell\mathrm{t}(f){\cdot}\ell\mathrm{t}(g)$ for any $v{\in}\,\mathrm{supp}\,(f)$, $w{\in}\,\mathrm{supp}\,(g)$ such that $v{\neq}\ell\mathrm{t}(f)$ or $w{\neq}\ell\mathrm{t}(g)$. Take $v{\in}\,\mathrm{supp}\,(f)$, $w{\in}\,\mathrm{supp}\,(g)$. Assume that $v{\neq}\ell\mathrm{t}(f)$. Then $v{<}\ell\mathrm{t}(f)$, $w{\leq}\ell\mathrm{t}(g)$ and, by the quasi-monotonicity of the order $<$ under multiplication of terms, $v{\cdot}w < \ell\mathrm{t}(f){\cdot}\ell\mathrm{t}(g)$. The case $w{\neq}\ell\mathrm{t}(g)$ is treated in the same way. □

Definition 2.3.11. The leading term function with the property stated in part a) of the latter lemma is called **quasi-stable** (under multiplication of polynomials).

The important particular cases of the latter lemma are as follows.

Corollary 2.3.6. Let $g{\in}\mathcal{P}(e)\backslash\{0\}$ and $s{\in}\mathbb{T}(e)^*$. The following holds:

- a) if $s{\cdot}\ell\mathrm{t}(g){\neq}0$ then $sg{\neq}0$ and $\ell\mathrm{t}(sg){=}s{\cdot}\ell\mathrm{t}(g)$,
- b) if $s{\in}\mathbb{T}$ then $sg{\neq}0$ and $\ell\mathrm{t}(sg){=}s{\cdot}\ell\mathrm{t}(g)$,
- c) if $\alpha{\in}\Bbbk^*$ and $s{\cdot}\ell\mathrm{t}(g){\neq}0$ then $\alpha{\cdot}sg{\neq}0$ and $\ell\mathrm{t}(\alpha{\cdot}sg){=}s{\cdot}\ell\mathrm{t}(g)$.

Proof.
a) Let $s{\cdot}\ell\mathrm{t}(g){\neq}0$. Then $sg{\neq}0$ and $\ell\mathrm{t}(sg){=}s{\cdot}\ell\mathrm{t}(g)$ by lemma 2.3.11 with $f{=}s$.
b) It is a particular case of part *a*). Indeed, if $s{\in}\mathbb{T}$ then

$$\deg_e(s){+}\deg_e\Big(\ell\mathrm{t}(g)\Big) = 0{+}\deg_e\Big(\ell\mathrm{t}(g)\Big) = \deg_e\Big(\ell\mathrm{t}(g)\Big){<}\mathcal{B}$$

and hence $s{\cdot}\ell\mathrm{t}(g){\neq}0$ by part *a*) of lemma 2.3.1.
c) It is an easy combination of part *a*) with lemma 2.3.9. Indeed, if $s{\cdot}\ell\mathrm{t}(g){\neq}0$ then, by part *a*), $s{\cdot}g{\neq}0$ and hence, by lemma 2.3.9,

$$\alpha{\cdot}sg{\neq}0 \quad\text{and}\quad \ell\mathrm{t}(\alpha{\cdot}sg) \overset{(L2.3.9)}{=} \ell\mathrm{t}(sg) \overset{a)}{=} s{\cdot}\ell\mathrm{t}(g). \qquad \square$$

Corollary 2.3.7. Let $g{\in}\mathcal{P}(e){\setminus}\{0\}$.

- *a*) If $f{\in}\mathcal{P}(e){\setminus}\{0\}$ and $\ell\mathrm{t}(f){\cdot}\ell\mathrm{t}(g){\neq}0$ then $f{\cdot}g{\neq}0$ and $\ell\mathrm{c}(f{\cdot}g){=}\ell\mathrm{c}(f){\cdot}\ell\mathrm{c}(g)$;
- *b*) If $s{\in}\mathbb{T}(e)^*$ and $s{\cdot}\ell\mathrm{t}(g){\neq}0$ then $sg{\neq}0$ and $\ell\mathrm{c}(sg){=}\ell\mathrm{c}(g)$;
- *c*) If $t{\in}\mathbb{T}$ then $tg{\neq}0$ and $\ell\mathrm{c}(tg){=}\ell\mathrm{c}(g)$;
- *d*) If $\alpha{\in}\Bbbk^*$, $s{\in}\mathbb{T}$ and $s{\cdot}\ell\mathrm{t}(g){\neq}0$ then $\alpha{\cdot}sg{\neq}0$ and $\ell\mathrm{c}(\alpha{\cdot}sg){=}\alpha{\cdot}\ell\mathrm{c}(g)$.

Proof.

- *a*) If $f,g{\in}\mathcal{P}(e){\setminus}\{0\}$ and $\ell\mathrm{t}(f){\cdot}\ell\mathrm{t}(g){\neq}0$ then, by lemma 2.3.11, $f{\cdot}g{\neq}0$ and, according to $(*)$ and the subsequent argumentation in the proof of lemma 2.3.11, the coefficient at the leading term $\ell\mathrm{t}(f){\cdot}\ell\mathrm{t}(g)$ in $f{\cdot}g$ is $\ell\mathrm{c}(f){\cdot}\ell\mathrm{c}(g)$.
- *b*) It is a particular case of part *a*) provided that the extended term $s{\in}\mathbb{T}(e)^*$ can be considered as a polynomials with $\ell\mathrm{c}(s){=}1$.
- *b*) It is a particular case of part *b*) provided that $t{\cdot}\ell\mathrm{t}(g){\neq}0$ by part *a*) of lemma 2.3.1.
- *d*) It is an easy combination of part *b*) with corollary 2.3.4. Indeed, if $s{\cdot}\ell\mathrm{t}(g){\neq}0$ then, by part *b*), $sg{\neq}0$ and hence, by corollary 2.3.4,

$$\alpha{\cdot}sg{\neq}0 \quad\text{and}\quad \ell\mathrm{c}(\alpha{\cdot}sg) \overset{(C2.3.4)}{=} \alpha{\cdot}\ell\mathrm{c}(sg) \overset{b)}{=} \alpha{\cdot}\ell\mathrm{c}(g). \qquad \square$$

2.3.3 Normal forms and reduction in $\mathcal{P}(e)$

Definition 2.3.12. The **top-reduces to** relation is defined as follows. Let $f, g \in \mathcal{P}(e)$ be arbitrary polynomials.

- Let $h \in \mathcal{P}(e) \setminus \{0\}$. Then f **top-reduces to** g **modulo** h **in one step** denoted
$$f \xrightarrow{h} g$$
if and only if $f \neq 0$, there exists $s \in \mathbb{T}(e)^*$ such that $\ell\mathrm{t}(f) = s \cdot \ell\mathrm{t}(h)$ and
$$g = f - \alpha \cdot s \cdot h, \quad \text{where} \quad \alpha = \frac{\ell\mathrm{c}(f)}{\ell\mathrm{c}(h)} \in \mathbb{k}^*.$$

- Let $F \subseteq \mathcal{P}(e) \setminus \{0\}$. Then f **top-reduces to** g **modulo** F **in one step** if and only if there exists $h \in F$ such that $f \xrightarrow{h} g$.

- f **top-reduces to** g **modulo** F if and only if
$$f \xrightarrow{F}_* g$$
where $\xrightarrow{F}_*$ is the reflexive-transitive closure of $\xrightarrow{F}$.

- A finite sequence of $N+1 \in \mathbb{N}$ polynomials $\{f_0, \ldots, f_N\} \subseteq \mathcal{P}(e)$ such that $f_o = f$, $f_i \xrightarrow{F} f_{i+1}$ for $0 \leq i < N$ and $f_N = g$
$$f = f_o \xrightarrow{F} f_1 \xrightarrow{F} \cdots \xrightarrow{F} f_i \xrightarrow{F} f_{i+1} \xrightarrow{F} \cdots \xrightarrow{F} f_N = g$$
is called **the top-reduction sequence from** f **to** g **modulo** F. It is clear that $f \xrightarrow{F}_* g$ if and only if there exists a top-reduction sequence from f to g modulo F according to what the reflexive-transitive closure of a relation is.

Remark 2.3.1. The essential part of the definition of the top-reducibility above is the condition $\ell\mathrm{t}(f) = s \cdot \ell\mathrm{t}(h)$. Because of the existence of zero-divisors among extended terms $\mathbb{T}(e)$, we must be careful about this. If we required $\ell\mathrm{t}(f) = \ell\mathrm{t}(s \cdot h)$, we would lose the vital criterion that $f \in \mathcal{P}(e)$ top-reduces to some $g \in \mathcal{P}(e)$ if and only if $\ell\mathrm{t}(f) \in \langle \ell\mathrm{t}(F) \rangle_{\mathcal{P}(e)}$ and get substantial difficulties in the subsequent theory of Gröbner bases for the algebra $\mathcal{P}(e)$ developed in this section. On the other hand, the equality $\ell\mathrm{t}(f) = \ell\mathrm{t}(s \cdot h)$ follows from $0 \neq \ell\mathrm{t}(f) = s \cdot \ell\mathrm{t}(h)$ by part $a)$ of corollary 2.3.6. The latter dependence can be formulated more precisely as the following rather technical but crucial statement.

Lemma 2.3.12. Let $g{\in}\mathcal{P}(e)$, $f,h{\in}\mathcal{P}(e)\backslash\{0\}$. Assume that $f \xrightarrow{h} g$ and let $s{\in}\mathbb{T}(e)^*$ be such that $\ell\mathrm{t}(f){=}s{\cdot}\ell\mathrm{t}(h)$ provided by the definition of one step top-reduction. Denote $\alpha{=}\ell\mathrm{c}(f)/\ell\mathrm{c}(h)$. Then

a) $\alpha{\cdot}s{\cdot}h \neq 0$,

b) $\ell\mathrm{t}(\alpha{\cdot}s{\cdot}h) = \ell\mathrm{t}(s{\cdot}h) = s{\cdot}\ell\mathrm{t}(h) = \ell\mathrm{t}(f)$,

c) $\ell\mathrm{c}(\alpha{\cdot}s{\cdot}h) = \alpha{\cdot}\ell\mathrm{c}(h) = \ell\mathrm{c}(f)$.

Proof. By definition of the leading term $\ell\mathrm{t}(f){\neq}0$. Then, by part *a)* of corollary 2.3.6

$$\ell\mathrm{t}(f) = s{\cdot}\ell\mathrm{t}(h) \neq 0 \quad \Longrightarrow \quad s{\cdot}h \neq 0 \;\text{ and }\; \ell\mathrm{t}(s{\cdot}h) = s{\cdot}\ell\mathrm{t}(h). \tag{1}$$

Therefore, by lemma 2.3.9, $\alpha{\neq}0$ and $s{\cdot}h{\neq}0$ follow

$$\alpha{\cdot}(s{\cdot}h){\neq}0 \;\text{ and }\; \ell\mathrm{t}\Big(\alpha{\cdot}(s{\cdot}h)\Big) = \ell\mathrm{t}(s{\cdot}h) \overset{(1)}{=} s{\cdot}\ell\mathrm{t}(h) = \ell\mathrm{t}(f) \tag{2}$$

which states parts *a)* and *b)*. Furthermore, by part *e)* of corollary 2.3.2,

$$\ell\mathrm{c}\Big(\alpha{\cdot}s{\cdot}h\Big) = \mathfrak{C}_{\alpha\cdot s\cdot h}\Big(\ell\mathrm{t}(\alpha{\cdot}s{\cdot}h)\Big) =$$
$$\alpha{\cdot}\,\mathfrak{C}_h\left(\frac{\ell\mathrm{t}(\alpha{\cdot}s{\cdot}h)}{s}\right) \overset{(2)}{=} \alpha{\cdot}\,\mathfrak{C}_h\left(\frac{s{\cdot}\ell\mathrm{t}(h)}{s}\right) = \alpha{\cdot}\,\mathfrak{C}_h\Big(\ell\mathrm{t}(h)\Big) = \alpha{\cdot}\ell\mathrm{c}(h) = \ell\mathrm{c}(f)$$

which gives part *c)*. $\square$

As a direct consequence of the technical statement above we get the usual property of the top-reduction relation which is the cornerstone for the termination of the subsequent reduction algorithm.

Lemma 2.3.13. If $f,g{\in}\mathcal{P}(e)\backslash\{0\}$, $F{\subseteq}\mathcal{P}(e)\backslash\{0\}$ are such that $f \xrightarrow{F} g$ then $\ell\mathrm{t}(g){<}\ell\mathrm{t}(f)$.

Proof.

Let $s{\in}\mathbb{T}(e)^*$ be such that $\ell\mathrm{t}(f){=}s{\cdot}\ell\mathrm{t}(h)$ provided by the definition of one step top-reduction. Denote $\alpha{=}\ell\mathrm{c}(f)/\ell\mathrm{c}(h)$. Then, by lemma 2.3.12,

$$\ell\mathrm{c}(\alpha{\cdot}s{\cdot}h) = \ell\mathrm{c}(f) \quad \text{and} \quad \ell\mathrm{t}(\alpha{\cdot}s{\cdot}h){=}\ell\mathrm{t}(f). \tag{1}$$

If $g{\neq}0$ then, by part *c)* of lemma 2.3.10,

$$\ell\mathrm{t}(g) = \ell\mathrm{t}(f{-}\alpha{\cdot}s{\cdot}h) < \max\Big(\ell\mathrm{t}(f), \ell\mathrm{t}(\alpha{\cdot}s{\cdot}h)\Big) \overset{(1)}{=} \ell\mathrm{t}(f).$$

$\square$

Corollary 2.3.8. Let $f, g \in \mathcal{P}(e)\backslash\{0\}$, $F \subseteq \mathcal{P}(e)\backslash\{0\}$ and $f \xrightarrow{F}_* g$. Then either $f = g$ or $\ell\mathrm{t}(g) < \ell\mathrm{t}(f)$.

Proof. The case $f = g$ is trivial. Otherwise proceed by inductive argument based on lemma 2.3.13. ☐

Definition 2.3.13. Let $f \in \mathcal{P}(e)\backslash\{0\}$ and $F \subseteq \mathcal{P}(e)\backslash\{0\}$. Then f is called **top-reducible modulo** F denoted $f \xrightarrow{F}$ if and only if there exists $g \in \mathcal{P}(e)$ such that f top-reduces to g modulo F in one step: $f \xrightarrow{F} g$.

Lemma 2.3.14. Let $f \in \mathcal{P}(e)\backslash\{0\}$, $F \subseteq \mathcal{P}(e)\backslash\{0\}$. Then $f \xrightarrow{F}$ if and only if $\ell\mathrm{t}(f) \in \left\langle \ell\mathrm{t}(F) \right\rangle_{\mathcal{P}(e)}$.

Proof.

$$\begin{array}{ccc}
\boxed{f \xrightarrow{F}} & \overset{\text{Def. 2.3.13}}{\Longrightarrow} & \boxed{\exists g \in \mathcal{P}(e) : f \xrightarrow{F} g} \\
\Uparrow \text{Def. 2.3.13} & & \Downarrow \text{Def. 2.3.12} \\
\boxed{\exists h \in F : f \xrightarrow{h} f - \frac{\ell\mathrm{c}(f)}{\ell\mathrm{c}(h)} \cdot \boxed{\frac{\ell\mathrm{t}(f)}{\ell\mathrm{t}(h)}} \cdot h} & \overset{\text{Def. 2.3.12}}{\Longleftarrow} & \boxed{\exists h \in F : \ell\mathrm{t}(h) | \ell\mathrm{t}(f)} \\
 & & \Updownarrow \text{Lemma 2.3.4} \\
 & & \boxed{\ell\mathrm{t}(f) \in \langle \ell\mathrm{t}(F) \rangle_{\mathcal{P}(e)}}
\end{array}$$

☐

Definition 2.3.14. A polynomial $g \in \mathcal{P}(e)$ is called **top-minimal modulo** $F \subseteq \mathcal{P}(e)\backslash\{0\}$ if and only if it is not top-reducible.

Lemma 2.3.15. Let $f \in \mathcal{P}(e)$ and $F \subseteq \mathcal{P}(e)\backslash\{0\}$. Then f is top-minimal modulo F if and only if

$$f = 0 \quad \text{or} \quad \ell\mathrm{t}(f) \notin \left\langle \ell\mathrm{t}(F) \right\rangle_{\mathcal{P}(e)}.$$

Proof. If $f = 0$ then f is not top-reducible according to definition 2.3.13. If $f \neq 0$ then f is not top-reducible if and only if $\ell\mathrm{t}(f) \notin \left\langle \ell\mathrm{t}(F) \right\rangle_{\mathcal{P}(e)}$ by lemma 2.3.14. ☐

The following direct consequence of lemma 2.3.13 is the last step towards the correctness and the termination of the reduction algorithm below.

Lemma 2.3.16. For any $f \in \mathcal{P}(e)$ and $F \subseteq \mathcal{P}(e)\backslash\{0\}$

a) there is no infinite sequence of polynomials $f_i \in \mathcal{P}(e)$ such that $f_o = f$ and $f_i \xrightarrow{F} f_{i+1}$ for all $i \in \mathbb{N}_o$;

b) there exists $g \in \mathcal{P}(e)$ such that $f \xrightarrow{F}_* g$ and g is top-minimal modulo F.

Proof. Let $\Gamma = \{f_o, \dots, f_i, \dots\}$ be an infinite sequence of polynomials $\mathcal{P}(e)$ such that $f_o = f$ and $f_i \xrightarrow{F} f_{i+1}$ for all $i \in \mathbb{N}_o$. Then, every polynomial in the sequence is top-reducible modulo F and hence, by definition 2.3.13, $\Gamma \subseteq \mathcal{P}(e) \setminus \{0\}$. By lemma 2.3.13, $\ell\mathrm{t}(f_{i+1}) < \ell\mathrm{t}(f_i)$ for all $i \in \mathbb{N}_o$ resulting to an infinite strictly descending chain

$$\cdots \ell\mathrm{t}(f_{i+1}) < \ell\mathrm{t}(f_i) < \cdots < \ell\mathrm{t}(f_o)$$

which contradicts the well-ordering property of the extended term order $<$. Therefore there exists $N \in \mathbb{N}_o$ such that f_N is top-minimal modulo F. Let $g = f_N$. By construction of Γ, $f \xrightarrow{F}_* g$ according to definition 2.3.12. □

In view of the latter lemma, given $f \in \mathcal{P}(e)$ and $F \subseteq \mathcal{P}(e) \setminus \{0\}$ the incremental construction of a top-reduction sequence of f modulo F can be turned to an algorithm to compute a top-minimal modulo F element. This algorithm can be considered as a simplification procedure on the equivalence classes modulo the ideal generated by F as follows.

Lemma 2.3.17. Let $f, g \in \mathcal{P}(e)$ and $F \subseteq \mathcal{P}(e) \setminus \{0\}$ be such that $f \xrightarrow{F}_* g$. Then $f - g \in \langle F \rangle_{\mathcal{P}(e)}$.

Proof. Let $\{f_0, \dots, f_N\} \subseteq \mathcal{P}(e)$ be a top-reduction sequence of f to g modulo F. Then $f_o = f$, $f_N = g$ and $f_i \xrightarrow{F} f_{i+1}$ for $0 \leq i < N$. By the definition of one step top-reduction, for each $0 \leq i < N$ there exist $\alpha_i \in \Bbbk^*$, $s_i \in \mathbb{T}(e)^*$ and $h_i \in F$ such that $f_{i+1} = f_i - \alpha_i \cdot s_i \cdot h_i$. Therefore

$$f - g = f_o - f_N = \sum_{0 \leq i < N} \alpha_i \cdot s_i \cdot h_i \in \langle F \rangle_{\mathcal{P}(e)}. \qquad \square$$

According to lemma 2.3.16 the following reduction algorithm terminates in finitely many steps for any $f \in \mathcal{P}(e)$ and $F \subseteq \mathcal{P}(e)$ resulting in a top-minimal modulo F polynomial.

The outcome of the algorithm has properties, important enough to deserve a separate definition which also explains the appellation "`NF`" of the algorithm.

Algorithm 2.3.1 The reduction algorithm `NF` in $\mathcal{P}(e)$

Require: $f\in\mathcal{P}(e)$, $F\subseteq\mathcal{P}(e)\setminus\{0\}$.
Ensure: $g\in\mathcal{P}(e)$ is a top-minimal modulo F polynomial.

$g := f$
while $g\neq 0$ and $\ell\mathrm{t}(g)\in\langle\ell\mathrm{t}(F)\rangle_{\mathcal{P}(e)}$ **do**
  choose $h\in F$ such that $\ell\mathrm{t}(h)|\ell\mathrm{t}(g)$
  $g := g - \frac{\ell\mathrm{m}(g)}{\ell\mathrm{m}(h)}\cdot h$
end while

Definition 2.3.15. Given $f\in\mathcal{P}(e)$ and $F\subseteq\mathcal{P}(e)\setminus\{0\}$. A polynomial $g\in\mathcal{P}(e)$ such that

$$f-g \in \langle F\rangle_{\mathcal{P}(e)} \quad \text{and} \qquad (NF_1)$$

$$g \quad \text{is top-minimal modulo} \quad F \qquad (NF_2)$$

is called a **normal form** of f modulo F. The set of normal forms is written $\mathrm{NF}_F(f)$ and, according to the forthcoming lemma 2.3.18, $\mathrm{NF}_F(f)\neq\emptyset$. Also note that, a normal form is not necessary unique. Despite this fact we allow a sloppy notation $g=\mathrm{NF}_F(f)$ instead of $g\in\mathrm{NF}_F(f)$ to indicate the fact that g is a normal form of f modulo F.

Lemma 2.3.18. Let $f\in\mathcal{P}(e)$ and $F\subseteq\mathcal{P}(e)\setminus\{0\}$. Then there exists $g\in\mathrm{NF}_F(f)$ such that $f \xrightarrow{F}_* g$. In particular, $\mathrm{NF}_F(f)\neq\emptyset$.

Proof. By part b) of lemma 2.3.16 there exists $g\in\mathcal{P}(e)$ such that $f \xrightarrow{F}_* g$ and g is top-minimal modulo F. Then, by lemma 2.3.17, $f-g\in\langle F\rangle_{\mathcal{P}(e)}$. Therefore $g\in\mathrm{NF}_F(f)$ since it satisfies both conditions of definition 2.3.15 of the normal form. □

Lemma 2.3.19. For any $f\in\mathcal{P}(e)$ and $F\subseteq\mathcal{P}(e)\setminus\{0\}$ the algorithm `NF` computes a normal form of f modulo F in finitely many steps.

Proof.

Correctness: By lemma 2.3.15, the condition $g\neq 0$ and $\ell\mathrm{t}(g)\in\left\langle\ell\mathrm{t}(F)\right\rangle_{\mathcal{P}(e)}$ at the while loop is equivalent to g is top-reducible and, by lemma 2.3.4, there exists $h\in F$ such that $\ell\mathrm{t}(h)|\ell\mathrm{t}(g)$, hence there exists a polynomial $g'=g-\frac{\ell\mathrm{t}(g)}{\ell\mathrm{t}(h)}\cdot h$ required in the body of the loop. Furthermore, by the definition of one step top-reduction, the workflow of a single run of the while loop body can be considered as $g \xrightarrow{F} g'$ and the iteration of the loop can be described by a sequence of polynomials $\Gamma=\{g_i\}\subseteq\mathcal{P}(e)$ such that $g_o=g$ and $g_i \xrightarrow{F} g_{i+1}$.

Termination: The infinitely many runs of the while loop would result to an infinite sequence

Γ which contradicts lemma 2.3.16. Hence Γ is finite.

(NF_1): By lemma 2.3.17.

(NF_2): If Γ is finite then there exists $N \in \mathbb{N}_o$ such that the condition at the while loop is false i.e. $g_N{=}0$ or $\ell\mathrm{t}(g_N) \notin \langle \ell\mathrm{t}(F) \rangle_{\mathcal{P}(e)}$ which is, by lemma 2.3.15, equivalent to the fact that $g{:=}g_N$ is top-minimal modulo F. $\square$

For each polynomial $f \in \mathcal{P}(e)$ and $F \subseteq \mathcal{P}(e) \backslash \{0\}$ the reduction algorithm computes a normal form g of f modulo F. As a matter of fact, this is a special kind of a normal form where the condition (NF_1) is replaced by a weaker statement. It is clear that $f-g \in \langle F \rangle_{\mathcal{P}(e)}$ required by (NF_1) is equivalent to the fact that $f-g = \sum_{h \in F} q_h \cdot h$ with $q_h \in \mathcal{P}(e)$. However g is not an arbitrary normal form of f modulo F but such that $f \xrightarrow{F}_* g$. This restriction confines the latter expression to what is called a Gröbner representation.

Definition 2.3.16. Let $f \in \mathcal{P}(e) \backslash \{0\}$ and $F \subseteq \mathcal{P}(e) \backslash \{0\}$. A representation

$$f = \sum_{h \in F} q_h h \quad \text{with} \quad q_h \in \mathcal{P}(e)$$

such that

$$q_h \neq 0 \quad \text{follows} \quad 0 \neq \ell\mathrm{t}(q_h)\ell\mathrm{t}(h) \leq \ell\mathrm{t}(f) \quad \text{for all} \quad h \in F$$

is called a **Gröbner representation** of f in terms of F. The sum with $q_h{=}0$ for all $h \in F$ is, by definition, the trivial Gröbner representation of 0 in terms of F.

Lemma 2.3.20. If $f, g \in \mathcal{P}(e)$ and $F \subseteq \mathcal{P}(e) \backslash \{0\}$ are such that $f \xrightarrow{F}_* g$ then either $f-g{=}0$ or there exist polynomials $\{q_h |\, h \in F\} \subseteq \mathcal{P}(e)$ such that the representation

$$f-g = \sum_{h \in F} q_h h \tag{$*$}$$

is a Gröbner representation of $f-g$ in terms of F.

Proof. If $f{=}g$ then the representation $(*)$ with $q_h{=}0$ for all $h \in F$ is trivially the desired Gröbner representation. Assume $f \neq g$. Then $f \neq 0$ and it is easy to see that

$$\ell\mathrm{t}(f-g) = \ell\mathrm{t}(f). \tag{1}$$

Indeed, either $g{=}0$ or, by corollary 2.3.8, $\ell\mathrm{t}(g) < \ell\mathrm{t}(f)$ and hence, by part *b)* of lemma 2.3.10, $\ell\mathrm{t}(f-g) = \max\big(\ell\mathrm{t}(f), \ell\mathrm{t}(g)\big) = \ell\mathrm{t}(f)$. Consider a a sequence of polynomials $\{f_o, \ldots, f_N\} \subseteq \mathcal{P}(e)$ such that $f_o{=}f$, $f_k \xrightarrow{F} f_{k+1}$ for $0 \leq k < N$ and $f_N{=}g$ i.e. it is a top-reduction sequence of f to g modulo F

$$f = f_o \xrightarrow{F} \cdots \xrightarrow{F} f_k \xrightarrow{F} f_{k+1} \xrightarrow{F} \cdots \xrightarrow{F} f_N = g.$$

Denote $J{=}\{0,\ldots,N{-}1\}{\subseteq}\mathbb{N}_o$, $\Upsilon{=}\{f_o,\ldots,f_{N-1}\}$. Note that by the definition of one step top-reduction $\Upsilon{\subseteq}\mathcal{P}(e)\backslash\{0\}$ and there is $h_k{\in}F$ such that $\ell\mathrm{t}(h_k)|\ell\mathrm{t}(f_k)$ and

$$f_{k+1} = f_k{-}\alpha_i{\cdot}s_i{\cdot}h_k \quad \text{with} \quad \alpha_k = \frac{\ell\mathrm{c}(f_k)}{\ell\mathrm{c}(h_k)} \in \mathbb{k}^*,\ s_k = \frac{\ell\mathrm{t}(f_k)}{\ell\mathrm{t}(h_k)} \in \mathbb{T}(e)^* \tag{2}$$

for each $k{\in}J$. Let us emphasize the fact that, by the definition of one step top-reduction,

$$s_k{\cdot}\ell\mathrm{t}(h_k) = \ell\mathrm{t}(f_k) \neq 0 \quad \text{for} \quad k \in J. \tag{3}$$

By lemma 2.3.13 the leading terms of the polynomials of Υ form a strictly descending chain

$$\ell\mathrm{t}(f_{N-1}) < \cdots < \ell\mathrm{t}(f_{k+1}) < \ell\mathrm{t}(f_k) < \cdots < \ell\mathrm{t}(f_o) = \ell\mathrm{t}(f). \tag{4}$$

Denote $J_h{=}\{k{\in}\ J|\, h_k{=}h\}$ and $q_h{=}\sum\limits_{k\in J_h}\alpha_k{\cdot}s_k$ for each $h{\in}F$. It is easy to see by (2) that

$$f{-}g = \sum_{k\in J}\alpha_k{\cdot}s_k{\cdot}h_k = \sum_{h\in F}\left(\sum_{k\in J_h}\alpha_k{\cdot}s_k\right)h = \sum_{h\in F}q_h h.$$

Take $h{\in}F$ and $i,j{\in}J_h$ such that $i{<}j$. Then, by construction of J_h and (4),

$$s_j{\cdot}\ell\mathrm{t}(h) = s_j{\cdot}\ell\mathrm{t}(h_j) \overset{(3)}{=} \frac{\ell\mathrm{t}(f_j)}{\ell\mathrm{t}(h_j)}\ell\mathrm{t}(h_j) = \ell\mathrm{t}(f_j) \overset{(4)}{<}$$
$$\ell\mathrm{t}(f_i) = \frac{\ell\mathrm{t}(f_i)}{\ell\mathrm{t}(h_i)}\ell\mathrm{t}(h_i) \overset{(3)}{=} s_i{\cdot}\ell\mathrm{t}(h_i) = s_i{\cdot}\ell\mathrm{t}(h)$$

which follows, by lemma 2.3.8, $s_j{<}s_i$, or, in other words, the terms $\{s_k|\, k{\in}J_h\}$ form a strictly descending sequence. Therefore all terms $\{s_k|\, k{\in}J_h\}$ are distinct and hence, by the linear independence of extended terms and the construction of coefficients in (2), $q_h{=}0$ if and only if $J_h{=}\emptyset$. Assume $J_h{\neq}\emptyset$. Denote $m{=}\min(J_h)$. Then it is clear (trivially by the definition of the leading term and the latter argumentation) that $\ell\mathrm{t}(q_h){=}s_m$ and finally

$$\ell\mathrm{t}(q_h)\ell\mathrm{t}(h) = s_m\ell\mathrm{t}(h_m) \overset{(3)}{=} \ell\mathrm{t}(f_m) \overset{(4)}{\leq} \ell\mathrm{t}(f) \overset{(1)}{=} \ell\mathrm{t}(f{-}g). \qquad \square$$

Another valuable property of a Gröbner representation is the following.

Lemma 2.3.21. Let $f{\in}\mathcal{P}(e)\backslash\{0\}$ and $F{\subseteq}\mathcal{P}(e)\backslash\{0\}$ be such that there exists a Gröbner representation of f in terms of F

$$f = \sum_{h\in F}q_h h \quad \text{with} \quad q_h{\in}\mathcal{P}(e).$$

Denote $F'{=}\{h{\in}F\,|\,q_h{\neq}0\}{\neq}\emptyset$. Then

$$\ell\mathrm{t}(f) = \max\Big(\{\ell\mathrm{t}(q_h)\ell\mathrm{t}(h) \mid h \in F'\}\Big).$$

Proof. It is clear that $F'{\neq}\emptyset$. Let $h{\in}F'$. Then, by definition of the Gröbner representation, $\ell\mathrm{t}(q_h)\ell\mathrm{t}(h){\leq}\ell\mathrm{t}(f)$ and $\ell\mathrm{t}(q_h)\ell\mathrm{t}(h){\neq}0$ [2]. Then, by lemma 2.3.11, $q_h h{\neq}0$ and $\ell\mathrm{t}(q_h h){=}\ell\mathrm{t}(q_h)\ell\mathrm{t}(h)$. Hence

$$\max\Big(\{\ell\mathrm{t}(q_h h)|\, h \in F'\}\Big) = \max\Big(\{\ell\mathrm{t}(q_h)\ell\mathrm{t}(h)|\, h \in F'\}\Big) \leq \ell\mathrm{t}(f).$$

On the other hand, by part a) of lemma 2.3.10,

$$\ell\mathrm{t}(f) \leq \max\Big(\{\ell\mathrm{t}(q_h h)|\, h \in F'\}\Big).$$

□

2.3.4 Gröbner bases in $\mathcal{P}(e)$

The machinery developed in the previous subsections is appropriate and sufficiently powerful to state the theory of Gröbner bases for the ring of extended polynomials $\mathcal{P}(e)$. Indeed, it is an amazing fact that the theoretical part of the Gröbner bases existence and characterization is smoothly translated to $\mathcal{P}(e)$ from the classical setup. It is easy to see that almost all results of the previous subsections have a direct counterpart in the classical setup although the existence of zero-divisors among extended terms surely imposes extra conditions in the definitions, somewhat distorts the arithmetic and requires extra care in the proofs, all these effects do not appear to be cumbersome though slightly burdensome in the construction of the Gröbner bases theory being mostly encapsulated inside the previous subsections. Throughout this section $<$ denotes an arbitrary extended term order.

Definition 2.3.17. Let $I{\lhd}\mathcal{P}(e)$ be an ideal in $\mathcal{P}(e)$. A set $G{\subseteq}I{\setminus}\{0\}$ such that $\big\langle\ell\mathrm{t}(G)\big\rangle_{\mathcal{P}(e)}{=}\big\langle\ell\mathrm{t}(I)\big\rangle_{\mathcal{P}(e)}$ is called a **Gröbner basis of** I. An arbitrary set $G{\subseteq}\mathcal{P}(e){\setminus}\{0\}$ is called a **Gröbner basis** if and only if it is a Gröbner basis of the ideal $\langle G\rangle_{\mathcal{P}(e)}$ it generates.

[2]Without this condition required in the definition of a Gröbner representation the proof would fail and moreover the claim of the lemma would be wrong. By the linear independence of extended terms and the uniqueness of the support it is quite clear that the leading term of f must appear in the right-hand sum but there must be a condition to ensure that $\ell\mathrm{t}(f)$ appears as the product of leading terms or, otherwise, we would not be able to use Gröbner representations for characterization of Gröbner bases in the spirit of the classical Gröbner bases theory

According to the definition, the existence of a Gröbner basis is trivial. It is clear that the whole ideal without zero is a Gröbner basis of itself. The true advantage of Gröbner bases in the algebra of commutative polynomials as well as in $\mathcal{P}(e)$ is the existence of a finite Gröbner basis for any ideal.

Lemma 2.3.22. Let $I \lhd \mathcal{P}(e)$ be an ideal. Then there exists a finite Gröbner basis $G \subseteq I \setminus \{0\}$ of I.

Proof. By corollary 2.3.3 there exists a finite set of terms $B \subseteq \ell\mathrm{t}(I)$ such that $\langle B \rangle_{\mathcal{P}(e)} = \big\langle \ell\mathrm{t}(I) \big\rangle_{\mathcal{P}(e)}$. For each $b \in B$ take a polynomial $g_b \in I$ such that $\ell\mathrm{t}(g_b) = b$ provided that $b \in \ell\mathrm{t}(I)$. Then the finite set of polynomials $\{g_b |\, b \in B\} \subseteq I \setminus \{0\}$ is a Gröbner basis of I according to definition 2.3.17. ☐

In view of the latter lemma, a Gröbner basis is always a finite set by convention. The next theorem states the equivalent conditions for an arbitrary set of extended polynomials to be a Gröbner basis of a given ideal.

Theorem 2.3.2. Let $I \lhd \mathcal{P}(e)$ be an ideal in $\mathcal{P}(e)$. Given an arbitrary set $G \subseteq I \setminus \{0\}$ of non-zero polynomials in I the following conditions are equivalent.

a) G is Gröbner basis of I.

b) For any $f \in I \setminus \{0\}$ there exists $g \in G$ such that $\ell\mathrm{t}(g) | \ell\mathrm{t}(f)$.

c) For any $f \in I \setminus \{0\}$ f is top-reducible modulo G.

d) For any $f \in \mathcal{P}(e)$ $f \in I$ if and only if $\mathrm{NF}_G(f) = \{0\}$.

e) For any $f \in \mathcal{P}(e)$ $f \in I$ if and only if $f \xrightarrow{G}_* 0$.

f) For any $f \in \mathcal{P}(e)$ $f \in I$ if and only if there exists a Gröbner representation of f in terms of G.

Proof. Note that, by $G \subseteq I \setminus \{0\}$, for any $f \in \mathcal{P}(e)$

$$f \in \langle G \rangle_{\mathcal{P}(e)} \ \text{ follows } \ f \in I. \tag{1}$$

a)⇒*b*) By definition 2.3.17 the set $\ell\mathrm{t}(G)$ is a basis of $\big\langle \ell\mathrm{t}(I) \big\rangle_{\mathcal{P}(e)}$. If $f \in I \setminus \{0\}$ then trivially $\ell\mathrm{t}(f) \in \big\langle \ell\mathrm{t}(I) \big\rangle_{\mathcal{P}(e)}$ and hence, by lemma 2.3.4, there exists $g \in G$ such that $\ell\mathrm{t}(g) | \ell\mathrm{t}(f)$.

b)⇒*a*) By part *b*) of lemma 2.3.5 it is sufficient to show that $\ell\mathrm{t}(I) \subseteq \big\langle \ell\mathrm{t}(G) \big\rangle_{\mathcal{P}(e)}$. Let

$f{\in}I\backslash\{0\}$. By assumption there exists $g{\in}G$ such that $\ell\mathrm{t}(g)|\ell\mathrm{t}(f)$. Hence

$$\ell\mathrm{t}(f) = \ell\mathrm{t}(g)\frac{\ell\mathrm{t}(f)}{\ell\mathrm{t}(g)} \in \big\langle \ell\mathrm{t}(G)\big\rangle_{\mathcal{P}(e)}.$$

$b)\Rightarrow c)$ Let $f{\in}I\backslash\{0\}$. Then the condition $\ell\mathrm{t}(g)|\ell\mathrm{t}(f)$ for some $g{\in}G$ follows that $\ell\mathrm{t}(f){\in}\big\langle\ell\mathrm{t}(G)\big\rangle_{\mathcal{P}(e)}$ and hence, by lemma 2.3.14, f is top-reducible.

$c)\Rightarrow d)$ The case $f{=}0$ is trivial. Let $f{\in}I\backslash\{0\}$ and $r{\in}\mathcal{P}(e)$ be a normal form of f modulo G. Then, by part (NF_2) of definition 2.3.15 of a normal form, r is *top-minimal* and, by part (NF_1) of the same definition, $f{-}r{\in}I$ which follows $r{\in}I$ and hence, if $r{\neq}0$ then, by part $c)$, r is *top-reducible* which is a contradiction. The converse is trivial by (1) and part (NF_1) of the definition of a normal form.

$d)\Rightarrow e)$ Let $f{\in}I$. By lemma 2.3.18 there exists $r{\in}\mathrm{NF}_G(f)$ such that $f \xrightarrow{G}_* r$. Then $r{=}0$ by part $d)$. The converse is trivial by (1) and lemma 2.3.17.

$e)\Rightarrow f)$ Let $f{\in}I$. Then, by part $e)$, $f \xrightarrow{G}_* 0$ and hence, by lemma 2.3.20, there exists a Gröbner representation of f in terms of G. The converse is trivial by (1) and definition 2.3.16 of the Gröbner representation.

$f)\Rightarrow b)$ Let $f{\in}I\backslash\{0\}$ and

$$f = \sum_{h\in G} q_h h \quad \text{with} \quad q_h{\in}\mathcal{P}(e) \quad \text{for all} \quad h{\in}\mathcal{P}(e)$$

be a Gröbner representation of f in terms of G provided by part $f)$. Clearly not all q_h are zero. By lemma 2.3.21 there exists $g{\in}G$ such that $\ell\mathrm{t}(f) = \ell\mathrm{t}(q_g)\ell\mathrm{t}(g)$ which is clearly equivalent to $\ell\mathrm{t}(g)|\ell\mathrm{t}(f)$. ☐

The characterization of Gröbner bases in terms of a Gröbner representation makes the following natural fact trivial.

Lemma 2.3.23. Let $I{\lhd}\mathcal{P}(e)$ be an ideal. If $G{\subseteq}I\backslash\{0\}$ is a Gröbner basis of I then it is a basis of the ideal: $I{=}\langle G\rangle_{\mathcal{P}(e)}$.

Proof. By part $f)$ of theorem 2.3.2. ☐

2.3.5 The completion procedure in $\mathcal{P}(e)$

There are definitely several ways to introduce and argue the validity of a completion procedure which expands an arbitrary set $F{\subseteq}\mathcal{P}(e)\backslash\{0\}$ to a Gröbner basis $G{\supseteq}F$ such that $\langle F\rangle_{\mathcal{P}(e)}{=}\langle G\rangle_{\mathcal{P}(e)}$. One approach could be based on the notion of the leading terms syzygy module which is, in fact, a convenient tool for extending the theory of Gröbner bases to modules and polynomials over a ring (see e.g. [Möl88], [MMT92]). However we prefer a more constructive approach based on the completion procedure with the appropriate definition of a set of critical polynomials much like in the spirit how it is done in the classical Gröbner bases theory [Buc65], [Buc98] and generalized in [Buc84], [Bün90], [KRK88], [Pan89], [Sti87], [Sti91] which seems to be more convincing and also better suited for the subsequent use of the theory as a foundation for computing extended Gröbner bases in practice.

To establish a counterpart of the celebrated Buchberger's criterion for an arbitrary set of extended polynomials to be a Gröbner basis we would need some rather technical although convenient to depict the computational meaning of the developed theory notions and statements. The proof of the criterion is based on the construction of a certain bound on the maximal extended term among the leaders of the summands in a particular representation of an arbitrary polynomial in $\mathcal{P}(e)$ which relies on the stability of the leading term function under multiplication with certain terms. It is easy to see from part d) of theorem 2.3.1 and part b) of corollary 2.3.6 that the only way the leading term function of an extended polynomial can loose stability under multiplication with a term is a sufficiently high e-degree of the term. The following definition is the key to segregate such situations.

Definition 2.3.18. Let $f{\in}\mathcal{P}(e)$ and $F{\subseteq}\mathcal{P}(e)$.

- The e-orbit of f is the set
$$\mathrm{E}\,(f) = \Big\{e^k{\cdot}f \mid 0{\le}k{<}\mathcal{B}\Big\} \setminus \Big\{0\Big\}.$$
- The e-orbit of F is the union of the individual e-orbits of the elements of F
$$\mathrm{E}\,(F) = \bigcup_{h\in F} \mathrm{E}\,(h).$$

The following definition of an L-standard representation generalizes the notion of a Gröbner representation. It can be used as a replacement for the Gröbner representation to get a slightly more general characterization of Gröbner bases in $\mathcal{P}(e)$ than it is stated in part f) of theorem 2.3.2 and will also be used in the first formulation of the analogue of the crucial S-polynomials-like criterion theorem exactly as it is usually done in the classical setup of the Gröbner bases theory but it can not be used in the classical fashion in the derivation of this theorem by construction of an L-standard representation "stripping off" the leading monomials of the given representation because the "inner" terms of the polynomial co-factors can vanish upon multiplication with the leading terms of the basis polynomials.

Definition 2.3.19. Let $L\in\mathbb{T}(e)^*$, $f\in\mathcal{P}(e)\backslash\{0\}$ and $F\subseteq\mathcal{P}(e)\backslash\{0\}$. A polynomial combination

$$f = \sum_{h\in F} q_h\cdot h \quad \text{with} \quad q_h\in\mathcal{P}(e) \quad \text{for all} \quad h\in F$$

is called **an L-standard representation of f in terms of F** if and only if.

$$0\neq\ell\mathrm{t}(q_h)\cdot\ell\mathrm{t}(h)\leq L \quad \text{for all} \quad h\in F \quad \text{such that} \quad q_h\neq 0.$$

The representation with $q_h=0$ for all $h\in F$ is, by definition, the trivial L-standard representation of 0 in terms of F. It is clear that a Gröbner representation of f is an L-standard representation with $L=\ell\mathrm{t}(f)$.

To proceed further we would need a "more stable" but weaker analogue of the L-standard representation. It is the crucial notion in the derivation of our version of the S-polynomials based characterization of Gröbner bases and the subsequent supplementary constructions heading towards the actual algorithm for computing Gröbner bases in the algebra of extended polynomials.

Definition 2.3.20. Let $L\in\mathbb{T}(e)^*$, $f\in\mathcal{P}(e)\backslash\{0\}$ and $F\subseteq\mathcal{P}(e)\backslash\{0\}$. A monomial combination

$$f = \sum_{i\in N} \alpha_i\cdot t_i\cdot h_i \quad \text{with} \quad N \text{ - a finite subset of } \mathbb{N}, \quad \alpha_i\in\Bbbk^*,\ t_i\in\mathbb{T},\quad h_i\in F \quad \text{for all} \quad i\in N$$

is called **an L-standard stabilized monomial representation of f in terms of F** if and only if.

$$t_i\cdot\ell\mathrm{t}(h_i)\leq L \quad \text{for all} \quad i\in N.$$

The empty sum is, by definition, the trivial L-standard stabilized monomial representation of 0 in terms of F.

The following lemma summarizes the basic properties of the L-standard stabilized monomial representation.

Lemma 2.3.24. Given $f\in\mathcal{P}(e)\backslash\{0\}$, $F\subseteq\mathcal{P}(e)\backslash\{0\}$ and the arbitrary monomial representation

$$f = \sum_{i\in N} \alpha_i\cdot t_i\cdot h_i \quad \text{with} \quad N \text{ - a finite subset of } \mathbb{N}, \quad \alpha_i\in\Bbbk^*,\ t_i\in\mathbb{T},\quad h_i\in F \quad \text{for all} \quad i\in N. \qquad (*)$$

Denote

$$M = \max_{i\in N}\Big(t_i\cdot\ell\mathrm{t}(h_i)\Big).$$

Then

a) $\alpha_i{\cdot}t_i{\cdot}h_i{\neq}0$, $t_i{\cdot}\ell\mathrm{t}(h_i){\neq}0$ and $\ell\mathrm{t}(\alpha_i{\cdot}t_i{\cdot}h_i){=}t_i{\cdot}\ell\mathrm{t}(h_i)$ for all $i{\in}N$;

b) $\ell\mathrm{t}(f){\leq}M$;

c) the representation $(*)$ is an M-standard stabilized monomial representation of f in terms of F.

Proof.

a) Take any index $i{\in}N$. By corollary 2.3.1, $t_i{\cdot}\ell\mathrm{t}(h_i){\neq}0$. By lemma 2.3.9 and part *b)* of corollary 2.3.6,

$$\begin{aligned}\alpha_i{\cdot}t_i{\cdot}h_i &\overset{(L2.3.9),(C2.3.6)}{\neq} 0,\\ \ell\mathrm{t}(\alpha_i{\cdot}t_i{\cdot}h_i) &\overset{(L2.3.9)}{=} \ell\mathrm{t}(t_i{\cdot}h_i) \overset{(C2.3.6)}{=} t_i{\cdot}\ell\mathrm{t}(h_i).\end{aligned}$$

b) By part *a)* of lemma 2.3.10,

$$\ell\mathrm{t}(f) \leq \max_{i\in N}\Big(\ell\mathrm{t}(\alpha_i{\cdot}t_i{\cdot}h_i)\Big) \tag{1}$$

provided that $\alpha_i{\cdot}t_i{\cdot}h_i{\neq}0$ for all $i{\in}N$ according to part *a)* and $f{\neq}0$ by assumption. Moreover,

$$\max_{i\in N}\Big(\ell\mathrm{t}(\alpha_i{\cdot}t_i{\cdot}h_i)\Big) \overset{a)}{=} \max_{i\in N}\Big(t_i{\cdot}\ell\mathrm{t}(h_i)\Big) = M. \tag{2}$$

Then $\ell\mathrm{t}(f){\leq}M$ by (1) and (2).

c) By construction, $t_i{\cdot}\ell\mathrm{t}(h_i){\leq}M$ for any $i{\in}N$. □

The next property is the main value of the L-standard stabilized monomial representation for the theory of Gröbner bases in $\mathcal{P}(e)$: a non-trivial L-standard stabilized monomial representation can be used as an upper bound on the leading term of the polynomial which it represents.

Lemma 2.3.25. Given $f{\in}\mathcal{P}(e)\backslash\{0\}$, $F{\subseteq}\mathcal{P}(e)\backslash\{0\}$ and $L{\in}\mathbb{T}(e)^*$ such that there exists an L-standard stabilized monomial representation of f in terms of F. Then $\ell\mathrm{t}(f){\leq}L$.

Proof. Let

$$f = \sum_{i\in N}\alpha_i{\cdot}t_i{\cdot}h_i \quad \text{with} \quad N \text{ - a finite subset of } \mathbb{N}, \\ \alpha_i{\in}\Bbbk^*,\ t_i{\in}\mathbb{T},\quad h_i{\in}F \quad \text{for all} \quad i{\in}N$$

be an L-standard stabilized monomial representation of f in terms of F. By definition 2.3.20 of the L-standard stabilized monomial representation,

$$t_i{\cdot}\ell\mathrm{t}(f_i){\le}L \ \text{ for all } \ i{\in}N. \tag{1}$$

Then, by part b) of lemma 2.3.24,

$$\ell\mathrm{t}(f) \overset{(L2.3.24)}{\le} \max_{i\in N}\Big(t_i{\cdot}\ell\mathrm{t}(h_i)\Big) \overset{(1)}{\le} L. \qquad \square$$

As it is mentioned before the notion of the L-standard stabilized monomial representation is weaker than the L-standard representation. The following pretty technical lemma states precisely the nature of this weakness and will not be used outside this chapter. It formulates a relation between the L-standard representation and the L-standard stabilized monomial representation. As a matter of fact, the absolutely trivial in the classical setup statement stays correct but requires a somewhat tricky argumentation. Recall that e^* denotes the set of all non-vanishing powers of e.

Notation. Let $s{\in}\mathbb{T}(e)^*$. **The e-part of** s is written $\varepsilon(s){=}e^{\deg_e(s)}$. Note that, by part d) of theorem 2.3.1, $\varepsilon(s)$ is the unique extended term such that $s{=}\mathrm{T}(s)\varepsilon(s)$.

Lemma 2.3.26. Let $f{\in}\mathcal{P}(e)\backslash\{0\}$, $F{\subseteq}\mathcal{P}(e)\backslash\{0\}$ and $L{\in}\mathbb{T}(e)^*$. If there exists a L-standard representation of f in terms of F then there exists an L-standard stabilized monomial representation of f in terms of $\mathrm{E}\,(F)$.

Proof. Let

$$f=\sum_{h\in F} q_h{\cdot}h \ \text{ with } \ q_h{\in}\mathcal{P}(e) \ \text{ for all } \ h{\in}F \tag{$*$}$$

be an L-standard representation of f in terms of F. Then, by definition 2.3.19 of an L-standard representation,

$$0{\ne}\ell\mathrm{t}(q_h){\cdot}\ell\mathrm{t}(h){\le}L \ \text{ for all } \ h{\in}F \ \text{ such that } \ q_h{\ne}0. \tag{1}$$

For each $h{\in}F$ and $s{\in}\operatorname{supp}(q_h)$ denote

$$\begin{aligned}\alpha_{h,s} &= \mathfrak{C}_{q_h}(s)\in\mathbb{k}^* && \text{- the coefficient in } q_h \text{ at } s,\\ t_s &= \mathrm{T}(s){\in}\mathbb{T} && \text{- the main part of } s,\\ \varepsilon_s &= \varepsilon(s){\in}e^* && \text{- the } e\text{-part part of } s.\end{aligned}$$

Then, according to the distributive and the associative laws in $\mathcal{P}(e)$ together with the unique decomposition $s{=}\mathrm{T}(s){\cdot}\varepsilon(s)$ for every $s{\in}\mathbb{T}(e)^*$, $(*)$ is equivalent to

$$f = \sum_{h\in F} \left(\sum_{s\in \mathrm{supp}(q_h)} \alpha_{h,s}\cdot t_s\cdot \varepsilon_s \right) \cdot h = \sum_{h\in F} \sum_{s\in \mathrm{supp}(q_h)} \alpha_{h,s}\cdot t_s\cdot \varepsilon_s\cdot h =$$

$$\sum_{h\in F} \sum_{\substack{s\in \mathrm{supp}(q_h)\\ \varepsilon_s\cdot h\neq 0}} \alpha_{h,s}\cdot t_s\cdot (\varepsilon_s\cdot h).$$

We claim that that the representation

$$f = \sum_{h\in F} \sum_{\substack{s\in \mathrm{supp}(q_h)\\ \varepsilon_s\cdot h\neq 0}} \alpha_{h,s}\cdot t_s\cdot (\varepsilon_s\cdot h) \tag{$**$}$$

is an L-standard stabilized monomial representation of f in terms of $\mathrm{E}\,(F)$.

It is clear by construction that the sum in $(**)$ is finite and $\alpha_{h,s}\in C^*$, $t_s\in \mathbb{T}$, $\varepsilon_s\cdot h\in \mathrm{E}\,(F)$ for all $h\in F$ and $s\in \mathrm{supp}\,(q_h)$ such that $\varepsilon_s\cdot h\neq 0$. Take any $h\in F$ such that $q_h\neq 0$ and $s\in \mathrm{supp}\,(q_h)$ such that $\varepsilon_s\cdot h\neq 0$ provided by $(*)$ and the assumption $f\neq 0$. Denote $\alpha=\alpha_{h,s}$, $t=t_s$, $\varepsilon=\varepsilon_s$, $\mathcal{H}=\varepsilon\cdot h$. It is clear that $h\neq 0$. According to definition 2.3.20 of the L-standard stabilized monomial representation it is sufficient to show that

$$t\cdot \ell\mathrm{t}(\mathcal{H}) \le L. \tag{2}$$

By definition 2.3.10 of the leading term and part d) of lemma 2.3.3

$$\ell\mathrm{t}(\mathcal{H}) \overset{(D2.3.10)}{\in} \mathrm{supp}\,(\mathcal{H}) = \mathrm{supp}\,(\varepsilon\cdot h) \overset{(L2.3.3)}{=} \varepsilon * \mathrm{supp}\,(h) \overset{(\mathcal{H}\neq 0)}{\neq} \emptyset$$

and hence there exists $w\in \mathrm{supp}\,(h)$ such that $\varepsilon\cdot w\neq 0$ and $\ell\mathrm{t}(\mathcal{H})=\varepsilon\cdot w$. By construction

$$t\cdot \ell\mathrm{t}(\mathcal{H}) = t\cdot \varepsilon\cdot w = t_s\cdot \varepsilon_s\cdot w = s\cdot w \tag{3}$$

Also, by definition of the leading term, $s\in \mathrm{supp}\,(q_h)$ and $w\in \mathrm{supp}\,(h)$ follow

$$s \overset{(D2.3.10)}{\le} \ell\mathrm{t}(q_h) \ \text{ and } \ w \overset{(D2.3.10)}{\le} \ell\mathrm{t}(h).^3 \tag{4}$$

On the other hand, by assumption (1) that $(*)$ is an L-standard representation of f in terms of F,

$$\ell\mathrm{t}(q_h)\cdot \ell\mathrm{t}(h) \neq 0 \ \text{ and } \ \ell\mathrm{t}(q_h)\cdot \ell\mathrm{t}(h) \le L. \tag{5}$$

[3] Note that due to the possibility $s\cdot \ell\mathrm{t}(h)=0$ the inequality $w\le \ell\mathrm{t}(h)$ can be strict.

Then, by (3), (4), (5) and definition 2.3.7 of the quasi-monotonicity of the order $<$,

$$t{\cdot}\ell\mathrm{t}(\mathcal{H}) \overset{(3)}{=} s{\cdot}w \overset{(4),(5),(D2.3.7)}{\leq} \ell\mathrm{t}(q_h)\ell\mathrm{t}(h) \overset{(5)}{\leq} L$$

and hence we have shown (2). □

As an easy consequence of the previous lemma a non-trivial L-standard representation can be used as a bound on the leading term of the polynomial which it represents though it is also not hard to derive this fact directly.

Lemma 2.3.27. Let $f{\in}\mathcal{P}(e)\backslash\{0\}$, $F{\subseteq}\mathcal{P}(e)\backslash\{0\}$ and $L{\in}\mathbb{T}(e)^*$ be such that there exists an L-standard representation of f in terms of F. Then $\ell\mathrm{t}(f){\leq}L$.

Proof. If there exists an L-standard representation of f in terms of F then, by lemma 2.3.26, there exists an L-standard stabilized monomial representation of f in terms of $\mathrm{E}\,(F)$ and hence, by lemma 2.3.25, $\ell\mathrm{t}(f){\leq}L$. □

As a matter of luck the possible effects resulting from the existence of zero-divisors among extended terms spare the notion of the least common multiple of extended terms which also allows to borrow the notion of an S-polynomial from the classical Gröbner bases theory almost unchanged.

Definition 2.3.21. Let $s_1, s_2{\in}\mathbb{T}(e)^*$. The **least common multiple of** s_1 **and** s_2 is the extended term

$$\mathrm{lcm}(s_1, s_2) = \mathrm{lcm}\Big(\mathrm{T}(s_1), \mathrm{T}(s_2)\Big){\cdot}e^{\max\left(\deg_e(s_1),\deg_e(s_2)\right)}.$$

Lemma 2.3.28. Let $s_1, s_2{\in}\mathbb{T}(e)^*$. Denote

$$d_1{=}\deg_e(s_1),\ d_2{=}\deg_e(s_2),\ t_1{=}\mathrm{T}(s_1),\ t_2{=}\mathrm{T}(s_2),\ L{=}\mathrm{lcm}(s_1, s_2).$$

Then

a) $L{\neq}0$, $\mathrm{T}(L){=}\mathrm{lcm}(t_1, t_2)$ and $\deg_e(L){=}\max(d_1, d_2)$,

b) $s_1|L$ and $s_2|L$,

c) if $s{\in}\mathbb{T}(e)^*$ is such that $s_1|s$ and $s_2|s$ then $L|s$.

Proof.

a) By part *a)* of theorem 2.3.1, $d_1{<}\mathcal{B}$ and $d_2{<}\mathcal{B}$, hence $\max(d_1, d_2){<}\mathcal{B}$ follows $L{\neq}0$. Then, by definition 2.3.21 of the least common multiple and part *d)* of theorem 2.3.1,

$$\mathrm{T}(L) = \mathrm{lcm}(t_1, t_2) \ \text{ and } \ \deg_e(L) = \max(d_1, d_2).$$

b) By part *a*) and lemma 2.3.2,

$$\begin{array}{llll} t_1|\mathrm{T}(L) & \text{and} \;\; d_1\le\deg_e(L) & \text{follows} & s_1|L, \\ t_2|\mathrm{T}(L) & \text{and} \;\; d_2\le\deg_e(L) & \text{follows} & s_2|L. \end{array}$$

c) Denote $t{=}\mathrm{T}(s)$, $d{=}\deg_e(s)$. If $s_1|s$ and $s_2|s$ then, by lemma 2.3.2, $t_1|t$, $t_2|t$, $d_1{\le}d$ and $d_2{\le}d$. Hence, by part *a*) and the characteristic property of the least common multiple in $\mathbb{T}$,

$$\begin{array}{rclcr} t_1|t & \text{and} & t_2|t & \text{follows} & \mathrm{T}(L) = \mathrm{lcm}(t_1,t_2)|t, \\ d_1\le d & \text{and} & d_2\le d & \text{follows} & \deg_e(L) = \max(d_1,d_2)\le d \end{array}$$

and therefore, by lemma 2.3.2, $L|s$. □

Notation. For any $f_1, f_2{\in}\mathcal{P}(e)\backslash\{0\}$ the least common multiple of the leading terms of f_1 and f_2 is written

$$\mathrm{L}\,(f_1,f_2) = \mathrm{lcm}\Big(\ell\mathrm{t}(f_1),\ell\mathrm{t}(f_2)\Big) \in \mathbb{T}(e)^*.$$

The notion of an S-polynomial must be modified for use in our version of Gröbner bases in conjunction with the stabilized monomial representation. The presented version provides an easy way to get a new stabilized monomial representation from the given one upon cancellation of terms via addition.

Definition 2.3.22. Let $f_1, f_2{\in}\mathcal{P}(e)\backslash\{0\}$. The **stabilized S-polynomial of f_1 and f_2** is

$$\mathcal{S}_{\mathbb{T}}\,(f_1,f_2) = \begin{cases} \frac{\mathrm{L}(f_1,f_2)}{\ell\mathrm{m}(f_1)}\cdot f_1 - \frac{\mathrm{L}(f_1,f_2)}{\ell\mathrm{m}(f_2)}\cdot f_2, & \text{if } \deg_e\Big(\ell\mathrm{t}(f_1)\Big){=}\deg_e\Big(\ell\mathrm{t}(f_2)\Big) \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

Definition 2.3.23. For any $F{\subseteq}\mathcal{P}(e)\backslash\{0\}$ the set of pairs

$$\mathrm{SP}_{\mathbb{T}}\,(F){=}\Big\{(g_1,g_2)\;\Big|\; g_1,g_2{\in}\mathrm{E}\,(F),\ \deg_e\big(\ell\mathrm{t}(g_1)\big){=}\deg_e\big(\ell\mathrm{t}(g_2)\big),\ \mathcal{S}_{\mathbb{T}}\,(g_1,g_2){\ne}0\Big\}$$

is called **the set of stabilized S-pairs of F.**

To establish the basic properties of the stabilized S-polynomial we would need the following easy facts about extended terms.

Lemma 2.3.29. If $s_1, s_2{\in}\mathbb{T}(e)^*$ are such that $\deg_e(s_1){=}\deg_e(s_2)$ then

$$a)\quad \deg_e(s_1) = \deg_e\Big(\mathrm{lcm}\big(s_1,s_2\big)\Big) = \deg_e(s_2),$$

$$b)\quad \frac{\mathrm{lcm}\big(s_1,s_2\big)}{s_1}, \frac{\mathrm{lcm}\big(s_1,s_2\big)}{s_2} \in \mathbb{T}.$$

Proof.

a) By part *a)* of lemma 2.3.28, the condition $\deg_e(s_1){=}\deg_e(s_2)$ follows

$$\begin{array}{c} \deg_e\Big(\mathrm{lcm}\big(s_1,s_2\big)\Big) \\ \| \text{ Lemma 2.3.28}, a) \\ \deg_e(s_1) = \max\Big(\deg_e(s_1), \deg_e(s_2)\Big) = \deg_e(s_2) \end{array}$$

b) Denote

$$d = \deg_e\Big(\mathrm{lcm}\big(s_1,s_2\big)\Big)$$

and

$$t_1 = \frac{\mathrm{lcm}\big(s_1,s_2\big)}{s_1}, \qquad t_2 = \frac{\mathrm{lcm}\big(s_1,s_2\big)}{s_2}.$$

By part *b)* of lemma 2.3.28 the quotients t_1 and t_2 exist. By construction,

$$\mathrm{lcm}\big(s_1,s_2\big) = t_1{\cdot}s_1 = t_2{\cdot}s_2. \tag{2}$$

Then, by (1), (2) and part *b)* of lemma 2.3.1,

$$\begin{array}{rcl} \deg_e(t_1{\cdot}s_1) \overset{(L2.3.1)}{=} \deg_e(t_1){+}\deg_e(s_1) & = & \deg_e(t_1) + d \\ \| \ (2) & & \| \ (1) \\ d = \deg_e\big(\mathrm{lcm}(s_1,s_2)\big) & & d \\ \| \ (2) & & \| \ (1) \\ \deg_e(t_2{\cdot}s_2) \overset{(L2.3.1)}{=} \deg_e(t_2){+}\deg_e(s_2) & = & \deg_e(t_2) + d \end{array}$$

and hence $\deg_e(t_1){=}\deg_e(t_2){=}d{-}d{=}0$ which is equivalent to $t_1, t_2{\in}\mathbb{T}$. ∎

Corollary 2.3.9 (from lemma 2.3.29). Let

$$F{\subseteq}\mathcal{P}(e)\backslash\{0\} \quad \text{and} \quad (f_1,f_2){\in}\mathrm{SP}_{\mathbb{T}}(F).$$

Then

a) $\deg_e\big(\ell\mathrm{t}(f_1)\big) = \deg_e\Big(\mathrm{L}\,(f_1,f_2)\Big) = \deg_e\big(\ell\mathrm{t}(f_2)\big)$,

b) $\dfrac{\mathrm{L}\,(f_1,f_2)}{\ell\mathrm{t}(f_1)}, \ \dfrac{\mathrm{L}\,(f_1,f_2)}{\ell\mathrm{t}(f_2)} \in \mathbb{T}$.

Proof. By definition 2.3.23 of the set of stabilized S-pairs of F,

$$f_1, f_2{\in}\mathcal{P}(e)\backslash\{0\} \ \text{ and } \ \deg_e\big(\ell\mathrm{t}(f_1)\big) = \deg_e\big(\ell\mathrm{t}(f_2)\big).$$

The rest is by lemma 2.3.29. □

The following property is the key to use the stabilized S-polynomials for the completion procedure in $\mathcal{P}(e)$ developed in a conventional way by analysis of the circumstances when the leading terms of a polynomial expression can cancel upon addition and with a special care towards the possibility of cancellation upon multiplication of terms.

Lemma 2.3.30. Let $F{\subseteq}\mathcal{P}(e)\backslash\{0\}$, $(f_1, f_2){\in}\mathrm{SP}_{\mathbb{T}}(F)$ and $g{=}\mathcal{S}_{\mathbb{T}}(f_1, f_2)$. Then

$$\ell\mathrm{t}(g) < \mathrm{L}\,(f_1, f_2).$$

Proof. Denote $L{=}\mathrm{L}\,(f_1, f_2)$,

$$\begin{aligned}
\alpha_1 &= \frac{1}{\ell\mathrm{c}(f_1)} \in \Bbbk^*, \quad t_1 = \frac{L}{\ell\mathrm{t}(f_1)}, \quad g_1 = t_1{\cdot}f_1,\\
\alpha_2 &= \frac{1}{\ell\mathrm{c}(f_2)} \in \Bbbk^*, \quad t_2 = \frac{L}{\ell\mathrm{t}(f_2)}, \quad g_2 = t_2{\cdot}f_2.
\end{aligned}$$

By part *b*) of corollary 2.3.9,

$$t_1, t_2 \in \mathbb{T} \tag{1}$$

and, by construction,

$$L = t_1{\cdot}\ell\mathrm{t}(f_1) = t_2{\cdot}\ell\mathrm{t}(f_2). \tag{2}$$

By definition of the stabilized S-polynomial,

$$g = \alpha_1{\cdot}g_1 - \alpha_2{\cdot}g_2. \tag{$*$}$$

By part *b*) of corollary 2.3.6, $g_1, g_2{\in}\mathcal{P}(e)\backslash\{0\}$ provided that $f_1, f_2{\in}\mathcal{P}(e)\backslash\{0\}$ by definition 2.3.23 of the set of stabilized $S-$pairs of F and $t_1, t_2{\in}T$ by (1). Our goal is to show that $\ell\mathrm{t}(g_1){=}\ell\mathrm{t}(g_2)$. Indeed, by (1), (2) and part *b*) of corollary 2.3.6,

$$\begin{aligned}
\ell\mathrm{t}(g_1) = \ell\mathrm{t}(t_1{\cdot}f_1) \overset{(1),(C2.3.6)}{=} t_1{\cdot}\ell\mathrm{t}(f_1) \overset{(2)}{=} L \overset{(2)}{=}&\\
t_2{\cdot}\ell\mathrm{t}(f_2) \overset{(1),(C2.3.6)}{=} \ell\mathrm{t}(t_2{\cdot}f_2) = \ell\mathrm{t}(g_2).&
\end{aligned} \tag{3}$$

Note that, by construction, $\alpha_1{\neq}0$, $\alpha_2{\neq}0$ and hence, by lemma 2.3.9, $\alpha_1{\cdot}g_1{\neq}0$ and $\alpha_2{\cdot}g_2{\neq}0$. Then, by (3) and lemma 2.3.9,

$$\ell\mathrm{t}(\alpha_1{\cdot}g_1) \overset{(L2.3.9)}{=} \ell\mathrm{t}(g_1) \overset{(3)}{=} \ell\mathrm{t}(g_2) \overset{(L2.3.9)}{=} \ell\mathrm{t}(\alpha_2{\cdot}g_2). \tag{4}$$

and, by corollary 2.3.4 and part c) of corollary 2.3.7,

$$\begin{array}{lr} \ell\mathrm{c}(\alpha_1{\cdot}g_1) \overset{(C2.3.4)}{=} \alpha_1{\cdot}\ell\mathrm{c}(g_1) = \alpha_1{\cdot}\ell\mathrm{c}(t_1{\cdot}f_1) \overset{(C2.3.7)}{=} \alpha_1{\cdot}\ell\mathrm{c}(f_1) = \dfrac{\ell\mathrm{c}(f_1)}{\ell\mathrm{c}(f_1)} = 1 & \\ \hfill \| & (5) \\ \ell\mathrm{c}(\alpha_2{\cdot}g_2) \overset{(C2.3.4)}{=} \alpha_2{\cdot}\ell\mathrm{c}(g_2) = \alpha_2{\cdot}\ell\mathrm{c}(t_2{\cdot}f_2) \overset{(C2.3.7)}{=} \alpha_2{\cdot}\ell\mathrm{c}(f_2) = \dfrac{\ell\mathrm{c}(f_2)}{\ell\mathrm{c}(f_2)} = 1. & \end{array}$$

By $(*)$, (3), (4), (5), the assumption $g{\neq}0$ and part c) of lemma 2.3.10,

$$\ell\mathrm{t}\Big(\mathcal{S}_{\mathbb{T}}\left(f_1,f_2\right)\Big) = \ell\mathrm{t}(g) \overset{(*)}{=} \ell\mathrm{t}(\alpha_1{\cdot}g_1{-}\alpha_2{\cdot}g_2) \overset{(4),(5),(L2.3.10),(3)}{<} L. \qquad \square$$

Now we have all the necessary ingredients to state and validate the analogue of the S-polynomials criterion for an arbitrary set of non-zero polynomials in $\mathcal{P}(e)$ to be a Gröbner basis. The statement of the theorem below formulated in terms of the L-standard stabilized monomial representations is not readily suitable to be used as a foundation for a computational procedure which checks either a given finite set of extended polynomials is a Gröbner basis because the possible algorithmic procedure for constructing the appropriate L-standard stabilized monomial representations is not likely to be used in practice, the presented formulation is, in a sense, the purest possible in the frame of our narration. The equivalent and more helpful for practical computing versions of the theorem are an easy implication.

Theorem 2.3.3. Let $I{\lhd}\mathcal{P}(e)$ be an ideal in $\mathcal{P}(e)$ generated by a finite set $G{\subseteq}I{\setminus}\{0\}$. Then G is Gröbner basis of the ideal I if and only if each polynomial in $\mathrm{E}\left(G\right)$ has a Gröbner representation in terms of G and for each pair $(g_1,g_2){\in}\mathrm{SP}_{\mathbb{T}}\left(G\right)$ there exists an L-standard stabilized monomial representation of $\mathcal{S}_{\mathbb{T}}\left(g_1,g_2\right)$ in terms of $\mathrm{E}\left(G\right)$ with $L{<}\mathrm{L}\left(g_1,g_2\right)$.

Proof.

$\Longrightarrow$ Let $G{\subseteq}I{\setminus}\{0\}$ be a Gröbner basis of I.

Take any pair $(g_1,g_2){\in}\mathrm{SP}_{\mathbb{T}}\left(G\right)$. Denote $f{=}\mathcal{S}_{\mathbb{T}}\left(g_1,g_2\right){\neq}0$. It is clear by definition 2.3.22 of the stabilized $S{-}$polynomial that $f{\in}I$. Then, by part f) of theorem 2.3.2, there exists a Gröbner representation of f in terms of G. Denote $L{=}\ell\mathrm{t}(f)$. It is clear by definition 2.3.19, that the Gröbner representation of f in terms of G is an L-standard representation of f in terms of G and hence, by lemma 2.3.26, there exists an L-standard stabilized monomial representation of f in terms of $\mathrm{E}\left(G\right)$. On the other hand, by lemma 2.3.30, $L{<}\mathrm{L}\left(g_1,g_2\right)$.

Take any $f{\in}\mathrm{E}\left(G\right)$. Then, by part f) of theorem 2.3.2, there exists a Gröbner representation of f in terms of G.

$\Longleftarrow$ Let $f{\in}\mathcal{P}(e)$. The case $f{=}0$ is trivial. Assume $f{\neq}0$. By part b) of theorem 2.3.2 it is sufficient to show that there exists $g{\in}G$ such that $\ell\mathrm{t}(g)|\ell\mathrm{t}(f)$. It is clear that f can be written as

$$f = \sum_{h\in G} q_h{\cdot}h \quad \text{with} \quad q_h{\in}\mathcal{P}(e) \quad \text{for each} \quad h{\in}G$$

and, using the distributivity law in $\mathcal{P}(e)$, the latter representation can be turned into a sum with monomial multiples

$$f = \sum_{i\in N} \alpha_i{\cdot}s_i{\cdot}h_i \quad \text{with} \quad N \text{ - a finite subset of } \mathbb{N},\ \alpha_i{\in}\Bbbk^*,\ s_i{\in}\mathbb{T}(e)^*, \quad h_i{\in}G \ \text{ not necessary pairwise different for each } \ i{\in}N.$$

Denote

$$t_i{=}\mathrm{T}(s_i){\in}\mathbb{T},\ \varepsilon_i{=}\varepsilon(s_i){\in}e^* \text{ and } \mathcal{H}_i{=}\varepsilon_i{\cdot}h_i{\in}\mathrm{E}\,(G){\cup}\{0\}$$

for each $i{\in}N$. Take

$$N_o = \{i{\in}N \mid \mathcal{H}_i{\neq}0\} \ \subseteq \ N.$$

Note that $N_o{\neq}\emptyset$ by assumption $f{\neq}0$. Then, using the identity $s{=}\mathrm{T}(s){\cdot}\varepsilon(s)$ for any $s{\in}\mathbb{T}(e)^*$ and the associativity law in $\mathcal{P}(e)$,

$$f = \sum_{i\in N} \alpha_i{\cdot}t_i{\cdot}(\varepsilon_i{\cdot}h_i) = \sum_{i\in N_o} \alpha_i{\cdot}t_i{\cdot}\mathcal{H}_i \quad \text{with} \quad \alpha_i{\in}\Bbbk^*,\ t_i{\in}\mathbb{T} \text{ and } \mathcal{H}_i{\in}\mathrm{E}\,(G) \quad \text{for each} \ \ i{\in}N_o. \qquad (*)$$

W.l.o.g. assume that either $t_i{\neq}t_j$ or $\mathcal{H}_i{\neq}\mathcal{H}_j$ for all $i,j{\in}N_o$ such that $i{\neq}j$. By lemma 2.3.9 and part b) of corollary 2.3.6

$$\begin{aligned} &\alpha_i{\cdot}t_i{\cdot}\mathcal{H}_i \neq 0 \ \text{ and} \\ &\ell\mathrm{t}(\alpha_i{\cdot}t_i{\cdot}\mathcal{H}_i) \overset{(L2.3.9)}{=} \ell\mathrm{t}(t_i{\cdot}\mathcal{H}_i) \overset{(C2.3.6)}{=} t_i{\cdot}\ell\mathrm{t}(\mathcal{H}_i) \qquad \text{for all} \quad i{\in}N_o. \end{aligned} \qquad (1)$$

Denote

$$M = \max_{i\in N_o}\Big(t_i{\cdot}\ell\mathrm{t}(\mathcal{H}_i)\Big). \qquad (2)$$

Then, by the assumption $f{\neq}0$ and part b) of lemma 2.3.24,

$$\ell\mathrm{t}(f){\leq}M. \qquad (3)$$

Take

$$N_1 = \Big\{i{\in}N_o \mid t_i{\cdot}\ell\mathrm{t}(\mathcal{H}_i){=}M\Big\} \overset{(2)}{\neq} \emptyset. \qquad (4)$$

Then $(*)$ can be written as

$$f = \sum_{i \in N_1} \alpha_i \cdot t_i \cdot \mathcal{H}_i \quad + \sum_{i \in N_o \setminus N_1} \alpha_i \cdot t_i \cdot \mathcal{H}_i \tag{$**$}$$

where M does not appear among the terms of the summands of the second sum and all leading terms of the summands of the second sum are equal to M. Indeed, for any index $i \in N_o \setminus N_1$

$$\operatorname{supp}(\alpha_i \cdot t_i \cdot \mathcal{H}_i) \overset{(1)}{=} t_i \cdot \ell\mathrm{t}(\mathcal{H}_i) \overset{(4)}{<} M$$

and hence, according to definition of the leading term as the maximal element of the support and the fact that $<$ is a total order on extended terms,

$$M \notin \operatorname{supp}(\alpha_i \cdot t_i \cdot \mathcal{H}_i).$$

Therefore

$$M \notin \bigcup_{i \in N_o \setminus N_1} \operatorname{supp}(\alpha_i \cdot t_i \cdot \mathcal{H}_i). \tag{5}$$

On the other hand,

$$\ell\mathrm{t}(\alpha_i \cdot t_i \cdot \mathcal{H}_i) \overset{(1)}{=} t_i \cdot \ell\mathrm{t}(\mathcal{H}_i) \overset{(4)}{=} M \quad \text{for any} \quad i \in N_1. \tag{6}$$

Consider the cases $\ell\mathrm{t}(f) < M$ and $\ell\mathrm{t}(f) = M$. According to (3), the case $M < \ell\mathrm{t}(f)$ is impossible.

$\boxed{\ell\mathrm{t}(f) < M}$: If $\ell\mathrm{t}(f) < M$ then, according to the fact that $<$ is a global order on extended terms, M does not appear in the left side of $(**)$ and hence, according to (5), (6) and the linear independence of extended terms, the sum of the leading monomials of all summands in the first sum of $(**)$ must add to zero and hence N_1 must contain at least two different indexes.
W.l.o.g. assume that $1, 2 \in N_1$. Denote

$$f_1 = \alpha_1 \cdot t_1 \cdot \mathcal{H}_1 \quad \text{and} \quad f_2 = \alpha_2 \cdot t_2 \cdot \mathcal{H}_2.$$

By construction of N_1, $\ell\mathrm{t}(f_1) = M = \ell\mathrm{t}(f_2)$ and hence, by part *b)* of lemma 2.3.1 and (1),

$$\begin{array}{ccccc}
\deg_e\big(\ell\mathrm{t}(\mathcal{H}_1)\big) = & \deg_e(t_1) + \deg_e\big(\ell\mathrm{t}(\mathcal{H}_1)\big) & \overset{(L2.3.1)}{=} & \deg_e\big(t_1 \cdot \ell\mathrm{t}(\mathcal{H}_1)\big) & \overset{(1)}{=} \deg_e\big(\ell\mathrm{t}(f_1)\big) \\
 & \| \; t_1 \in \mathbb{T} & & & \| \\
 & 0 & & & \deg_e(M) \\
 & \| \; t_2 \in \mathbb{T} & & & \| \\
\deg_e\big(\ell\mathrm{t}(\mathcal{H}_2)\big) = & \deg_e(t_2) + \deg_e\big(\ell\mathrm{t}(\mathcal{H}_2)\big) & \overset{(L2.3.1)}{=} & \deg_e\big(t_2 \cdot \ell\mathrm{t}(\mathcal{H}_2)\big) & \overset{(1)}{=} \deg_e\big(\ell\mathrm{t}(f_2)\big).
\end{array} \tag{7}$$

According to (7), the stabilized S-polynomial of $\mathcal{H}_1$ and $\mathcal{H}_2$ is defined. By part c) of lemma 2.3.28, $\mathrm{L}(\mathcal{H}_1,\mathcal{H}_2)|M$ provided that $\ell\mathrm{t}(\mathcal{H}_1)|M$, $\ell\mathrm{t}(\mathcal{H}_2)|M$ according to (1). Also, by part b) of the same lemma, $\ell\mathrm{t}(\mathcal{H}_1)|\mathrm{L}(\mathcal{H}_1,\mathcal{H}_2)$ and $\ell\mathrm{t}(\mathcal{H}_2)|\mathrm{L}(\mathcal{H}_1,\mathcal{H}_2)$. Denote

$$t{=}\frac{M}{\mathrm{L}(\mathcal{H}_1,\mathcal{H}_2)},\ \ v_1{=}\frac{\mathrm{L}(\mathcal{H}_1,\mathcal{H}_2)}{\ell\mathrm{t}(\mathcal{H}_1)},\ v_2{=}\frac{\mathrm{L}(\mathcal{H}_1,\mathcal{H}_2)}{\ell\mathrm{t}(\mathcal{H}_2)}, \qquad c_1{=}\frac{1}{\ell\mathrm{c}(\mathcal{H}_1)},\qquad c_2{=}\frac{1}{\ell\mathrm{c}(\mathcal{H}_2)}. \tag{8}$$

By (7) and definition 2.3.21 of the least common multiple of extended terms

$$\deg_e\Big(\mathrm{L}(\mathcal{H}_1,\mathcal{H}_2)\Big) = \max\Big(\deg_e\big(\ell\mathrm{t}(\mathcal{H}_1)\big),\deg_e\big(\ell\mathrm{t}(\mathcal{H}_2)\big)\Big) \overset{(7)}{=} \deg_e\Big(\ell\mathrm{t}(\mathcal{H}_1)\Big) \overset{(7)}{=} \deg_e(M) \overset{(7)}{=} \deg_e\Big(\ell\mathrm{t}(\mathcal{H}_2)\Big).$$

Then $\deg_e(t){=}\deg_e(M){-}\deg_e\big(\mathrm{L}(\mathcal{H}_1,\mathcal{H}_2)\big){=}0$ and hence $t{\in}\mathbb{T}$. By the same argument, $v_1,v_2{\in}\mathbb{T}$ (cf. lemma 2.3.29). Furthermore, it is easy to see that

$$t{\cdot}v_1{\cdot}\ell\mathrm{t}(\mathcal{H}_1) = \frac{M}{\mathrm{L}(\mathcal{H}_1,\mathcal{H}_2)}{\cdot}\frac{\mathrm{L}(\mathcal{H}_1,\mathcal{H}_2)}{\ell\mathrm{t}(\mathcal{H}_1)}{\cdot}\ell\mathrm{t}(\mathcal{H}_1) = M = \frac{M}{\mathrm{L}(\mathcal{H}_1,\mathcal{H}_2)}{\cdot}\frac{\mathrm{L}(\mathcal{H}_1,\mathcal{H}_2)}{\ell\mathrm{t}(\mathcal{H}_2)}{\cdot}\ell\mathrm{t}(\mathcal{H}_2) = t{\cdot}v_2{\cdot}\ell\mathrm{t}(\mathcal{H}_2)$$

and hence, by the quasi-cancellation law in $\mathbb{T}(e)^*$ (lemma 2.3.7),

$$t{\cdot}v_1 = t_1 \ \text{ and } \ t{\cdot}v_2 = t_2.$$

Then

$$\begin{aligned} f_1+f_2 &= \alpha_1{\cdot}t_1{\cdot}\mathcal{H}_1+\alpha_2{\cdot}t_2{\cdot}\mathcal{H}_2 = \\ &\alpha_1{\cdot}tv_1{\cdot}\mathcal{H}_1+\alpha_2{\cdot}tv_2{\cdot}\mathcal{H}_2 = t{\cdot}\Big(\alpha_1{\cdot}v_1{\cdot}\mathcal{H}_1+\alpha_2{\cdot}v_2{\cdot}\mathcal{H}_2\Big) = \\ &\frac{\alpha_1}{c_1}{\cdot}t{\cdot}\underbrace{\Big(c_1{\cdot}v_1{\cdot}\mathcal{H}_1-c_2{\cdot}v_2{\cdot}\mathcal{H}_2\Big)}_{\substack{\|\text{ by definition 2.3.22}\\ \mathcal{S}_{\mathbb{T}}(\mathcal{H}_1,\mathcal{H}_2)}}+c_2{\cdot}\left(\frac{\alpha_1}{c_1}+\frac{\alpha_2}{c_2}\right){\cdot}t_2{\cdot}\mathcal{H}_2. \end{aligned} \tag{9}$$

Let $\mathcal{S}_{\mathbb{T}}(\mathcal{H}_1,\mathcal{H}_2){\neq}0$. By assumption there exists an L-standard stabilized monomial representation of $\mathcal{S}_{\mathbb{T}}(\mathcal{H}_1,\mathcal{H}_2)$ in terms of $\mathrm{E}(G)$

$$\mathcal{S}_{\mathbb{T}}(\mathcal{H}_1,\mathcal{H}_2) = \sum_{j\in J}\beta_j{\cdot}\tau_j{\cdot}\chi_j \ \text{ with } \ J \text{ - a finite subset of } \mathbb{N}, \quad \beta_j{\in}\Bbbk^*,\ \tau_j{\in}\mathbb{T},\ \chi_j{\in}\mathrm{E}(G), \quad t_j{\cdot}\chi_j{\leq}L \quad \text{for all } j{\in}J \tag{10}$$

and $L\in\mathbb{T}(e)^*$ such that
$$L < \mathrm{L}\,(\mathcal{H}_1, \mathcal{H}_2). \tag{11}$$
Then (cf. (9))
$$\frac{\alpha_1}{c_1}\cdot t\cdot \mathcal{S}_{\mathbb{T}}\,(\mathcal{H}_1, \mathcal{H}_2) = \sum_{j\in J} \frac{\alpha_1}{c_1}\beta_j\cdot t\tau_j\cdot\chi_j \tag{12}$$
where $\frac{\alpha_1}{c_1}\beta_j\in\mathbb{k}^*$, $t\tau_j\in\mathbb{T}$, $\chi_j\in\mathrm{E}\,(G)$ and hence, by lemma 2.3.9, part $b)$ of corollary 2.3.6, (8), (10), (11) and quasi-monotonicity of the order $<$,
$$\frac{\alpha_1}{c_1}\beta_j\cdot t\tau_j\cdot\chi_j \neq 0 \quad\text{and}$$
$$\ell\mathrm{t}\left(\frac{\alpha_1}{c_1}\beta_j\cdot t\tau_j\cdot\chi_j\right) \overset{(L2.3.9)}{=} \ell\mathrm{t}(t\tau_j\cdot\chi_j) \overset{(C2.3.6)}{=}$$
$$t\cdot\Big(\tau_j\cdot\ell\mathrm{t}(\chi_j)\Big) \overset{(10)}{<} t\cdot L \overset{(11)}{<} t\cdot\mathrm{L}\,(\mathcal{H}_1, \mathcal{H}_2) \overset{(8)}{=} M \tag{13}$$
for all $j\in J$. According to (13), the leading terms of all summands in the representation (12) are strictly less than M. Then (12) plugged into (9) gives a representation of the same form as (9) but with the number of summands with the leading term M less than 2. Furthermore, the updated (9) plugged into the first sum of $(**)$ gives, according to (1) and (13), a representation of the same form as $(**)$ but with strictly less elements in N_1. By inductive argument based on this transformation of $(**)$, the expression $(**)$ can be turned to an expression of the same form but with a strictly smaller bound M and hence, by the argument based on the well-ordering property of $<$, there exists a representation of the same form as $(**)$ but with $M{=}\ell\mathrm{t}(f)$.

$\boxed{\ell\mathrm{t}(f){=}M}$: W.l.o.g assume that $1\in N_1$. By construction of $(**)$, $\mathcal{H}_1\in\mathrm{E}\,(G)$. Then, by assumption there exists a Gröbner representation of $\mathcal{H}_1$ in terms of G
$$\mathcal{H}_1 = \sum_{h\in G} p_h\cdot h \quad\text{with}\quad p_h\in\mathcal{P}(e) \quad\text{for each}\quad h\in\mathcal{P}(e)$$
and $0 \neq \ell\mathrm{t}(p_h)\cdot\ell\mathrm{t}(h) {\leq} \ell\mathrm{t}(\mathcal{H}_1)$ for all $h\in\mathcal{P}(e)$ such that $p_h{\neq}0$. By lemma 2.3.21, there exists $h_o\in G$ such that $p_{h_o}{\neq}0$ and
$$\ell\mathrm{t}(\mathcal{H}_1) = \ell\mathrm{t}(p_{h_o})\cdot\ell\mathrm{t}(h_o). \tag{15}$$
Furthermore, by construction of N_1, the assumption $\ell\mathrm{t}(f){=}M$, (1) and (15),
$$\ell\mathrm{t}(f) = M \overset{(1)}{=} t_1\cdot\ell\mathrm{t}(\mathcal{H}_1) \overset{(15)}{=} t_1\cdot\ell\mathrm{t}(p_{h_o})\cdot\ell\mathrm{t}(h_o).$$
Therefore,

$$\boxed{\ell\mathrm{t}(h_o)|\ell\mathrm{t}(f) \text{ and } h_o{\in}G}\,.$$

□

The subsequent applications of the latter theorem 2.3.3 towards computational aspects of Gröbner bases in $\mathcal{P}(e)$ would benefit from a refined condition on the existence of a Gröbner representation for any element of the e-orbit of a set of polynomials based on the following simple observation.

Lemma 2.3.31. Let $F{\subseteq}\mathcal{P}(e)\backslash\{0\}$, $s{\in}\mathbb{T}(e)^*$ and $f{\in}F$. If $s{\cdot}\ell\mathrm{t}(f){\neq}0$ then sf has a Gröbner representation in terms of F.

Proof. If $s{\cdot}\ell\mathrm{t}(f){\neq}0$ then, by part $a)$ of corollary 2.3.6, $sf{\neq}0$, $\ell\mathrm{t}(sf){=}s{\cdot}\ell\mathrm{t}(f)$ and hence $s{\cdot}f$ is a Gröbner representation of sf in terms of F according to definition 2.3.16 of a Gröbner representation. □

In view of the latter lemma the e-orbit in the formulation of theorem 2.3.3 can naturally be replaced by the subset of "critical" elements of the e-orbit.

Definition 2.3.24. Let $f{\in}\mathcal{P}(e)$ and $F{\subseteq}\mathcal{P}(e)$.

- The critical e-orbit of f is the set

$$\mathrm{EC}\,(f) = \Big\{e^k{\cdot}f \;\Big|\; 0{\le}k{<}\mathcal{B},\; e^k{\cdot}\ell\mathrm{t}(f){=}0\Big\} \setminus \Big\{0\Big\} \quad \Big[\subseteq \mathrm{E}\,(f)\Big].$$

- The critical e-orbit of F is the union of the individual critical e-orbits of the elements of F

$$\mathrm{EC}\,(F) = \bigcup_{h\in F} \mathrm{EC}\,(h) \quad \Big[\subseteq \mathrm{E}\,(F)\Big].$$

Lemma 2.3.32. Let $F{\subseteq}\mathcal{P}(e)\backslash\{0\}$. If any polynomial in $\mathrm{EC}\,(F)$ has a Gröbner representation in terms of F then any polynomial in $\mathrm{E}\,(F)$ has a Gröbner representation in terms of F.

Proof. Take any $h{\in}\mathrm{E}\,(F)$. Let e^k with $0{\le}k{<}\mathcal{B}$ and $f{\in}\mathrm{E}\,(F)$ be such that $h{=}e^k{\cdot}f$ $[{\neq}0]$ provided by definition 2.3.18 of an e-orbit. There arise two possibilities:

if $e^k{\cdot}\ell\mathrm{t}(f){=}0$ then $h{\in}\mathrm{EC}\,(F)$ according to definition 2.3.24 and hence h has a Gröbner representation in terms of F by assumption;

if $e^k{\cdot}\ell\mathrm{t}(f){\neq}0$ then h has a Gröbner representation in terms of F according to lemma 2.3.31. □

The following equivalent formulations of theorem 2.3.3 can appear to be more suitable for the practical computing. These all are the easy consequences of the relations described in the present and the previous sections.

Corollary 2.3.10 (from theorem 2.3.3). Let $I \triangleleft \mathcal{P}(e)$ be an ideal in $\mathcal{P}(e)$ generated by a finite set $G \subseteq I \setminus \{0\}$ such that each polynomial in $\mathrm{EC}\,(G)$ has a Gröbner representation in terms of G. Then the following conditions are equivalent.

a) The basis G is a Gröbner basis of I.

b) For each pair $(g_1, g_2) \in \mathrm{SP}_{\mathbb{T}}\,(G)$ the stabilized $S-$polynomial $\mathcal{S}_{\mathbb{T}}\,(g_1, g_2)$ top-reduces to 0 modulo G.

c) For each pair $(g_1, g_2) \in \mathrm{SP}_{\mathbb{T}}\,(G)$ there exists a Gröbner representation of $\mathcal{S}_{\mathbb{T}}\,(g_1, g_2)$ in terms of G.

d) For each pair $(g_1, g_2) \in \mathrm{SP}_{\mathbb{T}}\,(G)$ there exists an L-standard representation of $\mathcal{S}_{\mathbb{T}}\,(g_1, g_2)$ in terms of G with $L < \mathrm{L}\,(g_1, g_2)$.

Proof.

a)⇒*b*) Let G be a Gröbner basis of I. Take any pair $(g_1, g_2) \in \mathrm{SP}_{\mathbb{T}}\,(G)$. Then $\mathcal{S}_{\mathbb{T}}\,(g_1, g_2) \in I$ according to definition 2.3.22 of the stabilized $S-$polynomial. Hence $\mathcal{S}_{\mathbb{T}}\,(g_1, g_2) \xrightarrow{G}_* 0$ by part *e*) of theorem 2.3.2.

b)⇒*c*) Let $(g_1, g_2) \in \mathrm{SP}_{\mathbb{T}}\,(G)$. If $\mathcal{S}_{\mathbb{T}}\,(g_1, g_2) \xrightarrow{G}_* 0$ then, by lemma 2.3.20, there exists a Gröbner representation of $\mathcal{S}_{\mathbb{T}}\,(g_1, g_2)$ in terms of G.

c)⇒*d*) Let $(g_1, g_2) \in \mathrm{SP}_{\mathbb{T}}\,(G)$. If there exists a Gröbner representation of $\mathcal{S}_{\mathbb{T}}\,(g_1, g_2)$ in terms of G then this Gröbner representation is trivially an L-standard representation of $\mathcal{S}_{\mathbb{T}}\,(g_1, g_2)$ in terms of G with $L = \ell\mathrm{t}(\mathcal{S}_{\mathbb{T}}\,(g_1, g_2))$ by definition 2.3.19 of the L-standard representation. Furthermore, according to lemma 2.3.30, $\ell\mathrm{t}\big(\mathcal{S}_{\mathbb{T}}\,(g_1, g_2)\big) < \mathrm{L}\,(g_1, g_2)$.

d)⇒*a*) Let $(g_1, g_2) \in \mathrm{SP}_{\mathbb{T}}\,(G)$. If there exists an L-standard representation of $\mathcal{S}_{\mathbb{T}}\,(g_1, g_2)$ in terms of G with $L < \mathrm{L}\,(g_1, g_2)$ then, by lemma 2.3.26, there exists an L-standard stabilized monomial representation of $\mathcal{S}_{\mathbb{T}}\,(g_1, g_2)$ in terms of $\mathrm{E}\,(G)$ with the same L. On the other hand, by lemma 2.3.32, the assumption that any polynomial in $\mathrm{EC}\,(G)$ has a Gröbner representation in terms of G follows that each polynomial in $\mathrm{E}\,(G)$ has a Gröbner representation in terms of G. Therefore G is a Gröbner basis of I by theorem 2.3.3. □

The forthcoming algorithm for computing Gröbner bases in $\mathcal{P}(e)$ based on part *b*) of the latter corollary computes a normal form for each $\mathrm{E}\,(F)$ and $\mathrm{SP}_{\mathbb{T}}\,(F)$ of the given set F of basis polynomials and enlarges the basis by the normal form if it is not zero. The performance of the algorithm in practical computing depends a lot on the number of normal forms of the corresponding stabilized $S-$polynomials it has to compute in the flow. There are two well-known criteria in the conventional theory of Gröbner bases referred as **Buchberger's first and second criteria** introduced in [Buc79] (see also [AL94] lemma 3.3.1 on page 125 and corollary 3.3.3 on page 127, [BW93] lemma 5.66 on page 222 and proposition 5.70 on page 223, [CLO97] proposition 4 on page 101 and proposition 10 on page 106, [Win96] theorem 8.5.1 on page 201 and theorem 8.5.2 on page 202) to avoid the treatment of certain $S-$polynomials if the respective $S-$pair has certain properties in its own or in relation with other S-pairs appeared in the flow of the algorithm.

Buchberger's first criterion states that an S-pair should not be treated in the flow of a Gröbner basis computation if the leading terms of the polynomials of the pair are disjoint because the corresponding S-polynomial can be top-reduced to zero by the polynomials of the pair. The usual proof of this criterion relies on the fact that $a|b$ follows $a{\leq}b$ for any terms a, b and an admissible ordering $<$. Unfortunately the latter fact is not true any more if the condition that 1 is the smallest term[4] is removed from the definition of an admissible term order. According to definition 2.3.9 of an extended term order, 1 is not required to be the smallest non-zero extended term and, in fact, both instantiations of the Gröbner bases theory of the algebra of extended polynomials in the next sections operate with term orders where the variable e has a negative weight (due to the subsequent applications of the theory for computing Gröbner bases with numerical coefficients where the degree of e acts as a measure how small the coefficients are and controls the mechanism of pivoting) and hence any positive degree of e must be smaller than 1 in a quasi-monotonous order due to the compatibility of the order with multiplication in the sense of lemma 2.3.6. The presented proof states a weaker but, due to theorem 2.3.3, sufficient for utilization in a Gröbner bases computation algorithm statement in terms of an appropriate L-standard stabilized monomial representation of the respective stabilized S-polynomial. The author does not know if the stronger statement about the zero-top-reducibility of such an S-polynomial is generally valid in the case of a quasi-monotonous order and would be grateful to receive an argument either it is true or a counterexample.

To keep the presented theory as self contained and at the same time as short as possible we present a stand-alone version of the notion of disjoint terms needed to formulate the first Buchberger's criterion for $\mathcal{P}(e)$ and few basic properties as they appear due to the separate treatment of terms in main variables.

[4] A term order with this condition is usually called a *global order*.

Definition 2.3.25. Let $s_1, s_2 \in \mathbb{T}(e)^*$. Then s_1 and s_2 are **disjoint** if and only if

$$\mathrm{lcm}(s_1, s_2) = s_1 \cdot s_2 \in \mathbb{T}(e)^*.$$

Note that, by part a) of lemma 2.3.28, the product of disjoint terms can not be zero.

Lemma 2.3.33. If $s_1, s_2 \in \mathbb{T}(e)^*$ are disjoint then either $s_1 \in \mathbb{T}$ or $s_2 \in \mathbb{T}$.

Proof. Let s_1 and s_2 be disjoint terms

$$\mathrm{lcm}(s_1, s_2) = s_1 \cdot s_2. \tag{$*$}$$

Denote $d_1 = \deg_e(s_1)$ and $d_2 = \deg_e(s_2)$. By part a) of lemma 2.3.28,

$$\deg_e\Big(\mathrm{lcm}(s_1, s_2)\Big) = \max(d_1, d_2). \tag{1}$$

On the other hand, by part b) of lemma 2.3.1,

$$\deg_e(s_1 \cdot s_2) = d_1 + d_2. \tag{2}$$

Hence, the condition $(*)$ means that

$$\max(d_1, d_2) \overset{(1)}{=} \deg_e\Big(\mathrm{lcm}(s_1, s_2)\Big) \overset{(*)}{=} \deg_e(s_1 \cdot s_2) \overset{(2)}{=} d_1 + d_2. \tag{3}$$

W.l.o.g. let $d_2 \le d_1$. Then, by (3), $d_2 = d_1 + d_2$ follows $d_1 = 0$ which is equivalent to $s_1 \in \mathbb{T}$. $\square$

Corollary 2.3.11 (from lemma 2.3.33). If the terms $s_1, s_2 \in \mathbb{T}(e)^*$ are disjoint and $\deg_e(s_1) = \deg_e(s_2)$ then $s_1, s_2 \in \mathbb{T}$.

Proof. By lemma 2.3.33

$$0 = \min\Big(\deg_e(s_1), \deg_e(s_2)\Big) = \deg_e(s_1) = \deg_e(s_2)$$

which is equivalent to $s_1, s_2 \in \mathbb{T}$. $\square$

In view of the previous corollary the application of Buchberger's first criterion formulated in the lemma below is restricted to the terms in main variables. At least this is not worse than in the classical setup of Gröbner bases theory although it is a challenge for the future to find a more general statement capable of dealing with "truly extended" terms which essentially involve the extended variable.

Lemma 2.3.34 (Buchberger's first criterion in $\mathcal{P}(e)$). Let $F \subseteq \mathcal{P}(e) \setminus \{0\}$ and $(g_1, g_2) \in \mathrm{SP}_{\mathbb{T}}(F)$. If the leading terms $\ell\mathrm{t}(g_1)$ and $\ell\mathrm{t}(g_2)$ are disjoint then the stabilized S-polynomial $\mathcal{S}_{\mathbb{T}}(g_1, g_2)$ has an L-standard stabilized monomial representation in terms of $\mathrm{E}(F)$ with $L < \mathrm{L}(g_1, g_2)$.

Proof. If $(g_1, g_2) \in \mathrm{SP}_{\mathbb{T}}(F)$ then $g_1, g_2 \in \mathbb{T}(e)^*$. Assume for simplicity that $\ell\mathrm{c}(g_1) = \ell\mathrm{c}(g_2) = 1$. [5] Denote

$$p_1 = g_1 - \ell\mathrm{t}(g_1) \in \mathcal{P}(e) \quad \text{and} \quad p_2 = g_2 - \ell\mathrm{t}(g_2) \in \mathcal{P}(e).$$

Note that each p_k is either 0 or $\ell\mathrm{t}(p_k) < \ell\mathrm{t}(g_k)$ for $k = 1, 2$. If $\ell\mathrm{t}(g_1)$ and $\ell\mathrm{t}(g_2)$ are disjoint then

$$\begin{aligned} \ell\mathrm{t}(g_2) &= \frac{\mathrm{L}(g_1, g_2)}{\ell\mathrm{t}(g_1)} = \frac{\ell\mathrm{t}(g_1) \cdot \ell\mathrm{t}(g_2)}{\ell\mathrm{t}(g_1)}, \\ \ell\mathrm{t}(g_1) &= \frac{\mathrm{L}(g_1, g_2)}{\ell\mathrm{t}(g_2)} = \frac{\ell\mathrm{t}(g_1) \cdot \ell\mathrm{t}(g_2)}{\ell\mathrm{t}(g_2)}. \end{aligned}$$

Then it is easy to see by definition 2.3.22 of the stabilized S-polynomial that

$$\begin{aligned} \mathcal{S}_{\mathbb{T}}(g_1, g_2) = \ell\mathrm{t}(g_2) \cdot g_1 - \ell\mathrm{t}(g_1) \cdot g_2 = (g_2 - p_2) \cdot g_1 - (g_1 - p_1) \cdot g_2 = \\ g_2 \cdot g_1 - p_2 \cdot g_1 - g_1 \cdot g_2 + p_1 \cdot g_2 = p_1 \cdot g_2 - p_2 \cdot g_1. \end{aligned} \tag{1}$$

Let

$$\begin{aligned} p_1 &= \sum_{s \in \mathrm{supp}(p_1)} \alpha'_s \cdot s \quad \text{with} \quad \alpha'_s \in \Bbbk^* \quad \text{for each} \quad s \in \mathrm{supp}(p_1), \\ p_2 &= \sum_{s \in \mathrm{supp}(p_2)} \alpha''_s \cdot s \quad \text{with} \quad \alpha''_s \in \Bbbk^* \quad \text{for each} \quad s \in \mathrm{supp}(p_2). \end{aligned}$$

Denote

$$\begin{aligned} t_s &= \mathrm{T}(s) \in \mathbb{T}, \ \varepsilon_s = \varepsilon(s) \in e^* \quad \text{for all} \quad s \in \mathrm{supp}(p_1) \cup \mathrm{supp}(p_2), \\ \mathcal{H}'_s &= \varepsilon_s \cdot g_2 \in \mathrm{E}(F) \quad \text{for all} \quad s \in \mathrm{supp}(p_1), \\ \mathcal{H}''_s &= \varepsilon_s \cdot g_1 \in \mathrm{E}(F) \quad \text{for all} \quad s \in \mathrm{supp}(p_2). \end{aligned}$$

[5] This assumption does not restrict the consideration in the present proof based on construction of an appropriate L-standard stabilized monomial representation for $\mathcal{S}_{\mathbb{T}}(g_1, g_2)$ because, as it is quite easy to see from definition 2.3.22 of the stabilized S-polynomial, $\mathcal{S}_{\mathbb{T}}(g_1, g_2) = \mathcal{S}_{\mathbb{T}}(\alpha_1 \cdot g_1, \alpha_2 \cdot g_2)$ for any $\alpha_1, \alpha_2 \in \Bbbk^*$ (a rigorous proof is by lemma 2.3.9). Hence we can safely assume that g_1 and g_2 have been multiplied by the corresponding non-zero constants to make $\ell\mathrm{c}(g_1) = \ell\mathrm{c}(g_2) = 1$.

Then (1) is equivalent to

$$
\begin{aligned}
\mathcal{S}_{\mathbb{T}}(g_1,g_2) &\overset{(1)}{=} p_1 \cdot g_2 - p_2 \cdot g_1 \\
&= \left(\sum_{s\in\operatorname{supp}(p_1)} \alpha'_s \cdot s\right)\cdot g_2 - \left(\sum_{s\in\operatorname{supp}(p_2)} \alpha''_s \cdot s\right)\cdot g_1 \\
&= \sum_{s\in\operatorname{supp}(p_1)} \alpha'_s \cdot t_s \cdot (\varepsilon_s \cdot g_2) + \sum_{s\in\operatorname{supp}(p_2)} \alpha''_s \cdot t_s \cdot (\varepsilon_s \cdot g_1) \\
&= \sum_{s\in\operatorname{supp}(p_1)} \alpha'_s \cdot t_s \cdot \mathcal{H}'_s + \sum_{s\in\operatorname{supp}(p_2)} \alpha''_s \cdot t_s \cdot \mathcal{H}''_s = \sum_{\substack{s\in\operatorname{supp}(p_1)\\ \mathcal{H}'_s\neq 0}} \alpha'_s \cdot t_s \cdot \mathcal{H}'_s + \sum_{\substack{s\in\operatorname{supp}(p_2)\\ \mathcal{H}''_s\neq 0}} \alpha''_s \cdot t_s \cdot \mathcal{H}''_s. \quad (2)
\end{aligned}
$$

We claim that the representation

$$
\mathcal{S}_{\mathbb{T}}(g_1,g_2) \overset{(2)}{=} \sum_{\substack{s\in\operatorname{supp}(p_1)\\ \mathcal{H}'_s\neq 0}} \alpha'_s \cdot t_s \cdot \mathcal{H}'_s + \sum_{\substack{s\in\operatorname{supp}(p_2)\\ \mathcal{H}''_s\neq 0}} \alpha''_s \cdot t_s \cdot \mathcal{H}''_s \tag{3}
$$

is the desired L-standard stabilized monomial representation of $\mathcal{S}_{\mathbb{T}}(g_1,g_2)$ in terms of $\mathrm{E}(F)$ with $L<\mathrm{L}(g_1,g_2)$.

Indeed, take any $s\in\operatorname{supp}(p_1)$ such that $\mathcal{H}'_s\neq 0$. Then $\operatorname{supp}(\mathcal{H}'_s)\neq\emptyset$ and, by part d) of lemma 2.3.3,

$$
\operatorname{supp}(\mathcal{H}'_s) = \varepsilon_s * \operatorname{supp}(g_2).
$$

Hence, there exists $w\in\operatorname{supp}(g_2)$ such that $\varepsilon_s\cdot w\neq 0$ and

$$
\ell\mathrm{t}(\mathcal{H}'_s) = \varepsilon_s \cdot w. \tag{4}
$$

It is clear that

$$
s < \ell\mathrm{t}(g_1) \quad \text{and} \quad w \leq \ell\mathrm{t}(g_2). \tag{5}
$$

On the other hand, $(g_1,g_2)\in\mathrm{SP}_{\mathbb{T}}(F)$ implies $\deg_e\big(\ell\mathrm{t}(g_1)\big)=\deg_e\big(\ell\mathrm{t}(g_2)\big)$. Then, by corollary 2.3.9, $\ell\mathrm{t}(g_1),\ell\mathrm{t}(g_2)\in\mathbb{T}$ and hence

$$
0 \neq \ell\mathrm{t}(g_1)\cdot\ell\mathrm{t}(g_2) \in \mathbb{T}. \tag{6}
$$

Then, according to (4) and the quasi-monotonicity of $<$,

$$
t_s\cdot\ell\mathrm{t}(\mathcal{H}'_s) \overset{(4)}{=} t_s\cdot\varepsilon_s\cdot\ell\mathrm{t}(g_2) \overset{(5),(6)}{<} \ell\mathrm{t}(g_1)\cdot\ell\mathrm{t}(g_2) = \operatorname{lcm}\Big(\ell\mathrm{t}(g_1),\ell\mathrm{t}(g_2)\Big) = \mathrm{L}(g_1,g_2).
$$

It is clear that at least one sum in (3) must be not empty by the assumption $\mathcal{S}_{\mathbb{T}}(g_1,g_2)\neq 0$. W.l.o.g. assume that the first sum is not empty. It is well-known that the maximal element among finitely many terms keeps the strict inequality and therefore

$$
L_1 = \max\Big(\{t_s\cdot\ell\mathrm{t}(\mathcal{H}'_s) \mid s\in\operatorname{supp}(p_1), \mathcal{H}'_s\neq 0\}\Big) < \mathrm{L}(g_1,g_2).
$$

If the second sum in (3) is empty then set L_2=0. Otherwise, the same argumentation applied to the terms of the second sum in (3) gives

$$L_2 = \max\Big(\{t_s{\cdot}\ell\mathrm{t}(\mathcal{H}''_s) \mid s{\in}\,\mathrm{supp}\,(p_2), \mathcal{H}''_s{\neq}0\}\Big) < \mathrm{L}\,(g_1, g_2).$$

Now it is straightforward to check that (3) is an L-standard stabilized monomial representation of $\mathcal{S}_{\mathbb{T}}\,(g_1, g_2)$ in terms of $\mathrm{E}\,(F)$ with

$$0 \neq L = \max(L_1, L_2) < \mathrm{L}\,(g_1, g_2). \qquad \square$$

Buchberger's second criterion states another sufficient condition when an S-polynomial has an appropriate L-standard stabilized monomial representation. The formulation and the proof of the criterion are smoothly translated from the classical setup.

Lemma 2.3.35 (Buchberger's second criterion in $\mathcal{P}(e)$). Let $F{\subseteq}\mathcal{P}(e)\backslash\{0\}$. If $g_1, h, g_2{\in}F$ are such that $(g_1, g_2), (g_1, h), (h, g_2){\in}\mathrm{SP}_{\mathbb{T}}\,(F)$ and

- *(i)* there exists an L-standard stabilized monomial representation of $\mathcal{S}_{\mathbb{T}}\,(g_1, h)$ in terms of $\mathrm{E}\,(F)$ with $L{<}\mathrm{L}\,(g_1, h)$;
- *(ii)* there exists an L-standard stabilized monomial representation of $\mathcal{S}_{\mathbb{T}}\,(h, g_2)$ in terms of $\mathrm{E}\,(F)$ with $L{<}\mathrm{L}\,(h, g_2)$;
- *(iii)* $\ell\mathrm{t}(h)|\mathrm{L}\,(g_1, g_2)$.

Then there exists an L-standard stabilized monomial representation of $\mathcal{S}_{\mathbb{T}}\,(g_1, g_2)$ in terms of $\mathrm{E}\,(F)$ with $L{<}\mathrm{L}\,(g_1, g_2)$.

Proof. It is easy to see by definition 2.3.23 of the set of critical pairs that

$$\deg_e\Big(\ell\mathrm{t}(g_1)\Big) = \deg_e\Big(\ell\mathrm{t}(h)\Big) = \deg_e\Big(\ell\mathrm{t}(g_2)\Big) \qquad (1)$$

and hence, by definition 2.3.21 of the least common multiple in $\mathbb{T}(e)^*$ (cf.

corollary 2.3.9),

$$
\begin{array}{lcl}
\boxed{\deg_e\Big(\mathrm{L}\,(g_1,h)\Big)} & & \boxed{\deg_e\Big(\mathrm{L}\,(g_1,g_2)\Big)} \\
\| & & \| \\
\deg_e\Big(\mathrm{lcm}\big(\ell\mathrm{t}(g_1),\ell\mathrm{t}(h)\big)\Big) & & \deg_e\Big(\mathrm{lcm}\big(\ell\mathrm{t}(g_1),\ell\mathrm{t}(g_2)\big)\Big) \\
\|\ {\scriptstyle \text{Def. 2.3.21}} & & \|\ {\scriptstyle \text{Def. 2.3.21}} \\
\max\Big(\deg_e\big(\ell\mathrm{t}(g_1)\big),\deg_e\big(\ell\mathrm{t}(h)\big)\Big) & \overset{(1)}{=} & \max\Big(\deg_e\big(\ell\mathrm{t}(g_1)\big),\deg_e\big(\ell\mathrm{t}(g_2)\big)\Big) \\
\|\ {\scriptstyle (1)} & & \|\ {\scriptstyle (1)} \\
\boxed{\deg_e\Big(\ell\mathrm{t}(g_1)\Big)} & & \max\Big(\deg_e\big(\ell\mathrm{t}(h)\big),\deg_e\big(\ell\mathrm{t}(g_2)\big)\Big) \\
\|\ {\scriptstyle (1)} & & \|\ {\scriptstyle \text{Def. 2.3.21}} \\
\boxed{\deg_e\Big(\ell\mathrm{t}(h)\Big)} & & \deg_e\Big(\mathrm{lcm}\big(\ell\mathrm{t}(h),\ell\mathrm{t}(g_2)\big)\Big) \\
\|\ {\scriptstyle (1)} & & \| \\
\boxed{\deg_e\Big(\ell\mathrm{t}(g_2)\Big)} & & \boxed{\deg_e\Big(\mathrm{L}\,(h,g_2)\Big)}\,.
\end{array}
\tag{2}
$$

On the other hand, by part *b*) of lemma 2.3.28, $\ell\mathrm{t}(g_1)|\mathrm{L}\,(g_1,g_2)$ and $\ell\mathrm{t}(g_2)|\mathrm{L}\,(g_1,g_2)$. Then, by assumption (*iii*) that $\ell\mathrm{t}(h)|\mathrm{L}\,(g_1,g_2)$ and part *c*) of the same lemma,

$$\mathrm{L}\,(g_1,h)|\mathrm{L}\,(g_1,g_2) \quad \text{and} \quad \mathrm{L}\,(h,g_2)|\mathrm{L}\,(g_1,g_2). \tag{3}$$

Denote

$$t_1 \overset{(3)}{=} \frac{\mathrm{L}\,(g_1,g_2)}{\mathrm{L}\,(g_1,h)} \overset{(2)}{\in} \mathbb{T} \quad \text{and} \quad t_2 \overset{(3)}{=} \frac{\mathrm{L}\,(g_1,g_2)}{\mathrm{L}\,(h,g_2)} \overset{(2)}{\in} \mathbb{T}. \tag{4}$$

Again, by part *b*) of lemma 2.3.28,

$$
\begin{array}{cc}
\ell\mathrm{t}(g_1)|\mathrm{L}\,(g_1,g_2), & \ell\mathrm{t}(g_2)|\mathrm{L}\,(g_1,g_2), \\
\ell\mathrm{t}(g_1)|\mathrm{L}\,(g_2,h), & \ell\mathrm{t}(g_2)|\mathrm{L}\,(g_2,h), \\
\multicolumn{2}{c}{\ell\mathrm{t}(h)|\mathrm{L}\,(g_1,h),\ \ \ell\mathrm{t}(h)|\mathrm{L}\,(h,g_2)}
\end{array}
\tag{5}
$$

and hence there exist the corresponding quotients. Denote

$$
\begin{array}{ll}
\alpha_1 = \dfrac{1}{\ell\mathrm{c}(g_1)} \in \Bbbk^*, & \alpha_2 = \dfrac{1}{\ell\mathrm{c}(g_2)} \in \Bbbk^*, \\
w_1 \overset{(5)}{=} \dfrac{\mathrm{L}\,(g_1, g_2)}{\ell\mathrm{t}(g_1)} \overset{(2)}{\in} \mathbb{T}, & w_2 \overset{(5)}{=} \dfrac{\mathrm{L}\,(g_1, g_2)}{\ell\mathrm{t}(g_2)} \overset{(2)}{\in} \mathbb{T}, \\
v_1 \overset{(5)}{=} \dfrac{\mathrm{L}\,(g_1, h)}{\ell\mathrm{t}(g_1)} \overset{(2)}{\in} \mathbb{T}, & v_2 \overset{(5)}{=} \dfrac{\mathrm{L}\,(h, g_2)}{\ell\mathrm{t}(g_2)} \overset{(2)}{\in} \mathbb{T}, \\
u_1 \overset{(5)}{=} \dfrac{\mathrm{L}\,(g_1, h)}{\ell\mathrm{t}(h)} \overset{(2)}{\in} \mathbb{T}, & u_2 \overset{(5)}{=} \dfrac{\mathrm{L}\,(h, g_2)}{\ell\mathrm{t}(h)} \overset{(2)}{\in} \mathbb{T}, \\
\multicolumn{2}{c}{\beta = \dfrac{1}{\ell\mathrm{c}(h)} \in \Bbbk^*.}
\end{array}
\tag{6}
$$

The following mnemonic diagram can be helpful to navigate in the notation (4) and (6).

$$
\begin{array}{ccccc}
\alpha_1\ w_1 & & & & w_2\ \alpha_2 \\
t_1\ v_1\ g_1 & & \text{———} & & g_2\ v_2\ t_2 \\
 & \diagdown & & \diagup & \\
 & & u_1\ h\ u_2 & & \\
 & & \beta & &
\end{array}
$$

It is easy to see that

$$
\begin{array}{l}
w_1 \cdot \ell\mathrm{t}(g_1) \overset{(6)}{=} \mathrm{L}\,(g_1, g_2) \overset{(4)}{=} t_1 \cdot \mathrm{L}\,(g_1, h) \overset{(6)}{=} t_1 \cdot v_1 \cdot \ell\mathrm{t}(g_1) \overset{(6)}{=} t_1 \cdot u_1 \cdot \ell\mathrm{t}(h) \\
\qquad\qquad \| \\
w_2 \cdot \ell\mathrm{t}(g_2) \overset{(6)}{=} \mathrm{L}\,(g_1, g_2) \overset{(4)}{=} t_2 \cdot \mathrm{L}\,(h, g_2) \overset{(6)}{=} t_2 \cdot v_2 \cdot \ell\mathrm{t}(g_2) \overset{(6)}{=} t_2 \cdot u_2 \cdot \ell\mathrm{t}(h)
\end{array}
\tag{7}
$$

and then, by the quasi-cancellation law (lemma 2.3.7),

$$
t_1 \cdot u_1 = t_2 \cdot u_2,\ t_1 \cdot v_1 = w_1 \ \text{ and } \ w_2 = t_2 \cdot v_2 \tag{8}
$$

provided that $\mathrm{L}\,(g_1, g_2) \neq 0$ by part *a*) of lemma 2.3.28.

The key observation which makes the statement of the lemma valid is the fact that $\mathcal{S}_{\mathbb{T}}\,(g_1, g_2)$ can be expressed in terms of $\mathcal{S}_{\mathbb{T}}\,(g_1, h)$ and $\mathcal{S}_{\mathbb{T}}\,(h, g_2)$ in a particular way such that an appropriate L-standard stabilized monomial representation of $\mathcal{S}_{\mathbb{T}}\,(g_1, g_2)$ can be reconstructed from the corresponding representations of $\mathcal{S}_{\mathbb{T}}\,(g_1, h)$ and $\mathcal{S}_{\mathbb{T}}\,(h, g_2)$. Namely, by definition 2.3.22 of the stabilized $S-$polynomial and in view of the identities (8),

$$\begin{aligned}\mathcal{S}_{\mathbb{T}}(g_1,g_2) &= \alpha_1{\cdot}w_1{\cdot}g_1-\alpha_2{\cdot}w_2{\cdot}g_2 \\ &\overset{(8)}{=} \alpha_1{\cdot}t_1{\cdot}v_1{\cdot}g_1-\beta{\cdot}t_1{\cdot}u_1{\cdot}h+\beta{\cdot}t_2{\cdot}u_2{\cdot}h-\alpha_2{\cdot}t_2{\cdot}v_2{\cdot}g_2 \\ &= t_1{\cdot}\Big(\alpha_1{\cdot}v_1{\cdot}g_1-\beta{\cdot}u_1{\cdot}h\Big)+t_2{\cdot}\Big(\beta{\cdot}u_2{\cdot}h-\alpha_2{\cdot}v_2{\cdot}g_2\Big) \\ &= t_1{\cdot}\mathcal{S}_{\mathbb{T}}(g_1,h)+t_2{\cdot}\mathcal{S}_{\mathbb{T}}(h,g_2).\end{aligned} \tag{9}$$

Let

$$\mathcal{S}_{\mathbb{T}}(g_1,h)=\sum_{i\in N'}\alpha'_i{\cdot}t'_i{\cdot}\chi'_i \quad \text{with} \quad \begin{array}{l} N' \text{ - a finite subset of } \mathbb{N}, \\ \alpha'_i{\in}\Bbbk^*,\ t'_i{\in}\mathbb{T},\ \chi'_i{\in}\mathrm{E}(F), \\ t'_i{\cdot}\chi'_i{\le}L' \qquad \text{for all}\ \ i{\in}N' \end{array} \tag{10}$$

and

$$\mathcal{S}_{\mathbb{T}}(h,g_2)=\sum_{i\in N''}\alpha''_i{\cdot}t''_i{\cdot}\chi''_i \quad \text{with} \quad \begin{array}{l} N'' \text{ - a finite subset of } \mathbb{N}, \\ \alpha''_i{\in}\Bbbk^*,\ t''_i{\in}\mathbb{T},\ \chi''_i{\in}\mathrm{E}(F), \\ t''_i{\cdot}\chi''_i{\le}L'' \qquad \text{for all}\ \ i{\in}N'' \end{array} \tag{11}$$

be the L'-standard stabilized monomial representation of $\mathcal{S}_{\mathbb{T}}(g_1,h)$ in terms of $\mathrm{E}(F)$ and the L''-standard stabilized monomial representation of $\mathcal{S}_{\mathbb{T}}(h,g_2)$ in terms of $\mathrm{E}(F)$ with extended terms $L'{<}\mathrm{L}(g_1,h)$ and $L''{<}\mathrm{L}(h,g_2)$ respectively provided by assumptions (i) and (ii).
We claim that

$$t_1{\cdot}\mathcal{S}_{\mathbb{T}}(g_1,h)=\sum_{i\in N'}\alpha'_i{\cdot}t_1{\cdot}t'_i{\cdot}\chi'_i \tag{12}$$

is the $(t_1{\cdot}L')$-standard stabilized monomial representation of $t_1{\cdot}\mathcal{S}_{\mathbb{T}}(g_1,h)$ in terms of $\mathrm{E}(F)$. Indeed, by (4), $t_1{\in}\mathbb{T}$. Then, by corollary 2.3.1, $t_1{\cdot}L'{\neq}0$ and hence, by the quasi-monotonicity of $<$,

$$t_1{\cdot}t'_i{\cdot}\ell\mathrm{t}(\chi'_i)=t_1{\cdot}\Big(t'_i{\cdot}\ell\mathrm{t}(\chi_i)\Big)\overset{(10)}{\le}t_1{\cdot}L'. \text{ for any } \ i{\in}N' \tag{13}$$

By the same argument,

$$t_2{\cdot}\mathcal{S}_{\mathbb{T}}(h,g_2)=\sum_{i\in N''}\alpha''_i{\cdot}t_2{\cdot}t''_i{\cdot}\chi''_i \tag{14}$$

is the $(t_2{\cdot}L'')$-standard stabilized monomial representation of $t_2{\cdot}\mathcal{S}_{\mathbb{T}}(h,g_2)$ in terms of $\mathrm{E}(F)$: $t_2{\cdot}L''{\neq}0$ and

$$t_2{\cdot}t''_i{\cdot}\ell\mathrm{t}(\chi''_i)=t_2{\cdot}\Big(t''_i{\cdot}\ell\mathrm{t}(\chi''_i)\Big)\overset{(11)}{\le}t_2{\cdot}L''. \tag{15}$$

By (9), (12) and (14)

$$\mathcal{S}_{\mathbb{T}}\left(g_1,g_2\right) \overset{(9)}{=} t_1{\cdot}\mathcal{S}_{\mathbb{T}}\left(g_1,h\right)+t_2{\cdot}\mathcal{S}_{\mathbb{T}}\left(h,g_2\right) \overset{(12),(14)}{=} \sum_{i\in N'}\alpha_i'{\cdot}t_1{\cdot}t_i'{\cdot}\chi_i' \;+\; \sum_{i\in N''}\alpha_i''{\cdot}t_2{\cdot}t_i''{\cdot}\chi_i''. \quad (16)$$

Denote

$$L=\max\Big(t_1{\cdot}L',t_2{\cdot}L''\Big).$$

It is clear by (13) and (15) that (16) is an L-standard stabilized monomial representation of $\mathcal{S}_{\mathbb{T}}\left(g_1,g_2\right)$ in terms of $\mathrm{E}\left(F\right)$. To finish the proof we need to show that

$$L<\mathrm{L}\left(g_1,g_2\right).$$

By (4), $t_1,t_2{\in}\mathbb{T}$. Then, by corollary 2.3.1, $t_1{\cdot}\mathrm{L}\left(g_1,h\right){\neq}0$ and $t_2{\cdot}\mathrm{L}\left(h,g_2\right){\neq}0$. Hence, by the quasi-monotonicity of $<$,

$$L'<\mathrm{L}\left(g_1,h\right) \text{ follows } t_1{\cdot}L'<t_1{\cdot}\mathrm{L}\left(g_1,h\right)\overset{(4)}{=}\mathrm{L}\left(g_1,g_2\right),$$
$$L''<\mathrm{L}\left(h,g_2\right) \text{ follows } t_2{\cdot}L''<t_2{\cdot}\mathrm{L}\left(h,g_2\right)\overset{(4)}{=}\mathrm{L}\left(g_1,g_2\right)$$

and therefore

$$\boxed{L}=\max\Big(t_1{\cdot}L',t_2{\cdot}L''\Big)\boxed{<}\max\Big(t_1{\cdot}\mathrm{L}\left(g_1,h\right),t_2{\cdot}\mathrm{L}\left(h,g_2\right)\Big)=\boxed{\mathrm{L}\left(g_1,g_2\right)}. \quad \square$$

2.3.6 Gröbner bases computation in $\mathcal{P}(e)$

The following algorithm for computing Gröbner bases in the algebra of extended polynomials $\mathcal{P}(e)$ for an arbitrary extended term ordering $<$ basically results from part b) of corollary 2.3.10 which gives a criterion for an arbitrary set of polynomials to be a Gröbner basis in terms of top-reducibility to 0. The algorithm takes an arbitrary finite set of polynomials F and computes a Gröbner basis of $\langle F\rangle_{\mathcal{P}(e)}$. If the set F is infinite then the algorithm together with the correctness proof can be considered as an alternative mathematical Gröbner bases existence statement. The algorithm uses Buchberger's first and the second criteria for $\mathcal{P}(e)$ to eliminate certain redundant S-pairs. The use of criteria should result in a drastically improved performance of the algorithm implementation but also makes the correctness proof slightly more complicated since it is not possible to argue the correctness of the algorithm staying in the frame of part b) of corollary 2.3.10 but certain cases must be analyzed and argued using a "lower level" Gröbner basis criterion stated in theorem 2.3.3. Note that the presented layout of the algorithm can hardly be considered as a readily suited foundation for an optimal implementation. A more elaborated and optimized version is presented in the corresponding chapter with the description of our implementation. At the present point it is more important to introduce a shallow layout which reflects the transformation of the developed theory into an algorithm for computing Gröbner bases.

The layout of the algorithm `Gröbner` for computing Gröbner bases is listed on page 79. It uses a procedure `Update` which eliminates redundant S-pairs.

Algorithm 2.3.2 The redundant $S-$pairs elimination procedure `Update`

Require: C - the pending set of stabilized $S-$pairs.
Require: D - the set of processed $S-$pairs.
Ensure: C - such that no pair in C has disjoint leading terms and no triple of different pairs in C satisfies the "if" condition of lemma 2.3.35.

```
for (g1, g2)∈C do
  if ℓt(g1) and ℓt(g2) are disjoint then
    C := C\{(g1, g2)}
    D := D∪{(g1, g2)}
  else
    if ∃h∈P(e) such that
               (g1, h), (h, g2)∈D∪C\{(g1, g2)} and
               ℓt(h)|L (g1, g2)  then
      C := C\{(g1, g2)}
    end if
  end if
end for
```

Lemma 2.3.36. For any finite $F\subseteq\mathcal{P}(e)$ the algorithm `Gröbner` computes a Gröbner basis G of the ideal $\langle F\rangle_{\mathcal{P}(e)}$.

Proof.

For each iteration $k\in\mathbb{N}_o$ of the while-loop of the algorithm `Gröbner` denote $G_k\supseteq G$ - the basis at the beginning of the while-loop body, $h_k\in\mathcal{P}(e)$ - the chosen polynomial, r_k - the computed normal form modulo G_k, C_k - the pending set of stabilized pairs, E_k - the pending set of critical e-orbits. The index 0 also denotes the status right before the execution of the while-loop to avoid a separate treatment of initial conditions.

Termination of `Update`:

Algorithm 2.3.3 The algorithm `Gröbner` for computing Gröbner bases in the algebra of extended polynomials

Require: $F \subseteq \mathcal{P}(e) \backslash \{0\}$ - a set of extended polynomials.
Ensure: $G \subseteq \mathcal{P}(e) \backslash \{0\}$ - a Gröbner basis of $\langle F \rangle_{\mathcal{P}(e)}$.

```
G := F
C := SP_T(F)
D := ∅
E := EC(F)
Update(C, D)
while C≠∅ or E≠∅ do
   if E≠∅ then
      choose h∈E
      E := E \ {h}
   else
      choose (g1, g2)∈C
      C := C \ {(g1, g2)}
      D := D ∪ {(g1, g2)}
      h := S_T(g1, g2)
   end if
   r := a normal form of h modulo G computed by the reduction algorithm
   NF on page 47
   if r≠0 then
      G := G ∪ {r}
      C := C ∪ {(g, p)∈E(G)×E(r) | deg_e(ℓt(g))=deg_e(ℓt(p)), S_T(g, p)≠0}
      E := E ∪ EC(r)
      Update(C, D)
   end if
end while
```

In mathematical notation, the steps are:

$G := F$
$C := \mathrm{SP}_{\mathbb{T}}(F)$
$D := \emptyset$
$E := \mathrm{EC}(F)$
$\mathtt{Update}(C, D)$
while $C \neq \emptyset$ or $E \neq \emptyset$ **do**
  if $E \neq \emptyset$ **then**
    choose $h \in E$
    $E := E \backslash \{h\}$
  else
    choose $(g_1, g_2) \in C$
    $C := C \backslash \{(g_1, g_2)\}$
    $D := D \cup \{(g_1, g_2)\}$
    $h := \mathcal{S}_{\mathbb{T}}(g_1, g_2)$
  end if
  $r :=$ a normal form of h modulo G computed by the reduction algorithm `NF` on page 47
  if $r \neq 0$ **then**
    $G := G \cup \{r\}$
    $C := C \cup \left\{(g, p) \in \mathrm{E}(G) \times \mathrm{E}(r) \mid \deg_e\left(\ell\mathrm{t}(g)\right) = \deg_e\left(\ell\mathrm{t}(p)\right),\ \mathcal{S}_{\mathbb{T}}(g, p) \neq 0\right\}$
    $E := E \cup \mathrm{EC}(r)$
    $\mathtt{Update}(C, D)$
  end if
end while

At each call from the algorithm `Gröbner` the procedure `Update` iterates over a finite set of pairs resulting in a possibly smaller set and hence the for-loop terminates after finitely many steps.

Termination of `Gröbner`:

At each iteration of the while-loop the algorithm `Gröbner` computes a normal form r_k modulo the current basis. By part (NF_2) of definition 2.3.15 of the normal form, r_k is top-minimal modulo G_k and hence, by lemma 2.3.15, r_k is either 0 or $\ell\mathrm{t}(r_k) \notin \big\langle \ell\mathrm{t}(G_k) \big\rangle_{\mathcal{P}(e)}$ for each $k \in \mathbb{N}$. It is quite clear that the infinitely many iterations of the while-loop must result in an infinite set of indexes $\{k_i\}$ such that $h_{k_i} \neq 0$ and hence, according to the flow of the algorithm, $G_{k_i+1} = G_{k_i} \cup \{h_{k_i}\}$ but then the infinite strictly ascending chain of monomial ideals

$$\big\langle \ell\mathrm{t}(G_{k_1}) \big\rangle_{\mathcal{P}(e)} \subsetneq \big\langle \ell\mathrm{t}(G_{k_2}) \big\rangle_{\mathcal{P}(e)} \subsetneq \cdots \subsetneq \big\langle \ell\mathrm{t}(G_{k_i}) \big\rangle_{\mathcal{P}(e)} \subsetneq \cdots$$

contradicts the Hilbert basis theorem (cf. corollary 2.3.3 - the structure of monomial ideals).

Let $N \in \mathbb{N}_o$ be the number of runs of the while-loop. Denote $G = G_N$ - the final basis after the algorithm is complete. Then the flow of the algorithm results in an ascending chain of polynomial sets

$$F = G_o \subseteq \ldots \subseteq G_k \subseteq \ldots \subseteq G_N = G.$$

G is a basis of $\langle F \rangle_{\mathcal{P}(e)}$:

It is sufficient to show the following invariant of the while-loop:

$$F \subseteq G_k \subseteq \langle F \rangle_{\mathcal{P}(e)} \quad \text{for any} \quad 0 \leq k \leq N. \tag{1}$$

The first inclusion is true because initially $G_o = F$ and no element is ever removed from the basis during the execution of the while-loop.

For the second inclusion observe from the flow analysis of the algorithm (a strict derivation is not hard but a notational and logically tedious analysis of the ways how the new elements appear in the pending sets together with the fact that no polynomial is ever removed from the basis) that at each step $0 \leq k \leq N$ for any polynomial $\bar{g} \in E_k$ there exists $g \in G_k$ such that $\bar{g} \in \mathrm{EC}\,(g)$ and for any pair $(\bar{g_1}, \bar{g_2}) \in C_k$ there exist two (not necessary different though resulting to a non-zero stabilized S-polynomial!) polynomials $g_1, g_2 \in G_k$ such that $(\bar{g_1}, \bar{g_2}) \in \mathrm{SP}_{\mathbb{T}}\,(\{g_1, g_2\}) \subseteq \mathrm{E}\,(g_1) \times \mathrm{E}\,(g_2)$. Therefore

$$E_k \subseteq \mathrm{EC}\,(G_k) \Big[\subseteq \langle G_k \rangle_{\mathcal{P}(e)} \Big], \; C_k \subseteq \mathrm{SP}_{\mathbb{T}}\,(G_k) \Big[\subseteq \langle G_k \rangle^{\times 2}_{\mathcal{P}(e)} \Big] \quad \text{for} \quad 0 \leq k \leq N. \tag{2}$$

The second inclusion in (1) is clearly true before the while-loop starts: $F \subseteq F \subseteq \langle F \rangle_{\mathcal{P}(e)}$. On the other hand each iteration amounts to at most one normal form added to the basis. Assume the claim is true after iteration number k. Let h be the element chosen at the very beginning of the iteration $k+1$. Then h has been either chosen from

$$h \in E_k \overset{(2)}{\subseteq} \mathrm{EC}\left(G_k\right) \subseteq \langle F \rangle_{\mathcal{P}(e)}$$

where the last inclusion is by assumption, or it is the stabilized S-polynomial of a pair

$$(\bar{g}_1, \bar{g}_2) \in C_k \overset{(2)}{\subseteq} \mathrm{SP}_{\mathbb{T}}\left(G_k\right) \subseteq \langle F \rangle_{\mathcal{P}(e)}^{\times 2}$$

where the last inclusion is again by assumption and hence $h \in \langle F \rangle_{\mathcal{P}(e)}$ by the construction of a stabilized S-polynomial.

Then, in both cases, $h \in \langle F \rangle_{\mathcal{P}(e)}$ and hence the normal form $r \in \langle F \rangle_{\mathcal{P}(e)}$ by part (NF_1) of definition 2.3.15 of a normal form which follows

$$\langle F \rangle_{\mathcal{P}(e)} \supseteq G_{k+1} = \begin{cases} G_k \cup \{r\}, & r \neq 0 \\ G_k, & \text{otherwise.} \end{cases}$$

Let $G = G_N$ be the final basis attained when the while-loop is finished. Then, by (2),

$$F \subseteq G_N \subseteq \langle F \rangle_{\mathcal{P}(e)} \ \text{ follows } \ \langle F \rangle_{\mathcal{P}(e)} \subseteq \langle G_k \rangle_{\mathcal{P}(e)} \subseteq \langle F \rangle_{\mathcal{P}(e)}$$

and hence

$$\langle F \rangle_{\mathcal{P}(e)} = \langle G \rangle_{.}$$

$\boxed{G \text{ is a Gröbner basis of } \langle F \rangle_{\mathcal{P}(e)}:}$

It is sufficient to show that G satisfies the "if" condition of theorem 2.3.3. In particular, each non-zero stabilized S-polynomial $\mathcal{S}_{\mathbb{T}}\left(g_1, g_2\right)$ with $g_1, g_2 \in \mathrm{E}\left(G\right)$ must have an L-standard stabilized monomial representation in terms of $\mathrm{E}\left(G\right)$ with $L < \mathrm{L}\left(g_1, g_2\right)$. There is a number of reasons when the desired standard stabilized monomial representation exists. Indeed, for any $H \subseteq \mathcal{P}(e) \backslash \{0\}$ and an arbitrary non-zero extended polynomial $h \in \mathcal{P}(e) \backslash \{0\}$ there is the following (cf. corollary 2.3.10)

chain of deduction: $\qquad (CD)$

- if $h \xrightarrow{H}_* 0$ then, by lemma 2.3.20,
- there exists a Gröbner representation of h in terms of H; then this Gröbner representation is trivially
- an L-standard representation of h in terms of with $L{=}\ell\mathrm{t}(h)$ by definition 2.3.19 of the L-standard representation; then, by lemma 2.3.26,
- there exists an L-standard stabilized monomial representation of h in terms of $\mathrm{E}\,(H)$ with the same L.

We would also need a sufficient within the flow of the algorithm condition for a polynomial $\langle F\rangle_{\mathcal{P}(e)}$ to have a Gröbner representation in terms of the final basis G. Assume the element $h{\in}\langle F\rangle_{\mathcal{P}(e)}$ was taken at the beginning of the while-loop body at some iteration in the flow of the algorithm. Then

$$h \;\text{ has a Gröbner representation in terms of } G. \tag{GR}$$

Indeed, let i be the iteration number when h has been taken at the beginning of the while-loop and top-reduced to $h \xrightarrow{G_i}_* r$. If $r{=}0$ then it is not added to the basis, by (CD), h has a Gröbner representation in terms of G_i and hence in terms of $G_{i+1}{=}G_i$. If $r{\neq}0$ then it has been added to the basis resulting to $G_{i+1}{=}G_i{\cup}\{r\}$. Then

$$h \xrightarrow{G_i}_* r \xrightarrow{\{r\}} 0 \;\text{ follows }\; h \xrightarrow{G_{i+1}}_* 0$$

and again, by (CD), h has a Gröbner representation in terms of G_{i+1}.

In both cases h has a Gröbner representation in terms of G_{i+1} which is also a Gröbner representation of h in terms of G due to the inclusion $G_{i+1}{\subseteq}G_N{=}G$.

According to theorem 2.3.3 there can be two reasons when G fails to be a Gröbner basis:

- either there exists a polynomial $g{\in}\mathrm{E}\,(G)$ such that g does not have a Gröbner representation in terms of G
- or there are two different polynomials $g_1, g_2{\in}\mathrm{E}\,(G)$ which can be used to form a stabilized S-polynomial such that $\mathcal{S}_{\mathbb{T}}\,(g_1, g_2){\neq}0$ and does not have an L-standard stabilized monomial representation in terms of $\mathrm{E}\,(G)$ for any extended term $L{<}\mathrm{L}\,(g_1, g_2)$.

Consider these cases separately.

1. Assume there exist $g \in G$ and $\bar{g} \in \mathrm{E}(g)$ such that $\bar{g}$ does not have a Gröbner representation in terms of G. Then, it is easy to see by lemma 2.3.32 that $\bar{g} \in \mathrm{EC}(g)$. Let k be the iteration number when the element g has been inserted to the basis G_k. Then the element $\bar{g} \in \mathrm{EC}(g)$ must have been added to the pending set E_k of e-orbits. But $E = E_N = \emptyset$ is the necessary condition for the while-loop termination. The only place in the algorithm where an element is removed from the pending set of e-orbits is the beginning of the while-loop. Hence there must exist an iteration number k' such that $k < k' < N$ when the element $h = \bar{g}$ has been removed from the pending set of e-orbits and top-reduced. Then $\bar{g}$ has a Gröbner representation in terms of G according to (GR).

2. Assume there is a pair of polynomials $(\bar{g_1}, \bar{g_2}) \in \mathrm{SP}_{\mathbb{T}}(G)$ such that the stabilized S-polynomial $h = \mathcal{S}_{\mathbb{T}}(\bar{g_1}, \bar{g_2}) \neq 0$ does not have an L-standard stabilized monomial representation in terms of $\mathrm{E}(G)$ for any extended term $L < \mathrm{L}(\bar{g_1}, \bar{g_2})$. Let $g_1, g_2 \in G$ be polynomials such that $\bar{g_1} \in \mathrm{E}(g_1)$ and $\bar{g_2} \in \mathrm{E}(g_2)$. W.l.o.g. assume that g_2 has been added to the basis not earlier than g_1. If the polynomial g_2 was added to the basis at the iteration number k then the stabilized S-pair $(\bar{g_1}, \bar{g_2})$ must have been added to the pending set C_k. But $C = C_N = \emptyset$ is the necessary condition for the while-loop termination. Hence there exists the iteration number k' such that $k \leq k' < N$ when the pair $(\bar{g_1}, \bar{g_2})$ has been removed from the pending set $C_{k'}$. According to the flow of the algorithm the pair could have been taken from $C_{k'}$ at the beginning of the while-loop or it has been removed in the call of the `Update` procedure. Consider these cases separately.

   (a) Assume the pair $(\bar{g_1}, \bar{g_2})$ has been taken at the beginning of the while-loop body and the corresponding stabilized S-polynomial $h = \mathcal{S}_{\mathbb{T}}(\bar{g_1}, \bar{g_2})$ was top-reduced to a normal form modulo $G_{k'}$. Then h must have a Gröbner representation in terms of G according to (GR) and hence, by (CD), there exists an L-standard stabilized monomial representation of h in terms of $\mathrm{E}(G)$.

   (b) Assume the pair $(\bar{g_1}, \bar{g_2})$ has been removed from the pending set $C_{k'}$ in the call of the `Update` procedure at the iteration number k'. There are two places in the procedure when a pair can be removed form $C_{k'}$.

      i. Assume the pair $(\bar{g_1}, \bar{g_2})$ was removed by the first "if" condition. Then the leading terms $\ell\mathrm{t}(\bar{g_1})$ and $\ell\mathrm{t}(\bar{g_2})$ must be disjoint and hence, by lemma 2.3.34, there exists $L < \mathrm{L}(\bar{g_1}, \bar{g_2})$

such that h has an L-standard stabilized monomial representation in terms of $\mathrm{E}(G)$.

ii. Assume the pair $(\bar{g_1}, \bar{g_2})$ was removed by the second "if" condition. Then there exists a polynomial h such that the pairs $(\bar{g_1}, h), (h, \bar{g_2}) \in C_{k'} \cup D_{k'}$.
Consider the case when $(\bar{g_1}, h), (h, \bar{g_2}) \in D_{k'}$. But, according to the flow of the algorithm, any pair from D must have been in the pending set C at some earlier iterations and was moved to D because it either consists of polynomials with disjoint leading terms or the pair has been taken from C in the beginning of the while-loop body. Hence any pair in $D_{k'}$ has the desired standard stabilized monomial representation in terms of $\mathrm{E}(G)$ according to the cases analyzed above.
On the other hand, if either $(\bar{g_1}, h) \in C_{k'}$ or $(h, \bar{g_2}) \in C_{k'}$ then w.l.o.g assume that $(\bar{g_1}, \bar{g_2})$ is the chronologically last problematic (without an appropriate representation) pair removed by the second "if" condition. But then each pair $(\bar{g_1}, h), (h, \bar{g_2})$ has either been selected at the beginning of the while-loop body or has disjoint leading terms and hence in both cases both pairs have the appropriate stabilized monomial representations.
Then, in both cases, both pairs have the desired representations and hence the pair $(\bar{g_1}, \bar{g_2})$ must also have an L-standard stabilized monomial representation in terms of $\mathrm{E}(G)$ with $L < \mathrm{L}(\bar{g_1}, \bar{g_2})$ according to lemma 2.3.35. □

Although we delegate the discussion about the optimal implementation of the algorithm `Gröbner` to a separate chapter, one optimization should be considered now. For performance and common sense reasons it is desirable to have as few polynomials as possible in the computed Gröbner basis and therefore most implementations, including ours, of the algorithms for computing different kind of Gröbner bases deliver a minimal Gröbner basis in the sense that no proper subset of it is a Gröbner basis of the same ideal.

Definition 2.3.26. A Gröbner basis $G \subseteq \mathcal{P}(e) \setminus \{0\}$ is a **minimal Gröbner basis** if and only if it is the minimal by inclusion of sets Gröbner basis of the ideal $\langle G \rangle_{\mathcal{P}(e)}$.

To devise an algorithm which delivers a minimal Gröbner basis by excluding the redundant elements from an arbitrary input Gröbner basis we need a rule to detect redundant polynomials. The following lemma gives it.

Lemma 2.3.37. Let $G{\subseteq}\mathcal{P}(e)\backslash\{0\}$ be a Gröbner basis. If $g{\in}G$ is such that

$$\ell\mathrm{t}(g) \in \Big\langle \ell\mathrm{t}\big(G\backslash\{g\}\big)\Big\rangle_{\mathcal{P}(e)}$$

then $G\backslash\{g\}$ is a Gröbner basis of the ideal $\langle G\rangle_{\mathcal{P}(e)}$.

Proof. Denote $I{=}\langle G\rangle_{\mathcal{P}(e)}$ and $F{=}\big\langle G\backslash\{g\}\big\rangle_{\mathcal{P}(e)}$. If $\ell\mathrm{t}(g){\in}\big\langle \ell\mathrm{t}(F)\big\rangle_{\mathcal{P}(e)}$ then it is easy to see that

$$\Big\langle \ell\mathrm{t}(I)\Big\rangle_{\mathcal{P}(e)} = \Big\langle \ell\mathrm{t}(G)\Big\rangle_{\mathcal{P}(e)} = \Big\langle \ell\mathrm{t}(F)\Big\rangle_{\mathcal{P}(e)}$$

where the first equality is by definition 2.3.17 of a Gröbner basis and hence F is a Gröbner basis of I according to the same definition. □

The next lemma provides the termination condition for the minimal Gröbner bases computation algorithm.

Lemma 2.3.38. A Gröbner basis $G{\subseteq}\mathcal{P}(e)\backslash\{0\}$ is minimal if and only if

$$\ell\mathrm{t}(g) \notin \Big\langle \ell\mathrm{t}\big(G\backslash\{g\}\big)\Big\rangle_{\mathcal{P}(e)} \quad \text{for any } g \in G$$

Proof. The "only if" direction is given by lemma 2.3.37. To set the opposite assume that G is not minimal. Then there exists $g{\in}G$ such that $G\backslash\{g\}$ is a Gröbner basis of $\langle G\rangle_{\mathcal{P}(e)}$. Note that $\ell\mathrm{t}(g) \notin \big\langle \ell\mathrm{t}\big(G\backslash\{g\}\big)\big\rangle_{\mathcal{P}(e)}$ by assumption. It is clear that $g{\in}\langle G\rangle_{\mathcal{P}(e)}$ and hence, by part *b*) of theorem 2.3.2, there exists $h{\in}G\backslash\{g\}$ such that $\ell\mathrm{t}(h)|\ell\mathrm{t}(g)$ which follows $\ell\mathrm{t}(g){\in}\big\langle \ell\mathrm{t}\big(G\backslash\{g\}\big)\big\rangle_{\mathcal{P}(e)}$ and results in a contradiction. □

The algorithm `MinGröbner` on page 85 takes an arbitrary Gröbner basis G and computes a minimal Gröbner basis of $\langle G\rangle_{\mathcal{P}(e)}$.

Algorithm 2.3.4 The algorithm `MinGröbner` for computing minimal Gröbner bases

---

Require: $G{\subseteq}\mathcal{P}(e)\backslash\{0\}$ - a Gröbner basis
Ensure: G - a minimal Gröbner basis of the same ideal
  while $\exists g{\in}G$ such that $\ell\mathrm{t}(h)|\ell\mathrm{t}(g)$ for some $h{\in}G\backslash\{g\}$ **do**
    $G := G\backslash\{g\}$
  end while

Lemma 2.3.39. For any Gröbner basis $G{\subseteq}\mathcal{P}(e)\backslash\{0\}$ the algorithm `MinGröbner` computes a minimal Gröbner basis of $\langle G\rangle_{\mathcal{P}(e)}$.

Proof. The termination is clear because the input Gröbner basis G is a finite set by convention. At each iteration of the while loop the algorithm chooses $g \in G$ such that $\ell\mathrm{t}(h) | \ell\mathrm{t}(g)$ for some $h \in G \backslash \{g\}$ which is equivalent to the condition $\ell\mathrm{t}(g) \in \left\langle \ell\mathrm{t}(G \backslash \{g\}) \right\rangle_{\mathcal{P}(e)}$ according to part b) of lemma 2.3.5 and hence the strictly smaller updated basis $G \backslash \{g\}$ is a Gröbner basis of the same ideal by lemma 2.3.37. The termination condition of the while loop is $\ell\mathrm{t}(g) \notin \left\langle \ell\mathrm{t}(G \backslash \{g\}) \right\rangle_{\mathcal{P}(e)}$ for any $g \in G$ which is is exactly the condition in the criterion for an arbitrary Gröbner basis to be minimal stated in lemma 2.3.38. □

It is natural to use the algorithm `MinGröbner` as a postprocessing step in the algorithm `Gröbner` to obtain a possibly smaller output. On the other hand, one might also like to simplify and reduce the number of polynomials in the current basis at the intermediate steps of a Gröbner basis computation which can not be handled by the presented version of a minimal Gröbner basis computation algorithm due to the fact that lemma 2.3.37 is not valid without the assumption that the basis is a Gröbner basis. An alternative approach extensively used in computer algebra software including ours is based on a mutual top-reduction of the basis polynomials modulo each other.

Definition 2.3.27. A set $F \subseteq \mathcal{P}(e) \backslash \{0\}$ is called **auto-top-reduced** if and only if $\ell\mathrm{t}(f) \notin \left\langle \ell\mathrm{t}(F \backslash \{f\}) \right\rangle_{\mathcal{P}(e)}$ for any $f \in F$. Note that an auto-top-reduced Gröbner basis is exactly a minimal Gröbner basis in the sense of definition 2.3.26.

The algorithm `AutoReduce` on page 87 computes an auto-top-reduced set of polynomials starting from an arbitrary input. It can be used at an arbitrary point of a Gröbner basis computation although one might think of a reasonable from the point of view of practical computing strategy to apply the auto-top-reduction when it is more helpful. As a matter of fact the computer algebra community does not currently hold the conventional wisdom towards the optimal in a certain sense way to plug the auto-top-reduction in a Gröbner basis computation though it is considered to be helpful for the performance reasons. In our case the implementation for computing Gröbner bases with numerical values benefits from auto-top-reduction because it seems to reduce drastically the number of the arithmetical operations with polynomials performed in the flow of a Gröbner basis computation and probabilistically reduces the danger of a numerical blunder due to the accumulated round-off error in the coefficients which can easily result to an absurd structure of leading terms in the final result.

Lemma 2.3.40. For any finite set $F \subseteq \mathcal{P}(e) \backslash \{0\}$ the algorithm `AutoReduce` transforms F to an auto-top-reduced set $H \subseteq F$ such that $\langle H \rangle_{\mathcal{P}(e)} = \langle F \rangle_{\mathcal{P}(e)}$.

Algorithm 2.3.5 The algorithm `AutoReduce` for computing an auto-top-reduced set of polynomials

Require: $F \subseteq \mathcal{P}(e) \backslash \{0\}$
Ensure: F - auto-top-reduced
  while $\exists f \in F$ such that $\ell\mathrm{t}(f) \in \big\langle \ell\mathrm{t}(F \backslash \{f\}) \big\rangle_{\mathcal{P}(e)}$ **do**
    $F := F \backslash \{f\}$
    r := a normal form of f modulo $F \backslash \{f\}$ computed by the reduction algorithm on page 47
    if $r \neq 0$ **then**
      $F := F \cup \{r\}$
    end if
  end while

Proof. For each iteration $k \in \mathbb{N}_o$ of the while loop body denote F_k - the current basis at the beginning of the while-loop body execution, $f_k \in F_k$ - the element chosen in the flow of the kth iteration such that $\ell\mathrm{t}(f_k) \in \big\langle \ell\mathrm{t}(F_k \backslash \{f_k\}) \big\rangle_{\mathcal{P}(e)}$, $r_k \in \mathcal{P}(e)$ - the computed normal form of f_k. To avoid a separate treatment of the initial basis, set $F_o = F$. According to lemma 2.3.14, the prerequisite of the k-th while-loop body execution is equivalent to the condition that the current basis F_k contains an element f_k which is top-reducible modulo $F_k \backslash \{f_k\}$ and hence, by lemma 2.3.13, the normal form r_k computed in the loop body is either 0 or $\ell\mathrm{t}(r_k) < \ell\mathrm{t}(f_k)$. Then it is easy to see by the pigeonhole argument that the infinitely many runs of the loop body must result to the existence of at least one element $h \in F$ such that $h = h_{k_1}$ and

$$h_{k_1} \xrightarrow{F_{k_1}}_* h_{k_2} \xrightarrow{F_{k_2}}_* \cdots \xrightarrow{F_{k_{i-1}}}_* h_{k_i} \xrightarrow{F_{k_i}}_* \cdots$$

where $h_{k_i} \neq 0$ for each k_i in an infinite subset of iteration numbers $\{k_i | i \in \mathbb{N}\} \subseteq \mathbb{N}_o$ but then an infinite strictly descending sequence $\{\ell\mathrm{t}(h_{k_i}) | i \in \mathbb{N}\}$ contradicts the well-ordering property of an extended term order.

To show that the result of the algorithm execution generates the same ideal as the initial basis it is clearly sufficient to show that the ideal generated by the current basis is an invariant of the while-loop. Take an arbitrary iteration number k. Let $F = F_k$, $f = f_k$, $r = r_k$. According to the layout on page 87, the algorithm chooses f from the current basis F, computes a normal form r of f modulo $F \backslash \{f\}$ and transforms F to the set $\big(F \backslash \{f\}\big) \cup \big(\{r\} \backslash \{0\}\big)$. It is easy to see that the transformation of the current basis leads to a different generating set of the same ideal i.e.

$$\langle F \rangle_{\mathcal{P}(e)} = \Big\langle (F \backslash \{f\}) \cup \{r\} \Big\rangle_{\mathcal{P}(e)} \left[= \Big\langle (F \backslash \{f\}) \cup (\{r\} \backslash \{0\}) \Big\rangle_{\mathcal{P}(e)} \right].$$

Indeed, by part (NF_1) of definition 2.3.15 of a normal form, $f{-}r{\in}\langle F\backslash\{f\}\rangle_{\mathcal{P}(e)}$ and hence there exists $h{\in}\langle F\backslash\{f\}\rangle_{\mathcal{P}(e)}$ such that $f{=}r{+}h$. It is clear that

$$\langle F\backslash\{f\}\rangle_{\mathcal{P}(e)} + \langle h\rangle_{\mathcal{P}(e)} = \langle F\backslash\{f\}\rangle_{\mathcal{P}(e)} \tag{$*$}$$

and

$$\begin{aligned} \Big[\langle f\rangle_{\mathcal{P}(e)} =\Big]\ \langle r{+}h\rangle_{\mathcal{P}(e)} \subseteq \langle r\rangle_{\mathcal{P}(e)} + \langle h\rangle_{\mathcal{P}(e)}, \\ \Big[\langle r\rangle_{\mathcal{P}(e)} =\Big]\ \langle f{-}h\rangle_{\mathcal{P}(e)} \subseteq \langle f\rangle_{\mathcal{P}(e)} + \langle h\rangle_{\mathcal{P}(e)}. \end{aligned} \tag{$**$}$$

Therefore

$$\begin{aligned} \boxed{\langle F\rangle_{\mathcal{P}(e)}} &= \Big\langle F\backslash\{f\}\Big\rangle_{\mathcal{P}(e)} + \langle f\rangle_{\mathcal{P}(e)} \\ &= \Big\langle F\backslash\{f\}\Big\rangle_{\mathcal{P}(e)} + \langle r{+}h\rangle_{\mathcal{P}(e)} \\ &\overset{(**)}{\subseteq} \Big\langle F\backslash\{f\}\Big\rangle_{\mathcal{P}(e)} + \langle r\rangle_{\mathcal{P}(e)} + \langle h\rangle_{\mathcal{P}(e)} \overset{(*)}{=} \\ &\qquad \boxed{\Big\langle F\backslash\{f\}\Big\rangle_{\mathcal{P}(e)} + \langle r\rangle_{\mathcal{P}(e)}} = \\ &\qquad\qquad \Big\langle F\backslash\{f\}\Big\rangle_{\mathcal{P}(e)} + \langle f{-}h\rangle_{\mathcal{P}(e)} \overset{(**)}{\subseteq} \\ &\qquad\qquad \Big\langle F\backslash\{f\}\Big\rangle_{\mathcal{P}(e)} + \langle f\rangle_{\mathcal{P}(e)} + \langle h\rangle_{\mathcal{P}(e)} \overset{(*)}{=} \\ &\qquad\qquad\qquad \Big\langle F\backslash\{f\}\Big\rangle_{\mathcal{P}(e)} + \langle f\rangle_{\mathcal{P}(e)} = \boxed{\langle F\rangle_{\mathcal{P}(e)}}. \end{aligned}$$

At the end of the while-loop the resulting basis is auto-top-reduced because the termination condition of the while-loop is the condition in definition 2.3.27 of an auto-top-reduced set. □

The auto-top-reduction can be applied at an arbitrary point of the algorithm `Gröbner`. Indeed, the auto-top-reduction before the while-loop execution in the Gröbner basis computation algorithm results in a possibly different generating set of the same ideal. The auto-top-reduction inside the Gröbner basis computation while-loop eliminates the older polynomials which became top-reducible modulo the newly added. As a matter of fact the correctness proof of the Gröbner bases computation algorithm which uses the auto-top-reduction at intermediate steps becomes extremely long but not considerably reacher in content in our frame of argumentation. Because of this reason we omit the full proof but present a sketch of the possible "wording" of a complete proof which is easy though tiresome to attain by analogy with the argumentation used in [BW93] in lemma 5.73 on page 231.

The correctness of this transformation of the current basis in the flow of the Gröbner basis computation can be augured by Buchbereger's second criterion. Indeed, let G be the current basis and $f \in G$ is a polynomial top-reducible modulo $G \setminus \{f\}$. Then there must exist a polynomial $h \in G$ such that $\ell t(h) | \ell t(f)$. Then $\frac{\ell t(f)}{\ell t(h)} \cdot \ell t(h) = \ell t(f) \neq 0$ and, by part a) of corollary 2.3.6, $\ell t\left(\frac{\ell t(f)}{\ell t(h)} \cdot h\right) = \frac{\ell t(f)}{\ell t(h)} \cdot \ell t(h)$ which follows that f top-reduces in one step modulo h according to definition 2.3.12. Furthermore, again by part a) of corollary 2.3.6, $\ell t\left(\varepsilon\left(\frac{\ell t(f)}{\ell t(h)}\right) \cdot h\right) = \varepsilon\left(\frac{\ell t(f)}{\ell t(h)}\right) \cdot \ell t(h)$ and hence the one step top-reduction can be considered as the evaluation of a stabilized S-pair: $q = \mathcal{S}_{\mathbb{T}}\left(f, \varepsilon\left(\frac{\ell t(f)}{\ell t(h)}\right) \cdot h\right)$. On the other hand, the condition $\ell t(h) | \ell t(f)$ follows that $\ell t(h) | \mathrm{L}\,(f,g)$ for any $g \in \mathcal{P}(e) \setminus \{0\}$ and hence, if there exists a stabilized S-pair (f,g) then it is superfluous by Buchbereger's second criterion if the stabilized $S-$polynomial $\mathcal{S}_{\mathbb{T}}\left(g, \varepsilon\left(\frac{\ell t(f)}{\ell t(h)}\right) \cdot h\right)$ has been treated and the result of top-reduction of q has been added to the current basis (cf. the discussion at the end of page 228 in [BW93]). The latter condition on q is true because it is exactly the way how the algorithm `AutoReduce` works. On the other hand if g is a polynomial which arrived to the current basis at some later iteration after the current auto-top-reduction which effected f then the condition on $\mathcal{S}_{\mathbb{T}}\left(g, \varepsilon\left(\frac{\ell t(f)}{\ell t(h)}\right) \cdot h\right)$ is also true: according to the layout of the algorithm `Gröbner` the stabilized S-pair $(g, \varepsilon\left(\frac{\ell t(f)}{\ell t(h)}\right) \cdot h)$ is added to the pending list and will be eventually treated either by top-reduction of the corresponding stabilized S-polynomial or excluded by the call of the superfluous S -pairs elimination procedure which follows in either case that the S-polynomial has an appropriate for application of Buchberger's second criterion representation according to the proof of lemma 2.3.36. Note that the argumentation above is nothing but a hint how the complete proof can be established and it is very far from being the correctness proof of anything.

2.4 Gröbner bases in the ring $\mathcal{P}_e$

Let $\mathcal{P}_e = \mathcal{P}(e)$ and $\mathbb{T}_e = \mathbb{T}(e)$ be the instances of the algebra $\mathcal{P}(e)$ and the extended terms $\mathbb{T}(e)$ such that $\mathbb{T}_e \subseteq \mathcal{P}_e$. Denote ${\mathbb{T}_e}^*$ – the set of non-zero terms in $\mathbb{T}_e$, $\mathbb{M}_e$ – the monomials $\mathbb{M}(e)$ of the algebra $\mathcal{P}_e$ and $\mathbb{M}_{\not e}$ the set of $e-$free monomials of $\mathcal{P}_e$. Equip $\mathbb{T}_e$ with the term order $\underset{\mathbb{T}_e}{<}$ defined as the lexicographical product of the order $\underset{\mathbb{T}}{<}$ on $\mathbb{T}$ and the degree anti-compatible order on the (finitely many) powers of e with priority given to the main variables and "0" – the smallest term:

Definition 2.4.1 (the order on $\mathbb{T}_e$). For any $s_1, s_2 \in \mathbb{T}_e$ we define $\mathbf{s_1 \underset{\mathbb{T}_e}{<} s_2}$ if and only if

$$\begin{aligned} s_2 \neq 0 \ \text{ and } \ & a)\ s_1 = 0 \ \text{ or} \\ & b)\ s_1 \neq 0 \ \text{ and } \ b_1)\mathrm{T}(s_1) \underset{\mathbb{T}}{<} \mathrm{T}(s_2) \ \text{ or} \\ & \qquad\qquad\qquad\quad b_2)\mathrm{T}(s_1) = \mathrm{T}(s_2) \ \text{ and } \ \deg_e(s_2) < \deg_e(s_1). \end{aligned}$$

To borrow the theory of Gröbner bases for the algebra of extended polynomials to the algebra $\mathcal{P}_e$ we need the order $\underset{\mathbb{T}_e}{<}$ to be an extended term ordering: it must be quasi-monotonous and a well-ordering. It is quite clear that $\underset{\mathbb{T}_e}{<}$ is a total order (the rigorous verification of this fact is completely identical to the one for the block-orders, see e.g. [BW93] page 191 or [Win96] page 181, in the classical theory of Gröbner bases). To check the quasi-monotonicity property we need an easy technical lemma.

Lemma 2.4.1. Let $s_1, s_2 \in \mathbb{T}_e{}^*$. If $s_1 \underset{\mathbb{T}_e}{\leq} s_2$ then $\mathrm{T}(s_1) \underset{\mathbb{T}}{\leq} \mathrm{T}(s_2)$.

Proof. Let $s_1, s_2 \in \mathbb{T}_e{}^*$ be such that $s_1 \underset{\mathbb{T}_e}{\leq} s_2$. Assume for a contradiction that $\mathrm{T}(s_2) \underset{\mathbb{T}}{<} \mathrm{T}(s_1)$. Then, by part b_1) of definition (2.4.1), $s_2 \underset{\mathbb{T}_e}{<} s_1$. ☐

Lemma 2.4.2 (quasi-monotonicity of $\underset{\mathbb{T}_e}{<}$). The order $\underset{\mathbb{T}_e}{<}$ is quasi-monotonous.

Proof. Let $a_1, b_1 \in \mathbb{T}_e$ and $a_2, b_2 \in \mathbb{T}_e{}^*$ be terms such that

$$a_1 \underset{\mathbb{T}_e}{<} a_2, \tag{i}$$

$$b_1 \underset{\mathbb{T}_e}{\leq} b_2. \tag{ii}$$

We have to show that

$$\text{If} \quad a_2 \cdot b_2 \neq 0 \quad \text{then} \quad a_1 \cdot b_1 \underset{\mathbb{T}_e}{<} a_2 \cdot b_2.$$

Note that by part b) of lemma 2.3.1

$$\begin{aligned} \deg_e(a_2 \cdot b_2) &= \deg_e(a_2) + \deg_e(b_2), \\ \mathrm{T}(a_2 \cdot b_2) &= \mathrm{T}(a_2) \cdot \mathrm{T}(b_2). \end{aligned} \tag{1}$$

If $a_1{\cdot}b_1 = 0$ then, $a_1{\cdot}b_1 \underset{\mathbb{T}_e}{<} a_2{\cdot}b_2$ since, by part a) of definition 2.4.1, zero is the smallest element of $\mathbb{T}_e$. Assume $a_1{\cdot}b_1 \neq 0$. Then $a_1, b_1 \in {\mathbb{T}_e}^*$ and, by part b) of lemma 2.3.1,

$$\begin{aligned} \deg_e(a_1{\cdot}b_1) &= \deg_e(a_1) + \deg_e(b_1), \\ \mathrm{T}(a_1{\cdot}b_1) &= \mathrm{T}(a_1){\cdot}\mathrm{T}(b_1). \end{aligned} \tag{2}$$

By lemma 2.4.1

$$\begin{gathered} b_1 \underset{\mathbb{T}_e}{\leq} b_2 \quad \text{and} \quad a_1 \underset{\mathbb{T}_e}{<} a_2 \\ \Downarrow \\ \mathrm{T}(b_1) \underset{\mathbb{T}}{\leq} \mathrm{T}(b_2) \quad \text{and} \quad \mathrm{T}(a_1) \underset{\mathbb{T}}{\leq} \mathrm{T}(a_2). \end{gathered}$$

Since $\underset{\mathbb{T}}{<}$ is an admissible order on $\mathbb{T}$,

$$\begin{gathered} \mathrm{T}(b_1) \underset{\mathbb{T}}{\leq} \mathrm{T}(b_2) \quad \text{and} \quad \mathrm{T}(a_1) \underset{\mathbb{T}}{\leq} \mathrm{T}(a_2) \\ \Downarrow \\ \mathrm{T}(a_1){\cdot}\mathrm{T}(b_1) \underset{\mathbb{T}}{\leq} \mathrm{T}(a_1){\cdot}\mathrm{T}(b_2) \quad \text{and} \quad \mathrm{T}(a_1){\cdot}\mathrm{T}(b_2) \underset{\mathbb{T}}{\leq} \mathrm{T}(a_2){\cdot}\mathrm{T}(b_2) \\ \Downarrow \\ \mathrm{T}(a_1{\cdot}b_1) \overset{(2)}{=} \mathrm{T}(a_1){\cdot}\mathrm{T}(b_1) \quad \underset{\mathbb{T}}{\leq} \quad \mathrm{T}(a_2){\cdot}\mathrm{T}(b_2) \overset{(1)}{=} \mathrm{T}(a_2{\cdot}b_2). \end{gathered}$$

Consider two cases.

$(<)$: If $\mathrm{T}(a_1{\cdot}b_1) \underset{\mathbb{T}_e}{<} \mathrm{T}(a_2{\cdot}b_2)$ then $a_1{\cdot}b_1 \underset{\mathbb{T}_e}{<} a_2{\cdot}b_2$ by part b_1) of definition 2.4.1.

$(=)$: Otherwise $\mathrm{T}(a_1{\cdot}b_1) = \mathrm{T}(a_1{\cdot}b_2) = \mathrm{T}(a_2{\cdot}b_2)$ and, by the cancellation law in $\mathbb{T}$, $\mathrm{T}(b_1){=}\mathrm{T}(b_2)$, $\mathrm{T}(a_1){=}\mathrm{T}(a_2)$. Then, by part b_2) of definition 2.4.1,

$$\begin{gathered} \deg_e(b_1) \overset{(ii)}{\geq} \deg_e(b_2) \quad \text{and} \quad \deg_e(a_1) \overset{(i)}{>} \deg_e(a_2) \\ \Downarrow \\ \deg_e(a_1){+}\deg_e(b_1) \geq \deg_e(a_1){+}\deg_e(b_2) \quad \text{and} \quad \deg_e(a_1){+}\deg_e(b_2) > \deg_e(a_2){+}\deg_e(b_2) \\ \Downarrow \\ \deg_e(a_1{\cdot}b_1) \overset{(2)}{=} \deg_e(a_1) + \deg_e(b_1) \quad > \quad \deg_e(a_2) + \deg_e(b_2) \overset{(1)}{=} \deg_e(a_2{\cdot}b_2). \end{gathered}$$

Then $a_1{\cdot}b_1 \underset{\mathbb{T}_e}{<} a_2{\cdot}b_2$ by part b_2) of definition 2.4.1. □

Lemma 2.4.3. $\underset{\mathbb{T}_e}{<}$ is a well-ordering on $\mathbb{T}_e$ i.e. every non-empty subset $S \subseteq \mathbb{T}_e$ has the smallest element.

Proof. Let $\emptyset \neq S \subseteq \mathbb{T}_e$. If $0{\in}S$ then, by definition 2.4.1 of the order $\underset{\mathbb{T}_e}{<}$, "0" is the smallest element of S. Assume $S \subseteq {\mathbb{T}_e}^*$. Consider the set

$$M = \left\{ t{\in}\mathbb{T} \,\middle|\, \exists s{\in}S : \ \mathrm{T}(s){=}t \right\}$$

Clearly $M \neq \emptyset$ and, since $\underset{\mathbb{T}}{<}$ is a well-ordering, there exists $v{\in}M$ – the smallest element of M. Consider a finite (due to the bound on degrees of e) non-empty set

$$S_v = \{s{\in}S \ \mid \ \mathrm{T}(s){=}v\} \subseteq v{*}(e)^* \subseteq S.$$

Let $m{\in}S_v$ be the element with the highest degree of e among the elements of S_v. Then m is the smallest element of S. Indeed, take any $s{\in}S$. By assumption, $s{\neq}0$.

a) If $s{\in}S_v$ then $\mathrm{T}(s){=}\mathrm{T}(m)$ and $\deg_e(s){\leq}\deg_e(m)$ by the choice of m. Hence either $m{=}s$ or $m\underset{\mathbb{T}_e}{<}s$ by part b_2) of definition 2.4.1.

b) If $s{\in}S{\setminus}S_v$ then $v{=}\mathrm{T}(m)\underset{\mathbb{T}}{<}\mathrm{T}(s)$ by construction of S_v. Hence $m\underset{\mathbb{T}_e}{<}s$ by part b_1) of definition 2.4.1.

Then the smallest element w.r.t. $\underset{\mathbb{T}_e}{<}$ in the finite non-empty set

$$\left\{ s{\in}M \,\middle|\, \mathrm{T}(s){=}m \right\}$$

is the smallest element of S. □

Remark 2.4.1. The existence of the $e-$bound $\mathcal{B}$ is crucial for the well-ordering property of $\underset{\mathbb{T}_e}{<}$. Indeed, without the bound on the degrees of e the strictly descending w.r.t. $\underset{\mathbb{T}_e}{<}$ sequence $\{e^i\}_{i\in\mathbb{N}_o}$ of non-zero extended terms clearly does not have the smallest element.

Lemma 2.4.4. The order $\underset{\mathbb{T}_e}{<}$ is an extended term order.

Proof. By lemma 2.4.2 and 2.4.3. □

According to the latter lemma and the results of the previous section, the leading term function is quasi-stable on $\mathcal{P}_e$.

Remark 2.4.2. The condition $m{\cdot}\ell\mathrm{t}(f){\neq}0$ in the quasi-monotonicity property of an extended term order is crucial to keep the leading term function compatible with multiplication. Indeed, take $\mathcal{B}$=2, f=ex+y with y<x and m=e. Then

$$0 = e \cdot ex = e \cdot \ell\mathrm{t}(ex + y) = \\ m \cdot \ell\mathrm{t}(f) \neq \ell\mathrm{t}(m \cdot f) = \\ \ell\mathrm{t}\Big(e \cdot (ex+y)\Big) = \ell\mathrm{t}(e \cdot y) = e \cdot y.$$

2.4.1 The $\mathcal{B}$−free reduction in $\mathcal{P}_e$

We could borrow the reduction algorithm listed on page 47 for computing normal forms but it turns out that sometimes it is not reasonably efficient for the case of the polynomial ring $\mathcal{P}_e$ with the term order $\mathbb{T}_e$.

Example 2.4.1. Consider the ring $\mathcal{P}_e$=$\Bbbk[x, e]$. Let F=$\{x-ex\}$. The reduction algorithm applied to x modulo F would lead to the chain of $\mathcal{B}$ one step top-reductions

$$\underbrace{x \xrightarrow{F} e{\cdot}x \xrightarrow{F} \dots \xrightarrow{F} e^{\mathcal{B}}{\cdot}x}_{\mathcal{B}\text{ times}} = 0.$$

Is there a possibility to detect in few steps that x top-reduces to zero? The answer is yes.

The negative weight of the variable e in $\mathcal{P}_e$ allows to establish a more efficient version of the reduction algorithm as a modification of Teo Mora's tangent cone reduction [MT] which we call the $\mathcal{B}$−free reduction to reflect the fact that it does not rely on the existence of the e−bound $\mathcal{B}$ and moreover does not require the well-ordering property of $\underset{\mathbb{T}_e}{<}$ (which follows from the existence of the bound $\mathcal{B}$ due to the proof of lemma 2.4.3 and the subsequent remark) to argue the termination. It is similar to the generalization of the tangent cone reduction for the case of the *mixed* ordering type [GP96], [Gra] but does not require the homogenization of the input polynomials. The $\mathcal{B}$−free reduction can also be seen as a slightly modified "usual" top-reduction of extended polynomials (definition 2.3.12) but with a very special selection strategy.

To proceed with the construction of $\mathcal{B}$−free reduction we would need few technical facts about the properties of extended polynomials relative to the leading terms and specific to the order $\underset{\mathbb{T}_e}{<}$ listed in the lemmas below.

The order $\underset{\mathbb{T}_e}{<}$ is not a *global* order on extended terms in the sense that some terms are less than "1". In fact these are precisely terms which are a pure power of e.

Lemma 2.4.5. Let $s \in \mathbb{T}_e{}^*$. Then $s \leq 1$ if and only if $s \in (e)^* \Big(\subseteq \Bbbk[e] \Big)$.

Proof. By definition 2.4.1, $s \leq 1$ if and only if

$$\begin{aligned} &\mathrm{T}(s) \underset{\mathbb{T}}{<} \mathrm{T}(1) \text{ or} \\ &\mathrm{T}(s) = \mathrm{T}(1) \text{ and } \deg_e(s) \geq \deg_e(1). \end{aligned}$$

The first alternative is impossible because $\mathrm{T}(1)=1$ is the smallest term in $\mathbb{T}$ w.r.t. the admissible ordering $\underset{\mathbb{T}}{<}$. Thus

$$s = \mathrm{T}(s) \cdot e^{\deg_e(s)} = \mathrm{T}(1) \cdot e^{\deg_e(s)} = e^{\deg_e(s)}, \text{ where } \deg_e(s) \geq \deg_e(1) = 0.$$

□

An easy but valuable consequences of the latter lemma are as follows.

Corollary 2.4.1. Let $f \in \mathcal{P}_e \backslash \{0\}$ and $u \in \Bbbk[e]$. Then

a) $\ell\mathrm{t}(f) \underset{\mathbb{T}_e}{\leq} 1$ if and only if $f \in \Bbbk[e]$,

b) $f \in \mathrm{U}(\Bbbk[e])_{\mathcal{P}_e}$ if and only if $\ell\mathrm{t}(f)=1$,

c) $uf \neq 0$ and $\ell\mathrm{t}(uf) = \ell\mathrm{t}(f)$.

Proof.

a) Let $f \in \mathcal{P}_e \backslash \{0\}$. Assume $\ell\mathrm{t}(f) \underset{\mathbb{T}_e}{\leq} 1$. Take $s \in \mathrm{supp}\,(f)$. Then $s \underset{\mathbb{T}_e}{\leq} \ell\mathrm{t}(f) \underset{\mathbb{T}_e}{\leq} 1$ and, by the lemma, $s = e^{\deg_e(s)} \in \Bbbk[e]$. Thus f is a linear combination of terms from $\Bbbk[e]$. Conversely, if $f \in \Bbbk[e]$ then $\ell\mathrm{t}(f) = e^{\deg_e(\ell\mathrm{t}(f))} \underset{\mathbb{T}_e}{\leq} 1$.

b) Let

$$f = \sum_{0 \leq i < \mathcal{B}} \alpha_i \cdot e^i \in \mathrm{U}(\Bbbk[e])_{\mathcal{P}_e} \Big(\subseteq \Bbbk[e] \Big).$$

Then, by part *g)* of theorem 2.3.1, $\alpha_o \neq 0$. Hence

$$\ell\mathrm{t}(f) = \max_{\underset{\mathbb{T}_e}{<}} \Big(\mathrm{supp}\,(f) \Big) = \max_{\underset{\mathbb{T}_e}{<}} \Big(\{ e^i \mid \alpha_i \neq 0,\ 0 \leq i < \mathcal{B} \} \Big) =$$

$$e^{\min\left(\{ i \,\mid\, \alpha_i \neq 0,\, 0 \leq i < \mathcal{B} \} \right)} = e^o = 1.$$

Conversely, if $\ell\mathrm{t}(f)=1$ then, by part *a)*, $f \in \Bbbk[e]$. Since $\ell\mathrm{t}(f) \in \mathrm{supp}\,(f)$, the coefficient (at $\ell\mathrm{t}(f)=e^o$) $\alpha_o \neq 0$. Then $f \in \mathrm{U}(\Bbbk[e])_{\mathcal{P}_e}$ by part *g)* of theorem 2.3.1.

c) By part *b)* and the quasi-stability of the leading term function under multiplication in $\mathcal{P}_e$ (lemma 2.3.11)

$$\ell\mathrm{t}(u)\ell\mathrm{t}(f) = 1\ell\mathrm{t}(f) = \ell\mathrm{t}(f) \neq 0$$

follows

$$\ell\mathrm{t}(uf) = \ell\mathrm{t}(u)\ell\mathrm{t}(f) = \ell\mathrm{t}(f).$$

$\square$

The following notions will be used in the argumentation of the termination of the $\mathcal{B}-$free reduction. Given a non-zero polynomial in $\mathcal{P}_e$ we are interested in the $e-$degrees of the terms in the support with the same main part as the leading term. Namely, the maximal, the minimal degree and their difference are essential for the subsequent construction.

Notation. Let $f \in \mathcal{P}_e\backslash\{0\}$. Denote

$$L(f) = \Big\{v\in \operatorname{supp}(f) \;\Big|\; \mathrm{T}(v)=\mathrm{T}\Big(\ell\mathrm{t}(f)\Big)\Big\} \neq \emptyset.$$

Definition 2.4.2. Let $f \in \mathcal{P}_e\backslash\{0\}$. Then

- $\ell\deg_\varepsilon(f) = \max\Big(\deg_e(v) \mid v\in L(f)\Big)$ is the **leading** $e-$**degree** of f,
- $\ell\mathrm{ord}_\varepsilon(f) = \min\Big(\deg_e(v) \mid v\in L(f)\Big)$ is the **leading** $e-$**order** of f,
- $\ell\Delta_\varepsilon(f) = \ell\deg_\varepsilon(f)-\ell\mathrm{ord}_\varepsilon(f)$ is the **leading** $e-$**difference** of f.

It is easy to see that $\ell\mathrm{ord}_\varepsilon(f)=\deg_e\Big(\ell\mathrm{t}(f)\Big)$ according to the definition of the order $\underset{\mathbb{T}_e}{<}$.

In certain situations the leading $e-$degree and the leading $e-$order behave exactly like the usual degree and the order of univariate polynomials under addition.

Lemma 2.4.6. Let $f, g\in\mathcal{P}_e\backslash\{0\}$ such that $f+g\neq 0$ and

$$\mathrm{T}\Big(\ell\mathrm{t}(f)\Big) = \mathrm{T}\Big(\ell\mathrm{t}(g)\Big)=\mathrm{T}\Big(\ell\mathrm{t}(f+g)\Big). \qquad (*)$$

Then

a) $\ell\mathrm{ord}_\varepsilon(f+g)\geq \min\Big(\ell\mathrm{ord}_\varepsilon(f), \ell\mathrm{ord}_\varepsilon(g)\Big).$

b) $\ell\deg_\varepsilon(f+g) \leq \max\Big(\ell\deg_\varepsilon(f), \ell\deg_\varepsilon(g)\Big)$,

c) If $\ell\mathrm{ord}_\varepsilon(f) = \ell\mathrm{ord}_\varepsilon(g)$ then $\ell\Delta_\varepsilon(f+g) \leq \max\Big(\ell\Delta_\varepsilon(f), \ell\Delta_\varepsilon(g)\Big)$.

Proof.

a) By lemma 2.3.10. $\ell\mathrm{t}(f+g) \underset{\mathbb{T}_e}{\leq} \max\Big(\ell\mathrm{t}(f), \ell\mathrm{t}(g)\Big)$. Then, by $(*)$ and part b_2) of definition 2.4.1 of the order $\underset{\mathbb{T}_e}{<}$,

$$\ell\mathrm{ord}_\varepsilon(f+g) = \deg_e\Big(\ell\mathrm{t}(f+g)\Big) \geq$$
$$\min\Big(\deg_e\big(\ell\mathrm{t}(f)\big), \deg_e\big(\ell\mathrm{t}(g)\big)\Big) = \min\Big(\ell\mathrm{ord}_\varepsilon(f), \ell\mathrm{ord}_\varepsilon(g)\Big).$$

b) By part b) of lemma 2.3.3

$$\mathrm{supp}\,(f+g) \subseteq \mathrm{supp}\,(f) \cup \mathrm{supp}\,(g)\,. \tag{1}$$

Then

$$\begin{aligned} L(f+g) &= \Big\{v \in \mathrm{supp}\,(f+g) \mid \mathrm{T}(v) = \mathrm{T}\big(\ell\mathrm{t}(f+g)\big)\Big\} \\ &\overset{(1)}{\subseteq} \Big\{v \in \mathrm{supp}\,(f) \cup \mathrm{supp}\,(g) \mid \mathrm{T}(v) = \mathrm{T}\big(\ell\mathrm{t}(f+g)\big)\Big\} \\ &\overset{(*)}{=} \Big\{v \in \mathrm{supp}\,(f) \mid \mathrm{T}(v) = \mathrm{T}\big(\ell\mathrm{t}(f)\big)\Big\} \cup \\ &\quad \Big\{v \in \mathrm{supp}\,(g) \mid \mathrm{T}(v) = \mathrm{T}\big(\ell\mathrm{t}(g)\big)\Big\} \\ &= L(f) \cup L(g) \end{aligned} \tag{2}$$

which follows

$$\begin{aligned} \ell\deg_\varepsilon(f+g) &= \max\Big(\deg_e(v) \mid v \in L(f+g)\Big) \\ &\overset{(2)}{\leq} \max\Big(\deg_e(v) \mid v \in L(f) \cup L(g)\Big) \\ &\overset{(*)}{=} \max\Bigg(\max\Big(\deg_e(v) \mid v \in L(f)\Big), \max\Big(\deg_e(v) \mid v \in L(g)\Big)\Bigg) \\ &= \max\Big(\ell\deg_\varepsilon(f), \ell\deg_\varepsilon(g)\Big). \end{aligned}$$

c) It is an easy combination of parts a) and b). Let $d = \ell\mathrm{ord}_\varepsilon(f) = \ell\mathrm{ord}_\varepsilon(g)$. Then, by part a),

$$d \leq \max\Big(\ell\mathrm{ord}_\varepsilon(f), \ell\mathrm{ord}_\varepsilon(g)\Big) \tag{3}$$

and

$$\begin{aligned}
\boxed{\ell\Delta_\varepsilon(f{+}g)} &= \ell\mathrm{deg}_\varepsilon(f{+}g) - \ell\mathrm{ord}_\varepsilon(f{+}g)\\
&\boxed{\overset{b),(3)}{\leq}} \max\Big(\ell\mathrm{deg}_\varepsilon(f), \ell\mathrm{deg}_\varepsilon(g)\Big) - d\\
&= \max\Big(\ell\mathrm{deg}_\varepsilon(f){-}d, \ell\mathrm{deg}_\varepsilon(g){-}d\Big)\\
&= \max\Big(\ell\mathrm{deg}_\varepsilon(f){-}\ell\mathrm{ord}_\varepsilon(f), \ell\mathrm{deg}_\varepsilon(g){-}\ell\mathrm{ord}_\varepsilon(g)\Big)\\
&= \boxed{\max\Big(\ell\Delta_\varepsilon(f), \ell\Delta_\varepsilon(g)\Big)}\,.
\end{aligned}$$

□

Remark 2.4.3. The statement of the previous lemma is wrong without the condition

$$\mathrm{T}\Big(\ell\mathrm{t}(f)\Big) = \mathrm{T}\Big(\ell\mathrm{t}(g)\Big){=}\mathrm{T}\Big(\ell\mathrm{t}(f{+}g)\Big).$$

Indeed, let $\mathcal{B}{=}3$, take $f{=}ex{+}1{+}e^2$, $g{=}-ex$. Then $h{=}f{+}g{=}1{+}e^2$,

$$\ell\mathrm{ord}_\varepsilon(f) = \ell\mathrm{deg}_\varepsilon(f) = \ell\mathrm{ord}_\varepsilon(g) = \ell\mathrm{deg}_\varepsilon(g) = \mathrm{deg}_e(ex) = 1$$

and

$$\begin{aligned}
\ell\mathrm{ord}_\varepsilon(h) &= \ \mathrm{deg}_e(1) = 0 < 1 = \max\Big(\ell\mathrm{ord}_\varepsilon(f), \ell\mathrm{ord}_\varepsilon(g)\Big),\\
\ell\mathrm{deg}_\varepsilon(h) &= \mathrm{deg}_e\big(e^2\big) = 2 > 1 = \max\Big(\ell\mathrm{deg}_\varepsilon(f), \ell\mathrm{deg}_\varepsilon(g)\Big).
\end{aligned}$$

The situation with multiplication in $\mathcal{P}_e$ under the similar condition on the main parts of the leading terms is slightly different from the univariate polynomials case.

Lemma 2.4.7. Let $f{\in}\Bbbk^*$, $\alpha{\in}\Bbbk^*$. Then

1*a*) $\ell\mathrm{ord}_\varepsilon(\alpha{\cdot}f) = \ell\mathrm{ord}_\varepsilon(f)$,
1*b*) $\ell\mathrm{deg}_\varepsilon(\alpha{\cdot}f) = \ell\mathrm{deg}_\varepsilon(f)$,
1*c*) $\ell\Delta_\varepsilon(\alpha{\cdot}f) = \ell\Delta_\varepsilon(f)$.

If furthermore $s{\in}{\mathbb{T}_e}^*$ is such that $s{\cdot}\ell\mathrm{t}(f){\neq}0$ then $s{\cdot}f{\neq}0$ and

2*a*) $\ell\mathrm{ord}_\varepsilon(s{\cdot}f) = \mathrm{deg}_e(s){+}\ell\mathrm{ord}_\varepsilon(f)$,
2*b*) $\ell\mathrm{deg}_\varepsilon(s{\cdot}f) \leq \mathrm{deg}_e(s){+}\ell\mathrm{deg}_\varepsilon(f)$,
2*c*) $\ell\Delta_\varepsilon(s{\cdot}f) \leq \ell\Delta_\varepsilon(f)$.

Proof.

1*a*) By lemma 2.3.9

$$\ell\mathrm{t}(\alpha\cdot f) = \ell\mathrm{t}(f). \tag{1}$$

Then

$$\ell\mathrm{ord}_\varepsilon(\alpha\cdot f) = \deg_e\Big(\ell\mathrm{t}(\alpha\cdot f)\Big) \overset{(1)}{=} \deg_e\Big(\ell\mathrm{t}(f)\Big) = \ell\mathrm{ord}_\varepsilon(f).$$

1*b*) By part *a*) of lemma 2.3.3

$$\mathrm{supp}\,(\alpha\cdot f) = \mathrm{supp}\,(f) \tag{2}$$

and, by lemma 2.3.9,

$$\ell\mathrm{t}(\alpha\cdot f) = \ell\mathrm{t}(f) \tag{3}$$

Then

$$\begin{aligned}\ell\mathrm{deg}_\varepsilon(\alpha\cdot f) &= \max\Big(\deg_e(v) \mid v{\in}\,\mathrm{supp}\,(\alpha\cdot v)\,, \mathrm{T}(v){=}\mathrm{T}\big(\ell\mathrm{t}(\alpha\cdot f)\big)\Big)\\ &\overset{(2)}{=} \max\Big(\deg_e(v) \mid v{\in}\,\mathrm{supp}\,(v)\,, \mathrm{T}(v){=}\mathrm{T}\big(\ell\mathrm{t}(\alpha\cdot f)\big)\Big)\\ &\overset{(3)}{=} \max\Big(\deg_e(v) \mid v{\in}\,\mathrm{supp}\,(v)\,, \mathrm{T}(v){=}\mathrm{T}\big(\ell\mathrm{t}(f)\big)\Big)\\ &= \ell\mathrm{deg}_\varepsilon(f).\end{aligned}$$

1*c*) By definition of the leading $e-$difference it is an easy combination of parts 1*a*) and 1*b*)

$$\ell\Delta_\varepsilon(\alpha\cdot f) = \ell\mathrm{deg}_\varepsilon(\alpha\cdot f) - \ell\mathrm{ord}_\varepsilon(\alpha\cdot f) \overset{1a),1b)}{=} \ell\mathrm{deg}_\varepsilon(f) - \ell\mathrm{ord}_\varepsilon(f) = \ell\Delta_\varepsilon(f).$$

2*a*) By part *a*) of corollary 2.3.6 the condition $s\cdot\ell\mathrm{t}(f){\neq}0$ follows $s\cdot f{\neq}0$ and

$$\ell\mathrm{t}(s\cdot f) = s\cdot\ell\mathrm{t}(f). \tag{4}$$

Then, using part *a*) of lemma 2.3.1,

$$\ell\mathrm{ord}_\varepsilon(s\cdot f) = \deg_e\Big(\ell\mathrm{t}(s\cdot f)\Big) \overset{(4)}{=} \deg_e\Big(s\cdot\ell\mathrm{t}(f)\Big) = \deg_e(s) + \deg_e\Big(\ell\mathrm{t}(f)\Big).$$

2*b*) By part *d*) of lemma 2.3.3

$$\mathrm{supp}\,(s\cdot f) = s{*}\,\mathrm{supp}\,(f)\,, \tag{5}$$

by (4) and part b) of lemma 2.3.1

$$\mathrm{T}\Big(\ell\mathrm{t}(s{\cdot}f)\Big) \overset{(4)}{=} \mathrm{T}\Big(s{\cdot}\ell\mathrm{t}(f)\Big) = \mathrm{T}(s){\cdot}\mathrm{T}\Big(\ell\mathrm{t}(f)\Big) \tag{6}$$

and, by the cancellation law in $\mathbb{T}$, for any $w{\in}{\mathbb{T}_e}^*$

$$\mathrm{T}(s){\cdot}\mathrm{T}(w) = \mathrm{T}(s){\cdot}\mathrm{T}\Big(\ell\mathrm{t}(f)\Big) \iff \mathrm{T}(w) = \mathrm{T}\Big(\ell\mathrm{t}(f)\Big). \tag{7}$$

Then, using part a) of lemma 2.3.1,

$$\begin{aligned}
&\boxed{\ell\mathrm{deg}_\varepsilon(s{\cdot}f)} = \\
&\quad \max\Big(\mathrm{deg}_e(v) \mid v{\in}\,\mathrm{supp}\,(s{\cdot}f), \mathrm{T}(v){=}\mathrm{T}\big(\ell\mathrm{t}(s{\cdot}f)\big)\Big)\\
&\overset{(5)}{=} \max\Big(\mathrm{deg}_e(v) \mid v{\in}s{*}\,\mathrm{supp}\,(f), \mathrm{T}(v){=}\mathrm{T}\big(\ell\mathrm{t}(s{\cdot}f)\big)\Big)\\
&\overset{(6)}{=} \max\Big(\mathrm{deg}_e(v) \mid v{\in}s{*}\,\mathrm{supp}\,(f), \mathrm{T}(v){=}\mathrm{T}(s){\cdot}\mathrm{T}\big(\ell\mathrm{t}(f)\big)\Big)\\
&\overset{(7)}{=} \max\Big(\mathrm{deg}_e(sw) \mid w{\in}\,\mathrm{supp}\,(f),\ sw{\neq}0, \mathrm{T}(w){=}\mathrm{T}\big(\ell\mathrm{t}(f)\big)\Big)\\
&= \mathrm{deg}_e(s){+}\max\Big(\mathrm{deg}_e(w) \mid w{\in}\,\mathrm{supp}\,(f),\ sw{\neq}0, \mathrm{T}(w){=}\mathrm{T}\big(\ell\mathrm{t}(f)\big)\Big)\\
&\boxed{\leq}\ \mathrm{deg}_e(s){+}\max\Big(\mathrm{deg}_e(w) \mid w{\in}\,\mathrm{supp}\,(f), \mathrm{T}(w){=}\mathrm{T}\big(\ell\mathrm{t}(f)\big)\Big)\\
&= \boxed{\mathrm{deg}_e(s){+}\ell\mathrm{deg}_\varepsilon(f)}\ .
\end{aligned}$$

2c) It is an easy combination of parts 2a) and 2b):

$$\begin{aligned}
\ell\Delta_\varepsilon(sf) = \ell\mathrm{deg}_\varepsilon(sf) - \ell\mathrm{ord}_\varepsilon(sf) &\overset{2a),2b)}{\leq} \\
\ell\mathrm{deg}_\varepsilon(f){+}\mathrm{deg}_e(s) - \ell\mathrm{ord}_\varepsilon(f){+}\mathrm{deg}_e(s) &= \\
\ell\mathrm{deg}_\varepsilon(f) - \ell\mathrm{ord}_\varepsilon(f) &= \ell\Delta_\varepsilon(f).
\end{aligned}$$

□

Remark 2.4.4.

- The bound on the leading $e-$degree from the part 2a) could be taken more precise. Indeed, the existence of the bound $\mathcal{B}$ on $e-$degrees of extended terms imposes an extra constraint $\ell\deg_\varepsilon(sf)<\mathcal{B}$ and hence, a more precise and meaningful bound could be

$$\ell\deg_\varepsilon(sf)\leq\min\Big(\mathcal{B}-1,\deg_e(s)+\ell\deg_\varepsilon(f)\Big)$$

but we neglect this fact to keep the argumentation in the proof of the forthcoming statement about a bound on the result of one step top-reduction independent from the existence of the bound $\mathcal{B}$ on $e-$degrees and hence to keep the moral right to refer our alternative reduction as $\mathcal{B}-$free reduction. Strictly speaking we have used the existence of $\mathcal{B}$ in the derivation of part $2a$) but it is quite clear how the derivation can be modified (by eliminating the condition related to zero divisors among extended terms which is, in fact, influenced by the existence of the bound on $e-$degrees) for the $\mathcal{B}-$free case also resulting the to strict equality $\ell\deg_\varepsilon(sf)=\ell\deg_\varepsilon(f)+\deg_e(s)$ instead of the upper bound.

- The statement $2a$) does not hold without the condition $s\cdot\ell\mathrm{t}(f)\neq 0$. Indeed, take $\mathcal{B}=3$, $f=e^2x+1$, $s=e$. Then $s\cdot f=e$ and

$$\ell\mathrm{ord}_\varepsilon(s\cdot f)=\deg_e(e)=1<3=\\ \deg_e(e)+\deg_e\big(e^2x\big)=\deg_e(e)+\ell\mathrm{ord}_\varepsilon(f).$$

- The bound

$$\ell\mathrm{ord}_\varepsilon(s\cdot f)\leq\ell\mathrm{ord}_\varepsilon(f)+\deg_e(s)$$

on the leading $e-$order as well as the bound on the leading $e-$degree from part $2b$) and hence the bound on the leading $e-$difference from part $2c$) are valid under a weaker condition $s\cdot f\neq 0$ but the verification of these bounds gets slightly more complicated in our frame of argumentation while the presented statement is natural and sufficient for the sole purpose of the lemma as a placehold for technical details in the forming of the leading $e-$difference bound of the result of one step top-reduction.

The role of the leading $e-$difference for the $\mathcal{B}-$free reduction is quite similar to what the notion of **ecart** means for the original tangent cone reduction. Indeed, the bound on the ecart within the tangent cone reduction sequence ([MT] lemma 9) corresponds to the bound on the leading $e-$difference as follows.

Lemma 2.4.8. Let $f,g,h\in\mathcal{P}_e\backslash\{0\}$ such that $f\xrightarrow{h}g$. If

$$\mathrm{T}\Big(\ell\mathrm{t}(f)\Big)=\mathrm{T}\Big(\ell\mathrm{t}(g)\Big)\qquad(*)$$

then

$a)$ $\ell\mathrm{ord}_\varepsilon(f) < \ell\mathrm{ord}_\varepsilon(g)$,

$b)$ $\ell\Delta_\varepsilon(g) < \max\Big(\ell\Delta_\varepsilon(f), \ell\Delta_\varepsilon(h)\Big)$.

Proof.

$a)$ By lemma 2.3.13

$$\ell\mathrm{t}(g) \underset{\mathbb{T}_e}{<} \ell\mathrm{t}(f).$$

It is easy to see that the only possibility to have (1) in conjunction with the condition $\mathrm{T}\Big(\ell\mathrm{t}(f)\Big) = \mathrm{T}\Big(\ell\mathrm{t}(g)\Big)$ is the one provided by part $b_2)$ of definition 2.4.1 of the order $\underset{\mathbb{T}_e}{<}$ which is

$$\ell\mathrm{ord}_\varepsilon(f) = \deg_e\Big(\ell\mathrm{t}(f)\Big) < \deg_e\Big(\ell\mathrm{t}(g)\Big) = \ell\mathrm{ord}_\varepsilon(g).$$

$b)$ Let $\alpha\in\mathbb{k}^*$ and $s\in{\mathbb{T}_e}^*$ be such that

$$\alpha\ell\mathrm{c}(h) = \ell\mathrm{c}(f), s\ell\mathrm{t}(h) = \ell\mathrm{t}(f) \quad \text{and} \quad g = f-\alpha\cdot s\cdot h. \tag{1}$$

provided by definition 2.3.12 of one step top-reduction. We claim that

$$\ell\Delta_\varepsilon(f-\alpha\cdot s\cdot h) \overset{(!)}{\leq} \max\Big(\ell\Delta_\varepsilon(f), \ell\Delta_\varepsilon(\alpha\cdot s\cdot h)\Big) \overset{(!!)}{=}$$
$$\max\Big(\ell\Delta_\varepsilon(f), \ell\Delta_\varepsilon(s\cdot h)\Big) \overset{(!!!)}{\leq} \max\Big(\ell\Delta_\varepsilon(f), \ell\Delta_\varepsilon(h)\Big).$$

Indeed,

- **The first inequality**. By lemma 2.3.12

$$\alpha\cdot s\cdot h \neq 0, \tag{2}$$

$$\ell\mathrm{t}(\alpha\cdot s\cdot h) = \ell\mathrm{t}(s\cdot h) = s\cdot\ell\mathrm{t}(h) \overset{(1)}{=} \ell\mathrm{t}(f) \tag{3}$$

which follows

$$\mathrm{T}\Big(\ell\mathrm{t}(\alpha\cdot s\cdot h)\Big) \overset{(3)}{=} \mathrm{T}\Big(\ell\mathrm{t}(f)\Big) \overset{(*)}{=} \mathrm{T}\Big(\ell\mathrm{t}(g)\Big) \tag{4}$$

and

$$\ell\mathrm{ord}_\varepsilon(f) = \deg_e\Big(\ell\mathrm{t}(f)\Big) \overset{(3)}{=} \deg_e\Big(\ell\mathrm{t}(\alpha\cdot s\cdot h)\Big) = \ell\mathrm{ord}_\varepsilon(\alpha\cdot s\cdot h). \tag{5}$$

Then

$$\ell\Delta_\varepsilon(f-\alpha\cdot s\cdot h) \overset{(2),(4),(5)}{\leq} \max\Big(\ell\Delta_\varepsilon(f), \ell\Delta_\varepsilon(\alpha\cdot s\cdot h)\Big) \tag{!}$$

by part $c)$ of lemma 2.4.6.

- **The equality.** By part 1c) of lemma 2.4.7

$$\alpha \overset{(1)}{\neq} 0,\ s{\cdot}h \overset{(2)}{\neq} 0 \quad\Longrightarrow\quad \ell\Delta_\varepsilon(\alpha{\cdot}s{\cdot}h) = \ell\Delta_\varepsilon(s{\cdot}h)$$

  and hence

$$\max\Big(\ell\Delta_\varepsilon(f), \ell\Delta_\varepsilon(\alpha{\cdot}s{\cdot}h)\Big) = \max\Big(\ell\Delta_\varepsilon(f), \ell\Delta_\varepsilon(s{\cdot}h)\Big). \qquad (!!)$$

- **The second inequality.** By assumption, $h{\neq}0$. Then, by part 2c) of lemma 2.4.7,

$$s \overset{(1)}{\neq} 0,\ h \neq 0,\ s{\cdot}\ell\mathrm{t}(h) \overset{(3)}{\neq} 0 \quad\Longrightarrow\quad \ell\Delta_\varepsilon(s{\cdot}h) \leq \ell\Delta_\varepsilon(h)$$

  and hence

$$\max\Big(\ell\Delta_\varepsilon(f), \ell\Delta_\varepsilon(s{\cdot}h)\Big) \leq \max\Big(\ell\Delta_\varepsilon(f), \ell\Delta_\varepsilon(h)\Big). \qquad (!!!)$$

□

The next theorem (cf. lemma 10 in [MT]) is the key for the $\mathcal{B}-$free reduction although it might seem to be quite surprising because the statement of the theorem is absolutely trivial due to the well-ordering property of $\underset{\mathbb{T}_e}{<}$. To be more specific the proof of the theorem which does not rely on the well-ordering property is the key.

Theorem 2.4.1. Given $f{\in}\mathcal{P}_e$ and a finite set $F{\subseteq}\mathcal{P}_e\backslash\{0\}$. Consider a sequence $\Gamma{=}\{g_o,\ldots,g_i,\ldots\}{\subseteq}\mathcal{P}_e$ of non-zero polynomials defined as

(G_1) $g_0 = f$;

(G_2) $g_i \xrightarrow{h_i} g_{i+1}$, $h_i{\in}F_i$;

(G_3) if $h{\in}F_i$ and $\ell\mathrm{t}(h)|\ell\mathrm{t}(g_i)$ then $\ell\Delta_\varepsilon(h_i){\leq}\ell\Delta_\varepsilon(h)$.

where $F_0{=}F$, $F_i{=}F_{i-1}\cup\{g_{i-1}\}$ for $i{>}0$. Then Γ is finite.

Proof. If $g_k{=}0$ for some index $k{\in}\mathbb{N}_o$ then g_k does not top-reduce in one step by definition and hence the sequence Γ is finite. Let $f{\in}\mathcal{P}_e\backslash\{0\}$ and $\Gamma{\subseteq}\mathcal{P}_e\backslash\{0\}$ be an infinite sequence. Note that the leading terms $\{\ell\mathrm{t}(g_i)\}$ form a strictly descending chain w.r.t. $\underset{\mathbb{T}_e}{<}$. Consider the set

$$\emptyset \neq M = \Big\{\mathrm{T}\Big(\ell\mathrm{t}(g_i)\Big)\Big\}_{i\in\mathbb{N}_o} \subseteq \mathbb{T}.$$

Since $\underset{\mathbb{T}}{<}$ is a well-ordering there exists m – the smallest element in M. Let k_1 be an index such that $\mathrm{T}\Big(\ell\mathrm{t}(g_{k_1})\Big) = m$. Let $i{\geq}k_1$. Then, according to definition 2.4.1 of the order $\underset{\mathbb{T}_e}{<}$,

$$\mathrm{T}\Big(\ell\mathrm{t}(g_i)\Big) = \mathrm{T}\Big(\ell\mathrm{t}(g_{k_1})\Big)^6. \tag{2.1}$$

Consider the family of sets

$$D_i = \Big\{h \in F_i \;\Big|\; \ell\mathrm{t}(h)|\ell\mathrm{t}(g_i)\Big\} \subseteq F_i, \tag{2}$$

denote

$$\Delta_i = \min\Big\{\ell\Delta_\varepsilon(h) \;\Big|\; h \in D_i\Big\}. \tag{3}$$

Let $j{>}k_1$. By (G_2), $h_j{\in}D_j$, by (G_3), $\ell\Delta_\varepsilon(h_j){=}\Delta_j$. Furthermore, it is easy to see that

$$D_{j-1}\bigcup\{g_{j-1}\} \subseteq D_j. \tag{4}$$

Indeed, $h{\in}F_j$ and $\ell\mathrm{t}(h)|\ell\mathrm{t}(g_{j-1})$ for any $h{\in}D_{j-1}{\cup}\{g_{j-1}\}$ by construction of the sets F_j and D_{j-1}; also, $\ell\mathrm{t}(g_{j-1})|\ell\mathrm{t}(g_j)$ according to (1) and the inequality

$$\deg_e\Big(\ell\mathrm{t}(g_j)\Big) = \ell\mathrm{ord}_\varepsilon(g_j) > \ell\mathrm{ord}_\varepsilon(g_{j-1}) = \deg_e\Big(\ell\mathrm{t}(g_{j-1})\Big)$$

provided by part a) of lemma 2.4.8. Hence $\ell\mathrm{t}(h)|\ell\mathrm{t}(g_j)$ and, by (2), $h{\in}D_j$ which gives (4) as well as, by (3), an important consequence

$$\Delta_j \leq \min\Big(\Delta_{j-1}, \ell\Delta_\varepsilon(g_{j-1})\Big) \leq \Delta_{j-1}. \tag{5}$$

On the other hand, by part b) of lemma 2.4.8 and condition (G_2),

$$\ell\Delta_\varepsilon(g_j) < \max\Big(\Delta_{j-1}, \ell\Delta_\varepsilon(g_{j-1})\Big). \tag{6}$$

[6]Besides the well-ordering property of $\underset{\mathbb{T}_e}{<}$ we could now use part a) of Lemma 2.4.8 to show that the sequence $\{\ell\mathrm{ord}_\varepsilon(g_i)\}$ is a strictly ascending chain for $i > k_1$ and thus, due to the bound $\ell\mathrm{ord}_\varepsilon(g_i) < \mathcal{B}$, the sequence $\{g_i\}$ is finite. It is a more elaborated version of the argument that $\underset{\mathbb{T}_e}{<}$ is a well-order. In this case the condition (G_3) is, in fact, superfluous. However we prefer a bit less transparent argument based on part b) of lemma 2.4.8 which does not depend on the existence of the $e-$degree bound.

The bounds (5) and (6) give a bound on the leading $e-$difference of g_{j+1}:

$$\begin{aligned}
\boxed{\max\Big(\Delta_j, \ell\Delta_\varepsilon(g_j)\Big)} &\overset{(5)}{\leq} \max\Big(\Delta_{j-1}, \ell\Delta_\varepsilon(g_j)\Big) \\
&\overset{(6)}{\leq} \max\left(\Delta_{j-1}, \max\Big(\Delta_{j-1}, \ell\Delta_\varepsilon(g_{j-1})\Big)\right) \\
&= \boxed{\max\Big(\Delta_{j-1}, \ell\Delta_\varepsilon(g_{j-1})\Big)} \qquad (7)
\end{aligned}$$

and, by inductive argument based on (7),

$$\ell\Delta_\varepsilon(g_{j+1}) \overset{(6)}{<} \max\Big(\Delta_j, \ell\Delta_\varepsilon(g_j)\Big) \leq \overset{(7)}{\cdots} \leq \max\Big(\Delta_{k_1}, \ell\Delta_\varepsilon(g_{k_1})\Big) \ .$$

In virtue of the latter bound, the set of values $\Theta = \{\ell\Delta_\varepsilon(g_k) \mid k{>}k_1\}$ is finite. Let $\Delta{\in}\Theta$ be the minimal difference which repeats infinitely many times for $k{>}k_1$. Then there exists an index $k_2{>}k_1$ such that $\Delta_k{\geq}\Delta$ for all $k{>}k_2$. Take $j_2{>}j_1{>}k_2$ such that $\ell\Delta_\varepsilon(g_{j_2}){=}\ell\Delta_\varepsilon(g_{j_1}){=}\Delta$. Then, by inductive argument based on (7),

$$\ell\Delta_\varepsilon(g_{j_2+1}) \overset{(6)}{<} \max\Big(\Delta_{j_2}, \ell\Delta_\varepsilon(g_{j_2})\Big) \leq \overset{(7)}{\cdots} \leq \max\Big(\Delta_{j_1}, \ell\Delta_\varepsilon(g_{j_1})\Big)$$

which contradicts the choice of k_2. □

Remark 2.4.5. The argument used in the proof of the last part of the theorem above does not incorporate the existence of $\varepsilon-$degree bound $\mathcal{B}$. It has certain valuable advantages.

- It is possible to establish the generalization of Teo Mora's tangent cone reduction and eventually translate the complete theory of Gröbner bases to the algebra $\mathcal{P}_e$ with e – a free variable which is, in a sense, equivalent to $\mathcal{B}{=}\infty$. To establish the notion of a normal form w.r.t. the order $\underset{\mathbb{T}_e}{<}$ one has to consider $\mathcal{P}_e = \left\{\frac{f}{1+p} \mid f{\in}\mathcal{P}, p{\in}\langle e\rangle_{\mathbb{k}[e]}\right\}$. The proof of the theorem above does not require homogenization of the source polynomials and thus can be an alternative to the Lazard's method used in [GGM+96], [GP96], [Laz83] in the reduction part of the Gröbner basis computation algorithm.
- The conditions (G_1) and (G_2) mean that Γ is a kind of a top-reduction sequence while the extra condition (G_3) makes possible the argumentation of the finiteness of $\{g_i\}$ used in the proof based on stabilization

of the leading $e-$ difference of the elements of $\{g_i\}$. From the computational point of view (G_3) can be reconsidered as a reduction strategy which can, sometimes, result to a faster termination of the reduction procedure since it does not necessary have to "hit" the $e-$degree bound $\mathcal{B}$ when the main part of the leaders in the sequence $\{g_i\}$ has been "stabilized": every time there is choice the top-reduction should be done by an element with the lowest possible leading $e-$difference.

Example 2.4.2 (Example 2.4.1 continued). Let $g_o{=}x$, $h{=}x{-}ex$, $F_o{=}\{h\}$.

$$\begin{array}{lll} g_o & \xrightarrow{x-ex\in F_o} & ex{=}g_1, F_1{=}F_o\bigcup\{g_o\}{=}\{x{-}ex, x\} \\ g_1 & \xrightarrow{x\in F_1} & 0 \end{array}$$

by which we reconstruct

$$0 = g_o{-}h{-}e{\cdot}g_o = (1{-}e){\cdot}g_o{-}h.$$

By part b) of lemma 2.4.5, $1{-}e \in \mathrm{U}(\mathcal{P}_e)$. Thus

$$g_o{-}\frac{h}{1{-}e} = 0 = \mathrm{NF}_{g_o}(\langle h\rangle_{\mathcal{P}_e})\,.$$

Definition 2.4.3. A set $F{\subseteq}\mathcal{P}_e\backslash\{0\}$ and a sequence Γ of top-reductions which satisfies the conditions $G_{1,2,3})$ of theorem 2.4.1 is called the **sequence of $\mathcal{B}$-free reduction modulo F**.

The following lemma is a direct counterpart of corollary 2.3.8 for the case of $\mathcal{B}$-free reduction sequence.

Lemma 2.4.9. If $f{\in}\mathcal{P}_e\backslash\{0\}$, $F{\subseteq}\mathcal{P}_e\backslash\{0\}$ and $\{g_o,\ldots,g_i,\ldots\}$ is a $\mathcal{B}$-free reduction sequence of f modulo F then $\ell\mathrm{t}(g_k)\underset{\mathbb{T}_e}{\leq}\ell\mathrm{t}(f)$ for each $k{\in}\mathbb{N}_o$ such that $g_k{\neq}0$.

Proof. By inductive argument based on lemma 2.3.13. □

The fact that the sequence Γ of top-reductions in theorem 2.4.1 is necessary finite for any $f{\in}\mathcal{P}_e$ and $F{\subseteq}\mathcal{P}_e\backslash\{0\}$ brings the idea to attain a top-minimal modulo F element as the last element in a maximal sequence (i.e. the sequence that can not be extended further) starting from f and eventually derive the notion of a normal form in the same manner as it is done for a usual top-reduction sequence. As will be shown below such a top-minimal element is not necessary a normal form but has properties very similar to it.

Definition 2.4.4. Let $f \in \mathcal{P}_e$ and $F \subseteq \mathcal{P}_e$. A polynomial $g \in \mathcal{P}_e$ is called a **quasi-normal form of f modulo F** if and only if there exists $u \in \mathrm{U}(\mathbb{k}[e])_{\mathcal{P}_e}$ such that $\boxed{g \text{ is a normal form of } u \cdot f}$. The set of quasi-normal forms is written $\mathrm{QNF}_F(f)$.

The quasi-normal form is closely related and shares the essential properties with the usual normal form as it is shown in the following lemmas.

Lemma 2.4.10. Let $f \in \mathcal{P}_e$ and $F \subseteq \mathcal{P}_e \setminus \{0\}$. Then $\mathrm{NF}_F(f) \subseteq \mathrm{QNF}_F(f)$ and $\mathrm{QNF}_F(f) \neq \emptyset$.

Proof. It is clear that $1 \in \mathrm{U}(\mathbb{k}[e])_{\mathcal{P}_e}$ and therefore every normal form is a quasi-normal form i.e. $\mathrm{NF}_F(f) \subseteq \mathrm{QNF}_F(f)$. Other other hand, by lemma 2.3.18, $\mathrm{NF}_F(f) \neq \emptyset$. □

Lemma 2.4.11. Let $f \in \mathcal{P}_e$ and $F \subseteq \mathcal{P}_e \setminus \{0\}$. Then $\mathrm{NF}_F(f) = \{0\}$ if and only if $\mathrm{QNF}_F(f) = \{0\}$.

Proof.

⇒ Let $\mathrm{NF}_F(f) = \{0\}$. By lemma 2.4.10 there exists $r \in \mathrm{QNF}_F(f)$. Assume $r \neq 0$. Then, by definition 2.4.4 of a quasi-normal form, there exists $u \in \mathrm{U}(\mathbb{k}[e])_{\mathcal{P}_e}$ such that

$$uf - r \in \langle F \rangle_{\mathcal{P}_e} \tag{1}$$

and r is top-minimal modulo F which is, by the assumption $r \neq 0$ and lemma 2.3.15, equivalent to the condition

$$\ell\mathrm{t}(r) \notin \langle \ell\mathrm{t}(F) \rangle_{\mathcal{P}_e}. \tag{2}$$

It is well-known (see e.g. [BW93] lemma 1.18 on page 21) that

an invertible element can not be a zero divisor in any non-trivial ring. $(*)$

Therefore $u^{-1} r \neq 0$ and, by (1),

$$u^{-1} u f - u^{-1} r = f - u^{-1} r \in \langle F \rangle_{\mathcal{P}_e}. \tag{3}$$

Furthermore, by part c) of corollary 2.4.1,

$$\ell\mathrm{t}\big(u^{-1} r\big) = \ell\mathrm{t}(r) \overset{(2)}{\notin} \big\langle \ell\mathrm{t}(F) \big\rangle_{\mathcal{P}_e} \tag{4}$$

but then, by (3) and (4), $u^{-1} r$ is a normal form of f modulo F according to the definition 2.3.15 of the normal form. Therefore $u^{-1} r = 0$ and hence, according to $(*)$, $r = 0$.

⇐ Assume $\mathrm{QNF}_F(f) = \{0\}$. By lemma 2.4.10, $\mathrm{NF}_F(f) \subseteq \mathrm{QNF}_F(f)$. On the other hand, by lemma 2.3.18, $\mathrm{NF}_F(f) \neq \emptyset$. Hence $\mathrm{NF}_F(f) = \{0\}$. □

Now we can obtain one more characterization of Gröbner bases.

Corollary 2.4.2. Let $I \lhd \mathcal{P}_e$ be an ideal and $G \subseteq I \setminus \{0\}$. Then the following conditions are equivalent.

- G is a Gröbner basis of I.
- For any $f \in \mathcal{P}_e$, $f \in I$ if and only if $\mathrm{QNF}_G(f) = \{0\}$.

Proof. By part $d)$ of theorem 2.3.2 and lemma 2.4.11, G is a Gröbner basis of I if and only if for any $f \in \mathcal{P}_e$

$$f \in I \quad \Longleftrightarrow \quad \mathrm{NF}_G(f) = \{0\} \quad \Longleftrightarrow \quad \mathrm{QNF}_G(f) = \{0\}. \qquad \square$$

The following lemma (cf. lemma 11 in [MT]) and the corollary show that a top-minimal element attained as the last element of a $\mathcal{B}$-free reduction sequence is indeed a quasi-normal form.

Lemma 2.4.12. Let $f \in \mathcal{P}_e$, $F \subseteq \mathcal{P}_e \setminus \{0\}$ and $\Gamma = \{g_o, \ldots, g_i, \ldots, g_N\}$ is a (necessary finite by theorem 2.4.1) $\mathcal{B}$-free reduction sequence. Then there exists $u \in \mathrm{U}(\mathcal{P}_e)\Bbbk[e]$ and polynomials $\{q_h |\, h \in F\} \subseteq \mathcal{P}_e$ such that

$$uf - g_N = \sum_{h \in F} q_h h$$

is a Gröbner representation of $uf - g_N$ in terms of F.

Proof. Exclude the trivial case $f=0$. Denote $F_o = F$ and $F_k = F_{k-1} \cup \{g_{k-1}\}$ for $0<k<N$. By the condition (G_2) of theorem 2.4.1

$$g_k \xrightarrow{F_k} g_{k+1} \quad \text{for all} \quad 0 \leq k < N \tag{1}$$

and hence, by definition 2.3.12 of one step top-reduction $g_k \neq 0$ for all $0 \leq k < N$. We show by induction that for each index $0 \leq k < N$ there exists $u_k \in \mathrm{U}(\Bbbk[e])_{\mathcal{P}_e}$ and polynomials $\{q_{h,k} |\, h \in F_k\}$ such that

$$u_k g_k - g_N = \sum_{h \in F_k} q_{h,k} h, \tag{$*$}$$

$$0 \neq \ell\mathrm{t}(q_{h,k}) \ell\mathrm{t}(h) \underset{\mathbb{T}_e}{\leq} \ell\mathrm{t}(g_k) \text{ for all } h \in F_k \text{ such that } q_{h,k} \neq 0. \tag{$**$}$$

Let $k=N-1$ Then, by (1) and the definition of one step top-reduction there exists $h_{N-1} \in F_{N-1}$ such that

$$g_N = g_{N-1} - \alpha \cdot s \cdot h_{N-1}, \quad \text{with} \quad \alpha = \frac{\ell\mathrm{c}(g_{N-1})}{\ell\mathrm{c}(h_{N-1})} \in \Bbbk^* \quad \text{and} \quad s = \frac{\ell\mathrm{t}(g_{N-1})}{\ell\mathrm{t}(h_{N-1})} \in {\mathbb{T}_e}^*.$$

Denote $u_{N-1}{=}1$, $q_{h_{N-1},N-1}{=}\alpha{\cdot}s$. By lemma 2.3.9, $\alpha{\cdot}s{\neq}0$ and $\ell\mathrm{t}(\alpha{\cdot}s){=}s$. Then

$$u_{N-1}g_{N-1} - g_N = q_{h_{N-1},N-1}{\cdot}h_{N-1},$$
$$\ell\mathrm{t}\big(q_{h_{N-1},N-1}\big){\cdot}\ell\mathrm{t}(h_{N-1}) = s{\cdot}\ell\mathrm{t}(h_{N-1}) = \frac{\ell\mathrm{t}(g_{N-1})}{\ell\mathrm{t}(h_{N-1})}{\cdot}\ell\mathrm{t}(h_{N-1}) = \ell\mathrm{t}(g_{N-1})$$

and hence $(*)$ is satisfied.

Assume there exist $u_{k+1}{\in}\mathrm{U}(\Bbbk[e])_{\mathcal{P}_e}$ and polynomials $\{q_{h,k+1} |\, h{\in}F_{k+1}\}$ such that

$$u_{k+1}g_{k+1} - g_N = \sum_{h\in F_{k+1}} q_{h,k+1}h, \tag{2}$$

$$0{\neq}\ell\mathrm{t}(q_{h,k+1})\ell\mathrm{t}(h)\underset{\mathbb{T}_e}{\leq}\ell\mathrm{t}(g_{k+1}) \text{ for all } h{\in}F_{k+1} \text{ such that } q_{h,k+1}{\neq}0 \tag{3}$$

and transform it to the desired representation $(*)$ for k. By construction, $F_{k+1}{=}F_k{\cup}\{g_k\}$. Then

$$\sum_{h\in F_{k+1}} q_{h,k+1}{\cdot}h = \sum_{h\in F_k} q_{h,k+1}{\cdot}h + q_{g_k,k+1}{\cdot}g_k. \tag{4}$$

By (1) and the definition of one step top-reduction there exists $h_k{\in}F_k$ such that

$$g_{k+1} = g_k - \alpha{\cdot}s{\cdot}h_k, \quad \text{with} \quad \alpha = \frac{\ell\mathrm{c}(g_k)}{\ell\mathrm{c}(h_k)} \in \Bbbk^* \quad \text{and } s = \frac{\ell\mathrm{t}(g_k)}{\ell\mathrm{t}(h_k)} \in {\mathbb{T}_e}^*. \tag{5}$$

From (2), (4) and the latter equality,

$$u_{k+1}(g_k - \alpha{\cdot}s{\cdot}h_k) - g_N = \sum_{h\in F_k} q_{h,k+1}{\cdot}h + q_{g_k}{\cdot}g_k$$

or, equivalently,

$$\begin{aligned}(u_{k+1} - q_{g_k}){\cdot}g_k - g_N &= \sum_{h\in F_k} q_{h,k+1}{\cdot}h + \alpha{\cdot}u_{k+1}{\cdot}s{\cdot}h_k \\ &= \sum_{h\in F_k\setminus\{h_k\}} q_{h,k+1}{\cdot}h + (q_{h_k,k+1} + \alpha{\cdot}u_{k+1}{\cdot}s){\cdot}h_k.\end{aligned} \tag{6}$$

Denote

$$\begin{aligned}u_k &= u_{k+1} - q_{g_k},\\ q_{h,k} &= \begin{cases} q_{h,k+1}, & h{\in}F_k\setminus\{h_k\} \\ q_{h_k,k+1} + \alpha{\cdot}u_{k+1}{\cdot}s, & h = h_k.\end{cases}\end{aligned}$$

Then (6) turns into

$$u_k{\cdot}g_k - g_N = \sum_{h\in F_k} q_{h,k}{\cdot}h. \tag{7}$$

We have to show that $u_k{\in}\mathrm{U}(\Bbbk[e])_{\mathcal{P}_e}$ and $q_{h,k}$ meets the condition $(**)$ for each $h{\in}F_k$.

1. Clearly $u_k{\in}\mathrm{U}(\Bbbk[e])_{\mathcal{P}_e}$ if $q_{g_k,k+1}{=}0$. Otherwise, note that, by (1) and lemma 2.3.13,

$$\ell\mathrm{t}(g_{k+1}) \underset{\mathbb{T}_e}{<} \ell\mathrm{t}(g_k). \tag{8}$$

By (3) and (8),

$$0 \neq \ell\mathrm{t}(q_{g_k,k+1}){\cdot}\ell\mathrm{t}(g_k) \underset{\mathbb{T}_e}{\leq} \ell\mathrm{t}(g_{k+1}) \overset{(8)}{\underset{\mathbb{T}_e}{<}} \ell\mathrm{t}(g_k)$$

and hence, by lemma 2.3.8, $\ell\mathrm{t}(q_{g_k,k+1}) \underset{\mathbb{T}_e}{<} 1$. On the other hand, by part *b*) of lemma 2.4.5, $\ell\mathrm{t}(u_{k+1}){=}1$. Then, by part *c*) of lemma 2.3.10, $u_k{\neq}0$ and

$$\ell\mathrm{t}(u_k) = \max\Big(\ell\mathrm{t}(q_{g_k}), \ell\mathrm{t}(u_{k+1})\Big) = 1.$$

Therefore, using part *b*) of lemma 2.4.5 once again, $u_k{\in}\mathrm{U}(\Bbbk[e])_{\mathcal{P}_e}$.

2. Take $h{\in}F_k{\setminus}\{h_k\}$ such that $q_{h,k}{\neq}0$. Then, by (3),

$$\ell\mathrm{t}(q_{h,k})\ell\mathrm{t}(h) = \underbrace{\ell\mathrm{t}(q_{h,k+1})\ell\mathrm{t}(h) \underset{\mathbb{T}_e}{\leq} \ell\mathrm{t}(g_{k+1})}_{\overset{(3)}{\nparallel}\ 0} \overset{(3)}{\underset{\mathbb{T}_e}{\leq}} \ell\mathrm{t}(g_{k+1}) \overset{(8)}{\underset{\mathbb{T}_e}{<}} \ell\mathrm{t}(g_k)$$

3. Assume $h_k{\neq}0$. We claim that $q_{h_k,k}{\cdot}h_k{\neq}0$ and $\ell\mathrm{t}(q_{h_k,k}){\cdot}\ell\mathrm{t}(h_k){=}\ell\mathrm{t}(g_k)$. Recall that

$$q_{h_k,k} = q_{h_k,k+1} + \alpha{\cdot}u_{k+1}{\cdot}s \ \text{ with } \ \alpha = \frac{\ell\mathrm{c}(g_k)}{\ell\mathrm{c}(h_k)} \in \Bbbk^*, \quad s = \frac{\ell\mathrm{t}(g_k)}{\ell\mathrm{t}(h_k)} \in {\mathbb{T}_e}^*.$$

By lemma 2.3.9 and part *b*) of lemma 2.4.5, $\alpha{\cdot}u_{k+1}{\in}\mathrm{U}(\Bbbk[e])_{\mathcal{P}_e}$. Then, by part *c*) of lemma 2.4.5, $\alpha u_{k+1}{\cdot}s{\neq}0$ and $\ell\mathrm{t}(\alpha u_{k+1}{\cdot}s){=}s$. By construction,

$$s{\cdot}\ell\mathrm{t}(h_k) = \ell\mathrm{t}(g_k). \tag{9}$$

If $q_{h_k,k+1}{=}0$ then we are done. Otherwise, by (3) and (8),

$$0 \overset{(3)}{\neq} \ell\mathrm{t}(q_{h_k,k+1})\ell\mathrm{t}(h_k) \overset{(3)}{\underset{\mathbb{T}_e}{\leq}} \ell\mathrm{t}(g_{k+1}) \overset{(8)}{\underset{\mathbb{T}_e}{<}} \ell\mathrm{t}(g_k) \tag{10}$$

and then, by part c) of lemma 2.3.10,

$$q_{h_k,k} \neq 0 \;\text{ and }\; \ell\mathrm{t}(q_{h_k,k}) = \max\Big(\ell\mathrm{t}(q_{h_k,k+1}), \ell\mathrm{t}(\alpha{\cdot}u_{k+1}{\cdot}s)\Big) \overset{(9),(10)}{=} \ell\mathrm{t}(g_k).$$

According to 1), 2) and 3) the representation (7) meets the conditions $(*)$ and $(**)$. To finish the proof note that (7) with $k{=}0$ is exactly the statement of the lemma provided that $F_o{=}F$ by construction. □

Corollary 2.4.3. For $f{\in}\mathcal{P}_e$ and $F{\subseteq}\mathcal{P}_e\backslash\{0\}$ there exists a polynomial g such that g is a quasi-normal form of f modulo F and which can be attained from a $\mathcal{B}$-free reduction sequence. Furthermore, there exists a Gröbner representation of f in terms of $F{\cup}\{g\}\backslash\{0\}$.

Proof. Let $\Gamma{=}\{g_o, \ldots, g_i, \ldots\}{\subseteq}\mathcal{P}$ be a maximal (in the sense that it can not be continued) $\mathcal{B}$-free reduction sequence of f modulo F. By theorem 2.4.1, Γ is finite. Let $g_N{\in}\mathcal{P}$ be the last element of Γ. Then g_N is top-minimal modulo F. Indeed, if g_N is top-reducible modulo F then it is top-reducible modulo $F_N{=}F{\cup}\{g_o, \ldots, g_{N-1}\}$ and hence the next element $g_{N+1}{\in}\mathcal{P}$ such that $g_N \xrightarrow{F_N} g_{N+1}$ can be added to the sequence which contradicts the choice of Γ as maximal. Denote $g{=}g_N$. By lemma 2.4.12,

By lemma 2.4.12, there exists $u{\in}\mathrm{U}(\Bbbk[e])_{\mathcal{P}_e}$ and polynomials $\{q_h |\, h{\in}F\}$ such that

$$uf - g = \sum_{h\in F} q_h h \tag{1}$$

is a Gröbner representation of $uf{-}g$ in terms of F. In particular, $uf{-}g{\in}\langle F\rangle_{\mathcal{P}_e}$ and thus g is a quasi-normal form of f modulo F according to definition 2.4.4. Furthermore, by construction, g is obtained from the $\mathcal{B}$-free reduction sequence Γ.

Write the equation (1) as

$$f = \sum_{h\in F} \frac{q_h}{u}{\cdot}h + u^{-1}{\cdot}g. \tag{2}$$

Note that, by parts b) and c) of lemma 2.4.5

$$\ell\mathrm{t}(u) = 1,\; u{\cdot}p \neq 0 \;\text{ and }\; \ell\mathrm{t}(u{\cdot}p) = \ell\mathrm{t}(p) \;\text{ for any }\; p \in \mathcal{P}_e\backslash\{0\}. \tag{3}$$

Denote $F'{=}F{\cup}\{g\}\backslash\{0\}$. Our goal is to show that (2) is a Gröbner representation of f in terms of F'. Exclude two trivial cases

1) $f{=}0$.
2) $f{\neq}0,\; g{\neq}0$ and $\ell\mathrm{t}(f){=}\ell\mathrm{t}(g)$.

It is easy to see by construction of Γ that in both these cases $N{=}0$, $f{=}g$, $u{=}1$ and $q_h{=}0$ for all $h{\in}F$ and hence (2) is a Gröbner representation of f in terms of F'.

Assume $f{\neq}0$ and either $g{=}0$ or $\ell\mathrm{t}(f){\neq}\ell\mathrm{t}(g)$. Then

$$u{\cdot}f{-}g \neq 0 \quad\text{and}\quad \ell\mathrm{t}(u{\cdot}f{-}g) = \ell\mathrm{t}(f). \tag{4}$$

Indeed, by (3), $u{\cdot}f{\neq}0$ and $\ell\mathrm{t}(u{\cdot}f){=}\ell\mathrm{t}(f)$. Furthermore, by lemma 2.4.9,

$$g{=}0 \quad\text{or}\quad \ell\mathrm{t}(g)\underset{\mathbb{T}_e}{\leq}\ell\mathrm{t}(f). \tag{5}$$

If $g{=}0$ then (4) is clear. Otherwise, by lemma 2.3.10,

$$\ell\mathrm{t}(u{\cdot}f{-}g) = \max\Big(\ell\mathrm{t}(u{\cdot}f), \ell\mathrm{t}(g)\Big) \overset{(3)}{=} \max\Big(\ell\mathrm{t}(f), \ell\mathrm{t}(g)\Big) \overset{(5)}{=} \ell\mathrm{t}(f).$$

Take $h{\in}F$ such that $q_h{\neq}0$. By lemma 2.4.12, (1) is a Gröbner representation of $u{\cdot}f{-}g$ in terms of F and therefore

$$0 \neq \ell\mathrm{t}(q_h){\cdot}\ell\mathrm{t}(h)\underset{\mathbb{T}_e}{\leq}\ell\mathrm{t}(u{\cdot}f{-}g) \overset{(4)}{=} \ell\mathrm{t}(f). \tag{6}$$

By (3),

$$0 \overset{(3)}{\neq} \ell\mathrm{t}\Big(\frac{q_h}{u}\Big){\cdot}\ell\mathrm{t}(h) \overset{(3)}{=} \ell\mathrm{t}(q_h){\cdot}\ell\mathrm{t}(h) \overset{(6)}{=} \ell\mathrm{t}(f). \tag{7}$$

If $g{\neq}0$ then, again by (3),

$$u^{-1}{\cdot}g \overset{(3)}{\neq} 0 \quad\text{and}\quad \ell\mathrm{t}\big(u^{-1}\big){\cdot}\ell\mathrm{t}(g) \overset{(3)}{=} \ell\mathrm{t}(g) \underset{\mathbb{T}_e}{\overset{(5)}{\leq}} \ell\mathrm{t}(f). \tag{8}$$

By (7) and (8), (2) is a Gröbner representation of f in terms of F'. $\square$

In view of corollary 2.4.3, given any $f{\in}\mathcal{P}_e$ and $F{\subseteq}\mathcal{P}_e{\setminus}\{0\}$, the following algorithm computes a quasi-normal form of f modulo F.

Lemma 2.4.13. For any $f{\in}\mathcal{P}_e$ and $F{\subseteq}\mathcal{P}_e$ the algorithm `QNF` computes a quasi-normal form of f modulo F in finitely many steps.

Proof. It is easy to see according to the layout of the algorithm `QNF` that the sequence of values which the polynomial g takes during the execution of the while-loop is a $\mathcal{B}{-}$free reduction sequence modulo F and hence the termination of the algorithm is given by lemma 2.4.1. Then the final value of g after the execution of the while-loop is a quasi-normal form of f modulo F according to corollary 2.4.3. $\square$

Algorithm 2.4.1 The $\mathcal{B}$-free reduction algorithm `QNF` in $\mathcal{P}_e$

Require: $f\in\mathcal{P}_e$, $F\subseteq\mathcal{P}_e\backslash\{0\}$.
Ensure: $g\in\mathcal{P}_e$ is a quasi-normal form of f modulo F.

$g := f$
while $g\neq 0$ and $\ell\mathrm{t}(g)\in\langle\ell\mathrm{t}(F)\rangle_{\mathcal{P}_e}$ **do**
  choose $h\in F$ such that $\ell\mathrm{t}(h)|\ell\mathrm{t}(g)$ and $\ell\Delta_\varepsilon(h)\leq\ell\Delta_\varepsilon(h')$ for any $h'\in F$ such that $\ell\mathrm{t}(h')|\ell\mathrm{t}(g)$
  $F := F\cup\{g\}$
  $g := g - \frac{\ell\mathrm{m}(g)}{\ell\mathrm{m}(h)}\cdot h$
end while

2.4.2 The completion procedure in $\mathcal{P}_e$

The algorithm for computing extended Gröbner bases in the algebra $\mathcal{P}_e$ is the algorithm `Gröbner` on page 79 with the normal form computation procedure `NF` replaced by the quasi-normal form algorithm `QNF`. The correctness proof of this modified algorithm repeats the respective correctness proof of the algorithm `Gröbner` in lemma 2.3.36 with the conditions (NF_1) and (NF_2) of definition 2.3.15 of a normal form replaced by the statements (QNF_1) and (QNF_2) of lemma 2.4.14.

Lemma 2.4.14. Let $F\subseteq\mathcal{P}_e\backslash\{0\}$ and $f\in\langle F\rangle_{\mathcal{P}_e}$. Then, for any $r\in\mathrm{QNF}_F(f)$

$$f{-}r \in \langle F\rangle_{\mathcal{P}_e}, \qquad (QNF_1)$$
$$r \text{ is top-minimal modulo } F. \qquad (QNF_2)$$

Proof. By definition 2.4.4 of a quasi-normal form there exists $u\in\mathrm{U}(\Bbbk[e])_{\mathcal{P}_e}$ such that r is a normal form of $u\cdot f$. Then, by definition 2.3.15 of a normal form,

$$u\cdot f{-}r \in \langle F\rangle_{\mathcal{P}_e}, \qquad (NF_1)$$
$$r \text{ is top-minimal modulo F}. \qquad (NF_2)$$

The claim (QNF_2) is now clear in view of the condition (NF_2). On the other hand, $f\in\langle F\rangle_{\mathcal{P}_e}$ is equivalent to[7] $u\cdot f\in\langle F\rangle_{\mathcal{P}_e}$ by the definition of an ideal provided that u is a unit in $\mathcal{P}_e$ and hence $r\in\langle F\rangle_{\mathcal{P}_e}$ by (NF_1) which follows the statement $f{-}r\in\langle F\rangle_{\mathcal{P}_e}$ in the claim (QNF_1). □

[7]The condition $f\in\langle F\rangle_{\mathcal{P}_e}$ is important for this equivalence. It is easy to see that $f{-}uf$ with $u\in\mathrm{U}(\mathcal{P}_e)$ is not generally in $\langle F\rangle_{\mathcal{P}_e}$ if $f\in\mathcal{P}_e\backslash\langle F\rangle_{\mathcal{P}_e}$ (e.g. take $F{=}\{x\}$, $f{=}1$ and $u{=}2$).

2.5 Gröbner bases in the ring $\mathcal{P}_{e[X]}$

This section presents the basic theory of Gröbner bases for the ring $\mathcal{P}_{e[X]}$ - the ring of polynomials over the (primary ideal) ring $\mathbb{k}[e]$, where $\mathbb{k}$ is a field, in finitely many algebraically independent over $\mathbb{k}[e]$ indeterminates X. Our development does not involve any Gröbner bases like computation in $\mathcal{P}_{e[X]}$ but the interpretation of the result of a Gröbner basis computation in the algebra $\mathcal{P}_e$ under specialization of the extended variable e uses the properties of Gröbner bases in $\mathcal{P}_{e[X]}$ as a theoretical tool.

We start with a brief introduction to a more general theory of Gröbner bases for polynomials over an arbitrary commutative Nötherian ring. The presented facts are by and large known but scattered in the literature in somehow or other form (see e.g. [AL94] chapter 4, [Möl88], [TV78], [Zac78]). Then the general theory is specialized to the particular case of Gröbner bases for the polynomial ring over a principal ideal ring $\mathbb{k}[e]$ as a generalization of the respective all in all developed in the literature theory of Gröbner bases for the polynomials over a primary ideal domain (see [Möl88], [Pan89]).

2.5.1 Gröbner bases in a polynomial algebra over a ring

Let $R\,[X]$ be the algebra of polynomials over a commutative Nötherian ring R in finitely many indeterminates (variables) $X{=}\{x_1, \ldots, x_n\}$ algebraically independent over R.

Notation.

$R^* = R\backslash\{0\}$,

$R\,[X] \supsetneq \mathbb{T}$ – the terms (power products) in X,

$R\,[X] \supsetneq \mathbb{M} = \{\alpha{\cdot}t \mid \alpha{\in}R^*, t{\in}\mathbb{T}\}$ – the monomials in $R\,[X]$.

The elements of R also considered as the elements of $R\,[X]$ due to the canonical embedding $R{\subsetneq}R\,[X]$ are called **coefficients**.

Since indeterminates X are algebraically independent over R, each term in $\mathbb{T}$ has the unique (up to a permutation due to the commutativity) representation as a product of indeterminates and the terms $\mathbb{T}$ are linear independent over R in the sense that for any finite subset $T_o{\subseteq}\mathbb{T}$ and coefficients $\alpha_t{\in}R$, one for each $t{\in}\mathbb{T}_o$,

$$\sum_{t\in T_o} \alpha_t{\cdot}t = 0 \quad \text{if and only if} \quad \alpha_t{=}0 \text{ for all } t{\in}T_o.$$

A polynomial in $R[X]$ is often identified with the sum of monomials provided that such representation always exists and unique or with a linear combination of terms with coefficients from R^* provided that for each monomial there exists the unique representation as the product of the non-zero coefficient with the term. Let us formulate the latter facts more precisely also to fix the notation.

- Given an arbitrary term $t\in\mathbb{T}$ there exists the unique vector of non-negative integers $(d_1,\ldots,d_n)\in\mathbb{N}_o^{\times n}$ such that

$$t = x_1^{d_1}\cdots x_n^{d_n}.$$

  For each $1\leq i\leq n$ the value d_i denoted $\deg_{x_i}(t)$ is called **the degree of t in x_i**. The sum $d_1+\cdots+d_n$ denoted $\deg(t)$ is called **the (total) degree of t**.

- Given an arbitrary monomial $m\in\mathbb{M}$, there exists the unique representation of m as the product

$$m = \alpha{\cdot}t \quad\text{with}\quad \alpha\in R^*,\ t\in T$$

  denoted

  - $\mathfrak{C}(m)=\alpha$ - the coefficient of the monomial m,
  - $\mathfrak{T}(m)=t$ - the term of the monomial m.

- Given an arbitrary polynomial $f\in R[X]$, there exists the unique representation of f as the sum of monomials

$$f = \sum_{t\in\mathbb{T}_f} f_t \quad\text{with}\quad \begin{array}{l} T_f\subseteq\mathbb{T} - \text{a finite subset of } \mathbb{T}, \\ f_t\in\mathbb{M} \ \text{ for each }\ t\in\mathbb{T}_f. \end{array}$$

  - The well-defined set of terms T_f is called the **support** of f denoted $\operatorname{supp}(f)$ and sometimes $\mathfrak{T}(f)$ – the terms of f.
  - The monomials and the coefficients of monomials in the latter unique representation are denoted
    * $\mathfrak{M}(f) = \{f_t \mid t\in\operatorname{supp}(f)\}$ - the monomials of f,
    * $\mathfrak{C}(f) = \{\mathfrak{C}(f_t) \mid t\in\operatorname{supp}(f)\}$ - the coefficients of f.
  - The maximal degree among the terms of $f\neq0$ denoted $\deg(f)$ is called **the total degree of f**:

$$\deg(f) = \max_{t\in\operatorname{supp}(f)}\Big(\deg(t)\Big).$$

– To access the separate monomials and coefficients of f define

  * the monomial at a term t as

$$\mathfrak{M}_f(t) = \begin{cases} f_t & t \in \operatorname{supp}(f), \\ \text{undefined} & \text{otherwise}, \end{cases}$$

  * the coefficient at a term t as

$$\mathfrak{C}_f(t) = \begin{cases} \mathfrak{C}(f_t) & t \in \operatorname{supp}(f), \\ \text{undefined} & \text{otherwise}. \end{cases}$$

– To ease the manipulation with formal sums in $R[X]$ define

  * the formal monomial of f at t as

$$[\mathfrak{M}_f](t) = \begin{cases} \mathfrak{M}_f(t) & t \in \operatorname{supp}(f), \\ 0 & \text{otherwise}, \end{cases}$$

  * the formal coefficient of f at t as

$$[\mathfrak{C}_f](t) = \begin{cases} \mathfrak{C}_f(t) & t \in \operatorname{supp}(f), \\ 0 & \text{otherwise}. \end{cases}$$

The properties of the support

The presented properties of polynomials $R[X]$ under the ring operations are described in terms of set theoretic operations on the support of individual polynomials. The only exception must be done for the support of the product of polynomials which requires the following operation on the sets of terms.

Definition 2.5.1. Let $T_1, T_2 \subseteq \mathbb{T}$. The **free product** of T_1 and T_2 denoted by $T_1 * T_2$ is

$$T_1 * T_2 = \{v \cdot w \mid v \in T_1, w \in T_2\}.$$

Lemma 2.5.1. Let $T_1, T_2 \subseteq \mathbb{T}$. Then $T_1 * T_2 = \emptyset$ if and only if $T_1 = \emptyset$ or $T_2 = \emptyset$.

Proof. Let $T_1 \neq \emptyset$ and $T_2 \neq \emptyset$. Take $t_1 \in T_1$ and $t_2 \in T_2$. Then, according to definition 2.5.1 of the free product, $t_1 t_2 \in T_1 * T_2$ and hence $T_1 * T_2 \neq \emptyset$. On the other hand, if $T_1 * T_2 \neq \emptyset$. Take $t \in T_1 * T_2$. Then, by definition 2.5.1 of the free product, there exist $t_1 \in T_1$ and $t_2 \in T_2$ such that $t = t_1 t_2$. In particular, $T_1 \neq \emptyset$ and $T_2 \neq \emptyset$. ∎

Note that the "only if" direction of the statement of lemma 2.5.1 is not valid for the free product on the sets of extended terms $\mathbb{T}(e)$ in the sense of definition 2.3.5. As an example consider the algebra of extended polynomials $\mathcal{P}(e)$ with the e-bound $\mathcal{B}$=2. Take $\mathbb{T}(e)\supseteq S_1=S_2=\{e\}$. Then $S_1=S_2\neq\emptyset$ but $S_1{*}S_2=\{e{\cdot}e\}\setminus\{0\}=\{0\}\setminus\{0\}=\emptyset$.

The next lemma 2.5.2 summarizes the basic properties of the support for polynomials in $R\,[X]$. Note a subtle but considerable difference from the properties of the support for the extended polynomials $\mathcal{P}(e)$ in lemma 2.3.3 due to the fact that the terms $\mathbb{T}$ of polynomials $R\,[X]$ considered in this section is a free commutative monoid while the ring R can have zero-divisors. The difference can be summarized in the following table with the unified context from both lemmas and the $|\cdot|$ notation for the number of elements in a finite set.

	Lemma 2.3.3 for $\mathcal{P}(e)$	Lemma 2.5.2 for $R\,[X]$
$a)$	$\operatorname{supp}(\alpha{\cdot}f) = \operatorname{supp}(f)$	$\operatorname{supp}(\alpha{\cdot}f) \subseteq \operatorname{supp}(f)$
$b)$	no difference	
$c)$	the free product definition is different	
$d)$	$\lvert\operatorname{supp}(v{\cdot}g)\rvert \leq \lvert v{*}\operatorname{supp}(g)\rvert$	$\lvert\operatorname{supp}(v{\cdot}g)\rvert = \lvert v{*}\operatorname{supp}(g)\rvert$
$e)$	$\operatorname{supp}(\alpha{\cdot}v{\cdot}g) = v{*}\operatorname{supp}(g)$	$\operatorname{supp}(\alpha{\cdot}v{\cdot}g) \subseteq v{*}\operatorname{supp}(g)$

Lemma 2.5.2. Let $\alpha{\in}R^*$, $v{\in}\mathbb{T}$ and $f, g \in R\,[X]$. Then

$a)$ $\operatorname{supp}(\alpha{\cdot}f) \subseteq \operatorname{supp}(f)$,
$b)$ $\operatorname{supp}(f) \triangle \operatorname{supp}(g) \subseteq \operatorname{supp}(f{+}g) \subseteq \operatorname{supp}(f) \cup \operatorname{supp}(g)$,
$c)$ $\operatorname{supp}(fg) \subseteq \operatorname{supp}(f) * \operatorname{supp}(g)$,
$d)$ $\operatorname{supp}(v{\cdot}g) = v{*}\operatorname{supp}(g)$,
$e)$ $\operatorname{supp}(\alpha{\cdot}v{\cdot}g) \subseteq v{*}\operatorname{supp}(g)$.

Proof. All non-trivial parts follow the same pattern of argumentation based on the uniqueness of the support for each polynomial in $R\,[X]$. The uniqueness is used in the sense that given a representation of a polynomial as a linear combination of finitely many terms with non-zero coefficients this set of terms is the uniquely defined support which, in particular, does not depend on the way how the representation has been attained. The desired representation is achieved by a straightforward polynomial expressions manipulation provided that $R[X]$ is a ring.

Let $\{\alpha_t | t \in \operatorname{supp}(f)\}, \{\beta_t | t \in \operatorname{supp}(g)\} \subseteq R^*$ be the (uniquely defined) coefficients such that

$$f = \sum_{t \in \operatorname{supp}(f)} \alpha_t \cdot t, \qquad g = \sum_{t \in \operatorname{supp}(g)} \beta_t \cdot t.$$

a) Denote

$$N = \left\{ t \in \operatorname{supp}(f) \mid \alpha \cdot \alpha_t \neq 0 \right\} \subseteq \operatorname{supp}(f) \subseteq \mathbb{T}.$$

Then

$$\alpha \cdot f = \alpha \cdot \left(\sum_{t \in \operatorname{supp}(f)} \alpha_t \cdot t \right) = \sum_{t \in \operatorname{supp}(f)} (\alpha \alpha_t) \cdot t = \sum_{t \in N} (\alpha \alpha_t) \cdot t$$

where the coefficient $\alpha \cdot \alpha_t \neq 0$ for each $t \in N$ by construction. Hence $\operatorname{supp}(\alpha \cdot f) = N \subseteq \operatorname{supp}(f)$ by definition of the support.

b) For each $t \in \operatorname{supp}(f) \cup \operatorname{supp}(g)$ define

$$R \ni \gamma_t = \begin{cases} \alpha_t & \text{if } t \in \operatorname{supp}(f) \setminus \operatorname{supp}(g), \\ \alpha_t + \beta_t & \text{if } t \in \operatorname{supp}(f) \cap \operatorname{supp}(g), \\ \beta_t & \text{if } t \in \operatorname{supp}(g) \setminus \operatorname{supp}(f). \end{cases} \tag{$*$}$$

Let

$$N = \left\{ t \in \operatorname{supp}(f) \cup \operatorname{supp}(g) \;\middle|\; \gamma_t \neq 0 \right\}.$$

Then

$$f + g = \sum_{t \in \operatorname{supp}(f)} \alpha_t \cdot t + \sum_{t \in \operatorname{supp}(g)} \beta_t \cdot t = \sum_{t \in \operatorname{supp}(f) \cup \operatorname{supp}(g)} \gamma_t \cdot t = \sum_{t \in N} \gamma_t \cdot t$$

where $\gamma_t \neq 0$ for each $t \in N$ by construction of N and hence, by definition of the support, $\operatorname{supp}(f+g) = N \subseteq \operatorname{supp}(f) \cup \operatorname{supp}(g)$. On the other hand, $\operatorname{supp}(f) \triangle \operatorname{supp}(g) \subseteq N = \operatorname{supp}(f+g)$ because, by $(*)$, $\gamma_t \neq 0$ for $t \in \operatorname{supp}(f) \triangle \operatorname{supp}(g)$.

c) For each $t \in \operatorname{supp}(f) * \operatorname{supp}(g)$ define

$$R \ni \gamma_t = \sum_{\substack{t_1 \in \operatorname{supp}(f), t_2 \in \operatorname{supp}(g), \\ t_1 t_2 = t}} \alpha_{t_1} \beta_{t_2}.$$

Let
$$N = \Big\{ t \in \operatorname{supp}(f) * \operatorname{supp}(g) \;\Big|\; \gamma_t{\neq}0 \Big\}.$$
Then
$$fg = \Big(\sum_{t\in\operatorname{supp}(f)} \alpha_t{\cdot}t\Big)\Big(\sum_{t\in\operatorname{supp}(g)} \beta_t{\cdot}t\Big) = \sum_{t\in\operatorname{supp}(f)*\operatorname{supp}(g)} \gamma_t{\cdot}t \;\; = \sum_{t\in N} \gamma_s{\cdot}t$$
where $\gamma_t{\neq}0$ for each $t{\in}N$ by construction of N and hence, by definition of the support, $\operatorname{supp}(fg) = N \subseteq \operatorname{supp}(f) * \operatorname{supp}(g)$.

d) By
$$v{\cdot}g = v \cdot \left(\sum_{t\in\operatorname{supp}(g)} \beta_t{\cdot}t \right) = \sum_{t\in\,\operatorname{supp}(g)} \beta_t{\cdot}(vt)$$
and the definition of the free product on the sets of terms it follows that
$$\operatorname{supp}(v{\cdot}g) = \{vt \mid t{\in}\operatorname{supp}(g)\} = v{*}\operatorname{supp}(g)\,.$$

e) It is an easy combination of parts *a)* and *d)*. Indeed
$$\operatorname{supp}(\alpha{\cdot}v{\cdot}g) \overset{a)}{\subseteq} \operatorname{supp}(v{\cdot}g) \overset{d)}{=} v{*}\operatorname{supp}(g)\,.$$

□

The polynomial expressions attained in the proof of the support properties in the latter lemma result in the following rules to manipulate with (formal) coefficients at a term under the ring operations with polynomials in $R\,[X]$.

Corollary 2.5.1 (from the proof of lemma 2.5.2). Let $\alpha{\in}R^*$, $v{\in}\mathbb{T}$ and $f, g{\in}R\,[X]$. Then

a) $[\mathfrak{C}_{\alpha\cdot f}](t) = \alpha{\cdot}\, \mathfrak{C}_f(t)$ for each $t{\in}\operatorname{supp}(f)$,
$[\mathfrak{C}_{\alpha\cdot f}](t) = \alpha{\cdot}[\mathfrak{C}_f](t)$ for any $t{\in}\mathbb{T}$,

b) $[\mathfrak{C}_{f+g}](t) = [\mathfrak{C}_f](t) + [\mathfrak{C}_g](t)$ for each $t{\in}\operatorname{supp}(f) \cup \operatorname{supp}(g)$,

c) $[\mathfrak{C}_{fg}](t) = \sum\limits_{\substack{t_1\in\operatorname{supp}(f), t_2\in\operatorname{supp}(g),\\ t_1t_2=t}} \mathfrak{C}_f(t_1)\,\mathfrak{C}_g(t_2)$; for each $t{\in}\operatorname{supp}(f) * \operatorname{supp}(g)$,

d) $\mathfrak{C}_{vg}(vt) = \mathfrak{C}_g(t)$ for each $t{\in}\operatorname{supp}(g)$,
$[\mathfrak{C}_{vg}](vt) = [\mathfrak{C}_g](t)$ for any $t{\in}\mathbb{T}$,

e) $[\mathfrak{C}_{\alpha\cdot vg}](vt) = \alpha{\cdot}\,\mathfrak{C}_g(t)$ for each $t{\in}\operatorname{supp}(g)$,
$[\mathfrak{C}_{\alpha\cdot vg}](vt) = \alpha{\cdot}[\mathfrak{C}_g](t)$ for any $t{\in}\mathbb{T}$.

Proof. Each part is easy to check looking at the polynomial representation attained in the proof of the corresponding part of lemma 2.5.2. □

Note that according to parts a) and b) of corollary 2.5.1, the formal coefficient on $R\,[X]$ considered as a map

$$[\mathfrak{C}_{(.)}](v) : \begin{cases} \mathcal{P} \longrightarrow R{\cdot}1 \cong_R R \\ f \longrightarrow [\mathfrak{C}_f](v) \end{cases}$$

is a homomorphism of R-modules $R\,[X]_R$ and R_R.

The leading terms

Fix an arbitrary term order $<$ on $\mathbb{T}$. Recall that $<$ is a total order on terms such that $1{\le}t$ for all $t{\in}\mathbb{T}$ and (the monotonicity property) $t_1{<}t_2$ follows $t{\cdot}t_1{<}t{\cdot}t_2$. The monotonicity property has the inverse in the following sense.

Lemma 2.5.3 (monotonicity inverse). If $t, t_1, t_2{\in}\mathbb{T}$ are such that $t{\cdot}t_1{<}t{\cdot}t_2$ then $t_1{<}t_2$.

To make the proof of this simple fact even simpler let us point out a basic property of terms which has a value also in its own right.

Lemma 2.5.4 (the cancellation law of terms $\mathbb{T}$). If $t, t_1, t_2{\in}\mathbb{T}$ are such that $t{\cdot}t_1{=}t{\cdot}t_2$ then $t_1{=}t_2$.

Proof. Consider $t{\cdot}t_2{-}t{\cdot}t_1$ as a polynomial in $\Bbbk[X]$ where $\Bbbk$ is an arbitrary field. It is well-known that the polynomial ring $\Bbbk[X]$ does not have zero-divisors[8]. Then

$$t{\cdot}t_2{-}t{\cdot}t_1 = t{\cdot}(t_2{-}t_1) \;\text{ follows }\; t_2{-}t_1 = 0.$$ □

Proof of lemma 2.5.3. If $t_2{<}t_1$ then $t{\cdot}t_2{<}t{\cdot}t_1$ by the monotonicity of a term order. The equality case is excluded by the cancellation law. □

The **leading term**, the **leading coefficient** and **the leading monomial** are defined in the usual way on $\mathbb{T}$.

Definition 2.5.2. For any $f{\in}R\,[X]\backslash\{0\}$

- the **leading term of** f is the $<$-maximal term in the support of f:

$$\ell\mathrm{t}(f) = \max_{<}\Big(\mathrm{supp}\,(f)\Big);$$

[8] Here it is important that the cancellation law is the property of the terms $\mathbb{T}$ and does not depend on the polynomial ring where $\mathbb{T}$ is embedded.

- the **leading coefficient of** f is the coefficient at the leading term:

$$\ell\mathrm{c}(f) = \mathfrak{C}_f\big(\ell\mathrm{t}(f)\big)\ ;$$

- the **leading monomial of** f is the monomial at the leading term:

$$\ell\mathrm{m}(f) = \mathfrak{M}_f\big(\ell\mathrm{t}(f)\big) \ = \ \Big[\ell\mathrm{c}(f){\cdot}\ell\mathrm{t}(f)\Big].$$

Notation. For an arbitrary set $F{\subseteq}R\,[X]$

$$\begin{aligned}
\ell\mathrm{t}(F) &= \Big\{\ell\mathrm{t}(f) \mid f{\in}F{\setminus}\{0\}\Big\},\\
\ell\mathrm{c}(F) &= \Big\{\ell\mathrm{c}(f) \mid f{\in}F{\setminus}\{0\}\Big\},\\
\ell\mathrm{m}(F) &= \Big\{\ell\mathrm{m}(f) \mid f{\in}F{\setminus}\{0\}\Big\}.
\end{aligned}$$

The next lemma 2.5.5 summarizes the rules for computing the leading term function under the ring operations on polynomials in $R\,[X]$. The behavior of the leading term function in $R\,[X]$ is similar to the leading term in the algebra of extended polynomials $\mathcal{P}(e)$ but the stability conditions under multiplication with a term and a scalar are, in a sense, swapped. Certain significant discrepancies are listed in the following table.

	Lemma 2.5.5 for $R[X]$. $\alpha\in R^*$, $t\in\mathbb{T}$, $f,g\in R[X]\backslash\{0\}$.	The extended polynomials $\mathcal{P}(e)$. $\alpha\in\Bbbk^*$, $s\in\mathbb{T}(e)$, $f,g\in\mathcal{P}(e)\backslash\{0\}$.
a)	If$\alpha\cdot\ell\mathrm{c}(f)\neq 0$ then $\alpha\cdot f\neq 0$, $\ell\mathrm{t}(\alpha\cdot f)=\ell\mathrm{t}(f)$, $\ell\mathrm{c}(\alpha\cdot f)=\alpha\cdot\ell\mathrm{c}(f)$.	$\alpha\cdot f\neq 0$, $\ell\mathrm{t}(\alpha\cdot f)=\ell\mathrm{t}(f)$, (L. 2.3.9) $\ell\mathrm{c}(\alpha\cdot f)=\alpha\cdot\ell\mathrm{c}(f)$. (Cor. 2.3.4)
b)	$t\cdot f\neq 0$, $\ell\mathrm{t}(t\cdot f)=t\cdot\ell\mathrm{t}(f)$, $\ell\mathrm{c}(t\cdot f)=\ell\mathrm{c}(f)$.	If $s\cdot\ell\mathrm{t}(f)\neq 0$ then $s\cdot f\neq 0$, $\ell\mathrm{t}(s\cdot f)=s\cdot\ell\mathrm{t}(f)$, (Cor. 2.3.6) $\ell\mathrm{c}(s\cdot f)=\ell\mathrm{c}(f)$. (Cor. 2.3.7)
c)	If $\alpha\cdot\ell\mathrm{c}(f)\neq 0$ then $\alpha\cdot tf\neq 0$, $\ell\mathrm{t}(\alpha\cdot tf)=t\cdot\ell\mathrm{t}(f)$, $\ell\mathrm{c}(\alpha\cdot tf)=\alpha\cdot\ell\mathrm{c}(f)$.	If $s\cdot\ell\mathrm{t}(f)\neq 0$ then $\alpha\cdot sf\neq 0$, $\ell\mathrm{t}(\alpha\cdot sf)=s\cdot\ell\mathrm{t}(f)$, (Cor. 2.3.6) $\ell\mathrm{c}(\alpha\cdot sf)=\alpha\cdot\ell\mathrm{c}(f)$. (Cor. 2.3.7)
d)	no difference	

Lemma 2.5.5. Let $\alpha\in R^*$, $t\in\mathbb{T}$, $f,g\in R[X]\backslash\{0\}$. The following holds.

a) If $\alpha\cdot f\neq 0$ then $\ell\mathrm{t}(\alpha\cdot f)\leq\ell\mathrm{t}(f)$;
if $\alpha\cdot\ell\mathrm{c}(f)\neq 0$ then $\alpha\cdot f\neq 0$ and $\ell\mathrm{t}(\alpha\cdot f)=\ell\mathrm{t}(f)$, $\ell\mathrm{c}(\alpha\cdot f)=\alpha\cdot\ell\mathrm{c}(f)$, $\ell\mathrm{m}(\alpha\cdot f)=\alpha\cdot\ell\mathrm{m}(f)$.

b) $t\cdot f\neq 0$ and $\ell\mathrm{t}(t\cdot f)=t\cdot\ell\mathrm{t}(f)$, $\ell\mathrm{c}(t\cdot f)=\ell\mathrm{c}(f)$, $\ell\mathrm{m}(t\cdot f)=t\cdot\ell\mathrm{m}(f)$.

c) If $\alpha\cdot f\neq 0$ then $\alpha\cdot tf\neq 0$ and $\ell\mathrm{t}(\alpha\cdot tf)\leq t\cdot\ell\mathrm{t}(f)$;
if $\alpha\cdot\ell\mathrm{c}(f)\neq 0$ then $\alpha\cdot tf\neq 0$ and $\ell\mathrm{t}(\alpha\cdot tf)=t\cdot\ell\mathrm{t}(f)$, $\ell\mathrm{c}(\alpha\cdot tf)=\alpha\cdot\ell\mathrm{c}(f)$, $\ell\mathrm{m}(\alpha\cdot tf)=\alpha\cdot t\cdot\ell\mathrm{m}(f)$.

d) If $f+g\neq 0$ then $\ell\mathrm{t}(f+g)\leq\max\big(\ell\mathrm{t}(f),\ell\mathrm{t}(g)\big)$;
if $\ell\mathrm{t}(f)\neq\ell\mathrm{t}(g)$ or $\ell\mathrm{c}(f)+\ell\mathrm{c}(g)\neq 0$ then $f+g\neq 0$ and $\ell\mathrm{t}(f+g)=\max\big(\ell\mathrm{t}(f),\ell\mathrm{t}(g)\big)$;
if $\ell\mathrm{t}(f)=\ell\mathrm{t}(g)$, $\ell\mathrm{c}(f)+\ell\mathrm{c}(g)=0$ and $f+g\neq 0$ then $\ell\mathrm{t}(f+g)<\ell\mathrm{t}(g)$.

Proof. The condition that certain polynomial is not zero in the suppositions of each part of the lemma follows that the support of the polynomial is not the empty set by the uniqueness of the support, hence the maximal element function is defined on this support and therefore the polynomial has the leading term.

a) – If $\alpha{\cdot}f{\neq}0$ then, by part *a*) of lemma 2.5.2,

$$\ell\mathrm{t}(\alpha{\cdot}f) = \max\Big(\mathrm{supp}\,(\alpha{\cdot}f)\Big) \overset{(L2.5.2)}{\leq} \max\Big(\mathrm{supp}\,(f)\Big) = \ell\mathrm{t}(f). \tag{1a}$$

– By part *a*) of corollary 2.5.1,

$$[\mathfrak{C}_{\alpha\cdot f}]\Big(\ell\mathrm{t}(f)\Big) \overset{(C2.5.1)}{=} \alpha{\cdot}[\mathfrak{C}_f]\Big(\ell\mathrm{t}(f)\Big) = \alpha{\cdot}\ell\mathrm{c}(f). \tag{2a}$$

If $\alpha{\cdot}\ell\mathrm{c}(f){\neq}0$ then

$$\ell\mathrm{t}(f){\in}\,\mathrm{supp}\,(\alpha{\cdot}f) \tag{3a}$$

follows $\mathrm{supp}\,(\alpha{\cdot}f)\,{\neq}\emptyset$, hence

$$\alpha{\cdot}f{\neq}0 \text{ and } \boxed{\ell\mathrm{t}(f)} \overset{(3a)}{\leq} \max\Big(\mathrm{supp}\,(\alpha{\cdot}f)\Big) = \boxed{\ell\mathrm{t}(\alpha{\cdot}f)} \overset{(1a)}{\leq} \boxed{\ell\mathrm{t}(f)}\,. \tag{4a}$$

Furthermore

$$\ell\mathrm{c}(\alpha{\cdot}f) = \mathfrak{C}_{\alpha\cdot f}\Big(\ell\mathrm{t}(\alpha{\cdot}f)\Big) \overset{(4a)}{=} \mathfrak{C}_{\alpha\cdot f}\Big(\ell\mathrm{t}(f)\Big) = [\mathfrak{C}_{\alpha\cdot f}]\Big(\ell\mathrm{t}(f)\Big) \overset{(2a)}{=} \alpha{\cdot}\ell\mathrm{c}(f) \tag{5a}$$

and

$$\ell\mathrm{m}(\alpha{\cdot}f) = \ell\mathrm{c}(\alpha{\cdot}f){\cdot}\ell\mathrm{t}(\alpha{\cdot}f) \overset{(4a),(5a)}{=} \alpha{\cdot}\ell\mathrm{c}(f)\ell\mathrm{t}(f) = \alpha{\cdot}\ell\mathrm{m}(f). \tag{6a}$$

It is clear that (1*a*), (4*a*), (5*a*) and (6*a*) cover all propositions of part *a*) under respective assumptions.

b) By part *d*) of lemma 2.5.2 and lemma 2.5.1,

$$\mathrm{supp}\,(t{\cdot}f) \overset{(L2.5.2)}{=} t{*}\,\mathrm{supp}\,(f) \overset{\substack{f\neq0,\\(L2.5.1)}}{\neq} \emptyset \;\text{ follows }\; t{\cdot}f{\neq}0 \tag{1b}$$

by the uniqueness of the support. Furthermore, it is an easy consequence of the term order monotonicity inverse (lemma 2.5.3) that

$$\ell\mathrm{t}(t{\cdot}f) = \max\Big(\mathrm{supp}\,(t{\cdot}f)\Big) \overset{(1b)}{=} \max\Big(t{*}\,\mathrm{supp}\,(f)\Big) \overset{(L2.5.3)}{=} t{\cdot}\max\Big(\mathrm{supp}\,(f)\Big) {=} t{\cdot}\ell\mathrm{t}(f). \quad (2b)$$

Then, by part $d)$ of corollary 2.5.1,

$$\ell\mathrm{c}(t{\cdot}f) = \mathfrak{C}_{t\cdot f}\Big(\ell\mathrm{t}(t{\cdot}f)\Big) \overset{(2b)}{=} \mathfrak{C}_{t\cdot f}\Big(t{\cdot}\ell\mathrm{t}(f)\Big) \overset{(C2.5.1)}{=} \mathfrak{C}_{f}\Big(\ell\mathrm{t}(f)\Big) = \ell\mathrm{c}(f). \quad (3b)$$

From $(2b)$ and $(3b)$,

$$\ell\mathrm{m}(t{\cdot}f) = \ell\mathrm{c}(t{\cdot}f){\cdot}\ell\mathrm{t}(t{\cdot}f) \overset{(2b),(3b)}{=} \ell\mathrm{c}(f){\cdot}t\ell\mathrm{t}(f) = t{\cdot}\ell\mathrm{m}(f). \quad (4b)$$

It is clear that $(1b)$, $(2b)$, $(3b)$ and $(4b)$ cover all statements of part $b)$.

$c)$

- If $\alpha{\cdot}f{\neq}0$ then

$$\alpha{\cdot}tf{\neq}0 \quad (1c)$$

by part $b)$. Furthermore, by $(1a)$ and $(1b)$,

$$\ell\mathrm{t}(\alpha{\cdot}tf) \overset{(1a)}{\leq} \ell\mathrm{t}(tf) \overset{(2b)}{=} t{\cdot}\ell\mathrm{t}(f). \quad (2c)$$

- If $\alpha{\cdot}\ell\mathrm{c}(f){\neq}0$ then, by part $a)$, $\alpha{\cdot}f{\neq}0$ and hence, by part $b)$,

$$\alpha{\cdot}tf{\neq}0. \quad (3c)$$

Furthermore, by an easy combination of of parts $a)$ and $b)$ provided that $\alpha{\cdot}\ell\mathrm{c}(f){\neq}0$,

$$\ell\mathrm{t}(\alpha{\cdot}tf) \overset{b)}{=} t{\cdot}\ell\mathrm{t}(\alpha{\cdot}f) \overset{\alpha\cdot\ell\mathrm{c}(f)\neq0,\ a)}{=} t{\cdot}\ell\mathrm{t}(f) \quad (4c)$$

$$\ell\mathrm{c}(\alpha{\cdot}tf) \overset{b)}{=} \ell\mathrm{c}(\alpha{\cdot}f) \overset{\alpha\cdot\ell\mathrm{c}(f)\neq0,\ a)}{=} \alpha{\cdot}\ell\mathrm{c}(f) \quad (5c)$$

$$\ell\mathrm{m}(\alpha{\cdot}tf) \overset{b)}{=} t{\cdot}\ell\mathrm{m}(\alpha{\cdot}f) \overset{\alpha\cdot\ell\mathrm{c}(f)\neq0,\ a)}{=} \alpha{\cdot}t{\cdot}\ell\mathrm{m}(f) \quad (6c)$$

It is clear that $(1c)$, $(2c)$, $(3c)$, $(4c)$, $(5c)$ and $(6c)$ cover all propositions of part $c)$ under respective assumptions.

d) – If $f{+}g{\neq}0$ then, by part *b)* of lemma 2.5.2,

$$\begin{aligned}\ell\mathrm{t}(f{+}g) &= \max\Big(\mathrm{supp}\,(f{+}g)\Big) && (1d)\\ &\overset{(L2.5.2)}{\leq} \max\Big(\mathrm{supp}\,(f)\cup\mathrm{supp}\,(g)\Big) = \\ &\qquad \max\Big(\max\big(\mathrm{supp}\,(f)\big),\max\big(\mathrm{supp}\,(g)\big)\Big) = \\ &\qquad\qquad \max\Big(\ell\mathrm{t}(f),\ell\mathrm{t}(g)\Big).\end{aligned}$$

– Let $\ell\mathrm{t}(f){\neq}\ell\mathrm{t}(g)$ or $\ell\mathrm{c}(f){+}\ell\mathrm{c}(g){\neq}0$. Consider two cases

* Let $\ell\mathrm{t}(f){\neq}\ell\mathrm{t}(g)$. W.l.o.g. assume that $\ell\mathrm{t}(f){<}\ell\mathrm{t}(g)$. Then it is easy to see by transitivity of the total order $<$ and definition 2.5.2 of the leading term that

$$\begin{aligned}&\ell\mathrm{t}(g)\in\mathrm{supp}\,(g)\setminus\mathrm{supp}\,(f)\subseteq\\ &\qquad \mathrm{supp}\,(f)\triangle\mathrm{supp}\,(g) \overset{(L2.5.2)}{\subseteq} \mathrm{supp}\,(f{+}g)\,. && (2d)\end{aligned}$$

Then

$$\mathrm{supp}\,(f{+}g)\neq\emptyset \;\text{ follows }\; f{+}g{\neq}0. \qquad (3d)$$

Furthermore, by part *b)* of lemma 2.5.2,

$$\begin{aligned}&\boxed{\ell\mathrm{t}(g)} \overset{(2d)}{\leq} \max\Big(\mathrm{supp}\,(f)\triangle\mathrm{supp}\,(g)\Big) \overset{(L2.5.2)}{\leq}\\ &\qquad \max\Big(\mathrm{supp}\,(f+g)\Big) = \boxed{\ell\mathrm{t}(f{+}g)} \overset{(1d)}{\leq}\\ &\qquad\qquad \max\Big(\ell\mathrm{t}(f),\ell\mathrm{t}(g)\Big) = \boxed{\ell\mathrm{t}(g)}\,. && (4d)\end{aligned}$$

* Let $\ell\mathrm{t}(f){=}\ell\mathrm{t}(g)$ and $\ell\mathrm{c}(f){+}\ell\mathrm{c}(g){\neq}0$. Then

$$\begin{aligned}&[\mathfrak{C}_{f+g}]\Big(\ell\mathrm{t}(g)\Big) = [\mathfrak{C}_f]\Big(\ell\mathrm{t}(g)\Big) + [\mathfrak{C}_g]\Big(\ell\mathrm{t}(g)\Big) = \\ &\qquad \mathfrak{C}_f\Big(\ell\mathrm{t}(f)\Big) + \mathfrak{C}_g\Big(\ell\mathrm{t}(g)\Big) = \ell\mathrm{c}(f){+}\ell\mathrm{c}(g){\neq}0 && (5d)\end{aligned}$$

follows $\ell\mathrm{t}(g){\in}\mathrm{supp}\,(f{+}g)$, hence

$$\mathrm{supp}\,(f{+}g)\neq\emptyset \;\text{ and }\; f{+}g{\neq}0 \qquad (6d)$$

by the uniqueness of the support. Furthermore

$$\boxed{\ell\mathrm{t}(g)} \overset{(5d)}{\leq} \max\Big(\mathrm{supp}\,(f{+}g)\Big) = \boxed{\ell\mathrm{t}(f{+}g)} \overset{(1d)}{\leq}$$

$$\max\Big(\ell\mathrm{t}(f), \ell\mathrm{t}(g)\Big) = \boxed{\ell\mathrm{t}(g)}\,. \tag{7d}$$

In both cases

$$f{+}g \overset{(3d),(6d)}{\neq} 0, \tag{8d}$$

$$\ell\mathrm{t}(f{+}g) \overset{(4d),(7d)}{=} \max\Big(\ell\mathrm{t}(f), \ell\mathrm{t}(g)\Big). \tag{9d}$$

– Let $\ell\mathrm{t}(f){=}\ell\mathrm{t}(g)$, $\ell\mathrm{c}(f){+}\ell\mathrm{c}(g){=}0$ and $f{+}g{\neq}0$. Then

$$[\mathfrak{C}_{f+g}]\Big(\ell\mathrm{t}(g)\Big) = [\mathfrak{C}_f]\Big(\ell\mathrm{t}(g)\Big) + [\mathfrak{C}_g]\Big(\ell\mathrm{t}(g)\Big) =$$

$$\mathfrak{C}_f\Big(\ell\mathrm{t}(f)\Big) + \mathfrak{C}_g\Big(\ell\mathrm{t}(g)\Big) = \ell\mathrm{c}(f){+}\ell\mathrm{c}(g){=}0$$

follows

$$\ell\mathrm{t}(g){\notin}\,\mathrm{supp}\,(f{+}g) \quad \text{and hence} \quad \ell\mathrm{t}(g){\neq}\ell\mathrm{t}(f{+}g).$$

On the other hand, by ($1d$), $\ell\mathrm{t}(f{+}g){\leq}\max\big(\ell\mathrm{t}(f), \ell\mathrm{t}(g)\big){=}\ell\mathrm{t}(g)$. Therefore

$$\ell\mathrm{t}(f{+}g) < \ell\mathrm{t}(g) \tag{10d}$$

It is clear that ($1d$), ($8c$), ($9d$) and ($10d$) cover all propositions of part d) under respective assumptions. □

Part c) of lemma 2.5.5 can be reformulated in the language of monomials. It does not introduce any new content but gives a somewhat convenient notation. Namely, due to the fact that $\mathbb{T}$ is a free monoid and the linear independence of $\mathbb{T}$ over R, $m_1{\cdot}m_2{=}0$ if and only if $\mathfrak{C}(m_1)\cdot\mathfrak{C}(m_2)\,{=}0$ for any $m_1, m_2{\in}\mathbb{M}$.

Corollary 2.5.2 (reformulation of part c) of lemma 2.5.5).
Let $f{\in}R\,[X]{\setminus}\{0\}$ and $m{\in}M$.

If $m{\cdot}f{\neq}0$	then $m{\cdot}f{\neq}0$ and $\ell\mathrm{t}(m{\cdot}f){\leq}\mathfrak{T}(m)\cdot\ell\mathrm{t}(f)$;		
if $m{\cdot}\ell\mathrm{c}(f){\neq}0$	then $m{\cdot}f{\neq}0$ and	$\ell\mathrm{t}(m{\cdot}f)$	$=\mathfrak{T}(m)\cdot\ell\mathrm{t}(f)$,
		$\ell\mathrm{c}(m{\cdot}f)$	$=\mathfrak{C}(m)\cdot\ell\mathrm{c}(f)$,
		$\ell\mathrm{m}(m{\cdot}f)$	$=m{\cdot}\ell\mathrm{m}(f)$.

Proof. By part c) of lemma 2.5.5 with $\alpha{=}\,\mathfrak{C}(m)\in R^*$ and $t{=}\,\mathfrak{T}(m)\in\mathbb{T}$ provided that $m{\cdot}f{\neq}0$ if and only if $\alpha{\cdot}f{\neq}0$ and $m{\cdot}\ell\mathrm{c}(f){\neq}0$ if and only if $\alpha{\cdot}\ell\mathrm{c}(f){\neq}0$. □

Monomial ideals in $R[X]$

The structure of monomial ideals is typically used in the Gröbner bases theory for polynomials over a ring as a tool to study the top-reduction relation and argue the termination of the completion procedure.

Definition 2.5.3. An ideal $I \triangleleft R[X]$ is called a **monomial ideal** if and only if there exists a set of monomials in $R[X]$

$$M \subseteq \mathbb{M} \quad \text{such that} \quad I = \langle M \rangle_{R[X]}.$$

The following lemma states a characteristic property of monomial ideals. It is interchangeable with the definition of a monomial ideal and more convenient for theoretical considerations.

Lemma 2.5.6. Let $I = \langle M \rangle_{R[X]}$ be a monomial ideal generated by a set of monomials $M \subseteq \mathbb{M}$, $\mu \in \mathbb{M}$ is an arbitrary monomial, $f = \sum\limits_{t \in \mathrm{supp}(f)} f_t \in R[X]$ is an arbitrary polynomial presented by the (unique) sum of monomials $f_t = \mathfrak{M}_f(t)$, $t \in \mathrm{supp}(f)$. Then $f \in \langle M \rangle_{R[X]}$ if and only if $f_t \in \langle M \rangle_{R[X]}$ for each $t \in \mathrm{supp}(f)$. Furthermore, there exists a finite set of monomials $M_f \subseteq M$ such that f can be represented as a finite sum

$$f = \sum_{i \in N} a_i \cdot m_i \text{ with } N \text{ - a finite subset of } \mathbb{N},$$
$$a_i \in \mathbb{M},\ m_i \in \mathbb{M}_f \text{ not necessary pairwise different}$$
$$\text{and } \mathfrak{T}(m_i) \cdot \mathfrak{T}(m) \in \mathrm{supp}(f) \text{ for all } i \in N.$$

Proof. Let $f = \sum\limits_{t \in \mathrm{supp}(f)} f_t \in R[X]$ with $f_t \in \mathfrak{M}(f)$. Then there exists a finite set $M_f \subseteq \mathbb{M}$ and polynomials $q_m \in R[X]$, one for each $m \in M_f$, such that

$$f = \sum_{m \in M_f} q_m \cdot m. \qquad (*)$$

It is clear that using the distributive law the sum $(*)$ can be converted into a sum with monomial multiplies $\bigcup\limits_{m \in M_f} \mathbb{M}(q_m)$

$$\sum_{m \in M_f} q_m \cdot m = \sum_{m \in M_f} \sum_{v \in \mathrm{supp}(q_m)} \mathfrak{M}_{q_m}(v) \cdot m =$$
$$\sum_{t \in \mathbb{T}} \sum_{m \in M_f} \sum_{\substack{v \in \mathrm{supp}(q_m) \\ v \cdot \mathfrak{T}(m) = t}} \mathfrak{M}_{q_m}(v) \cdot m. \qquad (**)$$

Denote the innermost sums

$$A_{t,m} = \sum_{\substack{v\in\operatorname{supp}(q_m)\\ v\cdot\mathfrak{T}(m)=t}} \mathfrak{M}_{q_m}(v)\cdot m, \text{ for each } t\in\mathbb{T} \text{ and } m\in M_f$$

and the second sums

$$B_t = \sum_{m\in M_f} A_{t,m} \text{ for each } t\in\mathbb{T}.$$

Since the terms $\mathbb{T}$ are linear independent over R, the equation

$$f = \sum_{t\in\operatorname{supp}(f)} f_t = (**) = \sum_{t\in\operatorname{supp}(f)} B_t$$

follows that

$$B_t = \begin{cases} f_t & t\in\operatorname{supp}(f) \\ 0 & \text{otherwise.} \end{cases}$$

Fix $t\in\operatorname{supp}(f)$. It is clear that the (monoidal) equation $v\cdot\mathfrak{T}(m)=t$ with fixed $t\in T$, $m\in M_f$ and indeterminate $v\in\operatorname{supp}(q_m)$ has at most one solution in the free monoid $\mathbb{T}$. Thus each $A_{t,m}$ consists of at most one summand:

$$A_{t,m} = \begin{cases} \mathfrak{M}_{q_m}(w)\cdot m & \text{if } \exists w\in\operatorname{supp}(q_m) \text{ such that } w\cdot\mathfrak{T}(m)=t \\ 0 & \text{otherwise} \end{cases}$$

$$\text{for each } m\in M_f.$$

Define $M_t = \{m\in M_f \mid A_{t,m}\neq 0\}$ for each $t\in\mathbb{T}$. Note that for each $m\in M_t$ the sum $A_{t,m}$ consists of exactly one summand $\mathfrak{M}_{q_m}(w)\cdot m$ with $w\in\operatorname{supp}(q_m)$ such that $w\cdot\mathfrak{T}(m)=t$ (but the converse is not true since the product of monomials can be zero if there are zero divisors in R). Define $a_m = \mathfrak{M}_{q_m}(w)\in\mathbb{M}$. Then

$$\mathfrak{T}(a_m)\cdot\mathfrak{T}(m)=t \text{ and } f_t = B_t = \sum_{m\in M_t} A_{t,m} = \sum_{m\in M_t} a_m\cdot m \qquad (***)$$

Taking the representation $(***)$ for each $t\in\operatorname{supp}(f)$

$$f = \sum_{t\in\operatorname{supp}(f)} f_t = \sum_{t\in\operatorname{supp}(f)}\sum_{m\in M_t} a_m\cdot m \text{ with}$$

$$a_m \in \bigcup_{t\in\operatorname{supp}(f)} M_t \subseteq M_f \text{ and } \mathfrak{T}(a_m)\cdot\mathfrak{T}(m)\in\operatorname{supp}(f)$$

$$\text{for each } t\in\operatorname{supp}(f), m\in M_t.$$

which is the desired representation. □

Corollary 2.5.3. If $\mu\in M$ and $M\subseteq\mathbb{M}$ then $\mu\in\langle M\rangle_{\mathcal{P}}$ if and only if there is a finite subset $M_\mu\subseteq M$ such that μ can be represented as a finite sum with monomial co-factors

$$\mu=\sum_{m\in M_\mu} a_m\cdot m \text{ with } a_m\in\mathbb{M} \text{ and } \mathfrak{T}(a_m)\cdot\mathfrak{T}(m)=\mathfrak{T}(\mu) \text{ for each } m\in M_\mu.$$

Proof. It is a refined particular case of lemma 2.5.6. It is easy to see that the right-most sum with pairwise different elements from a subset of M in the equation $(***)$ in the proof of lemma 2.5.6 is exactly the claimed statement. □

The following lemma describes the special nature of monomial ideals. The presented property reflects the idea behind the forthcoming definition of the top-reduction relation in $R[X]$ in the next subsection.

Lemma 2.5.7. Let $M\subseteq\mathbb{M}$. Then there exists a finite subset $M_o\subseteq M$ such that

$$\langle M\rangle_{\mathcal{P}}=\langle M_o\rangle_{R[X]}.$$

Proof. Take $M\subseteq\mathbb{M}$ and let $I=\langle M\rangle_{R[X]}$. By Hilbert basis theorem the ring $R[X]$ is Nötherian. Hence there exists a finite set $F\subseteq I\backslash\{0\}$ such that $I=\langle F\rangle_{R[X]}$. Note that trivially

$$\langle f\rangle_{R[X]}\subseteq\sum_{\mu\in\mathfrak{M}(f)}\langle\mu\rangle_{R[X]}. \tag{$*$}$$

Take $f\in F$ and $\mu\in\mathfrak{M}(f)$. Then, by corollary 2.5.3 there is a finite subset $M_\mu\subseteq M$ such that $\mu\in\langle M_\mu\rangle_{R[X]}$. Thus

$$\langle\mu\rangle_{R[X]}\subseteq\langle M_\mu\rangle_{R[X]} \tag{$**$}$$

and

$$\langle M_\mu\rangle_{R[X]}\subseteq\langle M\rangle_{R[X]}. \tag{$***$}$$

From the above equations

$$\begin{aligned}\langle M\rangle_{R[X]}=\langle I\rangle_{R[X]}=\left\langle\bigcup_{f\in F}f\right\rangle_{R[X]}=\sum_{f\in F}\langle f\rangle_{R[X]}\overset{(*)}{\subseteq}\\ \sum_{f\in F}\left\langle\bigcup_{\mu\in\mathfrak{M}(f)}\mu\right\rangle_{R[X]}=\sum_{f\in F}\sum_{\mu\in\mathfrak{M}(f)}\langle\mu\rangle_{R[X]}\overset{(**)}{\subseteq}\\ \sum_{f\in F}\sum_{\mu\in\mathfrak{M}(f)}\langle M_\mu\rangle_{R[X]}=\left\langle\bigcup_{f\in F}\bigcup_{\mu\in\mathfrak{M}(f)}M_\mu\right\rangle_{R[X]}\overset{(***)}{\subseteq}\langle M\rangle_{R[X]}.\end{aligned} \tag{$\dagger$}$$

Take

$$M_o = \bigcup_{f\in F} \bigcup_{\mu\in\mathfrak{M}(f)} M_\mu.$$

By the choice of M_μ for each $f\in F$ and $\mu\in\mathfrak{M}(f)$, the set M_o is finite and $M_o\subseteq M$. By (†)

$$\langle M\rangle_{R[X]} \subseteq \langle M_o\rangle_{R[X]} \subseteq \langle M\rangle_{R[X]}$$

and thus M_o is the desired finite generating subset. $\square$

Reduction and reducibility in $R[X]$

The following definition of the one step top-reduction relation $\rightarrow$ (together with the reflexive-transitive closure $\rightarrow_*$) in $R[X]$ might look quite complicated although it is a natural generalization of the respective relation from the classical theory of Gröbner bases which preserves, as it is shown in the subsequent lemmas, the meaning of the relation for the theory of Gröbner bases: if $f\in R[X]\backslash\{0\}$, $g\in R[X]$, $F\subseteq R[X]\backslash\{0\}$ then the fact that f top-reduces to g in one step modulo F is a simple criterion for $\ell\mathrm{m}(f)\in\ell\mathrm{m}\Big(\langle F\rangle_{R[X]}\Big)$. Furthermore, the possible existence of zero-divisors in R requires special care in the definition of top-reduction to ensure that the leading term of the top-reducible (definition 2.5.5) polynomial actually cancels at the one step top-reduction.

Definition 2.5.4. Let $F\subseteq R[X]\backslash\{0\}$.

- For any $f\in R[X]\backslash\{0\}$ and $g\in R[X]$, f **top-reduces to g modulo F in one step**

  $$f \xrightarrow{F} g$$

  if and only if there is a finite subset $H\subseteq F$ and monomials $a_h\in\mathbb{M}$ one for each $h\in H$ such that

  $$\ell\mathrm{m}(f) = \sum_{h\in H} a_h\cdot\ell\mathrm{m}(h) \text{ with } \mathfrak{T}(a_h)\cdot\ell\mathrm{t}(h) = \ell\mathrm{t}(f) \quad \text{for all} \quad h\in H.$$

  and

  $$g = f - \sum_{h\in H} a_h\cdot h.$$

- For any $f,g\in R[X]$, f **top-reduces to g modulo F** if and only if

  $$f \xrightarrow{F}_* g$$

  where $\xrightarrow{F}_*$ is the reflexive-transitive closure of the relation $\xrightarrow{F}$.

- For any $f, g \in R[X]$, a sequence $\{f_o, \ldots, f_N\} \subseteq R[X]$ of $N+1 \in \mathbb{N}$ polynomials such that $f_o = f$, $f_i \xrightarrow{F} f_i+1$ for $0 \leq i < N$ and $f_N = g$

$$f = f_o \xrightarrow{F} f_1 \xrightarrow{F} \cdots \xrightarrow{F} f_i \xrightarrow{F} f_{i+1} \xrightarrow{F} \cdots \xrightarrow{F} f_N = g$$

is called **an N-top-reduction sequence of f to g modulo F** (note that, according to the definition of one step top-reduction, $f_i \neq 0$ for $0 \leq i < N$). A sequence of polynomials is called **a top-reduction sequence of f to g modulo F** if and only if it is an N-top-reduction sequence of f to g modulo F for some $N \in \mathbb{N}_o$. It is clear, according to what the reflexive-transitive closure of a relation is, that $f \xrightarrow{F}_* g$ if and only if there exists a top-reduction sequence of f to g modulo F.

Definition 2.5.5. A polynomial $f \in R[X]$ is called **top-reducible** modulo $F \subseteq R[X] \setminus \{0\}$ (denoted $f \xrightarrow{F}$) if and only if it top-reduces to some polynomial in $R[X]$ modulo F in one step i.e. there exists $g \in R[X]$ such that $f \xrightarrow{F} g$.

Definition 2.5.6. A polynomial $f \in R[X]$ is called **top-minimal** modulo $F \subseteq R[X] \setminus \{0\}$ if and only if f is not top-reducible modulo F.

Lemma 2.5.8. A polynomial $f \in R[X]$ is top-reducible modulo $F \subseteq R[X] \setminus \{0\}$ if and only if

$$f \neq 0 \quad \text{and} \quad \ell\mathrm{m}(f) \in \big\langle \ell\mathrm{m}(F) \big\rangle_{R[X]}. \tag{$\dagger$}$$

Proof. If f is top-reducible modulo F then it top-reduces in one step to some $g \in R[X]$, hence, by definition 2.5.4 of one step top-reduction, $f \neq 0$ and there exists a finite subset $H \subseteq F$ such that

$$\ell\mathrm{m}(f) = \sum_{h \in H} a_h \cdot \ell\mathrm{m}(h) \quad \text{with} \quad a_h \in \mathbb{M} \text{ for each } h \in H.$$

Therefore

$$\ell\mathrm{m}(h) \in \langle \ell\mathrm{m}(H) \rangle_{R[X]} \subseteq \langle \ell\mathrm{m}(F) \rangle_{R[X]}.$$

Conversely, if ($\dagger$) holds then by corollary 2.5.3, there is a finite subset $H \subseteq F$ and monomials $a_h \in \mathbb{M}$ one for each $h \in H$ such that

$$\ell\mathrm{m}(f) = \sum_{h \in H} a_h \cdot \ell\mathrm{m}(h) \quad \text{and} \quad \mathfrak{T}(a_h) \cdot \ell\mathrm{t}(h) = \ell\mathrm{t}(f) \quad \text{for each} \quad h \in H$$

which is exactly the condition on the leading monomial of f in the definition of one step top-reduction but then ir is clear that $f \xrightarrow{F} g$ with

$$g = f - \sum_{h \in H} a_h \cdot h.$$

Hence f is top-reducible by F according to definition 2.5.5. $\square$

Lemma 2.5.9. A polynomial $f{\in}R\,[X]$ is top-minimal modulo $F{\subseteq}R\,[X]\backslash\{0\}$ if and only if

$$f{=}0 \;\text{ or }\; \ell\mathrm{m}(f) \notin \langle \ell\mathrm{m}(F)\rangle_{R\,[X]}.$$

Proof. By definition 2.5.6, $f{\in}R\,[X]$ is top-minimal modulo F if and only if f is not top-reducible modulo F if and only if

$$f{=}0 \;\text{ or }\; \ell\mathrm{m}(f) \notin \langle \ell\mathrm{m}(F)\rangle_{R\,[X]}.$$

according to lemma 2.5.8. □

Being careful with the definition of the "top-reduces to" relation we get the usual property of the top-reducibility which is, together with the Nötherianity property of the top-reduction relation substantiated by the well-ordering property of $<$ on $\mathbb{T}$, the key for the reduction algorithm in $R\,[X]$.

Lemma 2.5.10. Let $f{\in}R\,[X]\backslash\{0\}$, $g{\in}R\,[X]$ and $F{\subseteq}R\,[X]\backslash\{0\}$. If f top-reduces to g by F then

$$\text{either } g{=}0 \text{ or } \ell\mathrm{t}(g){<}\ell\mathrm{t}(f).$$

Proof. The case $g{=}0$ is trivial. Assume $g{\neq}0$. Take a finite subset $H{\subseteq}F$ and monomials $a_h{\in}M$ one for each $h{\in}H$ such that

$$\ell\mathrm{m}(f) = \sum_{h\in H} a_h{\cdot}\ell\mathrm{m}(h) \quad \text{with} \quad \mathfrak{T}(a_h)\cdot\ell\mathrm{t}(h) = \ell\mathrm{t}(f) \;\text{ for each }\; h\in H, \tag{1}$$

$$g = f - \sum_{h\in H} a_h{\cdot}h \tag{2}$$

provided by definition 2.5.4 of the top-reduction relation. Take a subset of H

$$H' = \Big\{h{\in}H \mid a_h{\cdot}\big(h{-}\ell\mathrm{m}(h)\big){\neq}0\Big\} \subseteq H. \tag{3}$$

Then the equation (2) is clearly equivalent to

$$\begin{aligned} g = f - \sum_{h\in H} a_h{\cdot}h &= f - \sum_{h\in H} a_h{\cdot}\ell\mathrm{m}(h) - \sum_{h\in H} a_h{\cdot}\big(h - \ell\mathrm{m}(h)\big) \\ &\overset{(1)}{=} f - \ell\mathrm{m}(f) - \sum_{h\in H} a_h{\cdot}\big(h - \ell\mathrm{m}(h)\big) \\ &\overset{(3)}{=} \underbrace{\Big(f - \ell\mathrm{m}(f)\Big)}_{\substack{\|\\ g_1}} - \underbrace{\Big(\sum_{h\in H'} a_h{\cdot}\big(h - \ell\mathrm{m}(h)\big)\Big)}_{\substack{\|\\ g_2}}. \end{aligned} \tag{4}$$

We claim that each outermost summand of the last expression in (4) delivers a strict upper bound on $\ell\mathrm{t}(f)$ unless zero. Indeed, if $g_1{\neq}0$ then

$$\ell\mathrm{t}(g_1) = \ell\mathrm{t}\big(f{-}\ell\mathrm{m}(f)\big) < \ell\mathrm{t}(f) \tag{5}$$

by the linear independence of terms over R and definition of the support together with elementary properties of the maximal element function on a finite totally ordered set. On the other hand, for any $h{\in}H'$, $h{-}\ell\mathrm{m}(h){\neq}0$ by construction (3) of H' and hence

$$\ell\mathrm{t}\big(h{-}\ell\mathrm{m}(h)\big) < \ell\mathrm{t}(h) \tag{6}$$

by the same argument as (5). Then,

$$\begin{aligned}
\boxed{\ell\mathrm{t}\big(a_h{\cdot}\big(h{-}\ell\mathrm{m}(h)\big)\big)} &= \ell\mathrm{t}\Big(\mathfrak{C}(a_h)\cdot\mathfrak{T}(a_h)\cdot\big(h{-}\ell\mathrm{m}(h)\big)\Big) && \text{(by corollary 2.5.2)}\\
&\leq \ell\mathrm{t}\Big(\mathfrak{T}(a_h)\cdot\big(h{-}\ell\mathrm{m}(h)\big)\Big) && \text{(by part } b) \text{ of lemma 2.5.5)}\\
&= \mathfrak{T}(a_h)\cdot\ell\mathrm{t}\Big(\big(h{-}\ell\mathrm{m}(h)\big)\Big) && \text{(by (6) and monotonicity of } <)\\
&\boxed{<}\,\mathfrak{T}(a_h)\cdot\ell\mathrm{t}(h) && (\mathfrak{T}(a_h)\cdot\ell\mathrm{t}(h){=}\ell\mathrm{t}(f) \text{ by (1))}\\
&= \boxed{\ell\mathrm{t}(f)}\,. && (7)
\end{aligned}$$

If $g_2{\neq}0$ then, by inductive argument based on part d) of lemma 2.5.5 provided that $a_h{\cdot}\big(h{-}\ell\mathrm{m}(h)\big){\neq}0$ for all $h{\in}H'$ by construction (3) of H',

$$\ell\mathrm{t}(g_2){=}\ell\mathrm{t}\left(\sum_{h\in H'} a_h{\cdot}\big(h-\ell\mathrm{m}(h)\big)\right) \overset{(L2.5.5)}{\leq} \max_{h\in H'}\left(\ell\mathrm{t}\Big(a_h{\cdot}\big(h-\ell\mathrm{m}(h)\big)\Big)\right) \overset{(7)}{<} \ell\mathrm{t}(f). \tag{8}$$

According to (4), the case $g_1{=}0$ and $g_2{=}0$ is impossibly due to the assumption $g{\neq}0$. Then the desired strict inequality $\ell\mathrm{t}(g){<}\ell\mathrm{t}(f)$ is an easy consequence of (5) and (8). Indeed,

if $g_1{\neq}0$ and $g_2{=}0$ then

$$\ell\mathrm{t}(g) \overset{(4)}{=} \ell\mathrm{t}(g_1{+}g_2) = \ell\mathrm{t}(g_1) \overset{(5)}{<} \ell\mathrm{t}(f),$$

if $g_1{=}0$ and $g_2{\neq}0$ then

$$\ell\mathrm{t}(g) \overset{(4)}{=} \ell\mathrm{t}(g_1{+}g_2) = \ell\mathrm{t}(g_2) \overset{(8)}{<} \ell\mathrm{t}(f),$$

if $g_1{\neq}0$ and $g_2{\neq}0$ then, using part d) of lemma 2.5.5,

$$\ell\mathrm{t}(g) \overset{(4)}{=} \ell\mathrm{t}(g_1{+}g_2) \overset{(L2.5.5)}{\leq} \max\Big(\ell\mathrm{t}(g_1), \ell\mathrm{t}(g_2)\Big) \overset{(5),(8)}{<} \ell\mathrm{t}(f). \qquad \square$$

Lemma 2.5.11. Let $f{\in}R\,[X]$ and $F{\subseteq}R\,[X]\backslash\{0\}$. The following holds

a) *The Nötherianity property of the top-reduction relation:* there is no infinite sequence of polynomials $\{f_o,\ldots,f_i,\ldots\}{\subseteq}R\,[X]$ such that $f_o{=}f$ and $f_i \xrightarrow{F} f_{i+1}$ for all $i{\in}\mathbb{N}$.

b) There exists a polynomial $g{\in}R\,[X]$ such that $f \xrightarrow{F}_* g$ and g is top-minimal modulo F.

Proof. Let $\{f_o,\ldots,f_i,\ldots\}{\subseteq}R\,[X]$ be an infinite sequence of polynomials such that $f_o{=}f$ and $f_i \xrightarrow{F} f_{i+1}$ for all $i{\in}\mathbb{N}_o$.Then $f_i{\neq}0$ for all $i{\in}\mathbb{N}_o$ according to definition 2.5.4 of one step top-reduction and hence, by lemma 2.5.10, the leading terms $\{\ell\mathrm{t}(f_o),\ldots,\ell\mathrm{t}(f_i),\ldots\}{\subseteq}\mathbb{T}$ of the polynomials in the sequence form a strictly descending chain in $\mathbb{T}$

$$\cdots < \ell\mathrm{t}(f_{i+1}) < \ell\mathrm{t}(f_i) < \cdots < \ell\mathrm{t}(f_o)$$

which contradicts a well-known from the classical theory Gröbner bases fact that a term order $<$ is a well-ordering on $\mathbb{T}$ (the point here is that $<$ is a well-ordering of $\mathbb{T}$ as a monoid irrelative of the embedding $\mathbb{T}{\subsetneq}R\,[X]$). Then there exists an index $N{\in}\mathbb{N}_o$ such that f_N does not top-reduce to any $f_{N+1}{\in}R\,[X]$ modulo F in one step. Hence f_N is top-minimal modulo F according to definition 2.5.6. Furthermore, by construction

$$f = f_o \xrightarrow{F} f_1 \xrightarrow{F} \cdots \xrightarrow{F} f_i \xrightarrow{F} f_{i+1} \xrightarrow{F} \cdots \xrightarrow{F} f_N = g$$

and hence $f \xrightarrow{F}_* g$. ☐

Part *b)* of lemma 2.5.11 ascertains the existence of a polynomial with an extra property stated in the next lemma. Such a polynomial is usually called a normal form.

Lemma 2.5.12. If $f{\in}R\,[X]\backslash\{0\}$, $g{\in}R\,[X]$, $F{\subseteq}R\,[X]\backslash\{0\}$ are such that $f \xrightarrow{F}_* g$ then

$$f - g \in \langle F\rangle_{R\,[X]}$$

Proof. If $f \xrightarrow{F}_* g$ then there exists $N{\in}\mathbb{N}_o$ and a top-reduction sequence of $N{+}1$ polynomials $\{f_o,\ldots,f_i,\ldots,f_N\}{\subseteq}R\,[X]$ of f to g modulo F

$$f = f_o \xrightarrow{F} f_1 \xrightarrow{F} \cdots \xrightarrow{F} f_i \xrightarrow{F} f_{i+1} \xrightarrow{F} \cdots \xrightarrow{F} f_N = g. \tag{1}$$

Then, for any index $0{\leq}i{<}N$ there exists a finite subset $H_i{\subseteq}F$ and monomials $\{a_{h,i}|\, h{\in}H_i\}{\subseteq}\mathbb{M}$ such that

$$f_{i+1} = f_i - \sum_{h\in H_i} a_{h,i}{\cdot}h = f_i - \sum_{h\in F} a_{h,i}{\cdot}h \quad \text{where} \quad a_{h,i}{:=}0 \quad \text{for all} \quad h{\in}F\backslash H_i. \tag{2}$$

provided by definition 2.5.4 of one step top-reduction. Hence

$$\begin{aligned} f-g &\overset{(1)}{=} f_o - f_N \\ &= (f_o - f_1) + \cdots + (f_i - f_{i+1}) + \cdots + (f_{N-1} - f_N) \\ &\overset{(2)}{=} \sum_{i=0}^{N-1} \sum_{h\in F} a_{h,i}\cdot h = \sum_{h\in F} h\cdot \underbrace{\left(\sum_{i=0}^{N-1} a_{h,i}\right)}_{\substack{\parallel \\ q_h \in R\,[X]}} \in \langle F\rangle_{R\,[X]} \end{aligned}$$

due to the fact that

$$q_h \neq 0 \;\text{ follows }\; h \in \bigcup_{i=0}^{N-1} H_i \;\text{ for any }\; h\in F$$

by construction and hence the sum $\sum\limits_{h\in F}^{N-1} h\cdot q_h$ has only finitely many non-zero summands i.e. it is a polynomial. □

Definition 2.5.7. Let $f\in R\,[X]$ and $F\subseteq R\,[X]\setminus\{0\}$. A polynomial $g\in R\,[X]$ is called **a normal form of f modulo F** if and only if

$$f - g \in \langle F\rangle_{R\,[X]} \;\text{ and} \tag{NF_1}$$

$$g \text{ is top-minimal modulo } F. \tag{NF_2}$$

Lemma 2.5.13. For any $f\in R\,[X]$ and $F\subseteq R\,[X]\setminus\{0\}$ there exists a normal form of f modulo F

Proof. By part b) of lemma 2.5.11 there exists a polynomial $g\in R\,[X]$ such that $f \overset{F}{\longrightarrow}_* g$ and g is top-minimal modulo F. By lemma 2.5.12, $f \overset{F}{\longrightarrow}_* g$ follows $f-g\in\langle F\rangle_{R\,[X]}$. Hence g is a normal form of f modulo F according to definition 2.5.7. □

Note that it is customary to consider the top-reducibility only in the theory of Gröbner bases in $R\,[X]$ (at least for theoretical purposes, cf. [Möl88]) due to the fact that the top-reducibility is perfectly sufficient to construct the theory of Gröbner bases in $R\,[X]$ (like in the classical case when R is a field) and, on the other hand, the consideration of reducibility of an arbitrary monomial in a polynomial does not bring a new theoretical insight like the uniqueness of a (totally) reduced Gröbner basis in the classical theory ([AL94] theorem 1.8.7 on page 48) due to an intrinsic ambiguity of a normal form in $R\,[X]$. Indeed, consider the ideal $I=\langle 2\cdot x\rangle$ in $\mathbb{Z}[x]$. Then $x-1$ and $3\cdot x-1$ are clearly equivalent modulo I and they are both minimal modulo $2\cdot x$.

Gröbner bases in $\mathbf{R}[\mathbf{X}]$

The definition of a Gröbner basis in $R[X]$ is taken almost verbatim from the classical Gröbner bases theory taking into account slightly more complicated structure of monomial ideals in $R[X]$.

Definition 2.5.8. Let I be an ideal in $R[X]$. A set $G \subseteq I \setminus \{0\}$ such that

$$\Big\langle \ell\mathrm{m}(G) \Big\rangle_{R[X]} = \Big\langle \ell\mathrm{m}(I) \Big\rangle_{R[X]}.$$

is called a **Gröbner basis of** I. A set $G \subseteq R[X] \setminus \{0\}$ is called a **Gröbner basis** if and only if it is a Gröbner basis of the ideal $\langle G \rangle_{R[X]}$ which it generates.

Lemma 2.5.14. Let I be an ideal in $R[X]$. Then then there exists a finite Gröbner basis of I.

Proof. By lemma 2.5.7, there exists a finite set of monomials $M \subseteq \ell\mathrm{m}(I)$ such that

$$\Big\langle \ell\mathrm{m}(I) \Big\rangle_{R[X]} = \langle M \rangle_{R[X]}. \qquad (*)$$

For each monomial $m \in M$ take a polynomial $g_m \in I \setminus \{0\}$ such that $\ell\mathrm{m}(g_m) = m$ provided that $m \in \ell\mathrm{m}(I)$. Then

$$G = \Big\{ g_m \mid m \in M \Big\} \subseteq I \setminus \{0\}$$

is clearly finite. Furthermore, by construction, $\ell\mathrm{m}(G) = M$, then

$$\Big\langle \ell\mathrm{m}(G) \Big\rangle_{R[X]} = \Big\langle M \Big\rangle_{R[X]} \overset{(*)}{=} \Big\langle \ell\mathrm{m}(I) \Big\rangle_{R[X]}$$

and hence G is a Gröbner basis of I according to definition 2.5.8. □

In view of the latter lemma and according to a common convention we assume a Gröbner basis in a polynomial algebra over a ring to be finite set henceforth. The next lemma presents the characterization of Gröbner bases in $R[X]$ in terms of zero top-reducibility of the basis elements which is necessary and sufficient for our moderated usage of the theory of Gröbner bases in $R[X]$. See [AL94] (theorem 4.1.12 on page 208) for another equivalent condition on a (finite) subset in $R[X]$ to be a Gröbner basis.

Theorem 2.5.1. Let I be an ideal in $R[X]$ and $G \subseteq I \setminus \{0\}$. The following are equivalent.

a) G is a Gröbner basis of I.

b) For any $f{\in}R\,[X]$

$$f{\in}I \text{ if and only if } f \overset{G}{\longrightarrow}_* 0$$

Proof.

a)→*b*) If $f \overset{F}{\longrightarrow}_* 0$ then $f{\in}I$ by lemma 2.5.12. Conversely, let $f{\in}I$. By part *b*) of lemma 2.5.11, there exists $g{\in}R\,[X]$ such that $f \overset{G}{\longrightarrow}_* g$ and g is top-minimal modulo G. If $g{=}0$ then we are done. If $g{\neq}0$ then

$$\ell\mathrm{m}(g) \notin \Big\langle \ell\mathrm{m}(G) \Big\rangle_{R\,[X]} \tag{1}$$

by lemma 2.5.9. By assumption, G is a Gröbner basis of I and hence

$$\Big\langle \ell\mathrm{m}(G) \Big\rangle_{R\,[X]} = \Big\langle \ell\mathrm{m}(I) \Big\rangle_{R\,[X]} \tag{2}$$

by definition 2.5.8 of a Gröbner basis in $R\,[X]$. On the other hand, by lemma 2.5.12, $f{-}g{\in}\langle G\rangle_{R\,[X]}$. Hence $g{\in}\langle G\rangle_{R\,[X]}$ because $f{\in}I$ by assumption. But then

$$\Big\langle \ell\mathrm{m}(G) \Big\rangle_{R\,[X]} \overset{(1)}{\not\ni} \ell\mathrm{m}(f) \in \Big\langle \ell\mathrm{m}(I) \Big\rangle_{R\,[X]} \overset{(2)}{=} \Big\langle \ell\mathrm{m}(G) \Big\rangle_{R\,[X]}$$

which is a contradiction.

b)→*a*) It is clear that

$$\Big\langle \ell\mathrm{m}(G) \Big\rangle_{R\,[X]} \subseteq \Big\langle \ell\mathrm{m}(I) \Big\rangle_{R\,[X]}.$$

To show the converse let $m{\in}\ell\mathrm{m}(I)$. Then there exists $f_m{\in}I\backslash\{0\}$ such that $\ell\mathrm{m}(f_m){=}m$. By assumption, $f_m \overset{G}{\longrightarrow}_* 0$ and then it is easy to see that f_m is top-reducible modulo F. Hence, by lemma 2.5.8,

$$m = \ell\mathrm{m}(f_m) \in \Big\langle \ell\mathrm{m}(G) \Big\rangle_{R\,[X]}.$$

Therefore

$$\ell\mathrm{m}(I) \subseteq \Big\langle \ell\mathrm{m}(G) \Big\rangle \text{ and hence } \Big\langle \ell\mathrm{m}(I) \Big\rangle_{R\,[X]} \subseteq \Big\langle \ell\mathrm{m}(G) \Big\rangle_{R\,[X]}. \qquad \square$$

Lemma 2.5.15. Let I be an ideal in $R\,[X]$ and $G{\subseteq}I\backslash\{0\}$ be a Gröbner basis of I. Then G is a basis of I:

$$\langle G\rangle_{R\,[X]} = I.$$

Proof. It is clear that $\langle G\rangle_{R[X]}\subseteq I$. On the other hand, take any $f\in I$. Then, by part b) of theorem 2.5.1, $f \xrightarrow{G}_* 0$ and hence, by lemma 2.5.12, $f\in\langle G\rangle_{R[X]}$. Therefore

$$I \subseteq \langle G\rangle_{R[X]}. \qquad \square$$

Gröbner bases in a polynomial algebra $P[X]$ over a principal ideal domain

The general theory of Gröbner bases in a polynomial algebra over a ring can be refined if the ring has particular properties like it is a domain or a principal ideal ring. Let P be a principal ideal ring. As usual, $P^*=P\setminus\{0\}$, $P[X]$ denotes the algebra of polynomials over P in finitely many indeterminates X, $\mathbb{T}\subsetneq P[X]$ is the set of terms in X, the elements of P are considered as polynomials in $P[X]$ due to the canonical embedding $P\subsetneq P[X]$ and called coefficients, $\mathbb{M}=\{\alpha\cdot t|\,\alpha\in P^*,\ t\in\mathbb{T}\}\subsetneq P[X]$ are monomials of $P[X]$.

In this subsection we construct the theory of Gröbner bases in the algebra of polynomial $P[X]$ as a generalization of the known ([Möl88] section 4, [AL94] section 4.5) theory of Gröbner bases in a polynomial algebra over a principal ideal domain.

The following definition introduces the refined counterpart of the notion of a Gröbner basis for the case of the algebra $P[X]$ which is, in fact, a direct analogue of the classical Gröbner basis. It has been considered in the literature for the case when coefficients form a principal ideal domain ([Möl88] section 4, [AL94] page 251, [Pan89] definition 1.3 of a "D-basis", [KRK88] for an Euclidean domain of coefficients).

Definition 2.5.9. Let I be an ideal in $P[X]$. A Gröbner basis $G\subseteq I\setminus\{0\}$ of the ideal I is called a **strong Gröbner basis** of I if and only if for any $f\in I\setminus\{0\}$ there is $g\in G$ such that the leading monomial of g divides the leading monomial of f: $\ell\mathrm{m}(g)\,|\,\ell\mathrm{m}(f)$.

The definition of a strong Gröbner basis does not require any specific properties of the ring of coefficients but the existence of a strong Gröbner basis for an arbitrary ideal in $P[X]$ requires P to be a principal ideal ring (see exercise 4.5.16 on page 259 in [AL94] for the case when P is a domain). The construction of a strong Gröbner basis from an arbitrary Gröbner basis in $P[X]$ (in the sense of definition 2.5.8) needs the following definition of a saturated set of terms (introduced in [Möl88], also [AL94] page 213). Note that saturated sets play in important role in the general theory of Gröbner bases for polynomials over an arbitrary Nötherian ring of coefficients: it is a key to generalize the notion of an S-polynomial and compute Gröbner bases using e.g. Möller's technique [Möl88] in a very general setup or in a special case of a principal ideal domain for coefficients where Möller's technique specializes to a direct analogue of the classical S-polynomial criterion and Buchberger's algorithm for computing Gröbner bases (see section 4 in [Möl88], also page 250 in [AL94]). It is quite straightforward to adopt the latter technique to the case of polynomials over a principal ideal ring but due to the fact that we do not use the actual computation of a Gröbner basis for an arbitrary set of polynomials over a ring, the instance of the $S-$polynomial criterion and Buchberger's algorithm in $P[X]$ are not considered to save space.

Recall that for any finite non-empty set of terms $T \subseteq \mathbb{T}$.

- The least common multiple of the terms of T is the term

$$\operatorname{lcm}(T) = \prod_{i=1}^{n} x_i^{d_i} \quad \text{with} \quad d_i = \max_{t \in T}\big(\deg_{x_i}(t)\big) \tag{lcm}$$

  which is the unique term such that

  a) for any $t \in T$ $t \,|\, \operatorname{lcm}(T)$, (|lcm)

  b) if $s \in \mathbb{T}$ is such that $t \,|\, s$ for all $t \in T$ then $\operatorname{lcm}(T) \,|\, s$. (lcm↓)

- The greatest common divisor of the terms of T is the term

$$\gcd(T) = \prod_{i=1}^{n} x_i^{d_i} \quad \text{with} \quad d_i = \min_{t \in T}\big(\deg_{x_i}(t)\big) \tag{gcd}$$

  which is the unique term such that

  a) for any $t \in T$ $\gcd(T) \,|\, t$, (gcd|)

  b) if $s \in \mathbb{T}$ is such that $s \,|\, t$ for all $t \in T$ then $s \,|\, \gcd(T)$. (↑gcd)

By convention, $\operatorname{lcm}(\emptyset) = \gcd(\emptyset) = 1$. It is easy to see that the properties of the least common multiple and the greatest common divisor are trivially valid for the case of the empty set.

Definition 2.5.10. Given a finite set $F \subseteq P[X] \setminus \{0\}$.

- Let $t \in \mathbb{T}$ be a term. The subset of all elements $h \in F$ such that $\ell\mathrm{t}(h) \,|\, t$ is called the $t-$**saturated subset in** F denoted by F_t:

$$F_t = \{h \in F \;\mid\; \ell\mathrm{t}(h) \,|\, t\} \;\subseteq\; F$$

- The subset $H{\subseteq}F$ **is saturated** in F if and only if there exists a term $t{\in}\mathbb{T}$ such that H is the $t-$saturated subset in F:

$$H{=}F_t \ \text{ with } \ t{\in}\mathbb{T}.$$

- Given an arbitrary subset $H{\subseteq}F$ in F. The **saturation of** H **in** F (denoted $\mathrm{Sat}_F\,(H)$) is the smallest (by inclusion of sets) superset of H saturated in F.

- The set $F_{\mathbb{T}}$ is the set of **all saturated subsets in** F:

$$F_{\mathbb{T}} = \{F_t \ \mid \ t{\in}\mathbb{T}\}\,.$$

Notation. Given a finite set $H{\subseteq}P[X]\backslash\{0\}$. Denote

$$\mathrm{L}\,(H) = \mathrm{lcm}\Big(\ell\mathrm{t}(H)\Big) \ \in \ \mathbb{T}, \quad \text{where } \ell\mathrm{t}(H) \text{ is a shortcut for } \ \big\{\ell\mathrm{t}(h) \mid h{\in}H\big\}.$$

Lemma 2.5.16. The notation $\mathrm{L}\,(H)$ considered as a map from finite sets of non-zero polynomials to terms has the following properties inherited from the least common multiple of terms:

$$\mathrm{L}\,(\emptyset) = 1; \tag{$\pounds L_{\emptyset}$}$$

for any finite sets $H_1, H_2{\subseteq}\mathcal{P}\backslash\{0\}$

$$\text{if } \ H_1 \subseteq H_2 \ \text{ then } \ L(H_1)\,|\,L(H_2), \tag{$\pounds L_{\subseteq}$}$$

$$L(H_1 \cup H_2) = \mathrm{lcm}\Big(L(H_1), L(H_2)\Big), \tag{$\pounds L_{\cup}$}$$

$$\text{if } \ L(H_2)|L(H_1) \ \text{ then } \ L(H_1 \cup H_2) = L(H_1). \tag{$\pounds L_{\sqsubset}$}$$

Proof.

($\pounds L_{\emptyset}$): By convention

$$\mathrm{L}\,(\emptyset) = \mathrm{lcm}\Big(\ell\mathrm{t}(\emptyset)\Big) = \mathrm{lcm}(\emptyset) = 1.$$

($\pounds L_{\subseteq}$): If $H_1{\subseteq}H_2$ then $h{\in}H_1$ follows $h{\in}H_2$ and hence $\ell\mathrm{t}(h)\,|\,\mathrm{lcm}\big(\ell\mathrm{t}(H_2)\big)$ by ($|$lcm) property of the least common multiple. Therefore, by the property (lcm$\downarrow$)of the least common multiple

$$\mathrm{L}\,(H_1) = \ \mathrm{lcm}\Big(\ell\mathrm{t}(H_1)\Big) \ \Big| \ \mathrm{lcm}\Big(\ell\mathrm{t}(H_2)\Big) \ = \mathrm{L}\,(H_2).$$

($\pounds L_{\subseteq}$): Take any $h{\in}H_1{\cup}H_2$. Then $h{\in}H_1$ or $h{\in}H_2$ In either case, also using the fact that $\mathrm{L}\,(H_1)\,|\,\mathrm{L}\,(H_1{\cup}H_2)$, $\mathrm{L}\,(H_2)\,|\,\mathrm{L}\,(H_1{\cup}H_2)$ by the (|lcm) property and the transitivity of the | relation,

$$\left.\begin{array}{c} h{\in}H_1 \overset{(|\mathrm{lcm})}{\Longrightarrow} \ell\mathrm{t}(h)\,|\,\mathrm{L}\,(H_1) \\ \vee \\ h{\in}H_2 \overset{(|\mathrm{lcm})}{\Longrightarrow} \ell\mathrm{t}(h)\,|\,\mathrm{L}\,(H_2) \end{array}\right\} \overset{\substack{(|\mathrm{lcm}),\\ (\text{transitivity of}\,|)}}{\Longrightarrow} \ell\mathrm{t}(h)\,|\,\mathrm{lcm}\Big(\mathrm{L}\,(H_1),\mathrm{L}\,(H_2)\Big). \quad (1)$$

On the other hand, for any term $s{\in}\mathbb{T}$ such that $\ell\mathrm{t}(h)\,|\,s$ for any $h{\in}H_1{\cup}H_2$

$$\left.\begin{array}{c} \Big(\forall h{\in}H_1 \quad \ell\mathrm{t}(h)\,|\,s\Big) \overset{(\mathrm{lcm}\downarrow)}{\Longrightarrow} \mathrm{L}\,(H_1)|s \\ \wedge \\ \Big(\forall h{\in}H_2 \quad \ell\mathrm{t}(h)\,|\,s\Big) \overset{(\mathrm{lcm}\downarrow)}{\Longrightarrow} \mathrm{L}\,(H_2)|s \end{array}\right\} \overset{(\mathrm{lcm}\downarrow)}{\Longrightarrow} \mathrm{lcm}\Big(\mathrm{L}\,(H_1),\mathrm{L}\,(H_2)\Big)\,|\,s. \quad (2)$$

According to (1) and (2) the term $\mathrm{lcm}\big(\mathrm{L}\,(H_1),\mathrm{L}\,(H_2)\big)$ satisfies the properties (|lcm) and (lcm↓) of the least common multiple of the leading terms of $H_1{\cup}H_2$ and hence, it is the unique term such that

$$\mathrm{lcm}\Big(\mathrm{L}\,(H_1),\mathrm{L}\,(H_2)\Big) = \mathrm{lcm}\Big(\ell\mathrm{t}(H_1{\cup}H_2)\Big) = \mathrm{L}\,(H_1{\cup}H_2).$$

($\pounds L_{\sqsubset}$): If $\mathrm{L}\,(H_2)|\mathrm{L}\,(H_1)$ then it is easy to see by the definition of the least common multiple that $\mathrm{lcm}\big(\mathrm{L}\,(H_2),\mathrm{L}\,(H_1)\big){=}\mathrm{L}\,(H_1)$. Then, by the property ($\pounds L_{\cup}$),

$$\mathrm{L}\,(H_1{\cup}H_2) \overset{(\pounds L_{\cup})}{=} \mathrm{lcm}\Big(\mathrm{L}\,(H_2),\mathrm{L}\,(H_1)\Big) = \mathrm{L}\,(H_1). \qquad \square$$

The next lemma describes the basic properties of the saturated subsets and saturations.

Lemma 2.5.17. Let $F{\subseteq}\mathcal{P}\backslash\{0\}$ be a finite set.

a) The intersection of subsets saturated in F is a subset saturated in F. Furthermore, given subsets F_{t_1} and F_{t_2} saturated in F with $t_1,t_2{\in}\mathbb{T}$

$$F_{t_1} \cap F_{t_2} = F_{\mathrm{gcd}(t_1,t_2)}.$$

b) For any $t_1,t_2{\in}\mathbb{T}$

$$t_1\,|\,t_2 \implies F_{t_1} \subseteq F_{t_2}.$$

c) Given a subset $H{\subseteq}F$ there exists the unique $\mathrm{Sat}_F\,(H)$ – the saturation of H in F. Furthermore,

$$\mathrm{Sat}_F\,(H) = F_{L(H)}$$

d) The saturation of any $H \subseteq F$ in F is a set saturated in F:

$$\mathrm{Sat}_F\Big(\mathrm{Sat}_F(H)\Big) = \mathrm{Sat}_F(H).$$

e) Given $H_1, H_2 \subseteq F$.

$$\mathrm{Sat}_F\Big(\mathrm{Sat}_F(H_1) \cup \mathrm{Sat}_F(H_2)\Big) = \mathrm{Sat}_F(H_1 \cup H_2)$$

f) Given a term $t \in \mathbb{T}$

$$F_t = F_{L(F_t)}.$$

Proof.

a) Take any $h \in F$. By the property ($\uparrow$gcd) of the gcd and transitivity of the $|$ relation,

$$\ell\mathrm{t}(h) \,|\, t_1 \wedge \ell\mathrm{t}(h) \,|\, t_2 \implies \ell\mathrm{t}(h) \,|\, \gcd(t_1, t_2) \qquad (*)$$

and

$$\ell\mathrm{t}(h) \,|\, t_2 \overset{\gcd(t_1,t_2)\,|\,t_2}{\Longleftarrow} \ell\mathrm{t}(h) \,|\, \gcd(t_1, t_2) \overset{\gcd(t_1,t_2)\,|\,t_1}{\Longrightarrow} \ell\mathrm{t}(h) \,|\, t_1. \qquad (**)$$

Then

$$h \in F_{t_1} \cap F_{t_2} \iff \ell\mathrm{t}(h) \,|\, t_1 \wedge \ell\mathrm{t}(h) \,|\, t_2$$
$$\overset{(*),(**)}{\iff} \ell\mathrm{t}(h) \,|\, \gcd(t_1, t_2) \iff h \in F_{\gcd(t_1,t_2)}.$$

b) Take $t_1, t_2 \in \mathbb{T}$ such that $t_1 \,|\, t_2$. Then, for any $h \in F$

$$h \in F_{t_1} \iff \ell\mathrm{t}(h) \,|\, t_1 \overset{\substack{(t_1 | t_2),\\ (\text{transitivity of} |)}}{\Longrightarrow} \ell\mathrm{t}(h) \,|\, t_2 \iff h \in F_{t_2}.$$

c) *Uniqueness.* Let F_{t_1} and F_{t_2} be saturations of H in F. By definition of the saturation $H \subseteq F_{t_1}$ and $H \subseteq F_{t_2}$. Then, by part *a*), the subset

$$\Big(H \subseteq\Big) F_{t_1} \cap F_{t_2} = F_{\gcd(t_1,t_2)}$$

is a superset of H saturated in F and thus $F_{t_1} = F_{\gcd(t_1,t_2)} = F_{t_2}$ because, by definition, the saturation of H is the least (by inclusion of sets) saturated subset in F which contains H.
Existence. For any $h \in F$

$$h \in H \implies \ell\mathrm{t}(h) \,|\, L(H) \iff h \in F_{L(H)}$$

and thus $H \subseteq F_{L(H)}$. Consider a subset S saturated in F such that $H \subseteq S$. Take $t \in \mathbb{T}$ such that $S = F_t$. Then, for any $h \in F$

$$h \in H \subseteq F_t \implies \ell\mathrm{t}(h) \,|\, t.$$

Thus, by the property (lcm↓) of the lcm,

$$\underbrace{\mathrm{lcm}\Big(\ell\mathrm{t}(H)\Big)}_{\substack{\| \\ L(H)}} \Big| \; t \quad \text{and, by part } b), \quad F_{L(H)} \subseteq F_t.$$

d) By definition of $\mathrm{Sat}_F(H)$, for any $h \in F$

$$h \in \mathrm{Sat}_F(H) \setminus H \subseteq \mathrm{Sat}_F(H) \implies \ell\mathrm{t}(h) \,|\, L(H).$$

Thus

$$\underbrace{L\Big(\mathrm{Sat}_F(H) \setminus H\Big)}_{\substack{\| \text{ Def. of } L \\ \mathrm{lcm}\Big(\{\ell\mathrm{t}(h) \mid h \in \mathrm{Sat}_F(H) \setminus H\}\Big)}} \Big| \; L(H)$$

and

$$L\Big(\mathrm{Sat}_F(H)\Big) = L\Big(\big(\mathrm{Sat}_F(H) \setminus H\big) \cup H\Big) \overset{(\pounds L_{\sqsubset})}{=} L(H). \qquad (*)$$

Then

$$\mathrm{Sat}_F\Big(\mathrm{Sat}_F(H)\Big) \overset{\text{part } c)}{=} F_{L\left(\mathrm{Sat}_F(H)\right)} \overset{(*)}{=} F_{L(H)} \overset{\text{part } c)}{=} \mathrm{Sat}_F(H).$$

e) From the identity

$$\boxed{L\Big(\mathrm{Sat}_F(H_1) \cup \mathrm{Sat}_F(H_2)\Big)} \overset{(\pounds L_{\cup})}{=}$$

$$\mathrm{lcm}\Big(L(\mathrm{Sat}_F(H_1)), L(\mathrm{Sat}_F(H_2))\Big) \overset{\text{part } d)}{=}$$

$$\mathrm{lcm}\Big(L(H_1), L(H_2)\Big) \overset{(\pounds L_{\cup})}{=} \boxed{L\Big(H_1 \cup H_2\Big)} \qquad (**)$$

and part *c*)

$$\mathrm{Sat}_F\Big(\mathrm{Sat}_F(H_1) \cup \mathrm{Sat}_F(H_2)\Big) \overset{\text{part } c)}{=} F_{L\left(\mathrm{Sat}_F(H_1) \cup \mathrm{Sat}_F(H_2)\right)} \overset{(**)}{=}$$

$$F_{L\left(H_1 \cup H_2\right)} \overset{\text{part } c)}{=} \mathrm{Sat}_F(H_1 \cup H_2).$$

$f)$ Take any $t \in T$. Note that the subset F_t is saturated in F. Then

$$F_t \overset{\text{part } d)}{=} \mathrm{Sat}_F(F_t) \overset{\text{part } c)}{=} F_{L(F_t)}.$$

□

Remarks (to lemma 2.5.17).

1. The converse of part $b)$ is not generally true. Indeed, consider

$$F = \{x, x^2\} \subseteq \Bbbk[x, y] \backslash \{0\}.$$

Then

$$\{x\} = F_{xy} \subseteq F_{x^2} = \{x, x^2\} = F \quad \text{but } xy \nmid x^2.$$

2. Given a finite set $F \subseteq P[X] \backslash \{0\}$ the union of subsets saturated in F is not necessary a subset saturated in F. As a counterexample consider

$$F = \{x, y, xy\} \subseteq \Bbbk[x, y] \backslash \{0\}.$$

Then

$$\underbrace{F_x}_{\substack{\| \\ \{x\}}} \cup \underbrace{F_y}_{\substack{\| \\ \{y\}}} = \boxed{\{x, y\} \subsetneq} \mathrm{Sat}_F(F_x \cup F_y) \overset{\text{part } e)}{=} \mathrm{Sat}_F(\{x, y\}) \overset{\text{part } c)}{=} F_{xy} = \boxed{\{x, y, xy\}} \Big[= F\Big].$$

and hence the union $\{x, y\} \subseteq F$ is not a subset saturated in F because, by part $e)$, $F_{xy} = F$ is the least (by inclusion of sets) superset of $\{x, y\}$ saturated in F.

The construction of a strong Gröbner basis

The construction of a strong Gröbner basis in $P[X]$ from an arbitrary Gröbner basis (in the sense of definition 2.5.8) $G \subseteq P[X] \backslash \{0\}$ can be roughly described as follows: a special polynomial (see definition 2.5.11) is constructed for each non-empty subset saturated in G, the union of all these polynomials is a strong Gröbner basis of $\langle G \rangle_{P[X]}$.

Definition 2.5.11. Given a finite non-empty set $\emptyset{\neq}H{\subseteq}P[X]\backslash\{0\}$ let $d(H){\in}P^*$ be a principal generator of the ideal generated by the leading coefficients of the elements of H:

$$\langle d(H)\rangle_P = \langle \ell\mathrm{c}(H)\rangle_P.$$

Then there are coefficients $\alpha_h{\in}P$ one for each $h{\in}H$ such that

$$d(H) = \sum_{h\in H} \alpha_h{\cdot}\ell\mathrm{c}(h).$$

The **principal polynomial of** H is a polynomial

$$p_H = \sum_{h\in H} \alpha_h{\cdot}\frac{L(H)}{\ell\mathrm{t}(h)}h \quad \text{with} \quad L(H) = \mathrm{lcm}\Big(\ell\mathrm{t}(H)\Big).$$

Note that the principal polynomial is not uniquely defined since it depends on the choice of $d(H)$ and coefficients α_h.

The principal polynomial has important for construction of a strong Gröbner basis properties summarized in the following lemma.

Lemma 2.5.18. Let $\emptyset{\neq}H{\subseteq}P[X]\backslash\{0\}$ be a finite set and $p_H{\in}P[X]$ be a principal polynomial of H. Then

a) $p_H \quad {\neq}0$,

b) $\ell\mathrm{t}(p_H){=}L(H) \in \mathbb{T}$,

c) $\ell\mathrm{c}(p_H){=}d(H){\in}P\backslash\{0\}$ where $\langle d(H)\rangle_P{=}\langle \ell\mathrm{c}(H)\rangle_P$.

Proof. Choose $d(H){\in}P^*$ – a principal generator of the P-ideal of leading coefficients of H. Set $L{=}L(H){\in}\mathbb{T}$ and $d{=}d(H)$. Then

$$L = \mathrm{lcm}\Big(\ell\mathrm{t}(H)\Big), \qquad \langle d\rangle_P = \langle \ell\mathrm{c}(H)\rangle_P.$$

Choose coefficients $\alpha_h{\in}P$ one for each $h{\in}H$ such that

$$d = \sum_{h\in H} \alpha_h{\cdot}\ell\mathrm{c}(h). \tag{1}$$

Denote

$$\tau_h = \frac{L}{\ell\mathrm{t}(h)} \quad \text{for each } h \in H. \tag{2}$$

Then

$$p = \sum_{h\in H} \alpha_h{\cdot}\tau_h h \tag{\dagger}$$

is a principal polynomial of H.

a) Consider the formal coefficient at the term L in p:

$$
\begin{aligned}
[\mathfrak{C}_p]\Big(L\Big) = [\mathfrak{C}_{\left(\sum\limits_{h\in H}\alpha_h\cdot\tau_h h\right)}]\Big(L\Big) &= \qquad \text{(by parts } a) \text{ and } b) \text{ of corollary 2.5.1)}\\
\sum_{h\in H}\alpha_h\cdot[\mathfrak{C}_{\left(\tau_h\cdot h\right)}]\Big(L\Big) &\overset{(2)}{=}\\
\sum_{h\in H}\alpha_h\cdot[\mathfrak{C}_{\left(\tau_h h\right)}]\Big(\tau_h\cdot\ell\mathrm{t}(h)\Big) &= \qquad \text{(by part } d) \text{ of corollary 2.5.1)}\\
\sum_{h\in H}\alpha_h\cdot[\mathfrak{C}_{(h)}]\Big(\ell\mathrm{t}(h)\Big) &= \qquad \text{(by definition of } \ell\mathrm{c})\\
\sum_{h\in H}\alpha_h\cdot\ell\mathrm{c}(h) &\overset{(1)}{=} d \neq 0. \qquad (*)
\end{aligned}
$$

Then

$$
[\mathfrak{C}_p]\Big(L\Big) = d \neq 0 \implies \mathfrak{C}_p\Big(L\Big) = d \implies
L \in \operatorname{supp}(p) \implies \operatorname{supp}(p) \neq \emptyset \implies p \neq 0.
$$

b) Since $d(H){\neq}0$ there must exist $h{\in}H$ such that $\alpha_h\cdot\ell\mathrm{c}(h){\neq}0$ and thus (strictly speaking according to part *a*) of lemma 2.5.5) $\alpha_h\cdot h{\neq}0$. Then the set H of polynomials $h{\in}H$ not annihilated by the corresponding a_h is not empty:

$$
V = \{h{\in}H \mid \alpha_h\cdot h{\neq}0\} \neq \emptyset. \tag{3}
$$

Furthermore, for each $h{\in}V$, by part *c*) of lemma 2.5.5,

$$
\alpha_h\cdot\tau_h\cdot h{\neq}0 \ \text{ and } \ \ell\mathrm{t}\Big(\alpha_h\cdot\tau_h\cdot h\Big) \leq \ell\mathrm{t}\Big(\tau_h\cdot h\Big) = \tau_h\cdot\ell\mathrm{t}(h) \overset{(2)}{=} L. \tag{4}
$$

By part *a*), $p{\neq}0$ and thus $\ell\mathrm{t}(p)$ is defined. Furthermore, in the proof of part *a*) we found that $L{\in}\operatorname{supp}(p)$. Hence

$$
\begin{aligned}
\boxed{L \leq \ell\mathrm{t}(p)} &\overset{(\dagger)}{=} \ell\mathrm{t}\left(\sum_{h\in H}\alpha_h\cdot\tau_h\cdot h\right) \overset{(3)}{=}\\
&\ell\mathrm{t}\left(\sum_{h\in V}\alpha_h\cdot\tau_h\cdot h\right) \boxed{\overset{(4)}{\leq}} \max\Big(\{\ell\mathrm{t}(\tau_h\cdot h) \mid h{\in}V\}\Big) =\\
&\max\Big(\{\underbrace{\tau_h\cdot\ell\mathrm{t}(h)}_{\overset{(2)}{\|}\; L} \mid h{\in}V\}\Big) = \boxed{L}.
\end{aligned}
$$

c) In the proof of part a) we found that the coefficient at L in p is d. On the other hand, by part b), L is the leading term. Hence

$$\ell\mathrm{c}(p) = \mathfrak{C}_p\big(\ell\mathrm{t}(p)\big) = \mathfrak{C}_p(L) = d.$$

$\square$

The next theorem provides an idea how to construct a strong Gröbner basis.

Theorem 2.5.2. Let I be an ideal in $P[X]$ and $G \subseteq P[X] \backslash \{0\}$ be a Gröbner basis of I. For each non-empty subset $S \in G_{\mathbb{T}} \backslash \{\emptyset\}$ saturated in G choose p_S – a principal polynomial of S. Then the set

$$\overline{G} = \{p_S \mid S \in G_{\mathbb{T}} \backslash \{\emptyset\}\}.$$

is a strong Gröbner basis of I.

Proof. It is clear by the construction of a principal polynomial and part a) of lemma 2.5.18 that $p_S \in I \backslash \{0\}$ for any $S \in G_{\mathbb{T}} \backslash \{\emptyset\}$ and hence $\overline{G} \subseteq I \backslash \{0\}$. Take $f \in I \backslash \{0\}$. Since G is a Gröbner basis of I, the polynomial f is top-reducible modulo G by part b) of theorem 2.5.1. Then, by lemma 2.5.8,

$$\ell\mathrm{m}(f) \in \langle \ell\mathrm{m}(G) \rangle_{P[X]}.$$

and hence, by corollary 2.5.3, there is a subset $\emptyset \neq H \subseteq G$ and $\alpha_h \in P \backslash \{0\}$, $t_h \in \mathbb{T}$ one for each $h \in H$ such that

$$\ell\mathrm{m}(f) = \sum_{h \in H} \alpha_h \cdot t_h \cdot \ell\mathrm{m}(h) \quad \text{with } t_h \cdot \ell\mathrm{t}(h) = \ell\mathrm{t}(f) \quad \text{for each } h \in H.$$

The latter condition on leading terms of H follows, by the property (lcm$\downarrow$) of the least common multiple, that there exists the term $\tau \in \mathbb{T}$ such that

$$\tau \underbrace{\mathrm{L}\,(H)}_{\substack{\| \text{ by definition of } L \\ \mathrm{lcm}\left(\ell\mathrm{t}(H)\right)}} = \ell\mathrm{t}(f). \tag{1}$$

and thus there are terms $t_h \in \mathbb{T}$ one for each $h \in \mathrm{Sat}_F\,(H) \backslash H$ such that

$$t_h \cdot \ell\mathrm{t}(h) = \ell\mathrm{t}(f) \quad \text{for all } h \in \mathrm{Sat}_F\,(H) \backslash H$$

because, by part f) of lemma 2.5.17, $\mathrm{Sat}_F(H) = F_{\mathrm{L}(H)}$ – the saturation of H in F is the set of all polynomials of F whose leading term divides $\mathrm{L}(H)$. Define $\alpha_h = 0$ for all $h \in \mathrm{Sat}_F(H) \setminus H$. Then

$$\ell\mathrm{m}(f) = \sum_{h \in \mathrm{Sat}_F(H)} \alpha_h \cdot t_h \cdot \ell\mathrm{m}(h) \quad \text{and } t_h \ell\mathrm{t}(h) = \ell\mathrm{t}(f) \quad \text{for } h \in \mathrm{Sat}_F(H). \tag{$*$}$$

Since P is a principal ideal ring there exists $d \in P \setminus \{0\}$ such that

$$\langle d \rangle_P = \Big\langle \ell\mathrm{c}\big(\mathrm{Sat}_F(H)\big) \Big\rangle_P$$

and thus there exist coefficients $\gamma_h \in P \setminus \{0\}$ one for each $h \in \mathrm{Sat}_F(H)$ such that

$$\gamma_h d = \ell\mathrm{c}(h) \qquad \text{for all} \quad h \in \mathrm{Sat}_F(H). \tag{2}$$

Take $p \neq 0$ – a principal polynomial of $\mathrm{Sat}_F(H)$ (here we tacitly use the condition $\mathrm{Sat}_F(H) \neq \emptyset$ to have $p \neq 0$ by part a) of lemma 2.5.18) such that

$$\ell\mathrm{c}(p) = d \qquad \text{and} \qquad \ell\mathrm{t}(p) = L(H) \tag{3}$$

provided by lemma 2.5.18. W.l.o.g. assume that $p \in \overline{G}$ (this stipulation is required because strictly speaking the principal polynomial p depends on the choice of d). Then the sum $(*)$ is equivalent to

$$\ell\mathrm{m}(f) = \sum_{h \in \mathrm{Sat}_F(H)} \alpha_h \cdot t_h \cdot \ell\mathrm{m}(h) = \sum_{h \in \mathrm{Sat}_F(H)} \alpha_h \overbrace{\ell\mathrm{c}(h)}^{\gamma_h d \;\overset{(2)}{=}} \cdot \overbrace{t_h \cdot \ell\mathrm{t}(h)}^{\ell\mathrm{t}(f) \overset{(1)}{=} \tau L(H) \;\overset{(*)}{=}} =$$

$$\sum_{h \in \mathrm{Sat}_F(H)} \alpha_h \gamma_h \cdot \tau \cdot \underbrace{d}_{\overset{(3)}{=}\,\ell\mathrm{c}(p)} \cdot \underbrace{L(H)}_{=\,\ell\mathrm{t}(p)} = \left(\sum_{h \in \mathrm{Sat}_F(H)} \alpha_h \gamma_h \right) \cdot \tau \cdot \ell\mathrm{m}(p).$$

Thus $\ell\mathrm{m}(p) \,|\, \ell\mathrm{m}(f)$ with $p \in \overline{G}$. Since f is arbitrary chosen, $\overline{G}$ is a strong Gröbner basis of the ideal I. ☐

Corollary 2.5.4. Given an ideal I in $P[X]$ and a Gröbner basis G of I there exists an algorithm to compute a strong Gröbner basis $\overline{G}$ of I in finitely many steps starting from G.

Proof. By convention G is a finite set. Then clearly there are finitely many different subsets and, in particular, finitely many different saturated subsets in G and thus it is sufficient to treat finitely many principal polynomials one for each saturated subset in G to satisfy the condition of theorem 2.5.2 and get a finite strong Gröbner basis $\overline{G}$ of I.

With few minor refinements the reasoning above can be turned to an algorithm to compute a strong Gröbner basis starting from an arbitrary Gröbner basis of the ideal I. Indeed, denote

$$L = \{L(H) \mid \emptyset \neq H \subseteq G \text{ such that } \operatorname{Sat}_G(H) \neq \emptyset\} \subseteq \mathbb{T}$$

the set of all possible least common multiples of the leading terms of non-empty subsets saturated in G. Clearly L is a finite set. It is an easy consequence from part $f)$ of lemma 2.5.17 that

$$G_{\mathbb{T}} \backslash \{\emptyset\} = \{G_t \mid t{\in}L\}$$

which means that all non-empty subsets saturated in G have the form G_t with $t{\in}L$. The latter construction admits a further fine refinement: for each subset $S{\in}G_{\mathbb{T}}\backslash\{\emptyset\}$ choose a representative $t_S{\in}L$ such that $S{=}G_t$. Denote

$$\overline{L} \subseteq L \tag{$*$}$$

the (finite and non-empty) set of these representatives. Then the following algorithm 2.5.1 computes a strong Gröbner basis $\overline{G}$ of I starting from G.

Algorithm 2.5.1 An algorithm to compute a strong Gröbner basis

Require: G is a Gröbner basis of the ideal I
Ensure: $\overline{G}$ is a strong Gröbner basis of I.
  $\overline{L}$:= the set $(*)$ in the proof of corollary 2.5.4
  $\overline{G} := \emptyset$
  for all $t \in \overline{L} \subseteq \mathbb{T}$ **do**
    choose p_{G_t} – a principal polynomial for the subset G_t saturated in G
    $\overline{G} := \overline{G} \cup \{p_{G_t}\}$
  end for

□

The following definition presents a very particular form of the representation of a polynomial in terms of a strong Gröbner basis which is used as a technical tool in the next subsection.

Definition 2.5.12. Let $f{\in}P[X]\backslash\{0\}$ and $G{\subseteq}P[X]\backslash\{0\}$. A strong monomial Gröbner representation of f in terms of G is a representation of f as a sum in terms of G with monomial co-factors and pairwise different products of leading terms: there is $k{\in}\mathbb{N}$ and monomials $q_i{\in}\mathbb{M}$ one for each $1{\leq}i{\leq}k$ such that

$$f = \sum_{i=1}^{k} q_i{\cdot}g_i;\ g_i{\in}G,\ \ \ell\mathrm{t}(q_i){\cdot}\ell\mathrm{t}(g_i){\leq}\ell\mathrm{t}(f) \text{ and}$$
$$i{<}j \implies \ell\mathrm{t}(q_i){\cdot}\ell\mathrm{t}(g_i) < \ell\mathrm{t}(q_j){\cdot}\ell\mathrm{t}(g_j) \text{ for } i,j{\in}\{1,\dots,k\} \subseteq \mathbb{N}.$$

By convention, the empty sum is the trivial monomial Gröbner representation of 0 in terms of G.

Theorem 2.5.3. Let $G{\subseteq}P[X]\backslash\{0\}$ be a strong Gröbner basis of the ideal I in $P[X]$. Then for any $f{\in}I\backslash\{0\}$ there exists a strong monomial Gröbner representation of f in terms of G.

Proof. Take any $f{\in}I\backslash\{0\}$. Proceed by induction on the leading term $\ell\mathrm{t}(f)$ w.r.t. $<$. The inductive argument is applicable because $<$ is a well-ordering. Assume there exists a strong monomial Gröbner representation in terms of G of any $f'{\in}I\backslash\{0\}$ with $\ell\mathrm{t}(f'){\underset{\mathbb{T}}{<}}\ell\mathrm{t}(f)$. Since G is a strong Gröbner basis there exists $g{\in}G$ such that $\ell\mathrm{m}(g)|\ell\mathrm{m}(f)$. Then

$$f_< = f - \frac{\ell\mathrm{m}(f)}{\ell\mathrm{m}(g)}{\cdot}g \quad \text{and} \quad \ell\mathrm{m}(f) = \frac{\ell\mathrm{m}(f)}{\ell\mathrm{m}(g)}{\cdot}\ell\mathrm{m}(g). \tag{$\dagger$}$$

It is clear that

$$f \xrightarrow{g} f_<$$

by definition 2.5.4 of one step top-reduction. Hence, by lemma 2.5.10,

$$f_<{=}0 \ \text{ or } \ \ell\mathrm{t}(f_<){<}\ell\mathrm{t}(f). \tag{$*$}$$

If $f_<{=}0$ then $\frac{\ell\mathrm{m}(f)}{\ell\mathrm{m}(g)}{\cdot}g$ is clearly the desired strong monomial Gröbner representation of f in terms of G. Otherwise consider a strong monomial Gröbner representation of $f_<$ in terms of G provided by the inductive assumption

$$f_< = \sum_{i=1}^{k} q_i{\cdot}g_i;\ q_i{\in}\mathbb{M},\ g_i{\in}G,\ \ell\mathrm{t}(q_i){\cdot}\ell\mathrm{t}(g_i){\leq}\ell\mathrm{t}(f_<) \text{ and} \tag{$**$}$$
$$i{<}j \implies \ell\mathrm{t}(q_i){\cdot}\ell\mathrm{t}(g_i) < \ell\mathrm{t}(q_j){\cdot}\ell\mathrm{t}(g_j) \ \text{ for } \ i,j{\in}\{1,\dots,k\} \subseteq \mathbb{N}.$$

Set $q_{k+1}=\frac{\ell\mathrm{m}(f)}{\ell\mathrm{m}(g)}\in\mathbb{M}$, $g_{k+1}=g\in G$. Then, by construction, the representation $f=\sum_{i=1}^{k+1} q_i{\cdot}g_i$ is a strong monomial Gröbner representation of f in terms of G. Indeed, for any $i,j\in\{1\dots k{+}1\}$ such that $i<j$:

$$\begin{array}{ll}
\text{a) } i < k{+}1 \implies & \ell\mathrm{t}(q_i){\cdot}\ell\mathrm{t}(g_i) \overset{(**)}{\leq} \ell\mathrm{t}(f_<) \overset{(*)}{<} \ell\mathrm{t}(f);\\
\text{b) } i = k{+}1 \implies & \ell\mathrm{t}(q_{k+1}){\cdot}\ell\mathrm{t}(g_{k+1}) \overset{(\dagger)}{=} \ell\mathrm{t}(f);\\
\text{c) } j < k{+}1 \implies & \ell\mathrm{t}(q_i){\cdot}\ell\mathrm{t}(g_i) \overset{(**)}{<} \ell\mathrm{t}(q_j){\cdot}\ell\mathrm{t}(g_j);\\
\text{d) } j = k{+}1 \implies & \\
\quad i < k{+}1 \implies & \ell\mathrm{t}(q_i){\cdot}\ell\mathrm{t}(g_i) \overset{a)}{<} \ell\mathrm{t}(f) \overset{b)}{=} \ell\mathrm{t}(q_{k+1}){\cdot}\ell\mathrm{t}(g_{k+1}).
\end{array}$$ □

The specialization of a Gröbner basis in $P[X]$

Definition 2.5.13. Let K be an arbitrary ideal in P. The **specialization homomorphism** modulo K is a map

$$\phi_K : P[X] \longrightarrow \left({}^{P}/_{K}\right)[X]$$

from $P[X]$ to the ring of polynomials $\left({}^{P}/_{K}\right)[X]$ over the ring ${}^{P}/_{K}$ in the same indeterminates X and the same term order $<$ defined as the (unique) homomorphic extension of the canonical ring epimorphism

$$\psi_K : \begin{cases} P \longrightarrow {}^{P}/_{K} \\ \alpha \longrightarrow \alpha + K \end{cases}$$

such that $\phi_K(x)=x$ for any $x\in X$ provided by the universal property (see e.g. [BW93] lemma 2.15 on page 72) of polynomial rings: for any

$$f = \sum_{\mathrm{supp}(f)} c_t{\cdot}t \in P[X] \quad \text{with} \quad c_t\in P^* \quad \text{for each} \quad t\in \mathrm{supp}\,(f) \subseteq\mathbb{T}$$

the image of f under ϕ_K is

$$\phi_K(f) = \sum_{t\in\mathrm{supp}(f)} \psi_K(c_t){\cdot}t \quad \left[\quad = \sum_{\substack{t\in\mathrm{supp}(f)\\ \psi_K(c_t)\neq 0}} \psi_K(c_t){\cdot}t \right] \in \left({}^{P}/_{K}\right)[X].$$

It is clear by construction that ϕ_K is a surjective homomorphism of rings which trivially preserves the term order $<$: for any $t_1, t_2\in\mathbb{T}\subsetneq P[X]$

$$t_1<t_2 \quad \Longleftrightarrow \quad \left({}^{P}/_{K}\right)[X]\supsetneq\mathbb{T}\ni \quad \phi_K(t_1)=t_1<t_2=\phi_K(t_2) \quad \in\mathbb{T}\subsetneq \left({}^{P}/_{K}\right)[X].$$

The specialization of a Gröbner basis in $P[X]$ addresses the problem whether the image $\phi_K(G)$ (K is an ideal in P) of a Gröbner basis $G \subseteq P[X]$ is a Gröbner basis in $(P/K)[X]$ w.r.t. the same term order on $\mathbb{T}$. There is a series of results in the literature which deal with Gröbner bases in a polynomial ring over a principal ideal domain (see [Grä93], [Pau92], [Win88]) or over an arbitrary Nötherian ring (see [Kal97]) and give a sufficient condition for the image of a Gröbner basis to be a Gröbner basis under specialization modulo a maximal, prime or a primary ideal. We use the theoretical existence statement of a strong Gröbner basis for an arbitrary Gröbner basis in $P[X]$ and then follow the same pattern as in ([Pau92]) to state the condition on a (not necessary strong) Gröbner basis G in $P[X]$ to be a Gröbner basis $\phi_K(G)$ in $(P/K)[X]$ w.r.t. the same term order under the specialization ϕ_K modulo a maximal ideal $K \lhd P$.

Lemma 2.5.19. Let $K \lhd P$ be an ideal in P and $I \lhd P[X]$ be an ideal in $P[X]$. If there exists a Gröbner basis $G \subseteq I \backslash \{0\}$ of I such that $\ell c(G) \cap K = \emptyset$ then there exists a strong Gröbner basis $\overline{G} \subseteq I \backslash \{0\}$ such that $\ell c(\overline{G}) \cap K = \emptyset$.

Proof. Consider the set

$$\overline{G} = \left\{ p_S \mid S \in G_{\mathbb{T}} \backslash \{\emptyset\} \right\},$$

where p_S is a principal polynomial of a corresponding non-empty set $S \subseteq G$ saturated in G for each $S \in G_{\mathbb{T}} \backslash \{\emptyset\}$. By theorem 2.5.2, $\overline{G} \subseteq I \backslash \{0\}$ and $\overline{G}$ is a strong Gröbner basis of I. Take any $S \in G_{\mathbb{T}} \backslash \{\emptyset\}$. Then, by part c) of lemma 2.5.18,

$$\left\langle \ell c(p_S) \right\rangle_P = \left\langle \ell c(S) \right\rangle_P. \tag{$*$}$$

By assumption, $S \neq \emptyset$. Take any $h \in S \subseteq G$. By $(*)$, there exists $\alpha \in P^*$ such that $\alpha \cdot \ell c(p_S) = \ell c(h)$ and $\ell c(h) \notin K$ by assumption. Then it is clear that $\ell c(p_S) \notin K$. Since S is an arbitrary element of $G_{\mathbb{T}} \backslash \{\emptyset\}$,

$$\ell c(\overline{G}) \cap K = \emptyset. \qquad \square$$

Theorem 2.5.4 (the specialization of a Gröbner basis). Let $K \lhd P$ be a *maximal* ideal in P and $I \lhd P[X]$ be an ideal in $P[X]$. If $G \subseteq I \backslash \{0\}$ is a Gröbner basis of I such that $\ell c(G) \cap K = \emptyset$ then the image $\phi(G)$ of G under the specialization homomorphism $\phi := \phi_K$ modulo K is a Gröbner basis in $\left({}^{P}/_{K} \right)[X]$ of the image $\phi(I)$ of the ideal I under ϕ w.r.t. the same term order $<$.

Lemma 2.5.20. With terms and notation of theorem 2.5.4.

a) Let $m \in \mathbb{M} \subsetneq P[X]$ be a monomial in $P[X]$. Then

$$\phi(m) \neq 0 \iff \phi\Big(\mathfrak{C}(m)\Big) \neq 0. \tag{a}$$

b) Let $h \in P[X] \setminus \{0\}$ be such that $\phi\Big(\ell\mathrm{m}_X\,(h)\Big) \neq 0$. Then

$$\phi(h) \neq 0 \ \text{ and } \ \ell\mathrm{t}_X\,(h) = \ell\mathrm{t}\Big(\phi(h)\Big) \in \mathbb{T}. \tag{b}$$

c) Let m_1 and m_2 be monomials in $P[X]$. Then

$$\phi(m_1 m_2) = 0 \iff \phi\Big(\mathfrak{C}(m_1)\,\mathfrak{C}(m_2)\Big) = 0. \tag{c}$$

Proof.

a) By the definition of ϕ

$$\phi(m) = \phi\Big(\mathfrak{C}(m)\Big) \cdot \ell\mathrm{t}(h).$$

Thus

$$\phi(m) \neq 0 \iff \phi\Big(\mathfrak{C}(m)\Big) \cdot \ell\mathrm{t}(h) \neq 0 \iff \phi\Big(\mathfrak{C}(m)\Big) \neq 0. \tag{1}$$

b) Note that $\mathfrak{C}\big(\ell\mathrm{m}(h)\big) = \ell\mathrm{c}(h)$. Then, by part *a*),

$$\phi\Big(\ell\mathrm{m}(h)\Big) \neq 0 \overset{(1)}{\iff} \phi\Big(\ell\mathrm{c}(h)\Big) \neq 0 \implies$$
$$\ell\mathrm{t}(h) \in \mathrm{supp}\,\Big(\phi(h)\Big) \implies \mathrm{supp}\,\Big(\phi(h)\Big) \neq \emptyset \implies \phi(h) \neq 0.$$

Hence $\ell\mathrm{t}\big(\phi(h)\big)$ is defined and

$$\ell\mathrm{t}(h) \leq \max\Big(\mathrm{supp}\,\big(\phi(h)\big)\Big) = \ell\mathrm{t}\Big(\phi(h)\Big). \tag{2}$$

On the other hand, according to definition of the specialization homomorphism ϕ

$$\phi(h) = \phi\Big(\sum_{t \in \mathrm{supp}(h)} \mathfrak{C}_t(f) \cdot t\Big) = \sum_{t \in \mathrm{supp}(h)} \phi\Big(\mathfrak{C}_t(f)\Big) \cdot t = \sum_{\substack{t \in \mathrm{supp}(h) \\ \phi(\mathfrak{C}_t(f)) \neq 0}} \phi\Big(\mathfrak{C}_t(f)\Big) \cdot t.$$

Thus

$$\mathrm{supp}\,\Big(\phi(h)\Big) = \big\{t \in \mathrm{supp}\,(h) \mid \phi\big(\mathfrak{C}_t(f)\big) \neq 0\big\} \subseteq \mathrm{supp}\,(h)\,.$$

and

$$\ell\mathrm{t}\Big(\phi(h)\Big) = \max\Big(\mathrm{supp}\,(\phi(h))\Big) \leq \max\Big(\mathrm{supp}\,(h)\Big) = \ell\mathrm{t}(h). \qquad (3)$$

From (2) and (3)

$$\ell\mathrm{t}(h) \overset{(2)}{\leq} \ell\mathrm{t}\Big(\phi(h)\Big) \overset{(3)}{\leq} \ell\mathrm{t}(h) \quad \Longrightarrow \quad \ell\mathrm{t}(h) = \ell\mathrm{t}\Big(\phi(h)\Big).$$

c) Consider 2 cases.

c_1) Assume $m_1m_2{=}0$. Then, by construction of $P[X]$, $\mathfrak{C}(m_1)\,\mathfrak{C}(m_2){=}0$ and, since ϕ is a homomorphism,

$$\phi(m_1m_2) = 0 \;\text{ and }\; \phi\Big(\mathfrak{C}(m_1)\,\mathfrak{C}(m_2)\Big) = 0.$$

c_2) Assume $m_1m_2{\neq}0$. Then it is clear by construction of $P[X]$ that

$$\ell\mathrm{m}(m_1m_2) = m_1m_2,\;\; \ell\mathrm{c}(m_1m_2) = \mathfrak{C}(m_1m_2) = \mathfrak{C}(m_1)\,\mathfrak{C}(m_2)$$

and, by part *a*),

$$0 = \phi(m_1m_2) = \phi\Big(\ell\mathrm{m}(m_1m_2)\Big) \overset{(a)}{\Longleftrightarrow}$$
$$\phi\Big(\ell\mathrm{c}(m_1m_2)\Big) = \phi\Big(\mathfrak{C}(m_1)\,\mathfrak{C}(m_2)\Big) = 0. \quad \square$$

Proof of theorem 2.5.4. It is well-known that a maximal ideal is prime. Hence the ideal K has the following property

$$a{\cdot}b{\in}K \;\text{ if and only if }\; a{\in}K \;\text{ or }\; b{\in}K. \qquad (1)$$

Let

$$G \subseteq \;\; \widehat{G} = \overline{G} \cap G \;\; \subseteq I\backslash\{0\},$$

where $\overline{G}{\subseteq}I\backslash\{0\}$ is a strong Gröbner basis of I such that $\ell\mathrm{c}\big(\overline{G}\big){\cap}K{=}\emptyset$ provided by lemma 2.5.19 due to the assumption $\ell\mathrm{c}(G){\cap}K{=}\emptyset$. Then it is easy to see by definition 2.5.9 of a strong Gröbner basis that $\widehat{G}$ is a strong Gröbner basis of I. It is clear by construction that

$$\ell\mathrm{c}(\widehat{G}){\cap}K = \emptyset \qquad (2)$$

and then, by lemma 2.5.20, for any $g{\in}\widehat{G}$

$$\phi\Big(\ell\mathrm{c}(g)\Big) \neq 0 \overset{(a)}{\Longleftrightarrow} \phi\Big(\ell\mathrm{m}(g)\Big) \neq 0 \overset{(b)}{\Longrightarrow}$$
$$\phi(g) \neq 0 \quad \text{and} \quad \ell\mathrm{t}(g) = \ell\mathrm{t}\Big(\phi(g)\Big). \quad (3)$$

Denote $\overline{P}={}^{P}/_{K}$ and $\overline{\mathbb{M}}=\{\bar{\alpha}\cdot t \mid \bar{\alpha}\in\overline{P}, t\in\mathbb{T}\}\subsetneq\overline{P}\,[X]$ – the monomials of $\overline{P}\,[X]$. It is known ([Lan92] page 93) that a quotient-ring modulo a maximal ideal is a field and hence the theory of Gröbner bases in $\overline{P}\,[X]$ is the classical Gröbner bases theory.

Denote $I_o=\phi(I)$, $\widehat{G}_o=\phi(\widehat{G})$ – the images of I and $\widehat{G}$ under ϕ. Since the homomorphic image I_o of the ideal I is the ideal (no extra closure is required) it is sufficient to show that every element of $f_o\in I_o$ has a Gröbner representation in terms of $\widehat{G}_o$ of the form (cf. definition 5.59 on page 218 in [BW93])

$$f_o = \sum_{i\in N_f} \overline{m}_i\cdot\overline{g}_i \quad \text{with} \quad \begin{array}{l} N_f \text{ - a finite subset of } \mathbb{N}, \\ \overline{m}_i\in\overline{\mathbb{M}},\ \overline{g}_i\in\widehat{G}_o \ \text{ and} \\ \mathfrak{T}(\overline{m}_i)\cdot\ell\mathrm{t}(\overline{g}_i) \leq \ell\mathrm{t}(f_o) \\ \text{for each } \ i\in N_f. \end{array} \tag{$\dagger$}$$

provided that, by (3), $0\notin\overline{G}_o$. If it is true then G_o is a (classical) Gröbner basis in $\overline{P}\,[X]$, e.g. by theorem 5.62 on page 219 in [BW93], provided that $\overline{P}$ is a field.

Take any $f_o\in I_o$ together with the corresponding pre-image $f\in I$ of f_o under the specialization homomorphism:

$$f_o=\phi(f),$$

and show that there exists a Gröbner representation of f_o of the form (†) in terms of $\widehat{G}_o$. The case $f_o=0$ is trivial. Assume $f_o\neq 0$. Since $\widehat{G}$ is a strong Gröbner basis of the ideal I in $P[X]$, by theorem 2.5.3, there exists a strong monomial Gröbner representation of f in terms of $\widehat{G}$ i.e. there is $k\in\mathbb{N}$ and monomials $q_i\in\mathbb{M}$ one for each $1\leq i\leq k$ such that

$$f = \sum_{i=1}^{k} q_i\cdot g_i \quad \text{where} \quad \begin{array}{l} g_i\in\widehat{G},\ \mathfrak{T}(q_i)\cdot\ell\mathrm{t}(g_i) \leq \ell\mathrm{t}(f) \text{ and} \\ i<j \implies \mathfrak{T}(q_i)\cdot\ell\mathrm{t}(g_i) < \mathfrak{T}(q_j)\cdot\ell\mathrm{t}(g_j) \\ \text{for all } \ i,j\in\{1,\ldots,k\}\subseteq\mathbb{N}. \end{array} \tag{$*$}$$

Our next goal is to ascertain that the image of both sides of the Gröbner representation $(*)$ under the specialization homomorphism ϕ can be turned to a Gröbner representation of f_o in terms of $\widehat{G}_o$. The first step is to show that each monomial "shift" $q_i\cdot g_i$ either vanishes completely or retains the leading term under the action of ϕ.

Take $i\in\{1,\ldots,k\}$. Assume that $\phi\Big(q_i\cdot\ell\mathrm{m}(g_i)\Big)=0$. Then

$$\phi\Big(q_i\cdot\ell\mathrm{m}(g_i)\Big) = 0 \overset{(c)}{\iff} \phi\Big(\mathfrak{C}(q_i)\,\ell\mathrm{c}(g_i)\Big) = 0 \iff$$
$$\mathfrak{C}(q_i)\,\ell\mathrm{c}(g_i) \in K \overset{(1),(2)}{\Longrightarrow} \mathfrak{C}(q_i)\in K \iff \phi\Big(\mathfrak{C}(q_i)\Big) = 0.$$

and hence

$$\phi(q_i\cdot g_i) = \phi(q_i)\cdot\phi(g_i) \overset{(a)}{=} 0\cdot\phi(g_i) = 0.$$

The latter observation can be summarized as

$$\phi\Big(q_i\cdot\ell\mathrm{m}(g_i)\Big) = 0 \quad\Longrightarrow\quad \phi(q_i\cdot g_i) = 0. \tag{4}$$

On the other hand assume that $\phi\Big(q_i\ell\mathrm{m}_X\,(g_i)\Big)\neq 0$. Then, using part c) of lemma 2.5.5,

$$\phi\Big(q_i\ell\mathrm{m}(g_i)\Big) \neq 0 \overset{\phi \text{ is homo}}{\Longrightarrow} q_i\ell\mathrm{m}(g_i) \neq 0 \overset{(L2.5.5)}{\Longrightarrow}$$
$$q_i\cdot g_i \neq 0 \text{ and } \ell\mathrm{m}(q_ig_i) = q_i\ell\mathrm{m}(g_i) \Longrightarrow$$
$$\phi\Big(\ell\mathrm{m}(q_ig_i)\Big) = \phi\Big(q_i\ell\mathrm{m}(g_i)\Big) \neq 0 \overset{(b)}{\Longrightarrow}$$
$$\phi(q_i\cdot g_i) \neq 0 \text{ and } \ell\mathrm{t}(q_i\cdot g_i) = \ell\mathrm{t}\Big(\phi(q_i\cdot g_i)\Big).$$

Let us analyze the last equation carefully. Since ϕ is a homomorphism

$$\phi(q_i\cdot g_i) = \phi(q_i)\cdot\phi(g_i) \text{ and hence } \ell\mathrm{t}\Big(\phi(q_i\cdot g_i)\Big) = \ell\mathrm{t}\Big(\phi(q_i)\cdot\phi(g_i)\Big)$$

where the elements $\phi(q_i),\phi(g_i)\in\overline{P}\,[X]$. Note a well-known fact from the classical theory of Gröbner bases that the leading term function w.r.t. any term order is always monotonous under multiplication of non-zero elements in $\overline{P}\,[X]$ as it is well-known from the classical Gröbner bases theory provided that $\overline{P}$ is a field. Furthermore $\phi(q_i)\neq 0$ because $\phi(q_i)\cdot\phi(g_i)=\phi(q_i\cdot g_i)\neq 0$ and then

$$\phi(q_i) \neq 0 \overset{(b)}{\Longrightarrow} \mathfrak{T}\Big(\phi(q_i)\Big) = \ell\mathrm{t}\Big(\phi(q_i)\Big) = \ell\mathrm{t}(q_i) = \mathfrak{T}(q_i) \text{ and}$$
$$\phi(g_i) \overset{(3)}{\neq} 0 \overset{(b)}{\Longrightarrow} \ell\mathrm{t}\Big(\phi(g_i)\Big) = \ell\mathrm{t}(g_i) \qquad \text{follows}$$
$$\ell\mathrm{t}\Big(\phi(q_i)\cdot\phi(g_i)\Big) = \ell\mathrm{t}\Big(\phi(q_i)\Big)\cdot\ell\mathrm{t}\Big(\phi(g_i)\Big) = \mathfrak{T}(q_i)\cdot\ell\mathrm{t}(g_i).$$

Collecting the last statements together, the rule is

$$
\begin{aligned}
\phi\Big(q_i \ell\mathrm{m}(g_i)\Big) \neq 0 \quad \Longrightarrow \quad & \phi(q_i \cdot g_i) \neq 0, \\
& \mathfrak{T}\Big(\phi(q_i)\Big) = \mathfrak{T}(q_i)\,,\ \ \ell\mathrm{t}\Big(\phi(g_i)\Big) = \ell\mathrm{t}(g_i) \text{ and} \\
& \ell\mathrm{t}(q_i \cdot g_i) = \ell\mathrm{t}\Big(\phi(q_i \cdot g_i)\Big) \\
& \qquad\qquad = \ell\mathrm{t}\Big(\phi(q_i)\cdot\phi(g_i)\Big) \\
& \qquad\qquad = \mathfrak{T}\Big(\phi(q_i)\Big)\cdot\ell\mathrm{t}\Big(\phi(g_i)\Big) = \mathfrak{T}(q_i)\cdot\ell\mathrm{t}(g_i)
\end{aligned}
\tag{5}
$$

Let $J\subseteq\{1,\ldots,k\}$ be the set of all indexes i in the Gröbner representation $(*)$ such that $\phi\big(q_i \cdot \ell\mathrm{m}(g_i)\big)\neq 0$. Then $J\neq\emptyset$ due to (5) and because the left side of $(*)$ is mapped to $f_o\neq 0$ by assumption,

$$J = \Big\{ i\in\{1,\ldots,k\} \;\Big|\; \phi\big(q_i\cdot\ell\mathrm{m}(g_i)\big)\neq 0 \Big\} \neq \emptyset.$$

Apply the specialization homomorphism ϕ to both sides of the monomial Gröbner representation $(*)$. By (4),

$$f_o = \sum_{i=1}^{k} \phi\Big(q_i\cdot g_i\Big) \overset{(4)}{=} \sum_{i\in J} \phi\Big(q_i\cdot g_i\Big) \overset{\phi\text{-homo}}{=} \sum_{i\in J} \phi(q_i)\cdot\phi(g_i) \tag{$**$}$$

and, by (5),

$$i<j \implies \underbrace{\mathfrak{T}\Big(\phi(q_i)\Big)\,\ell\mathrm{t}\Big(\phi(g_i)\Big)}_{\| \, (5)\atop \overbrace{\mathfrak{T}(q_i)\,\ell\mathrm{t}(g_i)}} \quad \overset{(*)}{<} \quad \underbrace{\mathfrak{T}\Big(\phi(q_j)\Big)\,\ell\mathrm{t}\Big(\phi(g_j)\Big)}_{\| \, (5)\atop \overbrace{\mathfrak{T}(q_j)\,\ell\mathrm{t}(g_j)}} \quad \implies$$

$$\mathfrak{T}\Big(\phi(q_i)\Big)\,\ell\mathrm{t}\Big(\phi(g_i)\Big) < \mathfrak{T}\Big(\phi(q_j)\Big)\,\ell\mathrm{t}\Big(\phi(g_j)\Big) \quad \text{for all } i,j\in J. \tag{$***$}$$

Take $k_o = \max(J)$ and

$$v = \ell\mathrm{t}\Big(\phi(q_{k_o}g_{k_o})\Big) \overset{(5)}{=} \mathfrak{T}\Big(\phi(q_{k_o})\Big)\,\ell\mathrm{t}\Big(\phi(g_{k_o})\Big) \in \mathbb{T}.$$

Denote

$$S' = \sum_{i \in J \setminus \{k_o\}} \phi(q_i){\cdot}\phi(g_i).$$

and

$$S = \phi(q_{k_o}){\cdot}\phi(g_{k_o}) + S'.$$

Then the equation $(**)$ is equivalent to

$$f_o = S.$$

Note that the leading term of f_o is defined because, by assumption, $f_o{\neq}0$. Furthermore $\ell\mathrm{t}(f_o){=}v$. Indeed, if $S'{=}0$ then clearly

$$\ell\mathrm{t}(f_o) = \ell\mathrm{t}(S) = \ell\mathrm{t}\Big(\phi(q_{k_o}){\cdot}\phi(g_{k_o})\Big) = v;$$

otherwise

$$\mathfrak{T}\Big(\phi(q_i)\Big)\,\ell\mathrm{t}\Big(\phi(g_i)\Big) \overset{(***)}{<} \mathfrak{T}\Big(\phi(q_{k_o})\Big)\,\ell\mathrm{t}\Big(\phi(g_{k_o})\Big) \quad \text{for all} \quad i{\in}J\setminus\{k_o\}$$

and then [9]

$$\ell\mathrm{t}(f_o) = \ell\mathrm{t}(S) = \max\Big(\ell\mathrm{t}\big(\phi(q_{k_o}){\cdot}\phi(g_{k_o})\big), \ell\mathrm{t}(S')\Big) = \ell\mathrm{t}\Big(\phi(q_{k_o}){\cdot}\phi(g_{k_o})\Big) = v.$$

Hence, by $(***)$,

$$\mathfrak{T}\Big(\phi(q_i)\Big)\,\ell\mathrm{t}\Big(\phi(g_i)\Big) \overset{(***)}{\leq} \mathfrak{T}\Big(\phi(q_{k_o})\Big)\,\ell\mathrm{t}\Big(\phi(g_{k_o})\Big) = v = \ell\mathrm{t}(f_o) \quad \text{for} \quad i{\in}J.$$

Then the statements $(**)$ and $(***)$ can be rewritten as

$$f_o = \sum_{i \in J} \phi(q_i){\cdot}\phi(g_i) \;\; g_i{\in}\widehat{G},\; q_i{\in}\mathbb{M} \quad \text{and}$$
$$\mathfrak{T}\Big(\phi(q_i)\Big)\,\ell\mathrm{t}\Big(\phi(g_i)\Big) \leq \ell\mathrm{t}(f_o) \quad \text{for all} \;\; i{\in}J$$

which is a Gröbner representation of f_o of the form $(\dagger)$ in terms of $\widehat{G}_o = \phi(\widehat{G})$ with monomials $\phi(q_i){\in}\overline{P}\,[X]$, $i{\in}J$ w.r.t. the ordering $<$ in the polynomial algebra $\overline{P}\,[X]$. Then, since $f{\in}I$ is arbitrary chosen, $\widehat{G}_o$ is a (classical) Gröbner

[9] Here we essentially use the fact that $(*)$ is a strong monomial Gröbner representation provided that $\widehat{G}$ is a strong Gröbner basis. Otherwise there would be no guarantee that v does not cancel in the rightmost side of $(**)$ resulting to $\ell\mathrm{t}(f_o){<}v$ (the case $f_o{=}0$ is impossible by assumption) which would mean that there is a cancellation of terms in $(**)$ "protruding beyond" the leading term of f_o and the rest of the proof will fail.

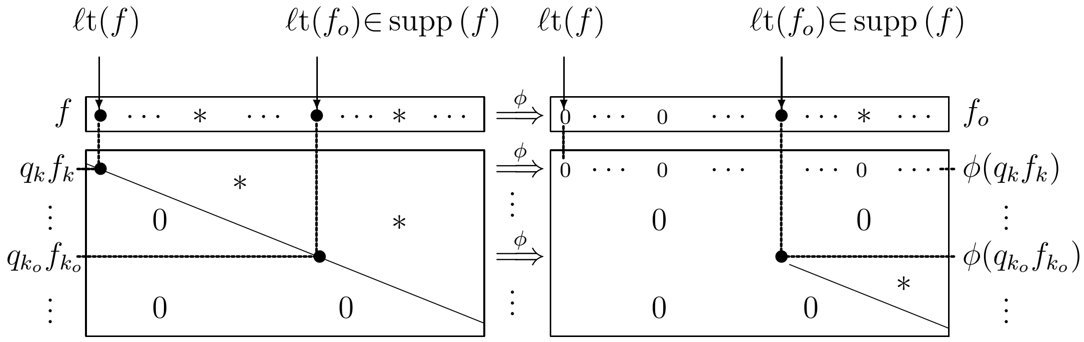
basis of I_o according to theorem 5.62 on page 219 in [BW93], provided that $\overline{P}$ is a field:

$$\left\langle \ell\mathrm{t}(\widehat{G}_o)\right\rangle_{\overline{P}[X]} = \left\langle \ell\mathrm{t}(I_o)\right\rangle_{\overline{P}[X]}.$$

The transition from the monomial Gröbner representation $(*)$ in $P[X]$ to the Gröbner representation $(**)$ in $\overline{P}[X]$ analytically attained above can be visualized in a graphical form as a linear combination of vectors in finite-dimensional vector spaces with a basis

$$U = \left(\bigcup_{i=1}^{k} \mathfrak{T}(q_i) * \mathrm{supp}\,(g_i)\right) \bigcup \mathrm{supp}\,(f) \subseteq \mathbb{T}.$$

over the ring P and the field $\overline{P}$ respectively provided that terms $\mathbb{T}$ are linear independent over P as well as over $\overline{P}$. The summation is carried out by rows resulting to the vectors which represent f and f_o respectively. The legend is: 0 – definitely zero, $\bullet$ – definitely not zero, $*$ – unknown.



The combination $(*)$ over P. The combination $(**)$ over $\overline{P}$.

The next step is to show that the image of the original Gröbner basis $G_o{=}\phi(G){\subseteq}\phi(\widehat{G}){=}\widehat{G}_o$ is a (classical) Gröbner basis of the ideal I_o in $\overline{P}[X]$. Recall that, by construction, the strong Gröbner basis $\widehat{G}{\subseteq}I\setminus\{0\}$ of the ideal I with $\ell\mathrm{c}(\widehat{G}){\cap}K{=}\emptyset$ is $\widehat{G}{=}G{\cup}\overline{G}$ where $\overline{G}{\subseteq}I\setminus\{0\}$ is a strong Gröbner basis of I such that $\ell\mathrm{c}(\widehat{G}){\cap}K{=}\emptyset$ provided by lemma 2.5.19. According to the proof of lemma 2.5.19, $\overline{G}$ is a set of principal polynomials:

$$p_S{\in}I\setminus\{0\} \;\text{ with }\; \ell\mathrm{c}(p){\notin}K \text{ and } \ell\mathrm{t}(q) \,\Big|\, \underbrace{\ell\mathrm{t}(p)}_{\substack{\| \text{ by part } b) \text{ of lemma 2.5.18} \\ L(S)}}$$

for each non-empty subset $S \subseteq G$ saturated in G and $q \in S$. Note that, by (3), $0 \notin \overline{G}_o$ and hence $\ell\mathrm{t}\big(\phi(g)\big)$ is defined for all $\phi(g)$, $g \in \overline{G}$. Take a principal polynomial $p \in \overline{G} \setminus G \subseteq \widehat{G}$ and the corresponding non-empty subset $S \subseteq G$ saturated in G. Then $\ell\mathrm{t}(p) = L(S)$ $\big[= \mathrm{lcm}\big(\ell\mathrm{t}(S)\big)\big]$. Take a polynomial $q \in S$. Then $\ell\mathrm{t}(q) | \ell\mathrm{t}(p)$ and hence, by (3),

$$\Big[\ell\mathrm{t}(q) \overset{(3)}{=}\Big]\ \ell\mathrm{t}\Big(\phi(q)\Big)\ \Big|\ \ell\mathrm{t}\Big(\phi(p)\Big)\ \Big[\overset{(3)}{=} \ell\mathrm{t}(p)\Big].$$

Consequently, by lemma 2.2.5, the element $\phi(p)$ can be safely removed from the Gröbner basis $\widehat{G}_o$ because it is top-reducible by $\phi(q) \in G_o \subseteq \widehat{G}_o$ and the remaining set $\widehat{G}_o \setminus \{\phi(p)\}$ is still a Gröbner basis of I_o. By iteration of the argument above for each $p \in \overline{G} \setminus G$

$$\widehat{G}_o \setminus \Big(\overline{G}_o \setminus G_o\Big) = G_o$$

is a Gröbner basis of the ideal I_o. $\square$

2.5.2 The e-specialization of a Gröbner basis in $\mathcal{P}_e$

The machinery of Gröbner bases for polynomials over a principal ideal ring presented in the previous subsections will now be used as a theoretical tool to show the invariance of the Gröbner basis property in the particular case of the algebra of extended polynomials $\mathcal{P}_e$ considered as the algebra of polynomials $\mathcal{P}_{e[X]}$ in the main variables X over the principal ideal ring $\Bbbk[e]$ ($\Bbbk$ is a field) and under the specialization of the extended variable e at 0 which is the core of the postprocessing step in our approach and implementation of the numerical computation of Gröbner bases in $\Bbbk[X]$ using the computation of a Gröbner basis in ${}_e\mathcal{P}$ as an intermediate step.

Let $\underset{\mathbb{T}}{<}$ be an arbitrary term order on $\mathbb{T}$. Recall that the extended term order $\underset{\mathbb{T}_e}{<}$ on the extended terms $\mathbb{T}_e \subsetneq \mathcal{P}_e$ is defined as the lexicographical product of the order $\underset{\mathbb{T}}{<}$ on $\mathbb{T}$ and the degree anti-compatible order on the (finitely many) powers of the extended variable e with priority given to the main variables and "0" – the smallest term (definition 2.4.1 on page 90). Let $\underset{\mathbb{T}}{<}$ be the term order on the terms of $\mathbb{T} \subsetneq \mathcal{P}_{e[X]}$.

It is clear that $\mathcal{P}_e$ and $\mathcal{P}_{e[X]}$ are indistinguishable as rings. Therefore a minor refinement of notation would be helpful to avoid confusion: for each polynomial $f \in \mathcal{P}_{e[X]} \setminus \{0\}$, let $\mathrm{supp}_X(f)$, $\ell\mathrm{t}_X(f)$, $\ell\mathrm{c}_X(f)$, $\ell\mathrm{m}_X(f)$ be the support, the leading term, the leading coefficient and the leading monomial of f w.r.t. the term order $\underset{\mathbb{T}}{<}$ and $\mathrm{supp}\,(f)$, $\ell\mathrm{t}(f)$, $\ell\mathrm{c}(f)$, $\ell\mathrm{m}(f)$ be the leading term, the leading coefficient and the leading monomial w.r.t. the extended term order $\underset{\mathbb{T}_e}{<}$ of the same polynomial f but considered as an element of $\mathcal{P}_e$.

The next lemma establishes a simple relation between the orders $\underset{\mathbb{T}}{<}$ and $\underset{\mathbb{T}_e}{<}$ as well as an important relation between the leading terms of a polynomial considered as an element of $\mathcal{P}_e$ and $\mathcal{P}_{e[X]}$.

Lemma 2.5.21. Let $f, g \in \mathcal{P}_e \backslash \{0\}$, $s_1, s_2 \in {\mathbb{T}_e}^*$. The following holds.

a) If $s_1 \underset{\mathbb{T}_e}{\leq} s_2$ then $\mathrm{T}(s_1) \underset{\mathbb{T}}{\leq} \mathrm{T}(s_2)$.

b) $\mathrm{T}\big(\ell\mathrm{t}(f)\big) = \ell\mathrm{t}_X(f)$.

c) If $\ell\mathrm{t}(f) \underset{\mathbb{T}_e}{\leq} \ell\mathrm{t}(g)$ then $\ell\mathrm{t}_X(f) \underset{\mathbb{T}}{\leq} \ell\mathrm{t}_X(g)$.

Proof.

a) Let $s_1 \underset{\mathbb{T}_e}{\leq} s_2$. Assume for a contradiction that $\mathrm{T}(s_2) \underset{\mathbb{T}}{<} \mathrm{T}(s_1)$. By part b_1) of definition 2.4.1 of the order $\underset{\mathbb{T}_e}{<}$, an extended term with a greater main part is greater w.r.t. $\underset{\mathbb{T}_e}{<}$ and hence $s_2 \underset{\mathbb{T}_e}{<} s_1$.

b) By definition of the order $\underset{\mathbb{T}_e}{<}$ on the extended terms $\mathbb{T}_e$ as a lexicographical product of the order $\underset{\mathbb{T}}{<}$ on the terms $\mathbb{T}$ and the degree anti-compatible order on the (finitely many) powers of e with priority given to the order $\underset{\mathbb{T}}{<}$, $\ell\mathrm{t}(f)$ is a term from $\mathrm{supp}\,(f)$ with the lowest power of e among the terms from $\mathrm{supp}\,(f)$ with the highest w.r.t. $\underset{\mathbb{T}}{<}$ main part (the part in the main variables X – the term in $\mathbb{T}$). Thus $\ell\mathrm{t}_X(f)$ – the leading term of f as an element of $\mathcal{P}_{e[X]}$ is $\mathrm{T}\big(\ell\mathrm{t}(x)\big)$ – the main part of the leading term of f as an element of $\mathcal{P}_e$ provided that the (formal) coefficient (an element in $\Bbbk[e]$) at $\mathrm{T}(\ell\mathrm{t}(x))$ is not zero – it has a non-zero summand $d_o = \ell\mathrm{c}(f) \cdot e^{\deg_e(\ell\mathrm{t}(f))}$ and possibly other summands with strictly greater powers of e and thus they can not add to zero with d_o because the (non-zero) powers of e are linear independent over the field $\Bbbk$.

c) Combine parts *a*) and *b*). ☐

The following theorem is a counterpart of theorem 4.1.18 on page 209 in [AL94]. It is a bridge between the Gröbner bases theory in $\mathcal{P}_e$ and $\mathcal{P}_{e[X]}$.

Theorem 2.5.5 (the translation principal of Gröbner bases in $\mathcal{P}_e$). Let I be an ideal in $\mathcal{P}_e$ and $G \subseteq I\backslash\{0\}$ be a Gröbner basis of I with respect to $\underset{\mathbb{T}_e}{<}$. Then G is a Gröbner basis of I considered as an ideal in $\mathcal{P}_{e[X]}$ with respect to $\underset{\mathbb{T}}{<}$.

Proof. Let $I{=}\langle G\rangle_{\mathcal{P}_{e[X]}}$. Take $f{\in}I\backslash\{0\}$ and show that f has a Gröbner representation in terms of G in $\mathcal{P}_{e[X]}$. By part f) of theorem 2.3.2 there exists a Gröbner representation of f in terms of G provided that G is a Gröbner basis of I in $\mathcal{P}_e$:

$$f=\sum_{h\in G} q_h{\cdot}h \text{ with } q_h{\in}\mathcal{P}_e \ \text{ for all } \ h{\in}G \tag{$*$}$$

and

$$0\neq \ell\mathrm{t}(q_h){\cdot}\ell\mathrm{t}(h)\underset{\mathbb{T}_e}{\leq}\ell\mathrm{t}(f) \ \text{ for all } \ h{\in}G \ \text{ such that } \ q_h{\neq}0. \tag{1}$$

It is sufficient to show that the representation $(*)$ is a Gröbner representation of f in terms of G considered as a polynomial combination in $\mathcal{P}_{e[X]}$. Take $h{\in}G$ such that $q_h{\neq}0$ and show that $\ell\mathrm{t}_X\,(q_h){\cdot}\ell\mathrm{t}_X\,(h)\underset{\mathbb{T}}{\leq}\ell\mathrm{t}_X\,(f)$. By lemma 2.3.11,

$$\ell\mathrm{t}(q_h){\cdot}\ell\mathrm{t}(h) \overset{(1)}{\neq} 0 \ \text{ follows } \ q_h{\cdot}h\neq 0 \ \text{ and } \ \ell\mathrm{t}(q_h{\cdot}h)=\ell\mathrm{t}(q_h){\cdot}\ell\mathrm{t}(h). \tag{2}$$

On the other hand, by part b) of lemma 2.5.21 and part b) of lemma 2.3.1,

$$\ell\mathrm{t}_X\,(q_h{\cdot}h) \overset{(2.5.21)}{=} \mathrm{T}\Big(\ell\mathrm{t}(q_h{\cdot}h)\Big) \overset{(2)}{=} \mathrm{T}\Big(\ell\mathrm{t}(q_h){\cdot}\ell\mathrm{t}(h)\Big) \overset{(2.3.1)}{=} \mathrm{T}\Big(\ell\mathrm{t}(q_h)\Big){\cdot}\mathrm{T}\Big(\ell\mathrm{t}(h)\Big) \overset{(2.5.21)}{=} \ell\mathrm{t}_X\,(q_h){\cdot}\ell\mathrm{t}_X\,(h). \tag{3}$$

Then, by part c) of lemma 2.5.21,

$$\ell\mathrm{t}(q_h{\cdot}h) \overset{(2)}{=} \ell\mathrm{t}(q_h){\cdot}\ell\mathrm{t}(h) \underset{\mathbb{T}_e}{\overset{(1)}{\leq}}\ell\mathrm{t}(f)$$

follows

$$\ell\mathrm{t}_X\,(q_h){\cdot}\ell\mathrm{t}_X\,(h) \overset{(3)}{=} \ell\mathrm{t}_X\,(q_h{\cdot}h) \underset{\mathbb{T}}{\overset{(2.5.21)}{\leq}} \ell\mathrm{t}_X\,(f).$$

Therefore, $(*)$ is a Gröbner representation of f in terms of G in $\mathcal{P}_{e[X]}$. Then, since f is arbitrary chosen, $G{\subsetneq}\mathcal{P}_{e[X]}\backslash\{0\}$ is a Gröbner basis of the ideal $I{\subsetneq}\mathcal{P}_{e[X]}$. □

Our application of the Gröbner bases theory of the algebra of extended polynomials for numerical computation of Gröbner bases uses the e-degree of a term as an indicator how "unreliable" the coefficient at the term is - a higher e-degree means that the coefficient at the term is either very small or it has been attained from a very small coefficient in the flow of the Gröbner basis computation. Therefore we need a measure how "reliable" a coefficient and, in particular, a leading coefficient in a Gröbner basis is provided by the following definition.

Definition 2.5.14. Let $s \in \mathbb{T}_e{}^*$ and $f \in \mathcal{P}_e \backslash \{0\}$. Then

a) $\mathrm{dec}_e(s)$ – the $e-$**decay of** α **in** $\mathcal{P}_e$ is the $e-$degree of s as an extended term in $\mathbb{T}_e$.
$$\mathrm{dec}_e(s) = \deg_e(s).$$

b) $\mathrm{dec}_e(f)$ – the $e-$**decay of** f **in** $\mathcal{P}_e$ is the $e-$decay of the leading term:
$$\mathrm{dec}_e(f) = \mathrm{dec}_e\Big(\ell\mathrm{t}(f)\Big).$$

By convention, $\mathrm{dec}_e(0)=0$.

Definition 2.5.15. A polynomial $f \in \mathcal{P}_e$ such that $\mathrm{dec}_e(f)=0$ is called **stable in** $\mathcal{P}_e$. A set $F \subseteq \mathcal{P}_e$ such that $\mathrm{dec}_e(f)=0$ for all $f \in F$ is called **stable in** $\mathcal{P}_e$.

The following counterpart of the e-decay for polynomials in $\mathcal{P}_{e[X]}$ is essentially the same map from polynomials to $\mathbb{N}_o$ as shown in the next lemma.

Definition 2.5.16. Let $\alpha \in \Bbbk[e] \backslash \{0\}$ and $f \in \mathcal{P}_{e[X]} \backslash \{0\}$. Then

a) $\mathrm{dec}_{X_e}(\alpha)$ – the $e-$**decay of** α **in** $\mathcal{P}_{e[X]}$ is the $e-$order of α as a polynomial in e.
$$\mathrm{dec}_{X_e}(\alpha) = \mathrm{ord}_e(\alpha).$$

b) $\mathrm{dec}_{X_e}(f)$ – the $e-$**decay of** f **in** $\mathcal{P}_{e[X]}$ is the $e-$decay of the leading coefficient:
$$\mathrm{dec}_{X_e}(f) = \mathrm{dec}_{X_e}\Big(\ell\mathrm{c}_X\,(f)\Big).$$

By convention, $\mathrm{dec}_{X_e}(0)=0$.

Definition 2.5.17. A polynomial $f \in \mathcal{P}_{e[X]}$ such that $\mathrm{dec}_{X_e}(f)=0$ is called **stable in** $\mathcal{P}_{e[X]}$. A set $F \subseteq \mathcal{P}_{e[X]}$ such that $\mathrm{dec}_{X_e}(f)=0$ is called **stable in** $\mathcal{P}_{e[X]}$.

Lemma 2.5.22. Let $f \in \mathcal{P}_e \backslash \{0\}$. Then $\mathrm{dec}_e(f)=\mathrm{dec}_{X_e}(f)$.

Proof. If $f=0$ then $\mathrm{dec}_e(f)=0=\mathrm{dec}_{X_e}(f)$ by convention. Assume $f \neq 0$. According to definitions of $\mathrm{dec}_e(f)$ and $\mathrm{dec}_{X_e}(f)$, it is sufficient to show that
$$\deg_e\big(\ell\mathrm{t}(f)\big) = \mathrm{ord}_e\big(\ell\mathrm{c}_X\,(f)\big).$$

Indeed, by definition 2.4.1 of the term order $\underset{\mathbb{T}_e}{<}$,
$$\deg_e\big(\ell\mathrm{t}(f)\big) = \min\Big(\deg_e(s) \mid s \in \mathrm{supp}\,(f)\,, \mathrm{T}(s)=\mathrm{T}\big(\ell\mathrm{t}(f)\big)\Big). \tag{1}$$

On the other hand, by part b) of lemma 2.5.21,

$$\ell\mathrm{t}_X(f) = \mathrm{T}\big(\ell\mathrm{t}(f)\big). \tag{2}$$

Therefore, by the linear independence of terms $\mathbb{T}$ over $\Bbbk[e]$,

$$\ell\mathrm{c}_X(f) = \sum_{\substack{s\in\operatorname{supp}(f)\\ \mathrm{T}(s)=\ell\mathrm{t}_X(f)}} \mathfrak{C}_f(s)\cdot e^{\deg_e(s)}$$

and hence

$$\operatorname{ord}_e\big(\ell\mathrm{c}_X(f)\big) = \min\Big(\deg_e(s)\mid s\in\operatorname{supp}(f),\mathrm{T}(s)=\ell\mathrm{t}_X(f)\Big). \tag{3}$$

Finally

$$\begin{aligned}
&\operatorname{dec}_e(f)\\
&\quad\| \text{ by def. } 2.5.14\\
&\deg_e\big(\ell\mathrm{t}(f)\big) \overset{(1)}{=} \min\Big(\deg_e(s)\mid s\in\operatorname{supp}(f),\mathrm{T}(s)=\mathrm{T}\big(\ell\mathrm{t}(f)\big)\Big)\\
&\qquad\qquad\overset{(2)}{=} \min\Big(\deg_e(s)\mid s\in\operatorname{supp}(f),\mathrm{T}(s)=\ell\mathrm{t}_X(f)\Big) \overset{(3)}{=} \operatorname{ord}_e\big(\ell\mathrm{c}_X(f)\big)\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\| \text{ by def. } 2.5.16\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\operatorname{dec}_{X_e}(f).
\end{aligned}$$ □

Corollary 2.5.5. A set $F\subseteq\mathcal{P}_e$ is stable in $\mathcal{P}_e$ if and only if it is stable in $\mathcal{P}_{e[X]}$

Proof. If $F=\emptyset$ then, by convention, F is stable in $\mathcal{P}_e$ and stable in $\mathcal{P}_{e[X]}$. If $F\neq\emptyset$ By lemma 2.5.22. □

In view of the latter corollary the e-decay of a polynomial is invariant of its interpretation in $\mathcal{P}_e$ or $\mathcal{P}_{e[X]}$.

Definition 2.5.18. The e**-specialization homomorphism** Φ_o is a map

$$\Phi_o : \mathcal{P}_{e[X]} \longrightarrow \Bbbk[X]$$

from the ring $\mathcal{P}_{e[X]}$ of polynomials over the ring $\Bbbk[e]$ to the ring $\Bbbk[X]$ of polynomials over a field $\Bbbk$ in the same indeterminates X and the same term order $\underset{\mathbb{T}}{<}$ defined as a homomorphic extension of the identity map on $\Bbbk$ such that $\Phi_o(e)=0$ and $\Phi_o(x)=x$ for any $x\in X$: for any

$$c(e) = \sum_{i=0}^{\mathcal{B}-1}\gamma_i\cdot e^i\in\Bbbk[e] \quad\text{with}\quad \gamma_i\in\Bbbk \quad\text{for each}\quad 0\leq i<\mathcal{B}$$

the image of $c(e)$ under Φ_o is

$$\Phi_o\bigl(c(e)\bigr) = c(0) = \gamma_o$$

and for any

$$f = \sum_{t\in\operatorname{supp}(f)} c_t(e)\cdot t \in \mathcal{P}_{e[X]} \quad\text{with}\quad 0{\neq}c_t(e){\in}\Bbbk[e] \quad\text{for each}\quad t{\in}\operatorname{supp}(f){\subseteq}\mathbb{T}$$

the image of f under Φ_o is

$$\Phi_o(f) = \sum_{t\in\operatorname{supp}(f)} c_t(0)\cdot t \quad \left[\quad = \sum_{\substack{t\in\operatorname{supp}(f)\\ c_t(0)\neq 0}} c_t(0)\cdot t\right] \in \Bbbk[X].$$

Lemma 2.5.23.

a) The restriction $\Phi_o|_{\Bbbk[e]}$ of the e-specialization homomorphism Φ_o on $\Bbbk[e]$ is a surjective ring homomorphism from $\Bbbk[e]$ to $\Bbbk$ with the kernel

$$\ker\bigl(\Phi_o|_{\Bbbk[e]}\bigr) = \langle e\rangle_{\Bbbk[e]} \quad\text{a maximal ideal in } \Bbbk[e].$$

b) The e-specialization homomorphism Φ_o is a surjective ring homomorphism from $\mathcal{P}_{e[X]}$ to $\Bbbk[X]$ which preserves the term order.

Proof.

a) Let y be an algebraically independent over $\Bbbk$ variable. Denote

$\xrightarrow{y\to e}$ – the canonical surjective homomorphism from $\Bbbk[y]$ to $\Bbbk[e]$ such that $\xrightarrow{y\to e}|_{\Bbbk}{=}\mathrm{id}_{\Bbbk}$ and $y \xrightarrow{y\to e} e$,

$\xrightarrow{y\to 0}$ – the canonical surjective homomorphism from $\Bbbk[y]$ to $\Bbbk$ such that $\xrightarrow{y\to 0}|_{\Bbbk}{=}\mathrm{id}_{\Bbbk}$ and $y \xrightarrow{y\to 0} 0$

provided by the universal property of polynomial rings. It is clear that $\bigl\langle y^{\mathcal{B}}\bigr\rangle_{\Bbbk[y]}{\subseteq}\langle y\rangle_{\Bbbk[y]}$ because $\mathcal{B}{\geq}1$ by convention. Then, by a well-known variation of the ring isomorphism theorem (see e.g. [Lan92] page 89), there exist unique homomorphisms

$$\lambda : \Bbbk[e] \longrightarrow \Bbbk \quad\text{and}\quad \lambda' : {}^{\Bbbk[y]}\!/_{\langle y^{\mathcal{B}}\rangle} \longrightarrow {}^{\Bbbk[y]}\!/_{\langle y\rangle}$$

which make the diagram

$$
\begin{tikzcd}
\Bbbk[y] \arrow[r, "y\to e"] \arrow[dr, "y\to 0"'] & \Bbbk[e] \arrow[r, "\cong"] \arrow[d, "\lambda"] & \Bbbk[y]/\langle y^{\mathcal{B}}\rangle \arrow[d, dotted, "\lambda'"] \\
& \Bbbk \arrow[r, "\cong"] & \Bbbk[y]/\langle y\rangle
\end{tikzcd}
\tag{$*$}
$$

commute. Let us take a closer look at the homomorphism λ. It is known that the evaluation homomorphism $\xrightarrow{y\to 0}$ is surjective. Hence λ is surjective by the commutativity of $(*)$. Furthermore, for any polynomial $f(y)\in\Bbbk[y]$

$$
\begin{tikzcd}
f(y) \arrow[r, "y\to e"] \arrow[dr, "y\to 0"'] & f(e) \arrow[d, "\lambda"] \\
& f(0)
\end{tikzcd}
$$

and hence the only possible value for $\lambda\big(f(e)\big)$ to keep the diagram $(*)$ commutative is $f(0)$. On the other hand the rule $f(e)\to f(0)$ defines the map $\lambda:\Bbbk[e]\to\Bbbk$ for any polynomial f in $\Bbbk[e]$ and this is exactly the way how the map $\Phi_o|_{\Bbbk[e]}$ works according to definition 2.5.18. Hence $\Phi_o|_{\Bbbk[e]}=\lambda$ is a surjective homomorphism from $\Bbbk[e]$ to C. Also it is easy to see by the linear independence of non-vanishing powers of e over $\Bbbk$ (cf. part c) of theorem 2.3.1) that for any polynomial $g(e)\in\Bbbk[e]$, $\Phi_o\big(g(e)\big)=g(0)=0$ if and only if $g\in\langle e\rangle_{\Bbbk[e]}$ and hence $\ker(\Phi_o)=\langle e\rangle_{\Bbbk[e]}$. By the second isomorphism theorem and the inclusion $\langle y^{\mathcal{B}}\rangle_{\Bbbk[y]}\subseteq\langle y\rangle_{\Bbbk[y]}$,

$$
\Bbbk[e]/\langle e\rangle \cong \frac{\Bbbk[y]/\langle y\rangle}{\langle y\rangle/\langle y^{\mathcal{B}}\rangle} \cong \Bbbk[y]/\langle y\rangle \cong \Bbbk \tag{1}
$$

and hence $\langle e\rangle_{\Bbbk[e]}$ is a maximal ideal in $\Bbbk[e]$ because the quotient modulo $\langle e\rangle_{\Bbbk[e]}$ is a field isomorphic to $\Bbbk$ (see [Lan92] page 93).

b) By part *a*), $\Phi_o:\Bbbk[e]\to\Bbbk$ is a surjective homomorphism. Then, by the universal property of polynomial rings ([BW93] lemma 2.17 on page 75) there exists the unique surjective homomorphic extension of $\Phi_o|_{\Bbbk[e]}$ to a homomorphism $\Lambda{:}\mathcal{P}_{e[X]}{\to}\Bbbk[X]$ such that $\Lambda|_{\Bbbk[e]}{=}\Phi_o|_{\Bbbk[e]}$ and $\Lambda(x){=}x$ for any $x{\in}X$. But $\Phi_o{:}\mathcal{P}_{e[X]}{\to}\Bbbk[X]$ is defined as a homomorphic extension of $\Phi_o|_{\Bbbk[e]}$. Hence $\Phi_o{=}\Lambda$. Also it is clear that $\Phi_o|_{\mathbb{T}}{=}\mathrm{id}|_{\mathbb{T}}$ considered as a monoid homomorphism and hence Φ_o trivially keeps the term order $\underset{\mathbb{T}}{<}$: for any $t_1, t_2{\in}\mathbb{T}{\subsetneq}\mathcal{P}_{e[X]}$

$$t_1 \underset{\mathbb{T}}{<} t_2 \quad\Longleftrightarrow\quad \Bbbk[X]{\supsetneq}\mathbb{T} \ni \quad \Phi_o(t_1){=}t_1 \underset{\mathbb{T}}{<} t_2{=}\Phi_o(t_2) \quad \in \mathbb{T}{\subsetneq}\Bbbk[X]. \quad \square$$

Remark 2.5.1. The evaluation map $\Phi_\alpha : \Bbbk[e]{\to}\Bbbk$ defined in the same fashion as Φ_o for an arbitrary $\alpha{\in}\Bbbk$ is not a ring homomorphism unless $\alpha{=}0$. For example take $\mathcal{B}{=}1$, $f{=}e$ and $g{=}1{+}e$. Fix any $0{\neq}\alpha{\in}\Bbbk$. Then $\alpha^2{\neq}0$ because $\Bbbk$ is a field and therefore

$$\begin{array}{c}\Phi_\alpha(f{\cdot}g) = \Phi_\alpha\big(e{\cdot}(1+e)\big) = \Phi_\alpha\big(e{+}e^2\big) \overset{(\mathcal{B}=1)}{=} \Phi_\alpha(e) = \alpha \\ \nparallel \\ \Phi_\alpha(f){\cdot}\Phi_\alpha(g) = \Phi_\alpha(e){\cdot}\Phi_\alpha(1{+}e) = \alpha{\cdot}(1{+}\alpha) = \alpha{+}\alpha^2\end{array}$$

The result of the next theorem is the core for the postprocessing step in the numerical algorithm for computing Gröbner bases $\Bbbk[X]$ via the computation of a Gröbner basis in $\mathcal{P}_e$.

Theorem 2.5.6 (The e-specialization of a Gröbner basis in ${}_e\mathcal{P}$). Let $G{\subseteq}\mathcal{P}_e{\setminus}\{0\}$ be a Gröbner basis (w.r.t. $\underset{\mathbb{T}_e}{<}$) in $\mathcal{P}_e$. If G is stable then the image $\Phi_o(G)$ of G under the e-specialization homomorphism Φ_o is a Gröbner basis in $\Bbbk[X]$ w.r.t. $\underset{\mathbb{T}}{<}$.

Proof. By the e-specialization theorem 2.5.5, G is a Gröbner basis w.r.t. $\underset{\mathbb{T}}{<}$ in the ring $\mathcal{P}_{e[X]}$ over $\Bbbk[e]$.

On the other hand, $\Bbbk[e]$ is a principal ideal ring. Indeed, let y be an algebraically independent over $\Bbbk$ variable. Then, by the universal property of polynomial rings ([BW93] lemma 2.15 on page 72) there exists the (unique) canonical surjective homomorphism $\theta : \Bbbk[y]{\to}\Bbbk[e]$ (such that $\theta|_{\Bbbk}{=}\mathrm{id}_C$ and $\theta(y){=}e$) and hence $\Bbbk[e]$ is a principal ideal ring as a homomorphic image of the principal ideal domain $\Bbbk[y]$ (see e.g. [Akr89] page 176 Theorem 4.1.19).

By corollary 2.5.5, G is a stable set in $\mathcal{P}_{e[X]}$. Then it is easy to see that $\ell c_X\,(g)(e) \xrightarrow{\Phi_o|_{\Bbbk[e]}} \ell c_X\,(g)(0){\neq}0$ which is equivalent to $\ell c_X\,(g)(e){\notin}\ker(\Phi_o|_{\Bbbk[e]})$ for any $g{\in}G{\subseteq}\mathcal{P}_{e[X]}$. On the other hand, the kernel of $\Phi_o|_{\Bbbk[e]}$ is, by part a) of lemma 2.5.23, $\ker\Phi_o|_{\Bbbk[e]}{=}\langle e\rangle_{\Bbbk[e]}$ and hence $\ell c_X\,(G){\cap}\langle e\rangle_{\Bbbk[e]}{=}\emptyset$. Then the image $\phi_{\langle e\rangle}(G)$ of the Gröbner basis $G\subseteq\mathcal{P}_{e[X]}$ under the specialization homomorphism $\phi_{\langle e\rangle}$ (definition 2.5.13) is a Gröbner basis w.r.t. the term order $\underset{\mathbb{T}}{<}$ in the polynomial ring $\left(\Bbbk[e]/\langle e\rangle\right)[X]$ over the principal ideal ring $\Bbbk[e]/\langle e\rangle$ by the specialization theorem 2.5.4. Furthermore, by $\ker\Phi_o|_{\Bbbk[e]}=\langle e\rangle_{\Bbbk[e]}$ and the same argument as in the proof of part a) of lemma 2.5.23, there exists a (uniquely defined) homomorphism $\lambda\colon\Bbbk[e]/\langle e\rangle\to\Bbbk$ which makes the diagram

$$\begin{tikzcd} \Bbbk[e] \arrow[r, "\Phi_o|_{\Bbbk[e]}"] \arrow[dr, "\phi_{\langle e\rangle}|_{\Bbbk[e]}"'] & \Bbbk \\ & \Bbbk[e]/\langle e\rangle \arrow[u, "\lambda"'] \end{tikzcd} \qquad (*)$$

commute. By the commutativity of $(*)$, λ is an isomorphism because $\langle e\rangle_{\Bbbk[e]}$ is a maximal ideal in $\Bbbk[e]$ and the surjective (by part a) of lemma 2.5.23) homomorphism $\Phi_o|_{\Bbbk[e]}$ is clearly not a zero map. Then, by the universal property of polynomial rings, there exists the unique homomorphic extension of the isomorphism λ to the ring isomorphism $\Lambda{:}\Bbbk[e]/\langle e\rangle\to\Bbbk[X]$ such that $\Lambda(x)=x$ for any $x\in X$ and the diagram

$$\begin{tikzcd} \mathcal{P}_{e[X]} \arrow[r, "\Phi_o"] \arrow[dr, "\phi_{\langle e\rangle}"'] & \Bbbk[X] \\ & \left(\Bbbk[e]/\langle e\rangle\right)[X] \arrow[u, "\Lambda"'] \end{tikzcd} \qquad (**)$$

commutes. It is clear that the ring isomorphism Λ is the identity map on terms, hence it trivially preserves the term order and hence it preserves the Gröbner basis property of a system of polynomials. Therefore the image

$$G \xrightarrow{\phi_{\langle e\rangle}} \phi_{\langle e\rangle}(G) \xrightarrow{\Lambda} \left(\Lambda\circ\phi_{\langle e\rangle}\right)(G) \overset{(**)}{=} \Phi_o(G)$$

is a (classical) Gröbner basis in $\Bbbk[X]$ w.r.t. the term order $\underset{\mathbb{T}}{<}$. $\square$

2.6 Gröbner bases in the algebra ${}_e\mathcal{P}$

If a Gröbner basis in the considered above algebra $\mathcal{P}_e$ contains a polynomial with the leading term which has a positive e-degree then the e-specialization theorem 2.5.4 considered in the previous section is not applicable and hence there is no guarantee that the result of the e-specialization of a Gröbner basis in ${}_e\mathcal{P}$ at e=0 is a Gröbner basis in $\Bbbk[X]$. To deal with such situations we recompute a Gröbner basis in a different algebra ${}_e\mathcal{P}$ which is again the algebra of extended polynomials also equal to $\mathcal{P}_e$ as a ring but with a different term order $\underset{e\mathbb{T}}{<}$ – in a sense, a mirror image of the order $\underset{\mathbb{T}_e}{<}$. The order $\underset{e\mathbb{T}}{<}$ is more "aggressive" towards elimination of leading terms with a positive e-degree.

Let ${}_e\mathcal{P}$=$\mathcal{P}(e)$ and ${}_e\mathbb{T} = \mathbb{T}(e)$ be the instances of the algebra $\mathcal{P}(e)$ and the extended terms $\mathbb{T}(e)$ (the terms in the main variables X and the extended variable e) such that $\mathbb{T}_e \subseteq \mathcal{P}_e$. Denote ${}_e\mathbb{T}^*$ – the set of non-zero terms $T(e)^*$ in ${}_e\mathbb{T}$, ${}_e\mathbb{M}$ – the monomials $\mathbb{M}(e)$ of the algebra ${}_e\mathcal{P}$ and ${}_{\not e}\mathbb{M}$ the set of $e-$free monomials of ${}_e\mathcal{P}$. Fix an arbitrary term order $\underset{\mathbb{T}}{<}$ on $\mathbb{T}$ (the terms in the main variables X). Equip ${}_e\mathbb{T}$ with the order $\underset{e\mathbb{T}}{<}$ defined as the lexicographical product of the degree anti-compatible order on the (finitely many) powers of e and the order $\underset{\mathbb{T}}{<}$ on $\mathbb{T}$ with priority given to the variable e and and "0" – the smallest term:

Definition 2.6.1 (the order on ${}_e\mathbb{T}$). For any $s_1, s_2 \in {}_e\mathbb{T}$ we define $\mathbf{s_1 \underset{e\mathbb{T}}{<} s_2}$ if and only if

$$
\begin{aligned}
s_2 \neq 0 \ \text{ and } \ & a)\ s_1 = 0 \ \text{ or} \\
& b)\ s_1 \neq 0 \ \text{ and } \ b_1) \deg_e(s_2) < \deg_e(s_1) \ \text{ or} \\
& \qquad\qquad\qquad b_2) \deg_e(s_2) = \deg_e(s_1) \ \text{ and } \ \mathrm{T}(s_1) \underset{\mathbb{T}}{<} \mathrm{T}(s_2).
\end{aligned}
$$

The next lemma describes the specific properties of the order $\underset{e\mathbb{T}}{<}$ under multiplication of terms.

Lemma 2.6.1. Let $a, b, c \in {}_e\mathbb{T}^*$ and $b \underset{e\mathbb{T}}{<} a$. Then

$$
\begin{aligned}
& a)\ \deg_e(a) \le \deg_e(b); \\
& b)\ ac = 0 \quad \Longrightarrow \quad bc = 0.
\end{aligned}
$$

Proof.

$a)$ Assume for a contradiction that $\deg_e(a) > \deg_e(b)$. Then, by part $b_1)$ of definition 2.6.1 of the order $\underset{e\mathbb{T}}{<}$, $b \underset{e\mathbb{T}}{<} a$.

$b)$ By part $a)$ of lemma 2.3.1, for any $s_1, s_2 \in {}_e\mathbb{T}$

$$
s_1 s_2 = 0 \ \text{ if and only if } \ \mathcal{B} \le \deg_e(s_1) + \deg_e(s_2). \qquad (*)
$$

On the other hand, by part a), $\deg_e(a) \le \deg_e(b)$. Therefore

$$ac = 0 \overset{(*)}{\iff} \mathcal{B} \le \deg_e(a) + \deg_e(c) \overset{a)}{\implies} \mathcal{B} \le \deg_e(b) + \deg_e(c) \overset{(*)}{\iff} bc = 0.$$

$\square$

To borrow the general theory of Gröbner bases in the algebra of extended polynomials to ${}_e\mathcal{P}$ the order $\underset{e\mathbb{T}}{<}$ must be quasi-monotonous (in the sense of definition 2.3.7) and a well-order. The next lemma states a slightly different from the quasi-monotonicity condition on $\underset{e\mathbb{T}}{<}$ which can be immediately turned to the quasi-monotonicity as it is done in the subsequent corollary.

Lemma 2.6.2 (refined quasi-monotonicity of $\underset{e\mathbb{T}}{<}$). Let $a, b, c, d \in {}_e\mathbb{T}^*$ and $b \underset{e\mathbb{T}}{<} a$, $d \underset{e\mathbb{T}}{<} c$.

$$\text{If } bd \neq 0 \text{ then } ac \neq 0 \text{ and } bd \underset{e\mathbb{T}}{<} ac.$$

Proof. Recall the rule from lemma 2.3.1 tacitly used further in the proof: for any $s_1, s_2 \in {}_e\mathbb{T}$

- $s_1 s_2 \neq 0$ if and only if $\deg_e(s_1) + \deg_e(s_2) < \mathcal{B}$,
- if $s_1 s_2 \neq 0$ then $\deg_e(s_1 s_2)$, $\mathrm{T}(s_1 s_2)$ are defined and

$$\begin{aligned} \deg_e(s_1 s_2) &= \deg_e(s_1) + \deg_e(s_2) && \in && \{0, \ldots, \mathcal{B}-1\} \subseteq \mathbb{N}_o, \\ \mathrm{T}(s_1 s_2) &= \mathrm{T}(s_1)\mathrm{T}(s_2) && \in && {}_e\mathbb{T}. \end{aligned}$$

Consider terms $a, b, c, d \in {}_e\mathbb{T}^*$ such that $b \underset{e\mathbb{T}}{<} a$, $d \underset{e\mathbb{T}}{<} c$ and $bd \neq 0$. Then, by part a) of lemma 2.6.1,

$$\deg_e(a) \le \deg_e(b), \quad \deg_e(c) \le \deg_e(d) \tag{1}$$

and hence

$$\deg_e(a) + \deg_e(c) \overset{(1)}{\le} \deg_e(b) + \deg_e(d) < \mathcal{B}. \tag{2}$$

By (2)

$$ac \neq 0 \text{ and } \deg_e(ac) \overset{ac \neq 0}{=} \deg_e(a) + \deg_e(c) \overset{(2)}{\le} \\ \deg_e(b) + \deg_e(d) \overset{bd \neq 0}{=} \deg_e(bd). \tag{3}$$

Consider two cases.

$(<):$ If $\deg_e(ac) < \deg_e(bd)$ then $bd \underset{e\mathbb{T}}{<} ac$ by part b_1) of definition 2.6.1 of $\underset{e\mathbb{T}}{<}$.

$(=):$ If $\deg_e(ac) = \deg_e(bd)$ then

$$\deg_e(a) \overset{(1)}{\leq} \deg_e(b) = \deg_e(ac) - \deg_e(d) \overset{(1)}{\leq} \deg_e(ac) - \deg_e(c) = \deg_e(a),$$
$$\deg_e(c) \overset{(1)}{\leq} \deg_e(d) = \deg_e(ac) - \deg_e(b) \overset{(1)}{\leq} \deg_e(ac) - \deg_e(a) = \deg_e(c).$$

Therefore

$$\deg_e(a) = \deg_e(b) \text{ and } \deg_e(c) = \deg_e(d). \tag{4}$$

Hence

$$ad \neq 0 \text{ because } \deg_e(a) + \deg_e(d) \overset{(4)}{=} \deg_e(a) + \deg_e(c) \overset{(2)}{<} \mathcal{B}.$$

It is easy to see that the only possibility left in definition 2.6.1 of $\underset{e\mathbb{T}}{<}$ to have $b \underset{e\mathbb{T}}{<} a$, $d \underset{e\mathbb{T}}{<} c$ together with (4) is the rule b_2). Thus

$$\mathrm{T}(b) \underset{\mathbb{T}}{<} \mathrm{T}(a), \quad \mathrm{T}(d) \underset{\mathbb{T}}{<} \mathrm{T}(c) \tag{5}$$

and

$$\mathrm{T}(bd) \overset{bd \neq 0}{=} \mathrm{T}(b)\mathrm{T}(d) \overset{(5)}{\underset{\mathbb{T}}{<}} \mathrm{T}(a)\mathrm{T}(d) \overset{ad \neq 0}{=} \mathrm{T}(ad),$$
$$\mathrm{T}(ad) \overset{ad \neq 0}{=} \mathrm{T}(a)\mathrm{T}(d) \overset{(5)}{\underset{\mathbb{T}}{<}} \mathrm{T}(a)\mathrm{T}(c) \overset{ac \neq 0}{=} \mathrm{T}(ac)$$

which follows

$$\mathrm{T}(bd) \underset{\mathbb{T}}{<} \mathrm{T}(ac)$$

and hence $bd \underset{e\mathbb{T}}{<} ac$ by part b_2) of definition 2.6.1.

□

Corollary 2.6.1 (quasi-monotonicity of ${}_e\mathbb{T}$). Let $b, d \in {}_e\mathbb{T}$, $a, c \in {}_e\mathbb{T}^*$ and $b \underset{e\mathbb{T}}{<} a$, $d \underset{e\mathbb{T}}{<} c$.

If $ac \neq 0$ then $bd \underset{e\mathbb{T}}{<} ac$.

Proof. If $bd{\neq}0$ then clearly $b,d{\in}\,{}_e\mathbb{T}^*$ and hence, by lemma 2.6.2, $bd\underset{e\mathbb{T}}{<}ac$. If $bd{=}0$ then, by part *a*) of definition 2.6.1 of the order $\underset{e\mathbb{T}}{<}$, $\big[0=\big]\ bd\underset{e\mathbb{T}}{<}ac\ \big[\neq 0\big]$. □

Now we show that the order $\underset{e\mathbb{T}}{<}$ is a well-ordering on extended terms.

Lemma 2.6.3. The order $\underset{e\mathbb{T}}{<}$ is a well-ordering on ${}_e\mathbb{T}$ i.e. every non-empty subset $S{\subseteq}\,{}_e\mathbb{T}$ has the smallest element.

Proof. Let $\emptyset{\neq}S{\subseteq}\,{}_e\mathbb{T}$. If $0{\in}S$ then, by definition 2.6.1 of the order $\underset{e\mathbb{T}}{<}$, 0 is the smallest element of S. Assume $S{\subseteq}\,{}_e\mathbb{T}^*$. For each $s{\in}S$, $\deg_e(s)$ takes a value from the finite set $\{0,\ldots,\mathcal{B}{-}1\}{\subseteq}\mathbb{N}$. Therefore there exists

$$d = \max\Big(\{\deg_e(s) \mid s{\in}S\}\Big).$$

Consider the set

$$S_d = \{s{\in}S \mid \deg_e(s){=}d\} \neq \emptyset.$$

Let

$$\emptyset \neq M = \mathrm{T}(S_d) = \{\mathrm{T}(s) \mid s{\in}S_d\} \subseteq \mathbb{T}.$$

Since the term order $\underset{\mathbb{T}}{<}$ on $\mathbb{T}$ is a well-ordering, there exists the smallest element of the non-empty set M:

$$v = \min_{\underset{\mathbb{T}}{<}}(M). \tag{1}$$

Then the element

$$m = v{\cdot}e^d \in S_d \tag{2}$$

is the smallest element of S w.r.t. the order $\underset{e\mathbb{T}}{<}$. Indeed, consider an arbitrary $s{\in}S$. By assumption, $s{\neq}0$.

a) If $s{\in}S{\setminus}S_d$ then $\deg_e(s){<}\deg_e(m)$ by construction of S_d and hence $m\underset{e\mathbb{T}}{<}s$ by part b_1) of definition 2.6.1.

b) If $s{\in}S_d$ then, by (2), $\deg_e(m){=}\deg_e(s)$ and hence, by (1), $\mathrm{T}(m)\underset{\mathbb{T}}{\leq}\mathrm{T}(s)$ and therefore either $m{=}s$ or $m\underset{e\mathbb{T}}{<}s$ by part b_2) of definition 2.6.1. □

Lemma 2.6.4. The order $\underset{e\mathbb{T}}{<}$ is an extended term order on ${}_e\mathbb{T}$.

Proof. By definition 2.6.1 of the order $\underset{e\mathbb{T}}{<}$, 0 is the $\underset{e\mathbb{T}}{<}$-smallest term. By corollary 2.6.1, $\underset{e\mathbb{T}}{<}$ is quasi-monotonous on ${}_e\mathbb{T}$ and, by lemma 2.6.3, $\underset{e\mathbb{T}}{<}$ is a well-ordering on ${}_e\mathbb{T}$. Hence $\underset{e\mathbb{T}}{<}$ is an extended term order on ${}_e\mathbb{T}$ according to definition 2.3.9. □

The particular properties of the order $\underset{e\mathbb{T}}{<}$ formulated in lemma 2.6.1 have an important consequence for multiplication of polynomials in ${}_e\mathcal{P}$ which generally means that the leading term function in ${}_e\mathcal{P}\backslash\{0\}$ behaves much like the "usual" leading term in the classical theory of Gröbner bases: for any pair of polynomials in ${}_e\mathcal{P}$ the leading term of the product keeps the stability unless the product is zero. It can be proved directly on the basis of definition 2.6.1 of the order $\underset{e\mathbb{T}}{<}$ but the fact that $\underset{e\mathbb{T}}{<}$ is quasi-monotonous makes the proof a good deal easier since it allows to borrow the properties of the leading term function under multiplication of polynomials in an arbitrary algebra of extended polynomials as it is written in lemma 2.3.11.

Corollary 2.6.2 (from part *b*) of lemma 2.6.1). Let $f, g \in {}_e\mathcal{P}\backslash\{0\}$. The following holds.

a) $f{\cdot}g{=}0$ if and only if $\ell\mathrm{t}(f){\cdot}\ell\mathrm{t}(g){=}0$;

b) if $f{\cdot}g{\neq}0$ then $\ell\mathrm{t}(f{\cdot}g) = \ell\mathrm{t}(f){\cdot}\ell\mathrm{t}(g)$.

Proof.

a)

($\Longleftarrow$) Assume $\ell\mathrm{t}(f){\cdot}\ell\mathrm{t}(g){=}0$. By part *c*) of lemma 2.3.3,

$$\operatorname{supp}(f{\cdot}g) = \operatorname{supp}(f) * \operatorname{supp}(g) = \Big\{s_1{\cdot}s_2 \mid s_1{\in}\operatorname{supp}(f)\,,\ s_2{\in}\operatorname{supp}(g)\Big\}\backslash\{0\}. \quad (*)$$

We claim that

$$\Big\{s_1{\cdot}s_2 \mid s_1{\in}\operatorname{supp}(f)\,,\ s_2{\in}\operatorname{supp}(g)\Big\} = \{0\}. \qquad (**)$$

Indeed, take any $s_1{\in}\operatorname{supp}(f)$ and $s_2{\in}\operatorname{supp}(g)$. If $s_1{=}\ell\mathrm{t}(f)$ and $s_2{=}\ell\mathrm{t}(g)$ then $s_1{\cdot}s_2{=}\ell\mathrm{t}(f){\cdot}\ell\mathrm{t}(g){=}0$ by assumption. Otherwise, by definition of the leading term of a polynomial as the maximal element of the support, $s_1 \underset{e\mathbb{T}}{<} \ell\mathrm{t}(f)$ or $s_2 \underset{e\mathbb{T}}{<} \ell\mathrm{t}(g)$. W.l.o.g assume that $s_1 \underset{e\mathbb{T}}{<} \ell\mathrm{t}(f)$. There arise two cases.

(1) If $s_2{=}\ell\mathrm{t}(g)$ then, by part $b)$ of lemma 2.6.1, the condition $s_1 \underset{e\mathbb{T}}{<} \ell\mathrm{t}(f)$ and the assumption $\ell\mathrm{t}(f){\cdot}s_2{=}\ell\mathrm{t}(f)\ell\mathrm{t}(g){=}0$ follow $s_1{\cdot}s_2{=}0$.

(2) Let $s_2 \underset{e\mathbb{T}}{<} \ell\mathrm{t}(g)$. According to case (1), $s_1{\cdot}\ell\mathrm{t}(g){=}0$. Then, by part $b)$ of lemma 2.6.1, the conditions $s_2 \underset{e\mathbb{T}}{<} \ell\mathrm{t}(g)$ and $s_1{\cdot}\ell\mathrm{t}(g){=}0$ follow $s_1{\cdot}s_2{=}0$.

It is easy to see that we have considered all possibilities for a pair of terms in $\mathrm{supp}\,(f) \times \mathrm{supp}\,(g)$ and hence (2) is true. Thus

$$\mathrm{supp}\,(f{\cdot}g) \overset{(*)}{=} \Big\{ s_1{\cdot}s_2 \mid s_1{\in}\,\mathrm{supp}\,(f)\,,\ s_2{\in}\,\mathrm{supp}\,(g) \Big\} \backslash \{0\} \overset{(**)}{=} \{0\} \backslash \{0\} = \emptyset$$

and therefore $f{\cdot}g{=}0$ by the uniqueness of the support.

$(\Longrightarrow)$ If $\ell\mathrm{t}(f){\cdot}\ell\mathrm{t}(g){\neq}0$ then, by lemma 2.3.11, $f{\cdot}g{\neq}0$.

$b)$ If $f{\cdot}g{\neq}0$ then, by part $a)$, $\ell\mathrm{t}(f){\cdot}\ell\mathrm{t}(g){\neq}0$ and hence, by lemma 2.3.11, $\ell\mathrm{t}(f{\cdot}g){=}\ell\mathrm{t}(f){\cdot}\ell\mathrm{t}(g)$. $\square$

Since $\underset{e\mathbb{T}}{<}$ is an extended term order on ${}_e\mathbb{T}{\subsetneq}{}_e\mathcal{P}$, the whole Gröbner bases theory and the algorithm for computing a Gröbner basis in the algebra of extended polynomials are applicable for computing Gröbner bases in ${}_e\mathcal{P}$. if the result of a Gröbner basis computation in ${}_e\mathcal{P}$ has a specific property discussed below then can be "e-specialized" to a Gröbner basis in $\Bbbk[X]$ by the use of a partial evaluation at $e{=}0$ in the same spirit as it is done for the algebra ${}_e\mathcal{P}$ in section 2.5.2. To establish the counterpart of the e-specialization for the algebra ${}_e\mathcal{P}$ we would need a kind of the e-grading structure in ${}_e\mathcal{P}$ discussed in the next subsection.

2.6.1 The $e-$graded structure of ${}_e\mathcal{P}$

It is easy to see by the linear independence of non-zero extended terms ${}_e\mathbb{T}^*$ (part c) of theorem 2.3.1) and the fact that the e-degree is a well-defined surjective map ${}_e\mathbb{T}{\supseteq}\,{}_e\mathbb{T}^*{\to}[0,\ldots,\mathcal{B}{-}1]{\subseteq}\mathbb{N}_o$ (according to part d) of theorem 2.3.1) that ${}_e\mathbb{T}^*$ can be represented as a disjoint union

$$\begin{aligned} {}_e\mathbb{T}^* &= \bigsqcup_{0\le k<\mathcal{B}} {}_e\mathbb{T}_k \quad \text{where} \quad {}_e\mathbb{T}_k \ \text{ denotes the set} \\ {}_e\mathbb{T}_k &= \Big\{ s{\in}\,{}_e\mathbb{T}^* \mid \deg_e(s){=}k \Big\} \ \Big[\subseteq {}_e\mathbb{T}^* \Big] \ \text{ for each } \ 0{\le}k{<}\mathcal{B}. \end{aligned}$$

On the other hand, by construction of the algebra of extended polynomials, the terms ${}_e\mathbb{T}^*$ generate ${}_e\mathcal{P}$ as a $\Bbbk$-linear vector space: ${}_e\mathcal{P}=\langle {}_e\mathbb{T}^*\rangle_{\Bbbk}$ and hence

$$
\begin{aligned}
{}_e\mathcal{P} &= \bigoplus_{0\le k<\mathcal{B}} {}_e\mathcal{P}_k \quad \text{where} \quad {}_e\mathcal{P}_k \quad \text{denotes the } \Bbbk\text{-linear vector space} \\
{}_e\mathcal{P}_k &= \langle {}_e\mathbb{T}_k\rangle_{\Bbbk} \qquad \text{spanned by the terms } {}_e\mathbb{T}_k,\ \ 0\le k<\mathcal{B}.
\end{aligned}
$$

It is clear by the direct $\Bbbk$-linear vector space sum decomposition of ${}_e\mathcal{P}$ into the *e-graded components* ${}_e\mathcal{P}_k$ with $0\le k<\mathcal{B}$ that every polynomial in $f\in{}_e\mathcal{P}$ has the unique representation as

$$
f=\sum_{0\le k<\mathcal{B}} \mathrm{gr}^e_k(f) \quad \text{with} \quad \mathrm{gr}^e_k(f)\in{}_e\mathcal{P}_k \quad \text{for each} \quad 0\le k<\mathcal{B}.
$$

Furthermore, $\mathrm{supp}\left(\mathrm{gr}^e_k(f)\right)\subseteq {}_e\mathbb{T}_k$ and hence

$$
\mathrm{supp}\left(\mathrm{gr}^e_{k_1}(f)\right)\bigcap\mathrm{supp}\left(\mathrm{gr}^e_{k_2}(f)\right)=\emptyset \quad \text{for any} \quad 0\le k_1,k_2<\mathcal{B} \quad \text{such that} \quad k_1\ne k_2.
$$

Then it is easy to see by the inductive argument based on part *b*) of lemma 2.3.3 that

$$
\bigsqcup_{0\le k<\mathcal{B}}\mathrm{supp}\left(\mathrm{gr}^e_k(f)\right)=\mathop{\triangle}_{0\le k<\mathcal{B}}\mathrm{supp}\left(\mathrm{gr}^e_k(f)\right)\subseteq \mathrm{supp}\,(f)\subseteq\bigsqcup_{0\le k<\mathcal{B}}\mathrm{supp}\left(\mathrm{gr}^e_k(f)\right)
$$

and hence

$$
\mathrm{supp}\,(f)=\bigsqcup_{0\le k<\mathcal{B}}\mathrm{supp}\left(\mathrm{gr}^e_k(f)\right).
$$

In particular,

$$
\mathrm{supp}\,(f)=\emptyset \quad \text{if and only if} \quad \mathrm{supp}\left(\mathrm{gr}^e_k(f)\right)=\emptyset \quad \text{for all} \quad 0\le k<\mathcal{B}
$$

and hence, by the uniqueness of the support,

$$
f=0 \quad \text{if and only if} \quad \mathrm{gr}^e_k(f)=0 \quad \text{for all} \quad 0\le k<\mathcal{B}
$$

which is also clear according to the direct sum decomposition of ${}_e\mathcal{P}$. Another useful consequence of the disjoint decomposition $\mathrm{supp}\,(f)=\bigsqcup_{0\le k<\mathcal{B}}\mathrm{supp}\left(\mathrm{gr}^e_k(f)\right)$ is the following:

$$
{}_e\mathbb{T}_k\supseteq\mathrm{supp}\left(\mathrm{gr}^e_k(f)\right)\subseteq\mathrm{supp}\,(f)\subseteq{}_e\mathbb{T}^*
$$

and hence

$$\begin{aligned}
\boxed{\operatorname{supp}\left(\mathrm{gr}_k^e(f)\right)} &= \operatorname{supp}\left(\mathrm{gr}_k^e(f)\right) \cap {}_e\mathbb{T}_k \\
&\subseteq \boxed{\operatorname{supp}(f) \cap {}_e\mathbb{T}_k} = \Big(\bigsqcup_{0\le i<\mathcal{B}} \operatorname{supp}\left(\mathrm{gr}_i^e(f)\right)\Big) \cap {}_e\mathbb{T}_k \\
&= \bigsqcup_{0\le i<\mathcal{B}} \Big(\operatorname{supp}\left(\mathrm{gr}_i^e(f)\right) \cap {}_e\mathbb{T}_k\Big) \\
&= \boxed{\operatorname{supp}\left(\mathrm{gr}_k^e(f)\right)}
\end{aligned}$$

which means

$$\operatorname{supp}\left(\mathrm{gr}_k^e(f)\right) = \operatorname{supp}(f) \cap {}_e\mathbb{T}_k \quad \text{for any} \quad 0\le k<\mathcal{B}.$$

The polynomials in the e-graded components ${}_e\mathcal{P}_k$ are important enough to deserve a separate notion.

Definition 2.6.2. A polynomial in ${}_e\mathcal{P}_k$ with $0\le k<\mathcal{B}$ is called ***e*-homogenous of degree** k. Note that 0 is trivially e-homogenous of any e-degree. A polynomial $f\in{}_e\mathcal{P}$ is called ***e*-homogenous** if and only if there exists an e-degree k such that $f\in{}_e\mathcal{P}_k$. It is clear by the direct decomposition ${}_e\mathcal{P}=\bigoplus_{0\le k<\mathcal{B}} {}_e\mathcal{P}_k$ that $f\neq 0$ is e-homogenous if an only if the decomposition $f=\sum_{0\le k<\mathcal{B}} \mathrm{gr}_k^e(f)$ consists of exactly one non-zero summand.

The notion of an e-homogenous polynomial has the following properties under the ring operations in ${}_e\mathcal{P}$.

Lemma 2.6.5. Let $f,g\in{}_e\mathcal{P}\setminus\{0\}$ be e-homogenous polynomials of e-degree k_1 and k_2 respectively.

a) For any $\alpha\in\Bbbk$ the product $\alpha\cdot f$ is e-homogeneous of degree k_1.

b) The sum $f+g$ is e-homogenous if and only if $k_1=k_2$. If $k_1=k_2$ then $f+g$ is e-homogenous of degree $k_1=k_2$.

c) The product $f\cdot g$ is e-homogenous. Furthermore, if $k_1+k_2<\mathcal{B}$ then $f\cdot g\neq 0$ is e-homogenous of degree k_1+k_2.

Proof. Note that $f\in{}_e\mathcal{P}_{k_1}$ and $g\in{}_e\mathcal{P}_{k_2}$.

a) Since ${}_e\mathcal{P}_{k_1}$ is a $\Bbbk$-linear vector space, $\alpha\cdot f\in{}_e\mathcal{P}_{k_1}$ and hence $\alpha\cdot f$ is e-homogenous of degree k_1.

$b)$ If $k{=}k_1{=}k_2$ then $f{+}g{\in}_e\mathcal{P}_k$ is e-homogenous of degree k provided that $_e\mathcal{P}_k$ is a $\Bbbk$-linear vector space. On the other hand, if $k_1{\neq}k_2$ then, from $_e\mathbb{T}_{k_1}{\cap}\,_e\mathbb{T}_{k_2}$ and part $b)$ of lemma 2.3.3,

$$\begin{aligned}\operatorname{supp}(f)\sqcup\operatorname{supp}(g) &= \operatorname{supp}(f)\triangle\operatorname{supp}(g)\\ &\subseteq \operatorname{supp}(f{+}g)\subseteq\operatorname{supp}(f)\sqcup\operatorname{supp}(g)\end{aligned}$$

end hence

$$\operatorname{supp}(f{+}g) = \operatorname{supp}(f)\sqcup\operatorname{supp}(g)\,.$$

Therefore, there exist $s_1, s_2{\in}\operatorname{supp}(f{+}g)$ such that $s_1{\in}\operatorname{supp}(f)\subseteq{}_e\mathbb{T}_{k_1}$ and $s_s{\in}\operatorname{supp}(g)\subseteq{}_e\mathbb{T}_{k_2}$ provided by the assumption $f{\neq}0$ and $g{\neq}0$ which follows $_e\mathbb{T}_{k_1}{\sqcup}\,_e\mathbb{T}_{k_2}{\subseteq}\operatorname{supp}(f{+}g)\nsubseteq{}_e\mathbb{T}_{k_1},\ _e\mathbb{T}_{k_2}$ and hence $f{+}g$ is not e-homogenous.

$c)$ Let $k{=}k_1{+}k_2$ By part $c)$ of lemma 2.3.3,

$$\operatorname{supp}(f{\cdot}g) = \operatorname{supp}(f)*\operatorname{supp}(g)\,. \tag{$*$}$$

Then, for any $s{\in}\operatorname{supp}(f{\cdot}g)$ there exist $s_1{\in}\operatorname{supp}(f)$ and $s_2{\in}\operatorname{supp}(g)$ such that $s{=}s_1{\cdot}s_2[{\neq}0]$ provided by the assumption $f{\neq}0$ and $g{\neq}0$ and hence, by part $b)$ of lemma 2.3.1,

$$\deg_e(s){=}\deg_e(s_1){+}\deg_e(s_2){=}k_1{+}k_2{=}k. \tag{$**$}$$

If $k{\geq}\mathcal{B}$ then, by $(*)$ and part $a)$ of lemma 2.3.1,

$$\begin{aligned}\operatorname{supp}(f{\cdot}g) &\overset{(*)}{=} \operatorname{supp}(f)*\operatorname{supp}(g)\\ &{=}\Big\{s_1{\cdot}s_2 \mid s_1{\in}\operatorname{supp}(f)\,,\ s_2{\in}\operatorname{supp}(g)\Big\}\backslash\{0\} \qquad \text{(by lemma 2.3.1)}\\ &{=}\{0\}\backslash\{0\} = \emptyset\end{aligned}$$

and then $f{\cdot}g{=}0$ is trivially e-homogenous. On the other hand, if $k{<}\mathcal{B}$ then, by $(*)$, $(**)$ and part $a)$ of lemma 2.3.1,

$$\begin{aligned}\operatorname{supp}(f{\cdot}g) &\overset{(*)}{=} \operatorname{supp}(f)*\operatorname{supp}(g) \qquad \text{(by } k{<}\mathcal{B} \text{ and lemma 2.3.1)}\\ &= \underbrace{\Big\{s_1{\cdot}s_2 \mid s_1{\in}\operatorname{supp}(f)\,,\ s_2{\in}\operatorname{supp}(g)\Big\}}_{\substack{\nparallel \text{ due to } f\neq0 \text{ and } g\neq0\\ \emptyset}} \overset{(**)}{\subseteq} {}_e\mathbb{T}_k.\end{aligned}$$

and hence

$$0 \neq f{\cdot}g{\in}\langle\,_e\mathbb{T}_k\rangle_{\Bbbk} = {}_e\mathcal{P}_k \ \text{ is } e\text{-homogenous of degree } k. \qquad \square$$

The particularly interesting for us e-homogenous polynomials arise as the lowest e-degree non-zero summand in the unique decomposition of a polynomial by the e-graded components.

Definition 2.6.3. For any $f \in {}_e\mathcal{P} \setminus \{0\}$ the **e-order of** f is the lowest e-degree within the terms in the support of f:

$$\mathrm{ord}_e(f) = \min\Big(\deg_e(s) \mid s \in \mathrm{supp}\,(f)\Big).$$

By convention, $\mathrm{ord}_e(0)=0$.

Lemma 2.6.6. Let $f \in {}_e\mathcal{P}$. Then $\mathrm{gr}^e_k(f)=0$ for any e-degree $k<\mathrm{ord}_e(f)$

Proof. The claim is trivial if $f=0$. Assume $f \neq 0$. Let $k<\mathrm{ord}_e(f)$. If $\mathrm{gr}^e_k(f) \neq 0$ then, by the uniqueness of the support, $\mathrm{supp}\,\big(\mathrm{gr}^e_k(f)\big) \neq \emptyset$, hence there exists $s \in \mathrm{supp}\,\big(\mathrm{gr}^e_k(f)\big) \subseteq {}_e\mathbb{T}_k$ and, by construction of ${}_e\mathbb{T}_k$, $\deg_e(s)=k<\mathrm{ord}_e(f)$ which is, due to the inclusion $\mathrm{supp}\,\big(\mathrm{gr}^e_k(f)\big) \subseteq \mathrm{supp}\,(f)$, a contradiction with the definition of $\mathrm{ord}_e(f)$ as the minimal e-degree within the terms in $\mathrm{supp}\,(f)$. $\square$

Corollary 2.6.3 (from lemma 2.6.6). For any $f \in {}_e\mathcal{P}$ there exists the unique decomposition

$$f = \sum_{\mathrm{ord}_e(f) \leq k < \mathcal{B}} \mathrm{gr}^e_k(f) \quad \text{with} \quad \mathrm{gr}^e_k(f) \in {}_e\mathcal{P}_k \quad \text{for each} \quad \mathrm{ord}_e(f) \leq k < \mathcal{B}.$$

Proof. For any polynomial $f \in {}_e\mathcal{P}$ there exists the unique decomposition

$$f = \sum_{0 \leq k < \mathcal{B}} \mathrm{gr}^e_k(f) \quad \text{with} \quad \mathrm{gr}^e_k(f) \in {}_e\mathcal{P}_k \quad \text{for each} \quad 0 \leq k < \mathcal{B}.$$

By lemma 2.6.6, $\mathrm{gr}^e_k(f)=0$ for each $0 \leq k < \mathrm{ord}_e(f)$. $\square$

Definition 2.6.4. For any $f \in {}_e\mathcal{P}$ the **e-initial of** f is the e-homogenous of degree $\mathrm{ord}_e(f)$ polynomial

$$\mathrm{in}_e(f) = \mathrm{gr}^e_{\mathrm{ord}_e(f)}(f) \ \big[\in {}_e\mathcal{P}_k\big].$$

It is clear that $\mathrm{supp}\,\big(\mathrm{in}_e(f)\big) \subseteq \mathrm{supp}\,(f)$. The e-initials of a set of polynomials $F \subseteq {}_e\mathcal{P}$ is the set of e-initials of the individual elements of F:

$$\mathrm{in}_e(F) = \Big\{\mathrm{in}_e(f) \mid f \in F\Big\}.$$

The e-grading and the notion of the e-initial of an extended polynomial do not incorporate the order $\underset{e\mathbb{T}}{<}$ and ${}_e\mathcal{P}$ and hence they can be defined for any algebra of extended of polynomials. But the order $\underset{e\mathbb{T}}{<}$ which gives the priority to an extended term with a lower $e-$degree makes the e-grading and the notion of the e-initial particularly meaningful in conjunction with the Gröbner bases theory in ${}_e\mathcal{P}$ since the order $\underset{e\mathbb{T}}{<}$ turns to be anti-compatible with the e-grading, the e-initial appears to be a kind of a generalization of the leading term function on ${}_e\mathcal{P}\backslash\{0\}$. The next lemma states these interrelationships precisely.

Lemma 2.6.7. For any $f{\in}_e\mathcal{P}$

a) $\mathrm{in}_e(f){=}0$ if and only if $f{=}0$,

b) if $f{\neq}0$ then $\ell\mathrm{m}\big(\mathrm{in}_e(f)\big){=}\ell\mathrm{m}(f)$,

c) if $f{\neq}0$ then $\deg_e\big(\ell\mathrm{t}(f)\big){=}\mathrm{ord}_e(f)$.

Proof.

a) If $f{=}0$ then

$$\mathrm{supp}\left(\mathrm{in}_e(f)\right)=\mathrm{supp}\left(\mathrm{gr}^e_{\mathrm{ord}_e(f)}(f)\right)\subseteq\mathrm{supp}\,(f)=\emptyset$$

and hence $\mathrm{in}_e(f){=}0$.

If $f{\neq}0$ then $\mathrm{supp}\,(f)\neq\emptyset$ and, by definition of $\mathrm{ord}_e(f)$, there exists $s{\in}\,\mathrm{supp}\,(f)$ such that $\deg_e(s){=}\mathrm{ord}_e(f)$, hence $s{\in}\,{}_e\mathbb{T}_{\mathrm{ord}_e(f)}$ which follows

$$\mathrm{supp}\left(\mathrm{in}_e(f)\right)=\mathrm{supp}\left(\mathrm{gr}^e_{\mathrm{ord}_e(f)}(f)\right)=\mathrm{supp}\,(f)\cap{}_e\mathbb{T}_{\mathrm{ord}_e(f)}\supseteq\{s\}\neq\emptyset$$

and hence $\mathrm{in}_e(f){\neq}0$ by the uniqueness of the support.

b) For any $e-$degree $k{<}\mathrm{ord}_e(f)$

$$\mathrm{supp}\left(\mathrm{gr}^e_k(f)\right)=\mathrm{supp}\,(f)\cap{}_e\mathbb{T}_k=\emptyset$$

(cf. lemma 2.6.6) and hence

$$\mathrm{supp}\,(f)=\bigsqcup_{\mathrm{ord}_e(f)\leq k<\mathcal{B}}\mathrm{supp}\left(\mathrm{gr}^e_k(f)\right).$$

Note that $\ell\mathrm{t}(f){\in}\,\mathrm{supp}\,(f)$. If

$$\ell\mathrm{t}(f){\notin}\,\mathrm{supp}\left(\mathrm{gr}^e_{\mathrm{ord}_e(f)}(f)\right)\quad\text{then}\quad\ell\mathrm{t}(f){\in}\bigsqcup_{\mathrm{ord}_e(f)<k<\mathcal{B}}\mathrm{supp}\left(\mathrm{gr}^e_k(f)\right)$$

and hence
$$\deg_e\Big(\ell\mathrm{t}(f)\Big) > \mathrm{ord}_e(f).$$
On the other hand, by part a),
$$\mathrm{supp}\left(\mathrm{gr}^e_{\mathrm{ord}_e(f)}(f)\right) \neq \emptyset \quad \text{provided that} \quad f \neq 0.$$
Hence there exists
$$s \in \mathrm{supp}\left(\mathrm{gr}^e_{\mathrm{ord}_e(f)}(f)\right) \subseteq \mathrm{supp}\,(f) \quad \text{and} \quad \deg_e(s) = \mathrm{ord}_e(f) < \mathrm{ord}_e\Big(\ell\mathrm{t}(f)\Big).$$
Then, by part b_1) of definition 2.6.1 of the order $\underset{e\mathbb{T}}{<}$, $\ell\mathrm{t}(f) \underset{e\mathbb{T}}{<} s$ which contradicts the choice of $\ell\mathrm{t}(f)$ as the $\underset{e\mathbb{T}}{<}$-maximal term in $\mathrm{supp}\,(f)$, therefore $\ell\mathrm{t}(f) \in \mathrm{supp}\,\big(\mathrm{in}_e(f)\big)$ due to $\mathrm{in}_e(f) = \mathrm{gr}^e_{\mathrm{ord}_e(f)}(f)$ by definition and hence, by the inclusion $\mathrm{supp}\,\big(\mathrm{in}_e(f)\big) \subseteq \mathrm{supp}\,(f)$,
$$\begin{array}{l} \boxed{\ell\mathrm{t}(f)} \underset{e\mathbb{T}}{\leq} \max\Big(\mathrm{supp}\,\big(\mathrm{in}_e(f)\big)\Big) \underset{e\mathbb{T}}{\leq} \max\big(\mathrm{supp}\,(f)\big) = \boxed{\ell\mathrm{t}(f)} \\ \qquad\qquad \| \\ \qquad \boxed{\ell\mathrm{t}\big(\mathrm{in}_e(f)\big)} \end{array}$$
which means $\ell\mathrm{t}\big(\mathrm{in}_e(f)\big) = \ell\mathrm{t}(f)$. On the other hand, for any e-degree k such that $k \neq \mathrm{ord}_e(f)$,
$$\begin{aligned} \Big\{\ell\mathrm{t}\big(\mathrm{in}_e(f)\big)\Big\} \bigcap \mathrm{supp}\left(\mathrm{gr}^e_k(f)\right) &\subseteq \mathrm{supp}\left(\mathrm{in}_e(f)\right) \bigcap \mathrm{supp}\left(\mathrm{gr}^e_k(f)\right) = \\ &\mathrm{supp}\left(\mathrm{gr}^e_{\mathrm{ord}_e(f)}(f)\right) \bigcap \mathrm{supp}\left(\mathrm{gr}^e_k(f)\right) \subseteq {}_e\mathbb{T}_{\mathrm{ord}_e(f)} \cap {}_e\mathbb{T}_k = \emptyset \end{aligned}$$
and hence $\ell\mathrm{m}(f) = \ell\mathrm{m}\big(\mathrm{in}_e(f)\big)$ by the linear independence of extended terms.

c) It is a direct consequence of the definition 2.6.1 of the term order $\underset{e\mathbb{T}}{<}$ which is degree-anti-compatible in e and hence the leading term of a (non-zero) polynomial must have the smallest e-degree within the terms in the support which is precisely the e-order of the polynomial according to definition 2.6.3. Alternatively, it is an easy consequence of part b) (in fact it is already proven!):
$$\deg_e\Big(\ell\mathrm{t}(f)\Big) = \deg_e\Big(\ell\mathrm{t}\big(\mathrm{in}_e(f)\big)\Big) = \deg_e\Big(\ell\mathrm{t}\big(\mathrm{gr}^e_{\mathrm{ord}_e(f)}(f)\big)\Big) = \mathrm{ord}_e(f)$$
due to the fact that $\mathrm{gr}^e_{\mathrm{ord}_e(f)}(f)$ is a $\Bbbk$-linear combination of extended terms of e-degree $\mathrm{ord}_e(f)$. $\square$

The following lemma summarizes the properties of the e-initial of a polynomial under the arithmetical operations with polynomials.

Lemma 2.6.8. Let $f, g \in {}_e\mathcal{P} \setminus \{0\}$.

a) For any $\alpha \in \Bbbk$ $\mathrm{in}_e(\alpha \cdot f) = \alpha \cdot \mathrm{in}_e(f)$.

b) If $\mathrm{ord}_e(f) = \mathrm{ord}_e(g)$ and $\mathrm{in}_e(f) + \mathrm{in}_e(g) \neq 0$ then $\mathrm{in}_e(f+g) = \mathrm{in}_e(f) + \mathrm{in}_e(g)$.

c) $\mathrm{in}_e(f \cdot g) = \mathrm{in}_e(f) \cdot \mathrm{in}_e(g)$.

Proof. The proof of each part uses the existence of the unique representation (cf. corollary 2.6.3)

$$h = \sum_{0 \le k < \mathcal{B}} \mathrm{gr}^e_k(h) = \sum_{\mathrm{ord}_e(h) \le k < \mathcal{B}} \mathrm{gr}^e_k(h) \quad \text{for any} \quad h \in {}_e\mathcal{P} \setminus \{0\} \tag{$*$}$$

a) If $\alpha = 0$ then

$$\mathrm{in}_e(0 \cdot f) = \mathrm{in}_e(0) = 0 = 0 \cdot \mathrm{in}_e(f).$$

Assume $\alpha \neq 0$. By $(*)$,

$$\alpha \cdot f \overset{(*)}{=} \alpha \cdot \sum_{\mathrm{ord}_e(f) \le d < \mathcal{B}} \mathrm{gr}^e_d(f) = \sum_{\mathrm{ord}_e(f) \le d < \mathcal{B}} \Big(\alpha \cdot \mathrm{gr}^e_d(f) \Big) \overset{(*)}{=} \sum_{0 \le d < \mathcal{B}} \mathrm{gr}^e_d(\alpha \cdot f)$$

and hence, by the uniqueness of $(*)$,

$$\mathrm{gr}^e_k(\alpha \cdot f) = 0 \quad \text{for any} \quad 0 \le k < \mathrm{ord}_e(f), \tag{1a}$$

$$\mathrm{gr}^e_{\mathrm{ord}_e(f)}(\alpha \cdot f) = \alpha \cdot \mathrm{gr}^e_{\mathrm{ord}_e(f)}(f). \tag{2a}$$

By part *a*) of lemma 2.6.7

$$\mathrm{in}_e(f) \neq 0 \text{ provided that } f \neq 0. \tag{3a}$$

Then, by definition 2.6.4 of the e-initial of a polynomial and lemma 2.3.9,

$$\mathrm{gr}^e_{\mathrm{ord}_e(f)}(\alpha \cdot f) \overset{(2a)}{=} \alpha \cdot \mathrm{gr}^e_{\mathrm{ord}_e(f)}(f) \overset{(\mathrm{D}\ 2.6.4)}{=} \alpha \cdot \mathrm{in}_e(f) \overset{\substack{(3a),\\ (L\ 2.3.9)}}{\neq} 0 \tag{4a}$$

and hence, by $(1a)$ and $(4a)$,

$$\mathrm{in}_e(\alpha \cdot f) \overset{\substack{(1a),\\ (4a)}}{=} \mathrm{gr}^e_{\mathrm{ord}_e(f)}(\alpha \cdot f) \overset{(4a)}{=} \alpha \cdot \mathrm{in}_e(f).$$

b) Denote $m{=}\deg_e(f){=}\deg_e(g)$. By $(*)$,

$$\begin{aligned} f+g &\overset{(*)}{=} \sum_{m\le i<\mathcal{B}} \mathrm{gr}^e_i(f) \ + \sum_{m\le j<\mathcal{B}} \mathrm{gr}^e_j(g) \\ &= \sum_{m\le d<\mathcal{B}} \Big(\mathrm{gr}^e_d(f) + \mathrm{gr}^e_d(g)\Big) \overset{(*)}{=} \sum_{0\le d<\mathcal{B}} \mathrm{gr}^e_d(f{+}g) \end{aligned}$$

and hence, by the uniqueness of $(*)$,

$$\mathrm{gr}^e_k(f{+}g) = 0 \quad \text{for any} \quad 0\le k<m, \tag{1b}$$

$$\mathrm{gr}^e_m(f{+}g) = \mathrm{gr}^e_m(f){+}\mathrm{gr}^e_m(g). \tag{2b}$$

Furthermore, according to definition 2.6.4 of the e-initial of a polynomial and the assumption $\mathrm{in}_e(f){+}\mathrm{in}_e(g){\neq}0$,

$$\mathrm{gr}^e_m(f{+}g) \overset{(2b)}{=} \mathrm{gr}^e_m(f){+}\mathrm{gr}^e_m(g) \overset{(\mathrm{D}\ 2.6.4)}{=} \mathrm{in}_e(f){+}\mathrm{in}_e(g){\neq}0 \tag{3b}$$

and hence, by (1*b*) and (3*b*),

$$\mathrm{in}_e(f{+}g) \overset{\substack{(1b),\\(3b)}}{=} \mathrm{gr}^e_m(f{+}g) \overset{(3b)}{=} \mathrm{in}_e(f){+}\mathrm{in}_e(g).$$

c) If $f{\cdot}g{=}0$ then, by part *a*) of corollary 2.6.2, $\ell\mathrm{t}(f){\cdot}\ell\mathrm{t}(g){=}0$, hence, by part *b*) of lemma 2.6.7,

$$\ell\mathrm{t}\Big(\mathrm{in}_e(f)\Big){\cdot}\ell\mathrm{t}\Big(\mathrm{in}_e(g)\Big) \overset{(L2.6.7)}{=} \ell\mathrm{t}(f){\cdot}\ell\mathrm{t}(g) = 0$$

and therefore, using part *a*) of corollary 2.6.2 once again, $\mathrm{in}_e(f){\cdot}\mathrm{in}_e(g){=}0$. By convention, $\mathrm{in}_e(0){=}0$ and hence

$$\mathrm{in}_e(f{\cdot}g) = \mathrm{in}_e(0) = 0 = \mathrm{in}_e(f){\cdot}\mathrm{in}_e(g).$$

Assume, $f{\cdot}g{\neq}0$. By $(*)$,

$$\begin{aligned} f{\cdot}g &\overset{(*)}{=} \left(\sum_{\mathrm{ord}_e(f)\le i<\mathcal{B}} \mathrm{gr}^e_i(f)\right) \cdot \left(\sum_{\mathrm{ord}_e(g)\le j<\mathcal{B}} \mathrm{gr}^e_j(g)\right) \\ &= \sum_{d=\mathrm{ord}_e(f)+\mathrm{ord}_e(g)}^{\mathcal{B}-1} \left(\sum_{\substack{\mathrm{ord}_e(f)\le i<\mathcal{B}\\ \mathrm{ord}_e(g)\le j<\mathcal{B}\\ i+j=d}} \mathrm{gr}^e_i(f){\cdot}\mathrm{gr}^e_j(g)\right) \\ &= \mathrm{gr}^e_{\mathrm{ord}_e(f)}(f){\cdot}\mathrm{gr}^e_{\mathrm{ord}_e(g)}(g) + \sum_{d=\mathrm{ord}_e(f)+\mathrm{ord}_e(g)+1}^{\mathcal{B}-1} \Big(\cdots\Big) \overset{(*)}{=} \sum_{0\le d<\mathcal{B}} \mathrm{gr}^e_d(f{\cdot}g) \end{aligned}$$

and hence, by the uniqueness of $(*)$,

$$\mathrm{gr}^e_k(f\cdot g) = 0 \quad \text{for any} \quad 0 \le k < \mathrm{ord}_e(f) + \mathrm{ord}_e(g), \tag{1c}$$

$$\mathrm{gr}^e_{\mathrm{ord}_e(f)+\mathrm{ord}_e(g)}(f\cdot g) = \mathrm{gr}^e_{\mathrm{ord}_e(f)}(f)\cdot\mathrm{gr}^e_{\mathrm{ord}_e(g)}(g). \tag{2c}$$

Furthermore, by part $a)$ of corollary 2.6.2, part $b)$ of lemma 2.6.7 and definition 2.6.4 of the e-initial of a polynomial,

$$\begin{aligned}
f\cdot g \neq 0 &\overset{\text{(C 2.6.2)}}{\Longrightarrow} \ell\mathrm{t}(f)\cdot\ell\mathrm{t}(g) \neq 0 \\
&\Longrightarrow \ell\mathrm{t}\Big(\mathrm{in}_e(f)\Big)\cdot\ell\mathrm{t}\Big(\mathrm{in}_e(g)\Big) \overset{\text{(L 2.6.7)}}{=} \ell\mathrm{t}(f)\cdot\ell\mathrm{t}(g) \neq 0 \\
&\overset{\text{(C 2.6.2)}}{\Longrightarrow} \mathrm{in}_e(f)\cdot\mathrm{in}_e(g) \neq 0 \\
&\Longrightarrow \mathrm{gr}^e_{\mathrm{ord}_e(f)+\mathrm{ord}_e(g)}(f\cdot g) \overset{(2c)}{=} \\
&\qquad \mathrm{gr}^e_{\mathrm{ord}_e(f)}(f)\cdot\mathrm{gr}^e_{\mathrm{ord}_e(g)}(g) \overset{\text{(D 2.6.4)}}{=} \\
&\qquad\qquad \mathrm{in}_e(f)\cdot\mathrm{in}_e(g) \neq 0.
\end{aligned} \tag{3c}$$

Finally, by $(1c)$ and $(3c)$,

$$\mathrm{in}_e(f\cdot g) \overset{(1c),\,(3c)}{=} \mathrm{gr}^e_{\mathrm{ord}_e(f)+\mathrm{ord}_e(g)}(f\cdot g) \overset{(3c)}{=} \mathrm{in}_e(f)\cdot\mathrm{in}_e(g). \qquad \square$$

It is quite straightforward to devise the characterization of e-graded ideals (i.e. the ideals generated e-homogenous elements) in analogy with [MK00] (proposition 1.7.10 on page 79) but our moderated usage of the e-grading does not require that general structural statement. Instead we present a pretty technical proposition about very particular e-graded ideals which is used in the subsequent Gröbner bases truncation theorem 2.6.1.

Lemma 2.6.9. Let I be an ideal in ${}_e\mathcal{P}$. Then the e-initials of the ideal in ${}_e\mathcal{P}$ generated by the e-initials of I are exactly the e-initials of I:

$$\mathrm{in}_e(I) = \mathrm{in}_e\Big(\big\langle \mathrm{in}_e(I)\big\rangle_{{}_e\mathcal{P}}\Big).$$

Proof.

$(\Longrightarrow)$ It is clear that

$$\mathrm{in}_e(I) \subseteq \big\langle \mathrm{in}_e(I)\big\rangle_{{}_e\mathcal{P}}$$

and hence

$$\mathrm{in}_e(I) = \mathrm{in}_e\Big(\mathrm{in}_e(I)\Big) \subseteq \mathrm{in}_e\Big(\big\langle \mathrm{in}_e(I)\big\rangle_{{}_e\mathcal{P}}\Big).$$

($\Longleftarrow$) Take any $0{\neq}g{\in}\big\langle \mathrm{in}_e(I)\big\rangle_{{}_e\mathcal{P}}$. Our goal is to show that $\mathrm{in}_e(g){\in}\mathrm{in}_e(I)$. Let $F{\subseteq}I\backslash\{0\}$ be a finite set such that

$$g=\sum_{f\in F} q_f{\cdot}\mathrm{in}_e(f) \quad \text{with} \quad q_f{\in}_e\mathcal{P}\backslash\{0\} \quad \text{for each} \quad f{\in}F. \tag{$*$}$$

Denote $d{=}\mathrm{ord}_e(g)$ and $d_f{=}\mathrm{ord}_e(f)$ for $f{\in}F$. By part definition 2.6.4 of the e-initial of a polynomial and part $a)$ of lemma 2.6.7, $0{\neq}\mathrm{in}_e(g){\in}_e\mathcal{P}_d$ is an $e-$homogeneous polynomial of degree d as well as each $0{\neq}\mathrm{in}_e(f){\in}_e\mathcal{P}_{d_f}$ is e-homogenous of degree d_f. By $(*)$

$$\begin{aligned} g \overset{(*)}{=} \sum_{f\in F} q_f{\cdot}\mathrm{in}_e(f) &= \sum_{f\in F}\left(\sum_{0\leq i<\mathcal{B}} \mathrm{gr}^e_i(q_f)\right){\cdot}\mathrm{in}_e(f) \\ &= \sum_{\substack{f\in F\\ 0\leq i<\mathcal{B}}} \mathrm{gr}^e_i(q_f){\cdot}\mathrm{in}_e(f) = \sum_{0\leq k<\mathcal{B}} \overbrace{\Big(\sum_{\substack{f\in F\\ 0\leq i<\mathcal{B}\\ d_f+i=k}} \mathrm{gr}^e_i(q_f){\cdot}\mathrm{in}_e(f)\Big)}^{g_k} = \sum_{0\leq k<\mathcal{B}} g_k. \end{aligned} \tag{$**$}$$

Take any $0{\leq}k{<}\mathcal{B}$. We claim that g_k is an e-homogenous polynomial of degree k and $g_k{\in}\mathrm{in}_e(I)$. The claim is trivial if $g_k{=}0$. Assume $g_k{\neq}0$. Consider any $f{\in}F$ and $0{\leq}i{<}\mathcal{B}$ such that $d_f{+}i{=}k$ and $\mathrm{gr}^e_i(q_f){\neq}0$ provided by $g_k{\neq}0$. Then, by part $c)$ of lemma 2.6.8 and part $c)$ of lemma 2.6.5,

$\mathrm{in}_e\big(\mathrm{gr}^e_i(q_f){\cdot}f\big) \overset{(\mathrm{L}\ 2.6.8)}{=} \mathrm{gr}^e_i(q_f){\cdot}\mathrm{in}_e(f)$ is e-homogenous of degree k (Lemma 2.6.5).

Hence, by the inductive argument based on part $b)$ of lemma 2.6.8,

$$g_k=\sum_{\substack{f\in F\\ 0\leq i<\mathcal{B}\\ d_f+i=k}} \mathrm{gr}^e_i(q_f){\cdot}\mathrm{in}_e(f) = \sum_{\substack{f\in F\\ 0\leq i<\mathcal{B}\\ d_f+i=k}} \mathrm{in}_e\big(\mathrm{gr}^e_i(q_f){\cdot}f\big) \overset{(\mathrm{L}\ 2.6.8)}{=} \mathrm{in}_e(f_k)$$

with

$$f_k=\sum_{\substack{f\in F\\ 0\leq i<\mathcal{B}\\ d_f+i=k}} \mathrm{gr}^e_i(q_f){\cdot}f \overset{(F\subseteq I)}{\in} I$$

Then g_k is homogenous of degree k by the inductive argument based on part $b)$ of lemma 2.6.5. Furthermore, $g_k=\mathrm{in}_e(f_k)$ with $f_k{\in}I$ and hence

$$g_k{\in}\mathrm{in}_e(I). \tag{$\dagger$}$$

On the other hand, by the uniqueness of decomposition

$$g=\sum_{d\leq k<\mathcal{B}} \mathrm{gr}^e_k(g)$$

and $(**)$

$$g_k = \mathrm{gr}^e_k(g) \quad \text{for each} \quad d{\le}k{<}\mathcal{B}.$$

In particular, by part b) of lemma 2.6.5,

$$\mathrm{in}_e(g) \overset{(\text{L } 2.6.5)}{=} \mathrm{gr}^e_d(g) = g_k \overset{(\dagger)}{\in} \mathrm{in}_e(I). \qquad \square$$

2.6.2 The symbolic perturbation of Gröbner bases in ${}_e\mathcal{P}$

The word "perturbation" being mostly associated with numerics and used in conjunction with the word "symbolic" might seem pretty confusing and hence requires an explanation. This subsection introduces two ways to transform a Gröbner basis in ${}_e\mathcal{P}$ to a Gröbner basis of a generally different ideal (also in a different algebra $\Bbbk[X]$ in theorem 2.6.2) which somewhat simulates the numerical perturbation of coefficients of a Gröbner basis computed ${}_e\mathcal{P}$ to get a numerically stable (without tiny leading coefficients and/or without coefficients which might have been considerably perturbed in the flow of the numerical Gröbner bases computation algorithm) under the backward translation of e to a numerical value which has a direct application to our algorithm for numerical computation of Gröbner bases where e stands for a stabilized order of magnitude of coefficients represented as a phantom variable e.

The first approach is based on the observation that the order $\underset{e\mathbb{T}}{<}$ can be considered as a kind of the anti-refinement of a weak order of terms induced by the e-grading: it is anti-compatible with the e-grading structure of ${}_e\mathcal{P}$ in the sense that $\deg_e(s_1){\le}\deg_e(s_2)$ follows $s_2\underset{e\mathbb{T}}{\le}s_1$ for all $s_1, s_2{\in}\,{}_e\mathbb{T}^*$ (according to part b) of definition 2.6.1 of $\underset{e\mathbb{T}}{<}$). There is a series of similar statements in the literature (e.g. [MT] proposition 4) for different domains with Gröbner bases and term orders when the initials of a Gröbner basis of the ideal form a Gröbner basis of the ideal generated by the initials of the ideal under a proper definition of the initial of a polynomials. The next theorem follows the same idea for the case of the algebra ${}_e\mathcal{P}$ and the notion of e-initials induced by the e-grading to state that the non-zero e-initials of a Gröbner basis is again a Gröbner basis (of the ideal generated by e-initials in ${}_e\mathcal{P}$). The result of the theorem can be used as a postprocessing step in the numerical Gröbner bases computation algorithm to get rid of the components in the polynomials in a Gröbner basis with a higher degree in e which are generally components with tiny or considerably perturbed in the flow of computation coefficients.

Theorem 2.6.1 (The truncation of Gröbner bases in ${}_e\mathcal{P}$). Let I be an ideal in ${}_e\mathcal{P}$. If $G \subseteq I \backslash \{0\}$ is a Gröbner basis of I then the e-initials is a Gröbner basis of the ideal generated by the e-initials of I:

$$\mathrm{in}_e(G) \ \text{ is a Gröbner basis of } \ \Big\langle \mathrm{in}_e(I) \Big\rangle_{{}_e\mathcal{P}}.$$

Proof. If G is a Gröbner basis of I then, according to definition 2.3.17 of a Gröbner basis,

$$\Big\langle \ell\mathrm{t}(G) \Big\rangle_{{}_e\mathcal{P}} = \Big\langle \ell\mathrm{t}(I) \Big\rangle_{{}_e\mathcal{P}}. \tag{1}$$

On the other hand, it is easy to see by part c) of lemma 2.6.7 (which reflects the anti-compatibility of $\underset{{}_e\mathbb{T}}{<}$ with the e-grading), that

$$\ell\mathrm{t}\Big(\mathrm{in}_e(G)\Big) = \ell\mathrm{t}(G) \ \text{ and } \ \ell\mathrm{t}(I) = \ell\mathrm{t}\Big(\mathrm{in}_e(I)\Big). \tag{2}$$

Also, by lemma 2.6.9,

$$\mathrm{in}_e(I) = \mathrm{in}_e\bigg(\Big\langle \mathrm{in}_e(I) \Big\rangle_{{}_e\mathcal{P}}\bigg). \tag{3}$$

Then

$$\begin{aligned} \Big\langle \ell\mathrm{t}\Big(\mathrm{in}_e(G)\Big) \Big\rangle_{{}_e\mathcal{P}} &\overset{(2)}{=} \Big\langle \ell\mathrm{t}(G) \Big\rangle_{{}_e\mathcal{P}} \overset{(1)}{=} \Big\langle \ell\mathrm{t}(I) \Big\rangle_{{}_e\mathcal{P}} \overset{(2)}{=} \Big\langle \ell\mathrm{t}\Big(\mathrm{in}_e(I)\Big) \Big\rangle_{{}_e\mathcal{P}} \overset{(3)}{=} \\ &\Bigg\langle \ \ell\mathrm{t}\bigg(\ \mathrm{in}_e\Big(\big\langle \mathrm{in}_e(I) \big\rangle_{{}_e\mathcal{P}}\Big)\bigg) \Bigg\rangle_{{}_e\mathcal{P}} \overset{(\dagger)}{=} \Bigg\langle \ \ell\mathrm{t}\Big(\big\langle \mathrm{in}_e(I) \big\rangle_{{}_e\mathcal{P}}\Big) \Bigg\rangle_{{}_e\mathcal{P}}, \end{aligned} \tag{4}$$

where the equivalence ($\dagger$) is by part c) of lemma 2.6.7. By (4),

$$\Big\langle \ell\mathrm{t}\Big(\mathrm{in}_e(G)\Big) \Big\rangle_{{}_e\mathcal{P}} = \Bigg\langle \ \ell\mathrm{t}\Big(\big\langle \mathrm{in}_e(I) \big\rangle_{{}_e\mathcal{P}}\Big) \Bigg\rangle_{{}_e\mathcal{P}}$$

and hence $\mathrm{in}_e(G)$ is a Gröbner basis of the ideal $\big\langle \mathrm{in}_e(I) \big\rangle_{{}_e\mathcal{P}}$ according to definition 2.3.17 of a Gröbner basis. □

The second approach to a transformation of Gröbner bases in ${}_e\mathcal{P}$ is a counterpart in ${}_e\mathcal{P}$ of the specialization of a Gröbner basis in the algebra ${}_e\mathcal{P}$ (theorem 2.5.4) under the partial evaluation of the polynomials of a Gröbner basis at e=0. It states that the result of the partial evaluation at e=0 of a Gröbner basis in ${}_e\mathcal{P}$ is a Gröbner basis in ${}_e\mathcal{P}$ under the order on the terms $\mathbb{T}$ (in the main variables X). This result is the core for the postprocessing step in the numerical algorithm for computing Gröbner bases $\Bbbk[X]$ via the computation of a Gröbner basis in ${}_e\mathcal{P}$.

Definition 2.6.5. The e-**specialization homomorphism** Φ_o is a map

$$\Phi_o : {}_e\mathcal{P} \longrightarrow \Bbbk[X]$$

from the ring ${}_e\mathcal{P}$ of extended polynomials in the indeterminates $X\cup\{e\}$ and the extended term order $\underset{e\mathbb{T}}{<}$ to the ring $\Bbbk[X]$ of polynomials in X equipped with the same term order $\underset{\mathbb{T}}{<}$. in the same indeterminates X and the term order $\underset{\mathbb{T}}{<}$ (which coincides with $\underset{e\mathbb{T}}{<}$ on $\underset{\mathbb{T}}{<}$) is defined as a homomorphic extension of the identity map on $\Bbbk$ such that $\Phi_o(e){=}0$ and $\Phi_o(x){=}x$ for any $x{\in}X$:

$$\Phi_o : \begin{cases} \quad {}_e\mathcal{P} & \longrightarrow \quad \Bbbk[X] \\ f(e,X) & \longrightarrow \quad f(0,X). \end{cases}$$

Lemma 2.6.10. The e-specialization homomorphism $\Phi_o{:}\,{}_e\mathcal{P}{\rightarrow}\Bbbk[X]$ is a surjective ring homomorphism.

Proof. Similar to lemma 2.5.23 on page 164. ☐

Remark 2.6.1. The map $\Phi_\alpha{:}\,{}_e\mathcal{P}{\rightarrow}\Bbbk[X]$ defined in the same fashion as Φ_o for an arbitrary $\alpha{\in}\Bbbk$ as a homomorphic extension of the identity map on $\Bbbk$ such that $\Phi_\alpha(e){=}\alpha$ and $\Phi_\alpha(x){=}x$ for any $x{\in}X$ is not a ring homomorphism unless $\alpha{=}0$. Indeed, the counterexample in remark 2.5.1 on page 166 applies verbatim provided that $\mathcal{P}(e)$, $\mathcal{P}_e$, $\mathcal{P}_{e[X]}$ and ${}_e\mathcal{P}$ is in fact the same ring.

Theorem 2.6.2 (the e-specialization of Gröbner bases in ${}_e\mathcal{P}$). Let $G{\subseteq}{}_e\mathcal{P}\backslash\{0\}$ be a Gröbner basis (w.r.t. the order $\underset{e\mathbb{T}}{<}$) such that $\mathrm{ord}_e(g) = 0$ for any $g{\in}G$. Then the image $\Phi_o(G)\backslash\{0\}{\subseteq}\Bbbk[X]\backslash\{0\}$ is a Gröbner basis of the ideal $\Phi_o(I)$ in $\Bbbk[X]$ w.r.t the term order $\underset{\mathbb{T}}{<}$.

Proof. Recall that due to the direct sum decomposition

$${}_e\mathcal{P} = \bigoplus_{0\leq k<\mathcal{B}} {}_e\mathcal{P}_k \text{ with } {}_e\mathcal{P}_k = \langle\, {}_e\mathbb{T}_k \rangle_{\Bbbk} \text{ - the } \Bbbk\text{-linear vector span of}$$
$${}_e\mathbb{T}_k = \Big\{ s{\in}\, {}_e\mathbb{T}^* \mid \deg_e(s) = k \Big\} \text{ for each } 0{\leq}k{<}\mathcal{B}$$

every polynomial $h{\in}{}_e\mathcal{P}$ has the unique decomposition

$$h = \sum_{0\leq k<\mathcal{B}} \mathrm{gr}^e_k(h) \text{ with } \mathrm{gr}^e_k(h){\in}{}_e\mathcal{P}_k \text{ for each } 0{\leq}k{<}\mathcal{B}.$$

Note that ${}_e\mathcal{P}_o{\subseteq}{}_e\mathcal{P}$ is an instance of $\Bbbk[X]$. W.l.o.g. assume that ${}_e\mathcal{P}_o{=}\Bbbk[X]$. It is easy to see that

$$\begin{aligned}\Phi_o(h) =& \Phi_o\Big(\sum_{0\leq k<\mathcal{B}} \mathrm{gr}^e_k(h)\Big) = \Phi_o\Big(\mathrm{gr}^e_0(h)\Big) + \Phi_o\Big(\sum_{1\leq k<\mathcal{B}} \mathrm{gr}^e_k(h)\Big) \\ =& \Phi_o\Big(\mathrm{gr}^e_0(h)\Big) + 0 = \Phi_o\Big(\mathrm{gr}^e_0(h)\Big) = \mathrm{gr}^e_0(h){\in}\Bbbk[X] \text{ for any } h{\in}{}_e\mathcal{P} \quad (*)\end{aligned}$$

provided that the homomorphism Φ_o is the (partial) evaluation at $e{=}0$.

Denote

$$G_o = \Phi_o(G)\backslash\{0\} \subseteq \Bbbk[X]\backslash\{0\}, \quad I_o = \Phi_o(I) \text{ - an ideal in } \Bbbk[X].$$

Our goal is to show that G_o is a Gröbner basis of I_o in $\Bbbk[X]$ w.r.t. the term order $\underset{\mathbb{T}}{<}$. Take any $0{\neq}f_o{\in}I_o$. Then, by $(*)$, there exists $f{\in}I$ such that $\mathrm{gr}_0^e(f){=}f_o$. Note that $f{\neq}0$ (e.g. because Φ_o is a homomorphism of rings). Furthermore, it clear that $\mathrm{ord}_e\big(\mathrm{gr}_0^e(f)\big){=}0$ (by $\mathrm{gr}_0^e(f){\in}\Bbbk[X]$) and then, due to the inclusion, $\emptyset{\neq}\,\mathrm{supp}\,\big(\mathrm{gr}_0^e(f)\big) \subseteq \mathrm{supp}\,(f)$,

$$0 \le \mathrm{ord}_e(f) = \min\Big(\deg_e(s) \mid s{\in}\,\mathrm{supp}\,(f)\Big) \le \\ \min\Big(\deg_e(s) \mid s{\in}\,\mathrm{supp}\,\big(\mathrm{gr}_0^e(f)\big)\Big) = \mathrm{ord}_e\big(\mathrm{gr}_0^e(f)\big) = 0$$

follows $\mathrm{ord}_e(f){=}0$ and then, by definition 2.6.4 of the e-initial of a polynomial, $\mathrm{in}_e(f){=}\mathrm{gr}_0^e(f){=}f_o$. The set G is a Gröbner basis of I in ${}_e\mathcal{P}$ (w.r.t. the order $\underset{e\mathbb{T}}{<}$) and therefore, by part $b)$ of theorem 2.3.2, there exists $g{\in}G$ such that

$$\ell\mathrm{t}(g)\,|\,\ell\mathrm{t}(f). \tag{1}$$

On the other hand, by part $c)$ of lemma 2.6.7, $\deg_e\big(\ell\mathrm{t}(f)\big){=}\mathrm{ord}_e(f){=}0$, hence $0{\le}\deg_e\big(\ell\mathrm{t}(g)\big){\le}\deg_e\big(\ell\mathrm{t}(g)\big){=}0$ and again, by part $c)$ of lemma 2.6.7, $\mathrm{ord}_e(g){=}0$, then, by definition 2.6.4 of the e-initial of a polynomial, $\mathrm{in}_e(g){=}\mathrm{gr}_0^e(g){\in}\Bbbk[X]$. Denote $g_o{=}\mathrm{in}_e(g)$. Note that, by part $a)$ of lemma 2.6.7, $g_o{\neq}0$ and hence, by $(*)$,

$$0 \neq g_o = \mathrm{gr}_0^e(g) \overset{(*)}{\in} G_o. \tag{2}$$

On the other hand, by part $b)$ of lemma 2.6.7,

$$\ell\mathrm{t}(g_o) = \ell\mathrm{t}\Big(\mathrm{in}_e(g)\Big) \overset{\text{(L 2.6.7)}}{=} \ell\mathrm{t}(g), \quad \ell\mathrm{t}(f_o) = \ell\mathrm{t}\Big(\mathrm{in}_e(f)\Big) \overset{\text{(L 2.6.7)}}{=} \ell\mathrm{t}(f)$$

and hence, by (1),

$$\ell\mathrm{t}(g_o)\,|\,\ell\mathrm{t}(f_o). \tag{3}$$

Furthermore, by part b_2) of definition 2.6.1 of the order $\underset{e\mathbb{T}}{<}$, the restriction of $\underset{e\mathbb{T}}{<}$ on $\mathbb{T}{=}\,{}_e\mathbb{T}_o$ is precisely the order $\underset{\mathbb{T}}{<}$:

$$\ell\mathrm{t}(h_o){=}\ell\mathrm{t}_{\underset{\mathbb{T}}{<}}(h_o) \;\text{ for any }\; 0{\neq}h_o{\in}\Bbbk[X] \tag{4}$$

where $\ell\mathrm{t}(h_o)$ is the "usual" leading term of h_o w.r.t. $\underset{e\mathbb{T}}{<}$ and $\ell\mathrm{t}_{\underset{\mathbb{T}}{<}}(h_o)$ is the leading term of h_o w.r.t. $\underset{\mathbb{T}}{<}$.
Since $g_o, f_o \in \Bbbk[X]$, by (2), (3) and (4),

$$\ell\mathrm{t}_{\underset{\mathbb{T}}{<}}(g_o) \,|\, \ell\mathrm{t}_{\underset{\mathbb{T}}{<}}(f_o) \quad \text{with} \quad g_o \in G_o \tag{5}$$

According to (5) we have shown that for any $0 \neq f_o \in I_o$ there exists $g_o \in G_o$ such that $\ell\mathrm{t}_{\underset{\mathbb{T}}{<}}(g_o) \,|\, \ell\mathrm{t}_{\underset{\mathbb{T}}{<}}(f_o)$ and therefore, by part *b*) of theorem 2.2.1, G_o is a Gröbner basis of I_o in $\Bbbk[X]$ w.r.t. the term order $\underset{\mathbb{T}}{<}$. □

Chapter 3

The implementation

This chapter describes in details our numerical algorithm for computing Gröbner bases. Our main concern are numerical Gröbner bases of zero-dimensional polynomial systems although the present algorithm is capable of computing a Gröbner basis numerically for an arbitrary system of polynomials. Given a zero-dimensional system, our ultimate goal is to compute the solutions of the system by exploiting the multiplicative structure of the associated quotient-algebra where a Gröbner basis is used as a tool to compute the representation of the multiplicative structure (by construction of so-called multiplication matrices, see section 4.2 for details).

The numerical computation of Gröbner bases is a complicated task as it inherits the potential computational inefficiency from the original "exact" Gröbner bases and also acquires the numerically specific difficulties. In short, the well-known doubly exponential complexity of a Gröbner basis computation (see e.g. [Dub90], [KM96], [MM82], [Giu84], [MM84]) turns the numerical computation of Gröbner bases into a hard problem at least due to the roundoff error accumulation phenomena that is likely (but not indeed necessary!) to become a striking issue in a lengthy computation. However, there is another, mathematically more striking, drawback pointed out by Hans Stetter in [Ste97]: for any fixed term order the (totally reduced) Gröbner basis of a polynomial system may depend discontinuously on the coefficients of the system in the sense that a small change in the coefficients may change the structure of leading terms and also the number of elements in the corresponding Gröbner basis; as the coefficients of the system approach such a discontinuous situation some coefficients of the Gröbner basis polynomials diverge to infinity and hence the Gröbner basis may be an extremely ill-conditioned representation of (the ideal of) the original system even if it is perfectly well-conditioned itself.

Let $\Bbbk$ be a field and X be a finite set of indeterminates (variables). Denote

$\mathcal{P}{=}\Bbbk[X]$ – a polynomial algebra equipped with an admissible term order $<_{\mathbb{T}}$ on the terms $\mathbb{T}{=}X^*{\subseteq}\mathcal{P}$. Consider a polynomial system $\mathcal{F}{\subseteq}P$. Assume $\mathcal{F}$ is zero-dimensional i.e. the *zero set* $Z(\mathcal{F})$ of $\mathcal{F}$ consists of finitely many isolated (the precise definition of the metric on the solution space is irrelevant) points. It is well-known that for some specific configurations of the zero set a small perturbation of the coefficients may cause a structural change in the geometry of the zero set e.g. change the number of solutions or blow points into curves. We assume that the geometry of the solution set is not structurally changing in a neighborhood of $\mathcal{F}$ as it is also the case for the analysis in [Ste97].

As has been analyzed in [Ste97], the possible discontinuity at $\mathcal{F}$ of the (totally reduced) Gröbner basis $\mathcal{G}$ of $\mathcal{F}$ is indeed caused by the possible discontinuity at $\mathcal{F}$ of the associated normal set $N(\mathcal{G})$. The proposed solution in [Ste97] is to locally eliminate the discontinuity by keeping the normal set, and consequently the leading terms of the Gröbner basis $\mathcal{G}$, invariant in a neighborhood of $\mathcal{F}$ in the case if the coefficients of $\mathcal{F}$ approach such a discontinuous situation. The basis $\widehat{\mathcal{G}}$ obtained in this fashion is generally no longer a Gröbner basis but a continuous extension of the genuine Gröbner basis of a *slightly perturbed* $\mathcal{F}$. It is also shown in [Ste97] that the extended Gröbner basis $\widehat{\mathcal{G}}$ retains the numerical condition of the original system $\mathcal{F}$ w.r.t. the determination of the zero set $Z(\mathcal{F})$ and hence it is a suitable representation of $\mathcal{F}$ for a numerical determination of $Z(\mathcal{F})$. The challenge stated in [Ste97] is the construction of a practically usable general purpose *stabilized Gröbner basis algorithm* which

- recognizes the closeness to a discontinuity of the genuine Gröbner basis $\mathcal{G}$ and the subsequent ill-conditioning which is to be expected;
- in this case, delivers an extended Gröbner basis $\widehat{\mathcal{G}}$ which does not share the ill-conditioning of $\mathcal{G}$;
- if no discontinuity of $\mathcal{G}$ occurs near $\mathcal{F}$ then no deviation from the standard approach is necessary and the result of the algorithm is the genuine Gröbner basis $\mathcal{G}$.

Our algorithm is a realized attempt to construct such a stabilized Gröbner basis algorithm executed in floating-point arithmetic using either the hardware supported single or double precision floats or the software emulated floating-point arithmetic with an arbitrary long (but fixed within a session of computation) mantissa from the `GNU Multiple Precision Arithmetic Library`. Besides the floating-point arithmetic our implementation is capable to deal with exact arithmetic presented by residue fields (modulo a prime integer) and arbitrary precision rational fractions. The implemented exact arithmetic is not our main concern but an extra feature intended for comparison of the results of computation.

Unfortunately the present algorithm is still quite far from being a suitable building block for a general purpose black-box floating-point software system for solving polynomial systems by the use of Gröbner bases since the flow of computation and the final result depend on certain computational model parameters which may be passed as a part of the input data. For each parameter omitted in the input, the algorithm uses a simple rule to set a reasonable default value; but – as common in numerical software – these rules can not be perfect. For a given system of polynomials the restricted experimentation with the parameters of computation generally leads to a meaningful result of the computation – an extended Gröbner basis which, for the case of a zero-dimensional system, represents the multiplicative structure of the quotient-algebra of the ideal of the initial system with an appropriate accuracy. A collection of experiments including the detailed description and analysis of the flow of computation is given in chapter 4.

The most striking complication in the numerical computation of a Gröbner basis of $\mathcal{F}\subseteq\mathcal{P}$ w.r.t. a fixed term order $<_{\mathbb{T}}$ by one of the variants of Buchberger algorithm is the appearance of very small leading coefficients in the intermediate polynomials. It can be an indication that

- the genuine Gröbner basis $\mathcal{G}$ of $\mathcal{F}$ is close to a discontinuity; but it may also be

- purely the matter of accumulated numerical roundoff which has created a tiny non-zero value for a coefficient which should have vanished; or it may be

- an issue related to the utilized selection and reduction strategies for S-polynomials: a very small coefficient at the leading term of some intermediate polynomial may not propagate to the final Gröbner basis being reduced (or "absorbed") at some later point of computation.

If a polynomial with a very small leading coefficient has been created in the flow of a numerical Gröbner basis computation (either through an S-polynomial formation or reduction) and it is combined, either in S-polynomial formation or reduction, with another polynomial whose matching coefficient is of order 1 then it must be multiplied with a very large scaling factor; this amplifies the numerical roundoff error in the polynomial and may result in the cancellation of leading digits – the well-known source of numerical instability. Our approach uses an elaborated version of a possible solution of the small leading coefficients problem proposed in [Ste97]: a very small leading coefficient should never be used for an S-polynomial formation or reduction; if a polynomial with a very small leading coefficient has some coefficients of order 1 at the same time then the leading term with the very small coefficient must be formally deprived of the function as the leading term – this function is taken over by the next (lower in the term order) term in the polynomial which has a coefficient of order 1[1].

This potential requirement to modify the leading term function (for the reason of numerical stability) raises a formidable problem: a fixed admissible term order on $\mathcal{P}$ which uniquely defines the leading term of any non-zero polynomial; the term order is defined on terms $\mathbb{T}\subseteq\mathcal{P}$ and hence does not take into account the order of magnitude of the coefficient at a term in the polynomial and leaves no possibility to plug the order of magnitude of coefficients into the term order and, in particular, into the leading term function staying in the frame of the classical theory of Gröbner bases. Although the term order is not an intrinsic property of $\mathcal{P}$, it is the driving force of the whole theory of Gröbner bases that makes it work: the term order makes possible the definition of a Gröbner basis, the reduction process, the completion procedure and finally Buchberger algorithm for computing Gröbner bases. One possible solution is presented in [Mou99] where a purely linear algebra technique is utilized to establish an analogue to the theory of Gröbner bases without an admissible term order for zero-dimensional polynomial systems. Our solution exploits a "less radical" approach: we do not deny the term order but extend the theory of Gröbner bases to the so-called algebra of extended polynomials $\mathcal{P}(e)$ (described in chapter 2) with an *extended term order* (definition 2.3.9 on page 37) which is again a purely symbolic theory in its own right without any regard towards numerical questions. The extended term order (which is a cornerstone of our theory just like the admissible term order for the classical theory of Gröbner bases) can be implemented as an order of terms which takes into account the size of the coefficients at the terms in a polynomial such that a term with a smaller coefficient (in the polynomial) is smaller in the order and therefore less likely to be the leading term of the polynomial;

[1]Indeed a very similar problem appears in the context of numerical solution of a system of linear equations by the use of some variant of the direct Gaussian elimination: at each stage of elimination it is usually not enough to just ask that the appropriate pivot element is non-zero but choose the pivot as an element with the maximal absolute value using either the partial or the complete pivoting technique (see e.g. page 444 in [Atk78]). The proposed solution in [Ste97] can actually be considered as a generalization of the complete pivoting for polynomial systems solving with Gröbner bases where the switching of the columns in the elimination matrix during Gaussian elimination can be associated with the local override of the demands of the term order during a Gröbner basis computation if required to maintain the numerical stability.

this way the theory of Gröbner bases in $\mathcal{P}(e)$ is turned into a theoretical foundation for the numerical computation of Gröbner bases as it is exploited in our algorithm.

The transition from the symbolic theory of Gröbner bases in the algebra $\mathcal{P}(e)$ of extended polynomials to the numerical computation of (generally extended) Gröbner bases in $\mathcal{P}$ can be established as follows. First of all, since we deal with numerical computation assume that the field $\Bbbk$ is either the real or complex numbers. Furthermore we assume that the system of polynomials $\mathcal{F}\subseteq\mathcal{P}$ whose (extended) Gröbner basis we wish to find is given with limited numerical precision $0<\delta\ll 1$. Consider a small real number $1\gg\varepsilon>\delta$ and the smallest integer $\mathcal{B}>1$ such that $\varepsilon^{\mathcal{B}}\leq\delta$ i.e. $\varepsilon^{\mathcal{B}}$ is indistinguishable from 0 in the given precision; $\mathcal{B}$ is typically 2 or 3.

We extend the set of variables X with a new variable e such that $e^{\mathcal{B}}=0$. The set of terms

$$\mathbb{T}(e)^* = \left\{t\cdot e^i \;\middle|\; t\in\mathbb{T}, 0\leq i<\mathcal{B}\right\},\ T(e) = T(e)^*\cup\{0\}$$

and the algebra

$$\mathcal{P}(e)=\Bbbk[X,e]\left(\cong \Bbbk[X,y]\big/\left\langle y^{\mathcal{B}}\right\rangle\right)$$

generated by $X\cup\{e\}$ are called the *set of non-zero extended terms*, the *set of extended terms* and the *algebra of extended polynomials* respectively by the author of this text. The original variables X are called the *main variables*. It is quite clear (cf. part d) of theorem 2.3.1 on page 26) that each non-zero extended term $s\in\mathbb{T}(e)^*$ can be uniquely represented as $s=t\cdot e^d$ with $t\in\mathbb{T}$, $0\leq d<\mathcal{B}$: t and d in the unique representation of s are referred as the *main part*, denoted $\mathrm{T}(s)=t$, and the *e-degree*, denoted $\deg_e(s)=d$, of s respectively. Sometimes we omit the word "extended" if it does not cause a confusion.

A degree of e is used in our approach to encode the size of the coefficient in a monomial $\alpha\cdot t\in\mathcal{P}$ with $\alpha\in\Bbbk^*$ and $t\in\mathbb{T}$ by the correspondence Ω defined as

$$\Omega(\alpha\cdot t) = \alpha\cdot\left(t\cdot e^{\mathcal{O}(\alpha)}\right)$$

where $\mathcal{O}\colon\Bbbk^*\rightarrow\mathbb{N}_o$ is a measure of how small is the absolute value of a coefficient such that for any $a,b\in\Bbbk^*$

$$\begin{aligned} |a|\leq|b| &\ \text{ implies }\ \mathcal{O}(a)\geq\mathcal{O}(b),\\ |a|\approx O(1) &\ \text{ implies }\ \mathcal{O}(a)=0,\\ |a|<\delta &\ \text{ implies }\ \mathcal{O}(a)\geq\mathcal{B}\ \text{ and hence }\ a\cdot t\cdot e^{\mathcal{O}(a)}\equiv 0\ \text{ in } \mathcal{P}(e). \end{aligned}$$

Now it is easy to define the translation of an arbitrary polynomial

$$p=\sum_{t\in\,\mathrm{supp}(p)}\alpha_t\cdot t\in\mathcal{P}\ \text{ with } \alpha_t\in\Bbbk^*\ \text{ for }\ t\in\mathrm{supp}\,(f)$$

to a polynomial $\Omega(p)\in\mathcal{P}(e)$ – the linear combination of extended terms where the degrees of e *encode* the order of magnitude of coefficients at respective terms and a higher degree corresponds to a smaller coefficient – using the formal linearity of Ω on monomials in $\mathcal{P}$:

$$\Omega(p) = \sum_{t\in\mathrm{supp}(p)} \Omega(\alpha_t\cdot t) = \sum_{t\in\mathrm{supp}(p)} \alpha_t\cdot\left(t\cdot e^{\mathcal{O}(\alpha_t)}\right) \ \in \mathcal{P}(e).$$

It seems natural to define $\mathcal{O}(\cdot)$ in terms of the ε-weighted measure of magnitude given as $\log_\varepsilon(|\alpha|)$, $\alpha\in\mathbb{k}^*$. Ideally we would like to have $\mathcal{O}(\alpha)=\log_\varepsilon(|\alpha|)$ but this not possible as long as we like to operate in a polynomial algebra. Therefore a proper definition can be

$$\mathcal{O}(\alpha)=\left\lfloor\log_\varepsilon(|\alpha|)\right\rfloor.$$

There is still a slight problem in the latter definition of $\mathcal{O}(\cdot)$ because clearly $\left\lfloor\log_\varepsilon(|\alpha|)\right\rfloor<0$ for any $\alpha\in\mathbb{k}^*$ such that $|\alpha|>1$. The problem disappears if we *normalize* a polynomial $p\in\mathcal{P}$ before the translation to $\Omega(p)\in\mathcal{P}(e)$ by multiplying it with an appropriate scaling factor $p:=\alpha\cdot p$, $\alpha\in\mathbb{k}^*$ such that the absolute value of any coefficient in p falls into the interval $(0,\ldots,1)$.

The translation scheme of the initial system $\mathcal{F}\subseteq\mathcal{P}$ to $F\subseteq\mathcal{P}(e)$ utilized in our implementation uses essentially the same definition of $\mathcal{O}(\cdot)$ as in the given sample but expressed in terms of the internal representation of floating point numbers (see section 3.1 for details). The normalization before the translation is performed for each polynomial $p\in\mathcal{F}$ whose $m(p)$ – the maximal absolute value of coefficients – is greater or equal than 1 in such a way that that $m(p)<1$ and $m(p)\approx O(1)$ after normalization. As a useful side effect, this kind of normalization turns $\mathcal{O}(\cdot)$ into a relative measure of magnitude of the coefficients in a polynomial.

After the translation of the input system to $F\subseteq\mathcal{P}(e)$ we compute a Gröbner basis in $\mathcal{P}(e)$ using one of the extensions of the original term order $<_\mathbb{T}$ on $\mathbb{T}\subseteq\mathcal{P}$ to an extended term order (definition 2.3.9 on page 37) $<$ on the extended terms $\mathbb{T}(e)\subseteq\mathcal{P}(e)$. The supported extensions of the order are

$\underset{\mathbb{T}_e}{<}$ defined as (cf. page 90) the lexicographical product of the order $<_\mathbb{T}$ on $\mathbb{T}$ and the degree anti-compatible order on the (finitely many) powers of e with priority given to the main variables and "0" – the smallest term;

$\underset{e\mathbb{T}}{<}$ defined as (cf. page 168) the lexicographical product of the degree anti-compatible order on the (finitely many) powers of e and the order $<_\mathbb{T}$ on $\mathbb{T}$ with priority given to the variable e and and "0" – the smallest term.

In either extension of the term order, e has a negative weight. On the other hand, the translation procedure of the initial polynomial system $\mathcal{F}\subseteq\mathcal{P}$ to $F\subseteq\mathcal{P}(e)$ encodes a term with a modulus smaller coefficient by a higher degree of e. In particular, if the utilized term order extension is $\underset{e\mathbb{T}}{<}$ (which is extra aggressive towards lessening the extended terms with a positive e-degree) then the leading term of $p\in\mathcal{F}$ with a very small leading coefficient α such that $\mathcal{O}(\alpha)>0$, and some other coefficients of order 1, is removed from the leading position in $\Omega(p)\in\mathcal{P}(e)$.

In short, we have established a mechanism where the order of terms is controlled by the order of magnitude of coefficients at the terms in a polynomial which is the cornerstone in our approach for the stabilized Gröbner basis algorithm.

In section 2.3.3 we have established the reduction algorithm in the algebra $\mathcal{P}(e)$ that can be readily applied as a building block in the computation of a Gröbner basis of F. The reduction procedure in $\mathcal{P}(e)$ has been developed as a purely symbolic process which does not take into account any numerical aspects like the apparent numerical stability problem if the reduction is performed modulo a polynomial with a very small leading coefficient. On the other hand, the system F satisfies certain prerequisites for a numerically stable reduction in the flow of a Gröbner basis computation in $\mathcal{P}(e)$ which are created during the translation from $\mathcal{F}\subseteq\mathcal{P}$:

- If the utilized term order extension is $\underset{e\mathbb{T}}{<}$ then the leading terms with very small coefficients of the polynomials $\mathcal{F}\subseteq\mathcal{P}$ are likely to be deprived of the leading position in the resulting system F.

- In the case when the translation fails to remove some small leading coefficients in $\mathcal{F}$ from the leading positions in F the situation is still not completely bad regarding the reduction in F. Indeed, let $f\in F$ be a polynomial with a very small leading coefficient such that $\mathcal{O}(\ell\mathrm{c}(f))>0$ (this can happen if all coefficients in the pre-image of f in $\mathcal{F}$ are very small or the utilized extension of the term order is $\underset{\mathbb{T}_e}{<}$). Then $\deg_e\big(\ell\mathrm{t}(f)\big)>0$ by construction of the translation process from $\mathcal{F}$ to F. But then it is automatically forbidden to use f to eliminate a term $s\in\operatorname{supp}(g)$ of some other polynomial $g\in F$ whose matching coefficient α_s in g has order 1 i.e. $\mathcal{O}(\alpha_s)=0$. Indeed, to eliminate s modulo f the leading term $\ell\mathrm{t}(f)$ must divide s (see the discussion below concerning the total reduction in $\mathcal{P}(e)$ and especially definition 3.0.6 on page 203 for the precise statement). On the other hand, $\mathcal{O}(\alpha_s)=0$ follows $\deg_e(s)=0$ and hence $\ell\mathrm{t}(f)$ can not divide s due to $\deg_e\big(\ell\mathrm{t}(f)\big) > \deg_e(s)$.

As we have seen in the discussion above, the term order extension $\underset{e\mathbb{T}}{<}$ provides an appropriate mechanism to remove the leading term of a polynomial with a small leading coefficient from the leading position during the translation from $\mathcal{F}\subseteq\mathcal{P}$ to $F\subseteq\mathcal{P}(e)$. Therefore it is *highly recommended* to use the order $\underset{e\mathbb{T}}{<}$ in our algorithm for numerical computation of an extended Gröbner basis (via a computation in $\mathcal{P}(e)$) especially when the genuine Gröbner basis is close to a discontinuity. Concerning the order $\underset{\mathbb{T}_e}{<}$, it simulates our early approach towards numerical computation of a Gröbner basis which is not capable to deal with extended Gröbner bases, performs the computation in the original algebra $\mathcal{P}$ with the term order $<_{\mathbb{T}}$ but uses a stabilized reduction process to improve the numerical stability in computing a genuine Gröbner basis under assumption that it is a well-conditioned representation of the original system (far from a discontinuity). The stabilization of the reduction process has been attained as follows. Assume that we reduce a polynomial $p\in P$ at some point of a Gröbner basis computation and some $t\in \operatorname{supp}(p)$ with an order 1 coefficient in p is reducible by the current basis but all possible reducers – the polynomials from the current basis whose leading terms divide t – have very small leading coefficients and therefore we would not like to reduce t in p at the current point of computation for the numerical stability reasons. Then the reduction of t in p is postponed and the (possibly partially reduced) p is moved to a separate set of delayed relations in the hope that there would appear a possibility to reduce t in p in a numerically stable way at some later point of computation due to a new polynomial added to the current basis: we wait for a polynomial with an order 1 leading coefficient and whose leading term divides t. This approach called *the reduction with delayed treatment* (which has been presented by the author at SNSC'01 conference) appears useful in numerical computing of Gröbner bases according to our experiments. But it has been hard to implement (due to the cumbersome data structures to keep track and update the reducibility status of the terms and polynomials whose reduction was postponed), the analysis of examples is difficult and the theoretical meaning of the delayed treatment is also not clear. On the other hand, according to our experiments, the current approach based on the theory of Gröbner bases in the algebra $\mathcal{P}(e)$ of extended polynomials, with $\underset{\mathbb{T}_e}{<}$ used as the term order extension, effectively reproduces the reduction with delayed treatment as well as it is also generally not able to deliver any extended Gröbner bases.

Indeed the computation of a Gröbner basis of the translated system F in $\mathcal{P}(e)$ would be of moderate value in the numerical respect if we had no mechanisms to keep the correspondence between the e-degree of a term and the true order of magnitude of the coefficient at a term in a polynomial in the flow of computation. Indeed, in the initial polynomials F the e-degrees are perfectly compliant with the true order of magnitude of the respective coefficients provided by construction of the translation process from $\mathcal{F}$ to F. The computation of a Gröbner basis of F in $\mathcal{P}(e)$ is naturally a sequence of operations with polynomials in $\mathcal{P}(e)$ where e acts like a conventional variable but the relation $e^{\mathcal{B}}{=}0$ is plugged into the polynomial arithmetic. In particular (cf. lemma 2.3.1 on page 29), for any $s_1, s_2{\in}\mathbb{T}(e)^*$

$$\begin{array}{ccc} s_1{\cdot}s_2 = 0 & \text{if and only if} & \deg_e(s_1){+}\deg_e(s_2){\geq}\mathcal{B}, \\ \text{if } s_1{\cdot}s_2{\neq}0 & \text{then} & \deg_e(s_1{\cdot}s_2){=}\deg_e(s_1){+}\deg_e(s_2). \end{array}$$

Such a behavior of e-degrees is not completely bad regarding the correspondence of the e-degree of a term and the true order of magnitude of the coefficient at the term in a polynomial during the computation of a Gröbner basis: if $\alpha_1{\cdot}s_1$ and $\alpha_2{\cdot}s_2$ are the monomials in some polynomials that appeared in the flow of computation and have been multiplied then the e-degree $\deg_e(s_1){+}\deg_e(s_2)$ of $\deg_e(s_1{\cdot}s_2)$ clearly simulates the true order of magnitude of $\alpha_1{\cdot}\alpha_2$ up to a certain extent; in particular, $\deg_e(s_1){+}\deg_e(s_2){\geq}\mathcal{B}$ indicates that we have hit the given numerical precision of computation and then the term $s_1{\cdot}s_2{\equiv}0$ is removed from the result of multiplication following the common numerical practice when a nearly degenerated situation is considered as degenerated.

But it is also clear that the order of magnitude of coefficients generally behaves in a more complex way under multiplication than the degrees of e. Furthermore, the *addition* of polynomials in $\mathcal{P}(e)$ can create new very small coefficients and this phenomenon is not reflected at all in the e-degrees. As an example consider a reduction of

$$f = x{+}(0.5{+}\alpha){\cdot}y{+}2{\in}\mathcal{P}(e) \quad \text{with} \quad \alpha{\in}\Bbbk^* \;\text{ such that } 0{<}|\alpha|{\ll}1 \;\text{ and }\; \mathcal{O}(\alpha){>}0$$

modulo

$$g = x{+}0.5{\cdot}y{+}1{\in}\mathcal{P}(e) \quad \text{where the order on terms is defined as } x{>}y{>}1.$$

The result of reduction

$$h = f{-}g = \alpha{\cdot}y{+}1$$

acquires a very small leading coefficient α which is not reflected at all by $\deg_e\big(\ell\mathrm{t}(h)\big){=}\deg_e(y){=}0$.

Our algorithm dynamically adjusts the e-degrees in the terms of intermediate polynomials computing a Gröbner basis. The corresponding mechanism is called the *trimming of e-degrees* (page 266). This is indeed the only mechanism which may cause the flow of computation to deviate from the flow prescribed by the theory of Gröbner bases in $\mathcal{P}(e)$, for the sake of numerical stability. The trimming of e-degrees occurs as a part of the implemented reduction process.

Surely we have to observe certain restrictions to keep the algorithm operational.

First of all it is unconditionally forbidden to decrease the e-degree of any term. This restriction is stipulated by the reduction process in $\mathcal{P}(e)$. Indeed, consider a monomial $\alpha{\cdot}s$, $\alpha{\in}\Bbbk^*$, $s{\in}\mathbb{T}(e)$ in some intermediate polynomial $f{\in}\mathcal{P}(e)$ that is being reduced at some point in the flow of computation. Both supported extensions ($\underset{\mathbb{T}_e}{<}$ and $\underset{e\mathbb{T}}{<}$) of the term order assign a negative weight to e so that decreasing the e-degree of s we create a higher in the order term. But the trimming of e-degrees is a part of the reduction process that is basically performed as a successive elimination of the reducible terms modulo the current basis by replacing them with a linear combination of strictly smaller terms. So, if we decrease the e-degree of s in the flow of reduction then we break the important principle that a reduction process can never introduce any higher terms than have originally been present in f; this would generally break the termination property of the reduction process provided by the well-ordering of the extended term order.

There is also a numerical reason against decreasing the e-degree of s: if s has a positive e-degree (otherwise we can not decrease the e-degree anyway) then the coefficient α at s in f is likely to stem from a very small coefficient which is the origin of the positive e-degree of s being assigned either at the initial translation from $\mathcal{P}$ to $\mathcal{P}(e)$ or by the trimming of e-degrees at an earlier point of computation. In this case α is an amplified small coefficient, with an amplified roundoff, and hence its numerical value should not be trusted; hence we should not decrease the e-degree of s creating a term higher in the extended term order and thus giving α an extra chance to become the leading coefficient in the result of reduction of f.
Two other restrictions on the trimming of e-degrees are the following.

- It is forbidden to change the e-degree of a term which is reducible modulo the current basis. Here the idea is pretty transparent: if we can reduce a term then we should do it instead of applying extra mechanisms to the term!

- It is forbidden to change the e-degrees in the terms of the polynomials that have been already added to the current basis, it is particularly important that no e-degree is ever changed in their leading terms. This restriction keeps the interpretation of a Gröbner basis computation as the successive extension of the ideal of leading terms of the current basis which is crucial for the termination argument of a Gröbner basis algorithm – both for the classical version in $\mathcal{P}$ and for our instantiation in $\mathcal{P}(e)$. Note that this restriction is not an important issue for our current implementation since we do not perform a reduction of polynomials that have been already added to the basis until the very final auto-reduction of the attained Gröbner basis; this is done without trimming of e-degrees.

A few words should also be said about the completion procedure used in the algorithm. Our implementation follows the completion procedure in $\mathcal{P}(e)$ as established in section 2.3.5 and elaborated in section 2.3.6. In short, to compute a Gröbner basis in $\mathcal{P}(e)$ one has to deal with S-polynomials as in the classical Buchberger algorithm but also with *critical e-shifts* – an extra type of critical elements. A critical e-shift is a shift $e^k \cdot f$ of the non-zero polynomial $f \in \mathcal{P}(e)$ by the pure power e^k with k – the smallest degree which annihilates the leading term of f: $e^k \cdot \ell\mathrm{t}(f)=0$.

The implementation of the Gröbner bases algorithm in $\mathcal{P}(e)$ utilized as the main step in computing numerical Gröbner bases in $\mathcal{P}$ is not principally different from a version of the classical Buchberger algorithm: starting from the initial polynomials $F \subseteq \mathcal{P}(e)$, the current basis is successively extend by the non-zero normal forms (modulo the current basis) of critical elements formed from the polynomials of the current basis until no more critical elements are left to be considered. Instead of keeping the current basis totally auto-reduced at any point of computation, we keep the polynomials added to the basis unchanged until the final auto-reduction of the attained Gröbner basis into the totally reduced Gröbner basis.

Our implementation also exploits two main ideas of the F_4 algorithm [Fau] by J. C. Faugere (a rearrangement of Buchberger algorithm) which appear attractive for numerical computing of Gröbner bases: the simultaneous reduction of a block of critical elements and the reduction process split into so-called *symbolic reduction* (see section 3.4.1 for details) and the purely linear algebra row echelon elimination procedure (described in section 3.4.2).

The attained Gröbner basis $G \subseteq \mathcal{P}(e)$ of $F \subseteq \mathcal{P}(e)$ is the subject for the final postprocessing in order to attain the desired extended Gröbner basis $\mathcal{G}$ of the input system $\mathcal{F} \subseteq \mathcal{P}$ together with the multiplicative structure of the quotient-algebra ${}^{\mathcal{P}}/_{\langle F \rangle}$ if F is zero-dimensional.

To summarize our approach, the computation of a (possibly extended) Gröbner basis $\mathcal{G}$ of a system $\mathcal{F} \subseteq \mathcal{P}$ consists of three major steps as follows.

1. *Preprocessing:* The input system $\mathcal{F}$ is *translated* to the system $F \subseteq \mathcal{P}(e)$ in the algebra of extended polynomials.

2. *Computation:* A sophisticated and highly specialized for the numerical setup version of a Gröbner basis algorithm in the algebra of extended polynomials (cf. section 2.3.6) is used to attain the totally reduced Gröbner basis $G\subseteq\mathcal{P}(e)$ for the user specified extension of the original term order $<_{\mathbb{T}}$: the extension of the order can be either $\mathbb{T}_e$ (page 90) or ${}_e\mathbb{T}$ (page 168). However the algorithm does not always rigidly follow the path prescribed by the theory of Gröbner bases in $\mathcal{P}(e)$ if the latter appears incompatible with the demands of numerical stability even at the cost of deviation from the ideal $\langle F\rangle_{\mathcal{P}(e)}$ of the initial system. The mechanism responsible for the possible deviation is implemented via so-called *trimming of e-degrees* is described in details on page 266. The resulting basis G is *always* the totally reduced genuine (not extended) Gröbner basis and typically $\langle G\rangle_{\mathcal{P}(e)}=\langle F'\rangle_{\mathcal{P}(e)}$ with F' – a polynomial system in a neighborhood of the initial system F unless the attained G is structurally wrong e.g. due to the insufficient precision of the floating point arithmetic used in the computation.

3. *Postprocessing:* If the input system $\mathcal{F}$ is zero-dimensional then the Gröbner basis $G\subseteq\mathcal{P}(e)$ is expanded to a basis $\Gamma\supseteq G$ that is *translated back* to the algebra $\mathcal{P}$ resulting in the *border basis* $\Upsilon\subseteq\mathcal{P}$ which is generally an extended Gröbner basis and also contains the totally reduced (possibly extended) Gröbner basis $\mathcal{G}\subseteq\Upsilon$ of the input system $\mathcal{F}$.

   If $\mathcal{F}$ is not zero-dimensional then we surely can not compute the border basis Υ but still deliver the totally reduced (possibly extended) Gröbner basis $\mathcal{G}$. This is the only step in our approach where the zero-dimensionality property of F is used, the previous two steps work as well for an arbitrary system of polynomials. Furthermore, this is the only step which may actually fail (see section 4.2 for details). The failure is considered as a bad luck for the whole numerical computation of an extended Gröbner basis for F and calls for a new computational session with different parameters e.g. a higher precision of the floating-point arithmetic.

The computation accepts the following parameters passed as a part of the input data together with the input polynomial system.

The alphabet: The main variables X.

The term order: The term order $<_{\mathbb{T}}$ on $\mathbb{T}=X^*$ – the terms in main variables.

The term order extension: The extension of the order $<_{\mathbb{T}}$ on the terms in the main variables to an order on extended terms $\mathbb{T}(e)$. The following extensions are supported.

> $\underset{\mathbb{T}e}{<}$: The lexicographical product of the order $<_{\mathbb{T}}$ and the degree anti-compatible order on the powers of e with the priority given to $<_{\mathbb{T}}$ (cf. page 90). This order provides an improved stability of the reduction process but is generally unable to handle the near-discontinuity problem. This extension is intended for the computation of the genuine Gröbner basis if it is a well-conditioned representation of the input polynomial system.
>
> $\underset{e\mathbb{T}}{<}$: The lexicographical product of the degree anti-compatible order on the powers of e and the order $<_{\mathbb{T}}$ with the priority given to e (cf. page 168). It is highly recommended to use this extension of the order as it fully unleashes the power of our approach for numerical Gröbner bases. According to our experiments the computation with $\underset{e\mathbb{T}}{<}$ results in the genuine Gröbner basis if it is a well-conditioned representation of the input system or the truly extended Gröbner basis if the genuine one is close to a discontinuity.
>
> $\underset{\mathbb{T}o}{<}$: The "fake" extension. The order $\underset{\mathbb{T}o}{<}$ is essentially the order $<_{\mathbb{T}}$ on the terms $\mathbb{T}$ in the main variables X as it does not take into account the powers of e comparing extended terms. This extension is intended for simulating the classical Buchberger algorithm and implemented for testing purposes.

The domain: The arithmetic domain for coefficients of the input polynomial system. The following domains are supported.

> *Float:* The single and double precision hardware floating point and the software emulated floating point with a user defined length of mantissa in bits [2].
>
> *Exact:*
>
> > *Rational:* The arbitrary precision rational fractions from the `GNU Multiple Precision Arithmetic Library`.
> >
> > *Residue:* The field $F_p \cong \mathbb{Z}/_{\langle p \rangle}$ with a user defined prime number p.

[2] Should be a multiple of 32 due to limitations of the `GNU Multiple Precision Arithmetic Library` used for the software floating point in our implementation.

The exact arithmetic is not our main concern in the implementation but an extra feature for testing purposes as well as to study the behavior of the algorithm and compare the results of computation. In particular, no e-degrees are assigned to any terms in the computation if the exact arithmetic is used.

The E-bound: An integer bound $\mathcal{B}>1$ on the degrees of e required for construction of the algebra $\mathcal{P}(e)$ of extended polynomials.

The zero bound: The numerical threshold $0<\zeta\ll 1$ for the zero test of polynomials in the flow of computation: if the absolute value of each coefficient in a polynomial is less than ζ then the polynomial is considered as zero. The utilization of the threshold in the algorithm is described in details on page 265.

The values of the e-bound and the zero bound are not used in the case of an exact arithmetic. In contrast, the time and the success of a computation with a floating point arithmetic strongly depends on the utilized e-bound and the zero bound parameters. So far we failed to establish reasonable rules to compute appropriate values of the parameters automatically and therefore it is the responsibility of a user to adjust the parameters for a specified polynomial system. If either value is omitted in the input, the algorithm uses the default settings

$$\mathcal{B} = 3, \quad \zeta = 2^{1-\left\lfloor \frac{d}{\mathcal{B}} \right\rfloor} \text{ with } d - \text{the length of mantissa in bits in the specified floating point domain.}$$

The last thing to be done before we proceed with the actual description of the algorithm in all details is the clarification of the total reduction process in $\mathcal{P}(e)$ to fix a slight incompliance between the theory of Gröbner bases in the algebra $\mathcal{P}(e)$ of extended polynomials as presented in chapter 2 and the utilization of the theory in our implementation of numerical Gröbner bases computation. On the one hand, our implementation uses the total reduction of polynomials. On the other hand, in the development of the theory of Gröbner bases in $\mathcal{P}(e)$, we have used *top-reducibility* and the respective *top-reduction algorithm* (see section 2.3.3) instead of the reducibility of an arbitrary term in a polynomial and the respective *total reduction algorithm* in order to avoid an extra, purely technical overhead in the reduction related part of the theory. But the particular variant of the reducibility notion and the respective reduction algorithm are not essential in the development of a theory of Gröbner bases, both in the classical version and in our instantiation for the algebra $\mathcal{P}(e)$, provided that in both cases the reducibility notion is essentially nothing more for the theory than a prerequisite to establish the reduction algorithm while the reduction algorithm is essentially nothing more but a possible way to compute normal forms (cf. definition 2.3.15 on page 47). In short, as long as the reduction algorithm corresponding to the used reducibility notion guarantees that the result of the reduction of a polynomial modulo a set of simplifiers is top-minimal (in the sense of definition 2.3.14 on page 45) modulo the simplifiers, such a reduction algorithm is as good for the theory as well as for the computation of Gröbner bases as any other variation of the reduction process; though the time of computation and, in the case of numerical Gröbner bases, the numerical stability may be considerably different.

The "total reducibility" of a polynomial modulo a given set of non-zero polynomials in the algebra $\mathcal{P}(e)$ of extended polynomials in the sense of reducibility of an arbitrary term in the polynomial modulo the set can be established as follows (cf. definition 2.3.12 on page 43).

Definition 3.0.6.

- For any $f, g \in \mathcal{P}(e)$ and $h \in \mathcal{P}(e) \setminus \{0\}$ we say that f **reduces in one step to g modulo h by eliminating** $s \in \operatorname{supp}(f)$ iff $\ell\mathrm{t}(h) \,|\, s$ and

  $$g = f - \alpha \cdot \frac{s}{\ell\mathrm{t}(h)} \cdot h \quad \text{with} \quad \alpha = \frac{\mathfrak{C}_f(s)}{\ell\mathrm{c}(h)} \in \mathbb{k}^*$$

  where $\mathfrak{C}_f(s)$ denotes the coefficient at s in f.

  The vital property that makes the definition meaningful is the stability of the leading term function: $\ell\mathrm{t}\left(\frac{s}{\ell\mathrm{t}(h)} \cdot h\right) = \frac{s}{\ell\mathrm{t}(h)} \cdot \ell\mathrm{t}(h) = s$ *by part a) of corollary 2.3.6 on page 41 provided that* $h \neq 0$ *and* $s \neq 0$*. Note however that the stability of the leading term function is not generally guaranteed under multiplication in* $\mathcal{P}(e)$*: it is possible that* $\ell\mathrm{t}(w \cdot q) \neq w \cdot \ell\mathrm{t}(q)$ *for* $q \in \mathcal{P}(e) \setminus \{0\}$*,* $w \in \mathbb{T}(e)^*$ *and* $w \cdot q \neq 0$*, in this case* $w \cdot \ell\mathrm{t}(q) = 0$*.*

- For any $f, g \in \mathcal{P}(e)$ and $F \subseteq \mathcal{P}(e) \setminus \{0\}$

  - f reduces in one step to g modulo F by eliminating $s \in \operatorname{supp}(f)$ iff there exists $h \in F$ such that f reduces in one step to g modulo h by eliminating s.
  - f reduces to g modulo F iff there exists a sequence $h_o, \ldots, h_m$ with $h_i \in \mathcal{P}(e)$ for $0 \leq i \leq m$, $h_o = f$, $h_m = g$ and for each $0 \leq i \leq m-1$ there exists $s_i \in \operatorname{supp}(h_i)$ such that h_i reduces in one step to h_{i+1} modulo F by eliminating s_i.

A possible version of the respective total reduction algorithm may be established as follows.

Require: $F \subseteq \mathcal{P}(e) \setminus \{0\}$
Require: $h \in \mathcal{P}(e)$ a polynomial to be totally reduced modulo F.
Ensure: h is totally reduced modulo F
  while $\exists s \in \operatorname{supp}(h)$ such that $\ell\mathrm{t}(r) \mid s$ with $r \in F$ **do**
    take $m = \max\left(\left\{ s \in \operatorname{supp}(h) \;\middle|\; \ell\mathrm{t}(r) \mid s \text{ with some } r \in F \right\}\right)$
    take $q \in F$ such that $\ell\mathrm{t}(q) \mid m$
    $w := \frac{\mathfrak{M}_h(m)}{\ell\mathrm{m}(q)}$ { where $\mathfrak{M}_h(m)$ is the monomial in h with the term m}
    $h := h - w \cdot q$ {eliminate m from h}
  end while

Let $h \in \mathcal{P}(e)$ and $F \subseteq \mathcal{P}(e) \setminus \{0\}$. Assume $r \in \mathcal{P}(e)$ is the result of total reduction of h modulo F performed by the foregoing algorithm. Then it is easy to show, by analogy with the flow of argumentation in lemma 2.3.19 on page 47 and lemma 2.3.20 on page 48 together with the observation that the sequence of the terms m, as they are chosen in the consecutive iterations of the while-loop in the total reduction algorithm, forms a strictly $<$-descending sequence of terms (and hence the sequence is finite by the well-ordering property of $<$), that

- the total reduction algorithm terminates,
- r is a normal form of h modulo F in the sense of definition 2.3.15 on page 47 i.e.

$$\begin{aligned} & h - r \in \langle F \rangle_{\mathcal{P}(e)} \quad \text{and} \\ & r \quad \text{is top-minimal modulo} \quad F, \end{aligned}$$

- there exists a Gröbner representation of $h - r$ in terms of F (in the sense of definition 2.3.16 on page 48 that contains a subtle difference from the "usual" notion of a Gröbner representation).

As a cheap way to improve the numerical stability, the reduction algorithm used in our implementation of the numerical Gröbner bases performs the *scaling* of the intermediate results in the reduction process during a Gröbner basis computation by multiplying the intermediate results with non-zero scalars. The result of such a reduction with scaling is not generally a normal form but a weak version of the so-called quasi-normal form (in the sense of definition 2.4.4 on page 106) provided that the multiplication of a non-zero polynomial with a non-zero scalar preserves both the leading term and the ideal membership of the polynomial: if $r \in \mathcal{P}(e)$ is the result of reduction with scaling of $h \in \mathcal{P}(e)$ modulo $F \subseteq \mathcal{P}(e) \setminus \{0\}$ then there exists a non-zero scalar $\alpha \in \Bbbk^*$ such that r is a normal form of $\alpha \cdot h$ modulo F (cf. lemma 2.4.14 on page 112). Note, however, that such a reduction with scaling is generally not feasible as a tool to compute the canonical representative of the residue class of a polynomial modulo an ideal with the already attained Gröbner basis since the scaling generally changes the residue class of the polynomial modulo the ideal unless the polynomial is in the ideal. Also note that our implementation uses a completely rearranged as compared with the aforementioned sample total reduction, and a highly sophisticated total reduction algorithm except of the very final total auto-reduction of the already attained Gröbner basis.

3.1 The preprocessing

Given a system of polynomials $\mathcal{F} \subseteq \mathcal{P}$ in the original polynomial algebra equipped with an admissible term order $<_{\mathbb{T}}$, the preprocessing step translates the system into $F \subseteq \mathcal{P}(e)$ in the algebra of extended polynomials with a specified extension $<$ of the original term order $<_{\mathbb{T}}$ by assigning an e-degree to each term $t \in \operatorname{supp}(p)$ in each polynomial $p \in F$ such that a smaller order of magnitude of the coefficient $\alpha_t \in \Bbbk^*$ at t in p is encoded by a higher e-degree.

Assume that the input system $\mathcal{F}$ is given with limited accuracy in the specified floating-point domain (otherwise no preprocessing is applied) with d bits used for the representation of mantissa. Let $\mathcal{B} > 1$ be the e-bound provided by the user. For each $\alpha \in \Bbbk^*$ such that $|\alpha| < 1$ we define the *infinitesimal order* of α as

$$\mathcal{O}(\alpha) = \left\lfloor \frac{-\mathcal{B} \cdot b}{d} \right\rfloor \in \mathbb{N}_o$$

where $b \in \mathbb{Z}$ is the binary exponent in the normalized representation of α in the specified floating-point domain.

This proposed infinitesimal order appears to be reasonably nice in our computational experiments but it represents purely the precision provided by the specified floating-point domain and does not include the precision of the system $\mathcal{F}$. Note that any $\alpha \in \Bbbk^*$ with $|\alpha| < 1$ such that $\mathcal{O}(\alpha) \geq \mathcal{B}$ is considered as indistinguishable from zero in the algorithm.

The polynomials $\mathcal{F}$ are translated to F just by preprocessing of each element of $\mathcal{F}$.

Require: $\mathcal{F} \subseteq \mathcal{P}$ – the input system of polynomials
Ensure: $F \subseteq \mathcal{P}(e)$ – the initial system for the subsequent computation of a Gröbner basis in $\mathcal{P}(e)$
  $F = \emptyset$
  for $p \in \mathcal{F}$ **do**

f:= preprocessed p
F:=F∪$\{f\}$
end for

The preprocessing of each polynomial

$$p = \sum_{t \in \operatorname{supp}(p)} \alpha_t \cdot t \in \mathcal{F} \quad \text{with} \quad \alpha \in \mathbb{k}^* \quad \text{for} \quad t \in \operatorname{supp}(p)$$

consists of two steps

Normalization: Let m be the smallest integer power of 2 such that

$$2^m > |\alpha_t| \quad \text{for any} \quad t \in \operatorname{supp}(p).$$

If $m>0$ then $p{:=}2^{-m} \cdot p$ otherwise nothing is done. In either case

$$|\alpha_t| < 1 \quad \text{for all} \quad t \in \operatorname{supp}(f)$$

after normalization. The scaling with a power of 2 is a well-known trick in numerical analysis to prevent the roundoff in numerical values on a binary machine.

Initialization of e-degrees: After normalization the polynomial p is translated to $f \in \mathcal{P}(e)$ by assigning the e-degree to each term $t \in \operatorname{supp}(p)$ as the infinitesimal order $\mathcal{O}(\alpha_t)$ of the coefficient at t in f or excluding t if $\mathcal{O}(\alpha_t) \geq \mathcal{B}$

$$\mathcal{P} \ni p = \sum_{t \in \operatorname{supp}(p)} \alpha_t \cdot t \longrightarrow \sum_{t \in \operatorname{supp}(p)} \alpha_t \cdot \left(t \cdot e^{\mathcal{O}(\alpha_t)}\right) = \sum_{\substack{t \in \operatorname{supp}(p) \\ \mathcal{O}(\alpha_t) < \mathcal{B}}} \alpha_t \cdot \left(t \cdot e^{\mathcal{O}(\alpha_t)}\right) = f \in \mathcal{P}(e).$$

3.2 The layout of the algorithm

This section deals with the upper-most layer in the description of our algorithm to compute a Gröbner basis in $\mathcal{P}(e)$. The algorithm takes a set of polynomials $F \subseteq \mathcal{P}(e)$ and computes the totally reduced Gröbner basis G of F. In the case of numerical coefficients with limited accuracy (which is our main concern) the resulting basis should be considered as the basis of a polynomial system in a neighborhood of the original system due to the inevitable roundoff errors that occur in the flow of computation if it has the same structure of the zero set.

The shallow layout of the algorithm presented below looks pretty much like the usual Buchberger algorithm for computing Gröbner bases except for two important particularities.

- The algorithm maintains an *extra pending set of critical elements* – the set of e-shifts. This extra type of critical elements has to be considered and treated in the flow of computation as a price for computing Gröbner bases in the algebra $\mathcal{P}(e)$ of extended polynomials, according to the results of section 2.3.6. In short, the completion procedure, devised as the result of the analysis of the ways how the leading term of a polynomial can cancel in a polynomial combination and the consequences of the cancellation, has to deal with the extra type of critical elements due to the extra possibility to cancel the leading term of a polynomial in the algebra $\mathcal{P}(e)$ of extended polynomials: the leading term of a polynomial can be cancelled by multiplication with a sufficiently high power of e.

- At each iteration of the while loop the algorithm takes a *block of critical elements* for *simultaneous* reduction. This idea is taken from the F_4 algorithm developed by J. Faugere and appears to be useful in the numerical setup: it is the key for our version of the reduction process that is also based on the reduction procedure of F_4 but has been considerably modified and extended for numerical stability reasons. Furthermore, the simultaneous reduction of several critical elements reduces the number of critical elements to be reduced in the flow of computation as it is also the case in the original F_4 algorithm: the simultaneous selection of several critical elements for reduction eliminates the necessity to stick to some linear, usually predefined in the implementation, order of selecting critical elements and hence reduces the overhead of the inappropriate for a particular input, order of treatment of the critical elements. For example, a well-known and widely used selection strategy for S-polynomials is a refinement of Buchberger's normal selection strategy where the ties are broken by the sugar degrees of the S-polynomials ([GMN$^+$91]); this selection strategy still leaves quite a freedom in the order of selection of S-polynomials and it is empirically observed that the number of S-polynomials which the algorithm has to consider in the flow of computation may depend very much on the way how this remaining freedom is exploited while the F_4 algorithm behaves like an advantageous for a particular problem selection strategy has been utilized (see the comparison charts in [Fau]).

The presented layout is intended to provide a crude idea of how the parts of the algorithm work together and also to help with navigation in the rest of the chapter. It is not supposed to provide a complete and self-sufficient description (like what are precisely the critical elements placed into the corresponding pending sets), the latter is delegated to the subsequent sections.

Require: $F \subseteq \mathcal{P}(e)$ – the input system of polynomials

Ensure: $G \subseteq \mathcal{P}(e) \setminus \{0\}$ – the totally reduced Gröbner basis of a system in a neighborhood of F.

$G := F \setminus \{0\}$
$G :=$ totally auto-reduced G { the initial auto-reduced basis}
$S :=$ the S-pairs of the elements of $G \times G$ { the pending set of S-pairs}
$E :=$ the e-shifts of the elements of G { the pending set of e-shifts}
eliminate the redundant elements in S and E
while $S \neq \emptyset$ or $E \neq \emptyset$ **do**
  if $S = \emptyset$ **then**
    take a block of e-shifts $E_o \subseteq E$
    $B :=$ the e-shifts E_o evaluated to the polynomial form from the internal representation used to maintain the pending set
    $E := E \setminus E_o$
  else
    take a block of S-pairs $S_o \subseteq S$
    $B :=$ the S-pairs S_o evaluated to half-S-pairs (see page 221) from the internal representation used to maintain the pending set
    $S := S \setminus S_o$
  end if
  $B :=$ the set B totally auto-reduced and totally reduced modulo G { the resulting B does not contain 0 – the elements that are reduced to zero are immediately removed in the flow of reduction following the common practice}
  $S_o :=$ the S-pairs from the elements of $B \times B$ and $B \times G$
  $E_o :=$ the e-shifts of the elements of B
  $G := G \cup B$
  if $S_o \neq \emptyset$ or $E_o \neq \emptyset$ **then**
    $S := S \cup S_o$
    $E := E \cup E_o$
    eliminate the redundant elements in S and E
  end if
end while
$G :=$ totally auto-reduced G { the final auto-reduction of the resulting Gröbner basis}

As can be seen from the layout there are three places in the algorithm where the reduction process is invoked:

- the initial auto-reduction of the original system,

- the reduction of a block of critical elements in the *main loop* of the algorithm and

- the final total auto-reduction of the computed Gröbner basis.

The initial auto-reduction and the reduction of critical elements in the main loop invoke a highly specific "$F_4 - type$" reduction process described in details in section 3.4. It also involves the trimming of e-degrees trick for the intermediate polynomials (see page 266) to maintain numerical stability especially in the case of a degenerated genuine Gröbner basis. The trimming is precisely the trick in our algorithm to deviate from the computation of the genuine Gröbner basis to that of a neighboring system.

In contrast, the final auto-reduction of the computed basis is done by the use of a very simple and very standard reduction algorithm without trimming of e-degrees to prevent a change in the leading terms of the basis during the final auto-reduction. Such a change would generally modify the ideal of leading terms of the basis elements and hence would almost inevitably ruin the Gröbner basis property of the result of execution of the main loop. On the other hand, the auto-reduction of the resulting Gröbner basis $G \subseteq \mathcal{P}(e)$ can not ruin a correct structure of the leading terms when it is performed in a well-known clever way easily transferred to $\mathcal{P}(e)$ from the classical theory of Gröbner bases where the auto-reduction is split in two parts as follows.

- Computation of a *minimal* Gröbner basis (cf. definition 2.3.26 on page 84) whose leading terms form the minimal generating set of the ideal of leading terms by excluding all polynomials $g_1 \in G$ such that $\ell\mathrm{t}(g_2) \,|\, \ell\mathrm{t}(g_1)$ for some $g_2 \in G \backslash \{g_1\}$. This operation is perfectly safe in the numerical respect since it does not include any operation with coefficients. This well-known from the classical Gröbner bases theory approach to obtain a minimal Gröbner basis from a Gröbner basis has been reestablished for the algebra $\mathcal{P}(e)$ of extended polynomials as the algorithm `MinGröbner` on page 85 and corroborated in lemma 2.3.39 on page 85.

- The total auto-reduction of the minimal Gröbner basis G attained at the previous step: each $g \in G$ is replaced by $\ell\mathrm{m}(g) + r$ where r is the result of total reduction (performed without scaling!) of $g - \ell\mathrm{m}(g)$ modulo G[3]. More formally the procedure can be established as the following algorithm.

[3] We can not use the usual approach when each $g \in G$ is replaced by the result of total reduction modulo $G \backslash \{g\}$ because, due to the negative weight of the variable e, it is generally possible that $\ell\mathrm{t}(h) \,|\, s$ with $s \in \mathrm{supp}\,(h) \backslash \{\ell\mathrm{t}(h)\}$, $h \in \mathcal{P}(e) \backslash \{0\}$ and therefore a polynomial $g \in G$ should also be used to eliminate certain terms $\mathrm{supp}\,(g) \backslash \{\ell\mathrm{t}(g)\}$ in itself!

Require: $G \subseteq \mathcal{P}(e) \setminus \{0\}$ – a minimal Gröbner basis
Ensure: $H \subseteq \mathcal{P}(e) \setminus \{0\}$ – the totally reduced Gröbner basis of $\langle G \rangle_{\mathcal{P}(e)}$
  $H := \emptyset$
  while $G \neq \emptyset$ **do**
    take any $g \in G$
    $r :=$ the result of total reduction of $g - \ell\mathrm{m}(g)$ modulo $G \cup H$ e.g. by the algorithm on page 204.
    $H := H \cup \{\ell\mathrm{m}(g) + r\}$
    $G := G \setminus \{g\}$
  end while

The termination and correctness of the algorithm are easy to validate by the observation that no top-reduction has to be performed for the input G provided that the input is a minimal Gröbner basis whose leading terms do not divide each other and then following the same way of argumentation as in lemma 2.3.40 on page 86 for the auto-top-reduction algorithm on page 87 provided that the the proof of the lemma is formulated in terms of the properties of the normal form (as defined on page 47) and hence the argumentation is independent of the particular type of reducibility since the respective reduction algorithm is a possible way to attain normal forms algorithmically.

3.3 The critical sets

According to section 2.3.6, the computation of a Gröbner basis in the algebra $\mathcal{P}(e)$ of extended polynomials has to consider two kinds of critical elements: the stabilized S-polynomials (definition 2.3.22) and the critical e-shifts - the elements of the e-orbit of a polynomial (definition 2.3.18). In actual fact the layout of the Gröbner basis computation algorithm in the algebra of extended polynomials $\mathcal{P}(e)$ presented in the theoretical discussion on page 79 uses a greatly oversized version of critical sets which is well-suited for the correctness proof of the Gröbner basis algorithm in $\mathcal{P}(e)$ (lemma 2.3.36) but likely to be made smaller for performance and numerical reliability reasons: For any polynomial f as well as for any pair of different polynomials f_1 and f_2 from the current basis, the version of the algorithm attained in section 2.3.6 considers

- the set of critical e-shifts of f

$$\mathrm{EC}\,(f) = \left\{ e^k \cdot f \;\middle|\; 0 \le k < \mathcal{B},\; e^k \cdot \ell\mathrm{t}(f) = 0 \right\} \setminus \left\{ 0 \right\},$$

- the set of stabilized S-polynomials of f_1 and f_2

$$\mathcal{S}_{\mathbb{T}}\left(\{f_1,f_2\}\right)=\Big\{\mathcal{S}_{\mathbb{T}}\left(g_1,g_2\right)\ \Big|\ g_1{\in}\mathrm{E}\left(f_1\right), g_2{\in}\mathrm{E}\left(f_2\right),\ \deg_e\big(\ell\mathrm{t}(g_1)\big){=}\deg_e\big(\ell\mathrm{t}(g_2)\big)\Big\}\setminus\Big\{0\Big\}$$

where

$$\mathcal{S}_{\mathbb{T}}\left(h_1,h_2\right)=\frac{\mathrm{lcm}\big(\ell\mathrm{t}(h_1),\ell\mathrm{t}(h_2)\big)}{\ell\mathrm{m}(h_1)}{\cdot}h_1-\frac{\mathrm{lcm}\big(\ell\mathrm{t}(h_1),\ell\mathrm{t}(h_2)\big)}{\ell\mathrm{m}(h_2)}{\cdot}h_2$$ – the stabilized S-polynomial of non-zero polynomials h_1 and h_2 such that $\deg_e\big(\ell\mathrm{t}(h_1)\big){=}\deg_e\big(\ell\mathrm{t}(h_2)\big)$

and

$$\mathrm{E}\left(h\right)=\Big\{e^k{\cdot}h\mid 0{\le}k{<}\mathcal{B}\Big\}\setminus\Big\{0\Big\}$$ — the e-orbit of a polynomial h.

Our implementation utilizes a **refined set of critical elements** as follows.

- For each polynomial f from the current basis, the algorithm considers the **e-shift**

$$E\mathcal{S}\left(f\right)=e^d{\cdot}f\quad\text{with}\quad d=\mathcal{B}{-}\deg_e\big(\ell\mathrm{t}(f)\big)\tag{1}$$

  unless $E\mathcal{S}\left(f\right){=}0$.

- For each pair of different polynomials f_1 and f_2 from the current basis the algorithm considers the **S-polynomial**

$$\mathcal{S}\left(f_1,f_2\right){=}\frac{L(f_1,f_2)}{\ell\mathrm{m}(f_1)}{\cdot}f_1{-}\frac{L(f_1,f_2)}{\ell\mathrm{m}(f_2)}{\cdot}f_2\ \text{with}\ L{=}\mathrm{lcm}\big(\ell\mathrm{t}(f_1),\ell\mathrm{t}(f_2)\big)\tag{2}$$

  unless $\mathcal{S}\left(f_1,f_2\right){=}0$. The least common multiple of extended terms above is taken in the sense of definition 2.3.21.

It is clear (also by part a) of 2.3.1) that the e-shift (1) is an element of the critical e-orbit of a polynomial. Furthermore the S-polynomial (2) can be considered as a stabilized S-polynomial as follows.

Lemma 3.3.1. For any non-zero distinct polynomials f_1 and f_2

$$\text{if } \mathcal{S}\left(f_1,f_2\right){\neq}0 \text{ then } \mathcal{S}\left(f_1,f_2\right){\in}\mathcal{S}_{\mathbb{T}}\left(\{f_1,f_2\}\right).$$

Proof. Let f_1, f_2 be (non-zero and distinct) polynomials and assume that $\mathcal{S}(f_1, f_2){\neq}0$. We have to show that $\mathcal{S}(f_1, f_2) = \mathcal{S}_{\mathbb{T}}(g_1, g_2)$ with $g_1{\in}\mathrm{E}(f_1)$ and $g_2{\in}\mathrm{E}(f_2)$ where $\mathrm{E}(h){=}\{e^k{\cdot}h \mid 0{\le}k{<}\mathcal{B}\}$ is the e-orbit of a polynomial $h{\in}\mathcal{P}(e)$ (cf. definition 2.3.18 on page 53). Denote

$$
\begin{aligned}
L(f_1, f_2) &= \mathrm{lcm}\big(\ell\mathrm{t}(f_1), \ell\mathrm{t}(f_2)\big),\\
t &= \mathrm{T}\big(L(f_1, f_2)\big) \ \text{ – the main part (without the power of } e\text{) of } L(f_1, f_2),\\
d &= \deg_e\big(L(f_1, f_2)\big) \ \text{ – the } e\text{-degree of } L(f_1, f_2),\\
\upsilon &= e^d \ \text{ – the } e\text{-part of } L(f_1, f_2),\\
t_1 &= \mathrm{T}\big(\ell\mathrm{t}(f_1)\big), \quad t_2 = \mathrm{T}\big(\ell\mathrm{t}(f_2)\big),\\
d_1 &= \deg_e\big(\ell\mathrm{t}(f_1)\big), \quad d_2 = \deg_e\big(\ell\mathrm{t}(f_2)\big),\\
\upsilon_1 &= e^{d_1}, \quad \upsilon_2 = e^{d_2}.
\end{aligned}
$$

Note that

$$t = \mathrm{lcm}(t_1, t_2),\ d = \max(d_1, d_2),\ L(f_1, f_2) = \upsilon{\cdot}t \tag{3}$$

according to definition 2.3.21 of the least common multiple of extended terms and, by part a) of lemma 2.3.28,

$$\upsilon{\neq}0. \tag{4}$$

Also, by part b) of lemma 2.3.28 and the unique decomposition of any extended term into the product of the main part and the power of e (part d) of theorem 2.3.1),

$$t_1 \,|\, t, \quad t_2 \,|\, t, \quad \upsilon_1 \,|\, \upsilon, \quad \upsilon_2 \,|\, \upsilon.$$

Take

$$g_1 = \frac{\upsilon}{\upsilon_1}{\cdot}f_1, \quad g_2 = \frac{\upsilon}{\upsilon_2}{\cdot}f_2. \tag{5}$$

Then, by corollary 2.3.1,

$$\frac{\upsilon}{\upsilon_i}\ell\mathrm{t}(f_i) = \frac{\upsilon}{\upsilon_i}{\cdot}\upsilon_i{\cdot}t_i = \upsilon{\cdot}t_i \overset{\substack{(4),\\ (\text{cor. } 2.3.1)}}{\neq} 0 \tag{6}$$

and hence, by part a) of corollary 2.3.6,

$$g_i{\neq}0, \quad \ell\mathrm{t}(g_i) \overset{(\text{cor. } 2.3.6)}{=} \frac{\upsilon}{\upsilon_i}\ell\mathrm{t}(f_i) \overset{(6)}{=} \upsilon{\cdot}t_i \quad \text{for } \ i = 1, 2. \tag{7}$$

Therefore $g_1 \in \mathrm{E}(f_1)$, $g_2 \in \mathrm{E}(f_2)$ and the S-polynomial can be rewritten as

$$\begin{aligned}
\mathcal{S}(f_1, f_2) &\overset{(2)}{=} \frac{L(f_1,f_2)}{\ell\mathrm{m}(f_1)} \cdot f_1 - \frac{L(f_1,f_2)}{\ell\mathrm{m}(f_2)} \cdot f_2 \\
&\overset{(3)}{=} \frac{\upsilon \cdot t}{\ell\mathrm{c}(f_1) \cdot \upsilon_1 \cdot t_1} f_1 - \frac{\upsilon \cdot t}{\ell\mathrm{c}(f_2) \cdot \upsilon_2 \cdot t_2} f_2 \\
&= \frac{t}{\ell\mathrm{c}(f_1) \cdot t_1} \cdot \frac{\upsilon}{\upsilon_1} f_1 - \frac{t}{\ell\mathrm{c}(f_2) \cdot t_2} \cdot \frac{\upsilon}{\upsilon_2} f_2 \\
&\overset{(5)}{=} \frac{\upsilon \cdot t}{\ell\mathrm{c}(f_1) \cdot \upsilon \cdot t_1} \cdot g_1 - \frac{\upsilon \cdot t}{\ell\mathrm{c}(f_2) \cdot \upsilon \cdot t_2} \cdot g_2 \overset{(7)}{=} \frac{L(f_1,f_2)}{\ell\mathrm{m}(g_1)} \cdot g_1 - \frac{L(f_1,f_2)}{\ell\mathrm{m}(g_2)} \cdot g_2 .
\end{aligned} \tag{8}$$

On the other hand,

$$\begin{aligned}
L(g_1, g_2) &:= \mathrm{lcm}\big(\ell\mathrm{t}(g_1), \ell\mathrm{t}(g_2)\big) \\
&\overset{(7)}{=} \mathrm{lcm}\big(\upsilon \cdot t_1, \upsilon \cdot t_2\big) \\
&\overset{(3)}{=} \upsilon \cdot \mathrm{lcm}\big(t_1, t_2\big) = e^d \cdot \mathrm{lcm}\big(t_1, t_2\big) \\
&= e^{\max(d_1, d_2)} \cdot \mathrm{lcm}\big(t_1, t_2\big) \\
&= e^{\max\left(\deg_e(\ell\mathrm{t}(f_1)), \deg_e(\ell\mathrm{t}(f_2))\right)} \cdot \mathrm{lcm}\Big(\mathrm{T}\big(\ell\mathrm{t}(f_1)\big), \mathrm{T}\big(\ell\mathrm{t}(f_2)\big)\Big) \\
&\overset{(?)}{=} \mathrm{lcm}\big(\ell\mathrm{t}(f_1), \ell\mathrm{t}(f_2)\big) = L(f_1, f_2)
\end{aligned} \tag{9}$$

where the equality (?) is by definition 2.3.21 of the least common multiple of a pair of terms in $\mathbb{T}(e)^*$. Therefore, by (8) and (9),

$$\mathcal{S}(f_1, f_2) = \frac{L(g_1,g_2)}{\ell\mathrm{m}(g_1)} \cdot g_1 - \frac{L(g_1,g_2)}{\ell\mathrm{m}(g_2)} \cdot g_2$$

which is the stabilized S-polynomial of $\mathcal{S}_{\mathbb{T}}(g_1, g_2)$ according to definition 2.3.22. This completes the proof since $g_1 \in \mathrm{E}(f_1)$ and $g_2 \in \mathrm{E}(f_2)$. ☐

According to the discussion above, the elements of the form (1) and (2) are among the critical elements which the Gröbner basis computation algorithm in $\mathcal{P}(e)$ has to consider according to the results of section 2.3.6. On the other hand, it is sufficient to consider the elements of the form (1) and (2) in the flow of the Gröbner basis algorithm; this is seen by constructing an appropriate standard representation

- for an arbitrary (non-zero) polynomial in the critical e-orbit of a polynomial from the current basis in terms of the e-shifts of the elements of the current basis as well as

- for an arbitrary (non-zero) stabilized S-polynomial for any pair of different polynomials from the current basis in terms of the e-shifts of the elements of the current basis

in the spirit of e.g. the proof of Buchberger's second criterion for the algebra $\mathcal{P}(e)$ of extended polynomials in lemma 2.3.35 following the algorithm correctness proof by analogy with lemma 2.3.36.

For an alternative, possibly simpler and even more natural way to see and argue the validity and adequacy of the smaller critical elements set, we may reinterpret the Gröbner basis computation of an ideal $\langle F \rangle$ in the algebra $\mathcal{P}(e) = \Bbbk[X, e]$ of extended polynomials as the computation of the Gröbner basis of the ideal $\left\langle F_y \cup \{y^{\mathcal{B}}\} \right\rangle$ in the algebra $\mathcal{P}(y) = \Bbbk[X, y]$ with a new free variable y; an arbitrary admissible term order $<_y$ on $\mathcal{P}(y)$ such that $\phi(w_1) < \phi(w_2)$ follows $w_1 <_y w_2$ for any terms $w_1, w_2 \in \mathcal{P}(y)$ with ϕ – the canonical epimorphism of $\mathcal{P}(y)$ on $\mathcal{P}(e)$ given as the unique homomorphic extension of the correspondence $y \rightarrow e$ provided by the universal property of polynomial rings, the reduction strategy which keeps all polynomials in the flow of the algorithm totally reduced with respect to $y^{\mathcal{B}}$ and F_y – an arbitrary pre-image of F in $\mathcal{P}(e)$ under ϕ e.g. attained by the substitution $e \rightarrow y$ in each polynomial $f = f(X, e) \in F$.

Recall that e is a local variable w.r.t. $<$ (i.e. $e < 1$) while $x > 1$ for any $x \in X$ in our setup and hence the appropriate Gröbner bases theory to describe and build up the artful verification of the Gröbner basis algorithm in $\mathcal{P}(y)$ is the variation of the Gröbner bases theory to the so-called mixed ordering case which includes term well-orderings (the classical theory ([Buc65],[Buc83]) and tangent cone term orderings ([Mor82],[MT]) as special cases. Such a theory together with the Gröbner basis algorithm has been attained not so long time ago by H. Grabe [Gra] and independently by the Singular group [GGM+96], [Sch96].

Unfortunately these instantiations of the mixed ordering case Gröbner bases theory are not readily suited for our situation because they require (at least, implicit) homogenization of an ideal (in the spirit of [Laz83], see also section 10.3 in [BW93]) while our approach deals with a "non-homogenized" version of a Gröbner basis computation and, in particular, with a "non-homogenized" reduction process due to the disregarding, experience towards the numerical computation of the Gröbner basis of the homogenized ideal of a zero dimensional system in the early experiments. Indeed, the homogenization effectively introduces an extra variable in the input system: it usually results in a considerably increased complexity of computation and, what is more striking, "erases the zero-dimensionality property" within the computation which complicates the (human) analysis of the intermediate result regarding the minimal terms modulo the currently attained basis.

Hence, the utilization of the mentioned theories as a theoretical foundation for our approach towards numerical computation of Gröbner bases would at least require a reconsideration of the reduction algorithm part at least to make it work without (at least implicitly assumed but nevertheless explicitly affecting the flow of computation) homogenization for the orderings like $\mathbb{T}_e$ (see page 90) and ${}_e\mathbb{T}$ (see page 168) which are vital for our numerical Gröbner bases computation. The author considers the development of such a modification of the mixed ordering case Gröbner bases theory as a challenge for the not so far future.

For now on we refer to the result of the forthcoming theory due to the formidable importance of the result for our implementation of the algorithm as follows: take any $f, g{\in}\mathcal{P}(e)$ and consider their pre-images $f_y, g_y{\in}\mathcal{P}(y)$ constructed by the substitution $e{\to}y$ as specified above. Note that the e-shifts $E\mathcal{S}\,(f)$ and $E\mathcal{S}\,(g)$ in $\mathcal{P}(e)$ correspond to S-polynomials $\mathcal{S}\left(y^{\mathcal{B}}, f_y\right)$ and $\mathcal{S}\left(y^{\mathcal{B}}, g_y\right)$ respectively in $\mathcal{P}(y)$ while the S-polynomial $\mathcal{S}\,(g, f){\in}\mathcal{P}(e)$ corresponds to $\mathcal{S}\,(g_y, f_y){\in}\mathcal{P}(y)$. Buchberger's second criterion applied to the triple

$$\begin{tikzcd} y^{\mathcal{B}} \arrow[rr, dash] \arrow[dr, dash] & & f_y \arrow[dl, dash] \\ & g_y & \end{tikzcd}$$

says that if $\ell\mathrm{t}(g_y)\,|\,\mathrm{lcm}(y^{\mathcal{B}}, \ell\mathrm{t}(f_y))$ then the S-polynomial $\mathcal{S}\left(y^{\mathcal{B}}, f_y\right)$ has a standard representation and hence can be safely excluded from further consideration computing a Gröbner basis (analogously to [BW93] page 218) in terms of $\mathcal{S}\left(y^{\mathcal{B}}, g_y\right)$ and $\mathcal{S}\,(g_y, f_y)$ (analogously to [BW93] Proposition 5.70 on page 223). Now assume that the polynomials f and g are generated in the flow of a Gröbner basis computation. Then, due to the assumption on the reduction strategy, f_y, g_y are totally reduced with respect to $y^{\mathcal{B}}$: in particular $\deg_y\bigl(\ell\mathrm{t}(f_y)\bigr), \deg_y\bigl(\ell\mathrm{t}(g_y)\bigr){<}\mathcal{B}$ and hence

$$\ell\mathrm{t}(g_y)\,|\,\mathrm{lcm}\bigl(y^{\mathcal{B}}, \ell\mathrm{t}(f_y)\bigr) \quad \text{if and only if} \quad \mathrm{T}\bigl(\ell\mathrm{t}(g_y)\bigr)\,|\,\mathrm{T}\bigl(\ell\mathrm{t}(f_y)\bigr)$$

since $\deg_y\bigl(\mathrm{lcm}(y^{\mathcal{B}}, \ell\mathrm{t}(f_y))\bigr){=}\mathcal{B}$, where $\deg_y(w)$ and $\mathrm{T}(w)$ are defined for any term

$$w{=}y^d{\cdot}\prod_{x\in X} x^{d_x},$$

in $\mathcal{P}(y)$ as

$$\deg_y(w) = d \text{ and } \mathrm{T}(w) = \prod_{x \in X} x^{d_x}.$$

Therefore, it is not necessary to reduce the S-polynomial $\mathcal{S}\left(y^{\mathcal{B}}, f_y\right)$ if

$$\mathrm{T}\big(\ell\mathrm{t}(g_y)\big) \,|\, \mathrm{T}\big(\ell\mathrm{t}(f_y)\big)$$

while the S-polynomials $\mathcal{S}\left(y^{\mathcal{B}}, g_y\right)$, $\mathcal{S}\left(g_y, f_y\right)$ are reduced or excluded either by another instance of Buchberger's second criteria or by any other rule to detect superfluous S-polynomials in the flow of the Gröbner basis algorithm (analogously to [BW93] theorem 5.71 on page 225 based on theorem 5.64 on page 219; [AL94] proposition 3.3.4 on page 128 based on theorem 3.2.5 on page 121).

Turning back to the algebra $\mathcal{P}(e)$ where our algorithm operates, the observation above results in the following rule to exclude certain e-shifts from the reduction computing a Gröbner basis.

Observation (Buchberger's second criterion for e-shifts). It is not necessary to reduce the e-shift $E\mathcal{S}\left(f\right)$ if the e-shift $E\mathcal{S}\left(g\right)$ as well as the S-polynomial $\mathcal{S}\left(g, f\right)$ are either reduced or excluded by another instance of whatever criteria to detect superfluous (for reduction) critical elements and $\mathrm{T}\big(\ell\mathrm{t}(g)\big) \,|\, \mathrm{T}\big(\ell\mathrm{t}(f)\big)$ – the main part (the part without the power of e) of $\mathrm{T}\big(\ell\mathrm{t}(g)\big)$ divides the main part of $\mathrm{T}\big(\ell\mathrm{t}(f)\big)$.

The algorithm maintains two pending sets of critical elements: the set E of e-shifts and the set S of S-pairs. The e-shifts are represented in E as the copies (internally represented as references) of the polynomials from the current basis, the evaluation to the "true" e-shift is performed according to (1) just before an element is submitted for reduction. Analogously the maintained representation of S stores the pairs (f, g) referred as S-**pairs** of the elements of the current basis instead of the "evaluated" S-polynomials. Furthermore, to avoid any arithmetical operations with coefficients in the maintenance of S, an S-pair is not evaluated completely according to (2) before submitting for (including mutual) reduction but evaluated partially to the pair of shifts

$$\frac{\mathrm{lcm}\big(\ell\mathrm{t}(f), \ell\mathrm{t}(g)\big)}{\ell\mathrm{t}(f)} \cdot f, \qquad \frac{\mathrm{lcm}\big(\ell\mathrm{t}(f), \ell\mathrm{t}(g)\big)}{\ell\mathrm{t}(g)} \cdot g.$$

The lazy representation of critical elements in the pending sets has the following advantages.

- It stores explicitly the source polynomials from the current basis whose parameters (like the total and the sugar degree, the absolute value of the leading coefficient etc.) are incorporated into the priority orders on S and E (described below) which control the critical elements selection strategy. Furthermore it is easy to access the leading terms of the source polynomials of a critical element which is important for the utilization of Buchberger's criteria (see lemma 2.3.34 and lemma 2.3.35 for Buchberger's first and second criteria respectively for the algebra $\mathcal{P}(e)$) as well as for our observation towards elimination of certain e-shifts.

- Since typically a good number of critical elements is eliminated in the flow of the Gröbner basis algorithm, hence the lazy representation eliminates the redundant operations with the source polynomials (like a shift with a term) required for evaluation to the respective critical elements.

The pending sets of critical elements maintained in our implementation are equipped with a *priority order* that turns them into a sort of a priority queue. This order is compatible with the aspects of numerical safety and computation complexity in the use of a critical element for reduction: if the result of the reduction of a "dangerous" element is not zero then the reduced form of the element typically inherits the "danger"; its adding to the current basis eventually raises the chance of a numerical blunder or at least increases the computation time. Therefore every time the algorithm requires a new critical element for reduction (or, indeed a bunch elements which our implementation reduces simultaneously in the spirit of Faugere's F_4 algorithm), we prefer to choose the safest critical elements from the pending critical sets currently attained in the flow of the algorithm. The idea is the following: when we postpone the reduction of the critical elements which we would not like to reduce, the non-zero reduced forms of the "better" critical elements will enlarge the current basis and *provide a chance* that certain dangerous critical elements appear redundant according to Buchberger's second criterion and hence can be safely eliminated from the further consideration.

From the numerical point of view, it is desirable to avoid the reduction of critical elements formed from source polynomials which have very small leading coefficients after normalization. Indeed, if a polynomial with a very small leading coefficient is combined with another polynomial whose matching coefficient is of order 1 in the S-polynomial formation, then it must be multiplied by a very large factor; this amplifies the numerical roundoff error, results in the cancellation of leading digits (a well-known source of numerical instability) and may also create some tiny coefficients after normalization of the attained polynomial. Moreover, a very small leading coefficient of a source polynomial may indeed be zero except for rounding errors that have occurred in calculation. If this very small coefficient stays the leading one in the reduced form (of an S-polynomial or an e-shift) which is added to the current basis then the ideal of leading terms may become structurally wrong or, at least, any elimination (reduction) by this reduced form will result in gross errors in further calculations. On the other hand, if the tiny leading coefficient disappears during reduction then it might indeed be safe to submit the critical element for reduction and insert the reduced form to the current basis; but our current implementation does not go that far in the analysis of the consequences in selecting a critical element for reduction.

Note that the trimming of e-degrees mechanism in the reduction part of the algorithm (see page 266 for details) partially solves the problem of a tiny leading coefficient, by assigning a higher e-degree to the terms with very small coefficients. But it suffers from certain limitations: the term with a higher e-degree might still remain the leading one and also the mechanism how the order of magnitude of a coefficient is translated to the e-degree of the term at the coefficient is far from perfection in the current implementation.

From the point of view of the computational complexity, so far there are no well founded and elaborated rules for an optimal (in whatever sense) selection strategy of critical elements for reduction available in the computer algebra literature. However, the computational experience gained by the computer algebra community recommends a certain good-showing principle for such a selection strategy in the frame of the classical Buchberger algorithm: first reduce the S-polynomials with a smaller w.r.t. the utilized admissible term order least common multiple of the leading terms of the source polynomials ([Buc79], [BW93] pages $228-229$) and break the ties by the sugar degree of S-polynomials ([GMN$^+$91]). We borrow this principle in our implementation with an extension towards the total degree orderings which are of main concern for us. According to our experiments, the selection by pure minimal sugar degree of the critical elements results in more critical elements to be considered in the flow of computation than the selection by minimal total degree of the leading terms incorporated into the selection strategy of critical elements. This also confirms the analogous observation by Faugere ([Fau], page 13) whose F_4 algorithm we used as a foundation in our implementation. Therefore we stick to the minimal total degree controlled selection strategy of critical elements. However the sugar degree proves to be useful in the selection strategy of reduction discussed in section 3.4.2 about linear elimination.

In the pending set of S-pairs, an S-pair has a higher priority to be submitted for reduction if it has a smaller e-degree or a smaller degree of the main part of the least common multiple of the leading terms or a greater maximal absolute value of the leading coefficients of the pair . More precisely, an S-pair (f_1, f_2) has a higher priority (is better) than (g_1, g_2) if and only if in the componentwise difference of tuples

$$\Big(\deg_e(L_g), \deg\big(\mathrm{T}(L_g)\big), \max\big(|\ell\mathrm{c}(f_1)|, |\ell\mathrm{c}(f_2)|\big)\Big) - \Big(\deg_e(L_f), \deg\big(\mathrm{T}(L_f)\big), \max\big(|\ell\mathrm{c}(g_1)|, |\ell\mathrm{c}(g_2)|\big)\Big)$$

with

$$L_f = \mathrm{lcm}\big(\ell\mathrm{t}(f_1), \ell\mathrm{t}(f_2)\big) \qquad \text{and} \qquad L_g = \mathrm{lcm}\big(\ell\mathrm{t}(g_1), \ell\mathrm{t}(g_2)\big),$$

the left-most non-zero entry is positive.

Analogously, in the pending set of e-shifts, an e-shift has a higher priority to be submitted for reduction if it has a higher (!) e-degree or a smaller degree of the main part of the leading term or a greater absolute value leading coefficient of the source polynomial. More precisely, an e-shift

$$e^{d_f} \cdot f \neq 0 \ \text{ with } \ d_f = \mathcal{B} - \deg_e\big(\ell\mathrm{t}(f)\big) < \mathcal{B}$$

has a higher priority (is better) than

$$e^{d_g} \cdot g \neq 0 \ \text{ with } \ d_g = \mathcal{B} - \deg_e\big(\ell\mathrm{t}(g)\big) < \mathcal{B}$$

if and only if in the componentwise difference of tuples

$$\Big(\deg_e\big(\ell\mathrm{t}(f)\big), \deg\big(\mathrm{T}(\ell\mathrm{t}(g))\big), \big|\ell\mathrm{c}(f)\big|\Big) - \Big(\deg_e\big(\ell\mathrm{t}(g)\big), \deg\big(\mathrm{T}(\ell\mathrm{t}(f))\big), \big|\ell\mathrm{c}(g)\big|\Big),$$

the left-most non-zero entry is positive.

The algorithm runs until the pending sets of both the S-pairs and the e-shifts are empty. When a new block of critical elements is required for reduction it is taken from the set of S-pairs if the set is not empty; otherwise the block is taken from the set of e-shifts.

The block of S-pairs is selected as follows: For any S-pair (h_1, h_2) assign the e-degree and the degree in main variables of the S-pair as

$$\begin{aligned} \deg_e\big((h_1, h_2)\big) &:= \deg_e(L) \ \text{ and} \\ \deg_X\big((h_1, h_2)\big) &:= \deg\big(\mathrm{T}(L)\big) \ \text{ where } \ L = \mathrm{lcm}\big(\ell\mathrm{t}(h_1), \ell\mathrm{t}(h_2)\big). \end{aligned}$$

Let S be the pending set of S-pairs at the moment when a block of S-polynomials is requested for reduction. Set

$$a = \min_{(h_1,h_2)\in S} \Big(\deg_e\big((h_1,h_2)\big)\Big),$$

$$b = \min_{\substack{(h_1,h_2)\in S\\ \deg_e((h_1,h_2))=a}} \Big(\deg_X\big((h_1,h_2)\big)\Big).$$

Then the block for reduction is defined as

$$\tilde{S} = \Big\{(h_1,h_2)\in S \;\Big|\; \deg_e\big((h_1,h_2)\big)=a,\ \deg_X\big((h_1,h_2)\big)=b\Big\}.$$

It is clear by construction that any S-pair in $\tilde{S}$ has a higher priority than any in $S\backslash\tilde{S}$. We set $S := S\backslash\tilde{S}$ and eliminate redundant S-pairs from $\tilde{S}$ according to Buchberger's criteria: first we apply the first criterion, then the second criterion is applied in the priority order which specializes to the comparison of the maximal absolute value of the leading coefficients in an S-pair among the elements of $\tilde{S}$. Furthermore, the S-pairs that have been delivered for reduction or recognized to be redundant according to Buchberger's first criterion are moved to the global within the run of the algorithm set S_{old} of the *processed* S-pairs which is used in the applicability check of Buchberger's second criterion. The following layout summarizes the described procedure.

Require: $\tilde{S}$ - a block of S-pairs.
Ensure: The redundant S-pairs are eliminated in $\tilde{S}$.

Apply Buchberger's first criterion.
$S_{old} := S_{old} \cup \Big\{(f_1,f_2)\in\tilde{S} \mid \mathrm{lcm}\big(\ell\mathrm{t}(f_1),\ell\mathrm{t}(f_2)\big) = \ell\mathrm{t}(f_1)\cdot\ell\mathrm{t}(f_2)\Big\}$
$\tilde{S} := \Big\{(f_1,f_2)\in\tilde{S} \mid \mathrm{lcm}\big(\ell\mathrm{t}(f_1),\ell\mathrm{t}(f_2)\big) \neq \ell\mathrm{t}(f_1)\cdot\ell\mathrm{t}(f_2)\Big\}$
$\mathrm{D}one := \emptyset$
while $\tilde{S} \neq Done$ **do**
  Choose $(f_1,f_2)\in\tilde{S}\setminus\mathrm{D}one$ with the least $\max\big(|\ell\mathrm{c}(f_1)|,|\ell\mathrm{c}(f_2)|\big)$.
  Apply Buchberger's second criterion.
  if $\exists (f_1,h),(h,f_2)\in S\cup\big(\tilde{S}\setminus\{(f_1,f_2)\}\big)\cup S_{old}$ such that $\ell\mathrm{t}(h) \mid \mathrm{lcm}\big(\ell\mathrm{t}(f_1),\ell\mathrm{t}(f_2)\big)$ **then**
    $\tilde{S} := \tilde{S}\setminus\{(f_1,f_2)\}$
  else
    $\mathrm{D}one := \mathrm{D}one\cup\{(f_1,f_2)\}$
  end if
end while
$S_{old} := S_{old}\cup\tilde{S}$

After the elimination of redundant S-pairs the remaining elements of $\tilde{S}$ are partially evaluated to the "shifts" which are delivered for reduction

$$\tilde{S} \;\longrightarrow\; \left\{ \frac{\mathrm{lcm}\big(\ell\mathrm{t}(h_1), \ell\mathrm{t}(h_2)\big)}{\ell\mathrm{t}(h_1)} \cdot h_1, \frac{\mathrm{lcm}\big(\ell\mathrm{t}(h_1), \ell\mathrm{t}(h_2)\big)}{\ell\mathrm{t}(h_2)} \cdot h_2 \;\middle|\; (h_1, h_2) \in \tilde{S} \right\}.$$

The selection of the block of e-shifts is done in a similar fashion. Recall that each e-shift $e^d \cdot f \neq 0$ with $d = \mathcal{B} - \deg_e\big(\ell\mathrm{t}(f)\big) < \mathcal{B}$ is represented in the pending set E of e-shifts as a copy (f) of the source polynomial f from the current basis. To each such a copy (f) assign the e-degree and the degree in main variables as

$$\deg_e\big((f)\big) := \deg_e\big(\ell\mathrm{t}(f)\big) \;\text{ and }\; \deg_X\big((f)\big) := \deg_X\big(\mathrm{T}(\ell\mathrm{t}(f))\big).$$

Let E be the pending set of e-shifts at the moment when a block of e-shifts is requested for reduction. Set

$$a = \max_{(f)\in E} \Big(\deg_e\big((f)\big)\Big) \;\text{ and }\; b = \min_{\substack{(f)\in E \\ \deg_e((f))=a}} \Big(\deg_X\big((f)\big)\Big).$$

Then the block for reduction is defined as

$$\tilde{E} = \Big\{ (f) \;\Big|\; \deg_e\big((f)\big) = a,\; \deg_X\big((f)\big) = b \Big\}.$$

It is clear by construction that any e-shift in $\tilde{E}$ has a higher priority than any in $E \backslash \tilde{E}$. We set $E := E \backslash \tilde{E}$ and eliminate redundant elements of $\tilde{E}$ according to Buchberger's second criterion for e-shifts, the remaining elements of $\tilde{E}$ are moved to the global within the run of the algorithm set of *processed* e-shifts E_{old} which is used in the applicability check of Buchberger's second criterion for e-shifts. The following layout summarizes the procedure.

Require: $\tilde{E}$ - a block of e-shifts.
Ensure: The redundant e-shifts are eliminated in $\tilde{E}$.

  D*one* := $\emptyset$
  while $\tilde{E} \neq Done$ **do**
    Choose $(f) \in \tilde{E} \backslash \mathrm{D}one$ with the least $|\ell\mathrm{c}(f)|$.
    Apply Buchberger's second criterion for e-shifts.
    if $\exists (g) \in E \cup \big(\tilde{E} \backslash \{(f)\}\big) \cup E_{old} \bigwedge \exists (f, g) \in S \cup S_{old}$ such that $\mathrm{T}(g) \,|\, \mathrm{T}(f)$ **then**
      $\tilde{E} := \tilde{E} \backslash \{(f)\}$
    else
      D*one* := D*one* $\cup \{(f)\}$

end if

end while

$E_{old} := E_{\text{old}} \cup \tilde{E}$

The remaining elements of $\tilde{E}$ are evaluated to the "true" e-shifts

$$\tilde{E} \;\longrightarrow\; \left\{ e^{\mathcal{B}-\deg_e\left(\ell t(f)\right)} \cdot f \;\middle|\; (f) \in \tilde{E} \right\} \setminus \{0\}$$

which are submitted for reduction.

As one could notice the author has paid a lot of attention and effort in the maintenance of the critical sets towards detection and elimination of critical elements redundant for reduction Indeed, the elimination of redundant critical elements is the main mechanism in our implementation to avoid the reduction of (at least certain) critical elements which we would prefer not to reduce due to the reasons of numerical stability and computation time. Except for the numerical aspects (described above in the discussion about the underlying ideas of our selection strategy for critical elements) our algorithm basically follows the usual patterns of virtually any implementation of the classical Buchberger algorithm.

Let us close the section with a closer look at these patterns. Clearly, the performance problem arises when a large number of reduced forms of critical elements has to be computed in the flow of the algorithm. This obvious statement is aggravated by the fact that the intermediate and even the resulting basis polynomials tend to become "complicated" (e.g. have a lot of terms in the support); this may drastically slow down the reduction process. This has been observed in real life computing and fairly recently also been confirmed by theoretical bounds on the complexity of Buchberger algorithm computational and intermediate polynomials ([Dub90], [KM96], [MM82], [Giu84], [MM84]).

The importance of mechanisms for detecting critical elements superfluous for reduction becomes more drastic with respect to performance due to some particularities of how the Gröbner basis algorithm works:

- Each non-zero reduced form of a critical element is eventually used to form new critical elements.

- Unless we have already attained a Gröbner basis, the result of reduction depends on the implemented reduction strategy. In particular, because of particularities of the implemented reduction strategy, a non-zero reduced form of a zero-reducible by the current basis critical element can be attained. According to the previous note, every such potentially superfluous but not reduced to zero critical element will eventually give rise to numerous new and also superfluous critical elements.

- A Gröbner basis computation tends to generate a lot of zero-reducible critical elements as the computation gets close to its end. In fact, a Gröbner basis is being attained at some point of the computation but the algorithm does not readily recognize it unless all possible critical elements are treated. Since the treatment of critical elements always amounts to finding their reduced forms, it is clear that a lot of computational efforts can be performed without any gain.

On the other hand, keeping the non-zero normal forms of certain superfluous S-polynomials and using them in reduction can sometimes improve the speed of computation for particular examples. Although there are no considerable theoretical facts towards the strategy of keeping such S-polynomials, the implementation of Buchberger algorithm and its extensions in some computer algebra systems provide an option to control a heuristical mechanism to keep superfluous but potentially useful in reduction elements. A good example is an implementation with an option *morePairs* in the computer algebra system "Singular" [GGM$^+$96, Remark 2.3, p. 13]. Our current implementation does not incorporate any mechanisms to recognize and keep such potentially useful critical elements but this might change in not so distant future.

3.4 Reduction

The reduction algorithm utilized in our implementation (except for the final total auto-reduction of the result) is an elaborated version of the reduction in the original F_4 algorithm. It inherits the original ideas of F_4 concerning

- the subdivision of the reduction process into the so-called symbolic reduction and the linear elimination to the row echelon form and

- the simultaneous reduction of a block of polynomials (which is typically a block of critical elements within a Gröbner basis computation).

Although the original F_4 has been developed for computing Gröbner bases with exact coefficients the ideas behind the algorithm appear attractive for numerical computing:

- The linear elimination (following the symbolic reduction of a block of polynomials) to the row echelon form is the only part of the algorithm where operations with coefficients of polynomials are performed;

therefore all numerical operations required in a numerical version of the algorithm can be clearly segregated and combined into a separate module which is purely a linear algebra routine.

- The simultaneous reduction of a block of polynomials provides an overview of the reduction process – what terms are reducible and how they can be reduced – which provides an extra freedom in the reduction. Even the crude version of a numerical algorithm translated verbatim from the original "exact" F_4 can immediately benefit from the numerical stability achieved by the use of partial pivoting where a polynomial to eliminate a term s is selected among the candidates with the leading term s according to the leading coefficient with maximal absolute value; a bit less transparent development of the same idea leads to a kind of the complete pivoting (with certain restrictions) by a *local override of the term order* during reduction as it is further elaborated in our approach and utilized in the present algorithm.

Before we discuss the details of the reduction procedure of our algorithm, we briefly outline the idea of reduction in the original F_4. Further details concerning the F_4 algorithm as well as the results of computational experiments can be found in [Fau].

Consider a polynomial algebra $\mathcal{P}{=}\Bbbk[X]$ equipped with an admissible term order $<$. We stick to a total reduction: given a polynomial $p{\in}\mathcal{P}$ and a (finite) set $F{\subseteq}\mathcal{P}\backslash\{0\}$ the reduction of p to a normal form modulo F is performed by the successive elimination of terms in the support of p which are reducible modulo F

while $\exists s{\in}\operatorname{supp}(p)$ such that $\ell\mathrm{t}(r)\,|\,s$ with $r{\in}F$ **do**
  take $s{\in}\operatorname{supp}(p)$ and $r{\in}F$ such that $\ell\mathrm{t}(r)\,|\,s$
  $p{:=}p{-}\frac{\mathfrak{M}_p(s)}{\ell\mathrm{m}(r)}{\cdot}r$ { where $\mathfrak{M}_p(s)$ is the monomial in p with the term s}
end while

until all terms in p become *minimal* modulo F i.e. $\ell\mathrm{t}(f)$ does not divide s for any $s{\in}\operatorname{supp}(p)$ and any $f{\in}F$. The termination of the reduction process is provided by the well-ordering property of $<$ since each elimination of a reducible term in $\operatorname{supp}(p)$ removes the term from p by replacing it with a linear combination of strictly smaller terms [4]. Each elementary elimination

[4]Note however that the precise formulation and argumentation of the termination statement is quite tricky unless we stick to a particular order in the elimination of reducible terms like prescribing the elimination of the maximal w.r.t. the term order $<$ reducible term in $\operatorname{supp}(p)$ as it is usually done in the textbooks on Gröbner bases theory to simplify the argumentation and verification of the reduction process. In general, without any a priory assumption on the order of elimination (as it is also stipulated by the real

in the reduction process is usually called a one-step reduction (cf. definition 3.0.6 on page 203): assume $s \in \operatorname{supp}(p)$ is reducible modulo F i.e. $\ell\mathrm{t}(r) \mid s$ for some $r \in F$, then p *reduces* in one step to $p := p - \frac{\mathfrak{M}_p(s)}{\ell\mathrm{m}(r)} \cdot r$ modulo F by eliminating s. The latter one step reduction can be rearranged as

$$p := p - \frac{\mathfrak{C}_p(s)}{\ell\mathrm{c}(r)} \cdot \left(\frac{s}{\ell\mathrm{t}(r)} \cdot r \right) \quad \text{where } \mathfrak{C}_p(s) \text{ is the coefficient at } s \text{ in } p$$

and subdivided into the composition of

- *the essentially polynomial operation:* computation of the shift

  $$h = \frac{s}{\ell\mathrm{t}(r)} \cdot r$$

  of r with the term $\frac{s}{\ell\mathrm{t}(r)}$ and

- *the purely linear algebra operation:* taking the linear combination

  $$p := p - \frac{\mathfrak{C}_p(s)}{\ell\mathrm{c}(h)} \cdot h$$

  with the appropriate "scaling factor" $\frac{\mathfrak{C}_p(s)}{\ell\mathrm{c}(h)}$ of h to eliminate the coefficient at the term s and therefore the term s from p provided that $\ell\mathrm{t}(h) = s$ by construction (due to the compatibility of the leading term function with multiplication of polynomials) and the linear independence of terms.

Now let several terms $\{s_1, \ldots, s_k\} \subseteq \operatorname{supp}(p)$ be reducible modulo F i.e. $\ell\mathrm{t}(r_i) \mid s_i$ with some $r_i \in F$ for $1 \leq i \leq k$. Then several one-step-reductions in the reduction process of p modulo F can be collected together and represented as a slightly overblown version of the partial reduction of p considered as a two-phase process:

- computation of the shifts

  $$h_i = \frac{s_i}{\ell\mathrm{t}(r_i)} \cdot r_i \quad \text{for} \quad 1 \leq i \leq k,$$

life computing where the mentioned simple order of elimination appears too restrictive for computational efficiency and, for the numerical instantiation, in numerical stability respects) the termination statement would require e.g. to consider the lexicographical order $<^{\times}$ induced by the term order $<$ on the set of finite sets of terms considered as the $<$-descending tuples, in this case the termination of the reduction process is provided by the well-ordering property of $<^{\times}$ given by the famous so-called *Cantor's second diagonal argument* (see e.g. [BW93] page 171) and the observation that each elementary elimination results in a strictly $<^{\times}$-smaller $\operatorname{supp}(p)$.

- the subsequent linear elimination by the use of the linear combination

$$p := p - \sum_{i=1}^{k} \alpha_i \cdot h_i$$

with the appropriate coefficients $\alpha_1, \ldots, \alpha_k \in \Bbbk$ such that $\{s_1, \ldots, s_k\}$ are eliminated from the support of the resulting p.

Following the original terminology of Faugere, the computation of shifts is referred as **the symbolic reduction** process. At the present point, it is also convenient to separate out a particular kind of reducibility: given $H \subseteq \mathcal{P} \setminus \{0\}$, a term s such that $\ell\mathrm{t}(h) = s$ for some $h \in H$ is called **linearly reducible** modulo H and h is called the **linear reducer** of s in H. Then the described partial reduction of p modulo F can be seen as a kind of "translation" of the reducibility of reducible terms $\{s_1, \ldots, s_k\} \subseteq \mathrm{supp}\,(p)$ to the linear reducibility modulo the linear reducers $\{h_1, \ldots, h_k\} \subseteq \langle F \rangle_{\mathcal{P}(e)}$ generated in the symbolic reduction process by computing the linear reducers as the appropriate shifts of $\{r_1, \ldots, r_k\} \subseteq F$ (the essentially polynomial operation) followed by the purely linear elimination of the terms from $\mathrm{supp}\,(p)$ by linear combinations with $\{h_1, \ldots, h_k\}$; this linear elimination is likely to be realized as a kind of Gaussian elimination in the $\Bbbk$-linear span of the terms

$$\mathrm{supp}\,(\{h_1, \ldots, h_k\}) \bigcup \mathrm{supp}\,(p)$$

sorted in the $<$-descending order instead of the explicit construction of the linear combination of linear reducers.

It is clear by construction that any reducible term s_i is the leading term of one and only one $h_i \in H$ but it can also appear in the support of another linear reducer $h_j \in H \setminus \{h_i\}$. In this case the linear elimination can also perform a simplification of h_j by eliminating s_i from it. Such a simplification is nothing but a useless extra work if the elimination of reducible terms is performed from the higher to lower w.r.t. the term order reducible terms (as it is likely to be done using the usual version of the reduction). But it appears useful if the elimination is performed from the lower to the higher w.r.t. the term order reducible terms; this is also done in our improved version of the "F_4-style" reduction due to the outstanding performance benefits provided by this order of elimination vs. the usual reduction process shown in practice. Note that the result p of the former linear elimination is not necessary totally reduced (or even top-reduced) modulo F due to the possibility of the new (not present in the support of p before the partial reduction has been applied) terms introduced in the linear reducers and propagated to the support of the resulting p while the reducibility of the new terms modulo F has not been checked so far. The complete analogue of the classical (total) reduction process is attained by an iteration of the symbolic reduction and the linear elimination until no reducible terms are left in the support of p.

It is straightforward to generalize the attained rearrangement of the total reduction of a single polynomial p into the simultaneous reduction of a set of polynomials $P \subseteq \mathcal{P}$ modulo F. The symbolic reduction turns into the procedure

$RT := \emptyset$ { the set of reducible modulo F terms in $\operatorname{supp}(P)$}
$H := \emptyset$ { the set of linear reducers of the terms in RT}
for $s \in \operatorname{supp}(P) \left(= \bigcup_{p \in P} \operatorname{supp}(p)\right)$ **do**
  if s is reducible modulo F i.e. $\ell\mathrm{t}(r) \mid s$ with some $r \in F$ **then**
    $RT := RT \cup \{s\}$
    $h := \frac{s}{\ell\mathrm{t}(r)} \cdot r$ { compute the linear reducer of s}
    $H := H \cup \{h\}$
  end if
end for

which constructs the set of reducible terms $RT \subseteq \operatorname{supp}(P)$ and the corresponding linear reducers $H = \{h_s\}_{s \in RT} \subseteq \mathcal{P} \setminus \{0\}$ with pairwise different leading terms such that $\ell\mathrm{t}(H) = RT$.

The output of the symbolic reduction process is then used for the linear elimination of the terms RT from P modulo H

for $s \in RT$ **do**
  take $h \in H$ with $\ell\mathrm{t}(h) = s$
  for $p \in P \cup \big(H \setminus \{h\}\big)$ such that $s \in \operatorname{supp}(p)$ **do**
    $p := p - \frac{\mathfrak{C}_p(s)}{\ell\mathrm{c}(h)} \cdot h$ {eliminate s from p; $\mathfrak{C}_p(s)$ denotes the coefficient at the term s in p}
  end for
end for

which can be seen as a linear elimination in the $\Bbbk$-linear span of the terms $\operatorname{supp}(P) \cup \operatorname{supp}(H)$ sorted in the $<$-descending order of the upper triangular matrix corresponding to H to the row echelon form and the elimination of RT from P.

Analogous to the reduction of a single polynomial, the iteration of the symbolic reduction and the row echelon elimination, until no reducible modulo F term remains in $\operatorname{supp}(P)$, transforms P into the block of normal forms of the original polynomials P modulo F. Indeed the linear elimination of P modulo H can be naturally realized as a customary row echelon elimination (usually called Jordan-Gauss elimination) for $P \cup H$ in the $\Bbbk$-vector space $\langle P \cup H \rangle_{\Bbbk}$ with a $\Bbbk$-basis given by $\operatorname{supp}(P \cup H)$ again in the $<$-descending order. This approach has the following advantages:

- The absence of artificial restrictions on the flow of elimination which is especially important in the view of numerical instantiation of the method; e.g. it allows the elimination of a term by the subtraction of a polynomial $p \in P$ from a polynomial $h \in H$. This is a preferable way of elimination with numerical (and even exact rational) coefficients if p has the leading coefficient of a considerably greater absolute value.

- The possibility to perform an auto-reduction of P, our development heavily utilizes this possibility by an appropriate modification of the symbolic reduction process.

- The possibility to use an external linear algebra solver for the linear elimination.

However such a "combined" row echelon elimination of $P \cup H$ raises the problem how to separate the reduced forms of the original P from the linearly reduced linear reducers in the result Q of the row echelon elimination since the information about the origin of the reduced form has generally disappeared when the polynomials P and H are "merged" for the combined row echelon elimination. The original F_4 algorithm overcomes this issue by the observation (see [Fau] theorem 2.1 and corollary 2.1) that the (also linearly auto-reduced) reduced forms of the elements of P are precisely the polynomials $q \in Q$ such that $\mathrm{supp}\,(q) \cap RT = \emptyset$. This is equivalent to the easy to check in the flow of computation condition $\ell\mathrm{t}(q) \notin RT$ (straightforward by $RT = \ell\mathrm{t}(H)$, Q is in the row echelon form and the uniqueness, up to scaling factors, of the row echelon form).

Our algorithm improves the original F_4 reduction process in a number of important ways. First of all note that the transition to the computation of a Gröbner basis in the algebra $\mathcal{P}(e)$ of extended polynomials does not introduce any particularities into the "F_4-style" reduction process:

- The extended term order is a well-ordering and therefore the reduction termination argument is easily transferred to the new setup.

- The translation of the reducibility of reducible terms in a block of polynomials to linear reducibility by computing the appropriate polynomial shifts, one for each reducible term, performed at the stage of symbolic reduction, does not require any change as long as the leading term function is stable i.e. the leading term of a shift is the shift of the leading term of a polynomial; but, it is easy to see from part a)

of corollary 2.3.6, that this is the case for any shift computed during symbolic reduction.

- The row echelon elimination can also be applied as is due to the linear independence of extended terms $\mathbb{T}(e)$.

However the numerical computation of Gröbner bases demands a careful analysis of the results of a computation for potential numerical problems and a subsequent research for the possible algorithmic alternatives to avoid or at least to reduce the hazardous effect of such problems; this has led to a considerable deviation from the original F_4 reduction process. The important changes can be categorized as follows.

- The simultaneous reduction of a block of polynomials includes the auto-reduction of the block implemented as a modification of the symbolic reduction. Since the block of polynomials typically consists of the critical elements to be reduced whose non-zero reduced forms are added to the current basis in the flow of a Gröbner basis computation, such an approach presents a compromise between the complete auto-reduction of the intermediate basis modulo each newly introduced reduced form and the widely used version of the Gröbner bases algorithm where the auto-reduction is performed at the very end of the computation when a Gröbner basis has already been attained. The first approach appears not feasible for numerical computation of Gröbner bases due to the numerical instability since it insists on every possible reduction to be done immediately while it appears advantageous to keep the already introduced in the current basis polynomials as they are until the final auto-reduction of the attained Gröbner basis.

  For example, the intermediate basis as well as the final Gröbner basis of a a zero-dimensional system (which is our main concern) is typically over-determined and therefore the ideal of the system tends to degenerate to $\langle 1\rangle_{\mathcal{P}(e)}=\mathcal{P}(e)$ under an arbitrary small perturbation of coefficients; this happens in the numerical computation due to the (accumulated) roundoff errors and/or the inaccuracy in the coefficients of the initial system. Assume that at some point of computation the intermediate Gröbner basis has a correct structure of leading terms (i.e. the leading terms are in the ideal of leading terms structurally equivalent to the ideal of leading terms of the initial system). The subsequent auto-reduction may create, through a near-zero reduction of a polynomial that should indeed be zero, a polynomial with a leading term which is "too close" to the origin, or, as an extreme case, just a constant such that the leading term should actually be in the normal set. Clearly the rest of the computation has no chance to arrive at a correct Gröbner basis: in the best case, we attain (after the back translation of the result from $\mathcal{P}(e)$ to $\mathcal{P}$) the Gröbner basis of a system with a strictly smaller, counting multiplicities, number of solutions (equivalent to the number of terms in the associated normal set) or, as it usually happens according to our experiments, the computation ends up with an inconsistent system $\{1\}$. On the other hand, the auto-reduction of the resulting Gröbner basis $G \subseteq \mathcal{P}(e)$ as it is performed in our algorithm can not ruin the (assuming correct) structure of the leading terms being performed in a well-known from the classical theory of Gröbner bases fashion and transferred verbatim to our setup (see page 209).

- The symbolic reduction incorporates a kind of the recursive search-in-depth for reducible terms approach by looking for new reducible terms in the already created linear reducers.

- The row echelon elimination is performed with complete pivoting among reducible terms recognized at the stage of the symbolic reduction. The numerical advantage of the complete vs. the partial (readily available in the original F_4 approach) pivoting is well-known: a reasonably slow propagation of the roundoff error. Note that most numerical implementations of whatever variant of direct linear elimination stick to the partial pivoting because it is easier to implement and at the same time the complete pivoting does not show a particular advantage against it's partial counterpart in most real-life applications despite much better theoretical estimates. Yet the linear elimination in the frame of a Gröbner basis computation appears to be an application where the complete pivoting provides an essential advantage.

- The trimming of e-degrees during linear elimination is done as described at the end of section 3.4.2. This is the only mechanism in the whole implementation of the algorithm where the demands of numerical stability require an essential deviation from the path of computation prescribed by the theory of Gröbner bases in the algebra $\mathcal{P}(e)$ of extended polynomials.

Simplifying only slightly, the shallow layout of the reduction process can be described as follows. Let $G \subseteq \mathcal{P}(e) \setminus \{0\}$ be the current basis attained at the point when the reduction process has been called to reduce a block of polynomials $B \subseteq \mathcal{P}(e) \setminus \{0\}$, for the initial auto-reduction of the input polynomials performed before the execution of the main loop of the Gröbner basis algorithm assume $G = \emptyset$.

Require: $B \subseteq \mathcal{P}(e) \setminus \{0\}$ – the input polynomials.
Ensure: R – the input polynomials B totally auto-reduced and totally reduced modulo G.

$R := B \setminus \{0\}$
$Simp := \emptyset$
repeat
$(R, RT) :=$ the result of symbolic reduction of R modulo itself and modulo $G \cup Simp$ $\{RT \subseteq R$ is the set of reducible terms in $R\}$
$R :=$ the result of the row echelon linear elimination of R with 0 polynomials removed
$Simp := Simp \cup \{$polynomials in R whose leading terms are in $RT\}$
$R :=$ polynomials in R that do not have any terms RT in the support
until nothing to reduce in R

The only concept in the layout that has not been mentioned so far is a computational trick based on the incremental construction of the set of simplifiers *Simp* during the reduction process. It is indeed a purely computational trick:

- To reuse the results of linear elimination by keeping the polynomials, which have been subjected to linear elimination at the previous iteration of the loop and which have kept reducible leading terms, instead of throwing them away which would indeed have been just a waste of the results of the row echelon elimination.
- To improve the numerical stability of the algorithm by a temporary extension of the current basis G that empirically creates extra alternatives in the search for preferable (in the sense of the implemented reducer selection strategy that surely takes into account the "estimated" potential numerical stability consequence of a selection) reducers at the subsequent calls to the symbolic reduction.

The current version of the algorithm treats the simplifiers exactly as it is presented in the layout: the set of simplifiers is flushed at each call of the reduction process and then constructed incrementally within the iteration of symbolic reduction and linear elimination. We have also experimented with a global set of simplifiers i.e. the set that is never flushed in the flow of computation; we have found such an option more memory and time consuming without a noticeable gain in the numerical stability respect.

Another interesting alternative to treat the simplifiers could maintain the simplifiers as a global set but use clever rules to decide which polynomials in each result of linear elimination should be added to the simplifiers. This is not done in the algorithm. Indeed something very similar to the latter alternative, but with a completely different utilization of the simplifiers for an improvement of efficiency of the algorithm has been used by J. C. Faugere in the original F_4 algorithm (see section 2.4 in [Fau]). This could also be an option for our numerical Gröbner bases algorithm after a more careful analysis and experimental studies of its numerical effects.

Although a plenty of other considerations including the theoretical analysis of correctness as well as computational benefits and drawbacks can be drawn concerning the iteration in the reduction process and the maintenance of the set of simplifiers, we leave these issues aside as they are not the key mechanisms in our algorithm to maintain computational efficiency and numerical stability. Instead we concentrate on the symbolic reduction and row echelon elimination as these are the crucial, and also considerably different from their original F_4 counterparts, mechanisms in the reduction part of our algorithm.

The next two subsections deal in details with the symbolic reduction and the linear elimination respectively as implemented in our algorithm.

3.4.1 Symbolic reduction

Recall that, according to definition 2.3.12, a non-zero polynomial $f{\in}\mathcal{P}(e)\backslash\{0\}$ **reduces** a non-zero term $s{\in}\mathbb{T}(e)^*$ if $s{=}a{\cdot}\ell\mathrm{t}(f)$ with $a{\in}\mathbb{T}(e)^*$ which is, by the definition of divisibility of terms, equivalent to the condition $\ell\mathrm{t}(f)\,|\,s$. In this case the reduction operation $s \xrightarrow{f}$ of s modulo f is defined as

$$s \xrightarrow{f} s - \frac{a}{\ell\mathrm{c}(f)} f \quad \text{with} \quad a = \frac{s}{\ell\mathrm{t}(f)}.$$

Following the usual terminological convention a term s is called *reducible* modulo $F{\subseteq}\mathcal{P}(e)\backslash\{0\}$ if s is reducible modulo some $f{\in}F$, otherwise s is called *minimal* modulo F (sometimes also called *irreducible* modulo F but we avoid that terminology because it collides with the established use of the word as "not allowing a proper factorization" though we do not deal with any factorizations computing a Gröbner basis). For the sequel it is convenient to slightly extend the "reducibility-related" terminology as follows.

Definition 3.4.1. For any $s{\in}\mathbb{T}(e)^*$ and any $f{\in}\mathcal{P}(e)\backslash\{0\}$ such that $\ell\mathrm{t}(f)\,|\,s$, the polynomial f is called a **reducer** of s. Furthermore, we distinguish two different types of reducibility: the reducer f is called a **linear reducer** of f and s is called **linearly reducible** modulo f if $\ell\mathrm{t}(f){=}s$, otherwise f is called a **proper reducer** and s is called **properly reducible** modulo f.

Given a non-zero term $s{\in}\mathbb{T}(e)^*$ and a set $F{\subseteq}\mathcal{P}(e)\backslash\{0\}$ of non-zero polynomials denote as $R_s(F)$ – the set of all reducers of s in F:

$$R_s(F) = \Big\{ r{\in}F \;\Big|\; \ell\mathrm{t}(r)\,|\,s \Big\}.$$

A term s is called **properly reducible** modulo F if F contains a proper reducer of s, if F contains a linear reducer of s then s is called **linearly reducible** modulo F.

The meaning of the linear and proper reducibility of a term s modulo a reducer f becomes transparent when the reduction of s modulo f is considered as a composition of two operations: a shift

$$r = \frac{s}{\ell\mathrm{t}(f)} \cdot f$$

and an appropriate linear combination of f and r like

$$s - \frac{1}{\ell\mathrm{c}(r)} \cdot r.$$

If f is a linear reducer of s then the computation of the shift r is indeed superfluous – the reduction specializes to the computation of a linear combination which is purely a linear algebra operation. Also note that the computation of r does not change the coefficients originating from f, on the other hand the formation of the linear combination deals only with coefficients. The latter subdivision appears useful in the numerical computation of Gröbner bases since it separates the numerical operations with generally inexact coefficients from the exact algebraic operations with terms; it allows the implementation of a Gröbner basis algorithm to collect the numerical operations into a single separate module which is indeed a linear algebra routine.

The transition of the idea to the reduction of a polynomial modulo a set of polynomials is immediate. One of the possible realizations of this idea is implemented in Faugere's F_4 algorithm; it utilizes a slightly more far-reaching generalization by reducing a set of polynomials modulo a set of polynomials that is also replicated in a slightly more complex form in our approach. Following the original F_4 algorithm the reduction process in $\mathcal{P}(e)$ can be formulated in the following way. Given a set of polynomials $H \subseteq \mathcal{P}(e) \setminus \{0\}$ to be reduced modulo $F \subseteq \mathcal{P}(e) \setminus \{0\}$.

- The computation of polynomial shifts turns into a procedure called the **symbolic reduction** which constructs a set $R \subset \langle F \rangle_{\mathcal{P}(e)}$ of linear reducers of the terms in $\mathrm{supp}\,(H)$ by traversing through all $s \in \mathrm{supp}\,(H)$: If s is reducible modulo F then the set R is extended with a corresponding linear reducer of s attained as $\frac{s}{\ell\mathrm{t}(f)} f$ with $f \in F$ – a reducer (either proper or linear) of s. The reduction of H modulo F can now be performed as a linear elimination of terms in $Q := H \cup R$.

To make the analogy with linear algebra more transparent, the set Q can be considered as a matrix M_Q with columns corresponding one-to-one to the terms in $\operatorname{supp}(Q)$ written as a descending in the term order tuple and rows corresponding to the polynomials in Q as $\Bbbk$-linear combinations in $\operatorname{supp}(Q)$ under the natural isomorphism of $\Bbbk$-linear spaces $\langle \operatorname{supp}(Q)\rangle_{\Bbbk} \cong \Bbbk^d$ with $d = |\operatorname{supp}(Q)|$.

- The computation of linear combinations turns into the **row echelon elimination** in this matrix M_Q attained at the previous step. The reduced forms of the elements of H are read off the rows of the row echelon form of M_Q as linear combinations in $\operatorname{supp}(Q)$ which do not contain any term $s \in \operatorname{supp}(P)$ with a non-zero coefficient reducible modulo F.

The output set of reduced forms of P can be seen as a basis of the intersection of the $\Bbbk$-linear vector space $\langle Q\rangle_{\Bbbk}$ and the $\Bbbk$-vector span $\langle \operatorname{supp}(Q) \setminus RT\rangle_{\Bbbk}$ where $RT \subseteq \operatorname{supp}(P)$ are terms reducible modulo F.

Note that the symbolic reduction generally introduces new terms in the output set Q which were not present in the initial set P; they are not checked for reducibility in the outlined crude version of the symbolic reduction. Hence the final set of reduced forms of P is not necessary totally reduced modulo F after the row echelon elimination.

Given a reducible term, it is customary to have several polynomials whose leading terms divide the term. In this case the algorithm has to make a decision which element in the set should be used to reduce the term. Most implementations of the Gröbner basis algorithm would just take the first found element $f \in F$ such that $f \in R_s(F)$ attained by the linear search in a structure used for the representation of F which is typically implemented as a sequential container of polynomials (a list, an array, etc) where the elements are stored in the chronological order as they appeared in the flow of computation. Such a *"pick the first"* strategy appears to be unacceptable for numerical computation of Gröbner bases: According to our experiments and experience, this virtually random choice of a reducer almost inevitably leads to a structurally wrong final basis – typically the inconsistent system $\{1\}$. Therefore our implementation uses a pretty sophisticated (a nice synonym for complicated) selection strategy for a reducer by inducing a priority structure on the set of reducers of a term much like it has been done for the critical elements selection strategy as described in section 3.3:

Let p be a polynomial to be reduced modulo F and $s \in \operatorname{supp}(p)$ a reducible term. The priority among the elements of $R_s(F)$ to be used as a reducer of s in p is defined as follows. Let $r_1, r_2 \in R_s(F)$ be two different possible reducers of s. First of all r_1 is unconditionally a better reducer if

- r_1 is a monomial and r_2 is not (in this case s is simply replaced by 0 in the reduction modulo r_1); or
- $\deg_e\bigl(\ell t(r_1)\bigr) < \deg_e\bigl(\ell t(r_2)\bigr)$ – the leading term of r_1 has a smaller e-degree (in this case the leading coefficient of r_1 is more reliable numerically according to our interpretation of the meaning of the e-degree of a term within a polynomial in the flow of computation).

Otherwise r_1 is better than r_2 if the leading coefficient of r_1 has a greater absolute value $|\ell c(r_1)| > |\ell c(r_2)|$ or $|\ell c(r_1)| = |\ell c(r_2)|$ and r_1 has a lower *heuristical coefficient dispersion*

$$\mathrm{hcd}(r_1) = \left|\frac{\mathrm{pc}(r_1)}{\ell c(r_1)}\right| < \left|\frac{\mathrm{pc}(r_2)}{\ell c(r_2)}\right| = \mathrm{hcd}(r_2)$$

where $\mathrm{pc}(h)$ denotes the coefficient in h at the term in the support of h *preceding* the leading term in the term order $<$ or formally set to zero if h is a monomial:

$$\mathrm{pc}(h) = \begin{cases} \ell c\bigl(h - \ell m(h)\bigr) & \text{if } h - \ell m(h) \neq 0, \\ 0 & \text{otherwise} \end{cases}$$

for any non-zero polynomial h.

The idea behind the choice of a reducer with a higher absolute value of the leading coefficient is quite transparent: it is the direct counterpart of pivoting in the Gaussian elimination process to slow down the propagation of the roundoff error during the subsequent row echelon elimination and to avoid the reduction modulo a polynomial from the current basis whose leading monomial may be zero except for rounding errors; on the other hand the choice of a reducer with a lower heuristical coefficient dispersion limits the dispersion in the order of magnitude of the coefficients within a polynomial during row echelon elimination.

If the priority between r_1 and r_2 is still indistinguishable according to the rules above and if the utilized term order $<$ is $\underset{\mathbb{T}_e}{<}$ (as defined on page 90) then r_1 is better if it has a smaller leading e-difference (definition 2.4.2 on page 95) $\ell\Delta_\varepsilon(r_1) < \ell\Delta_\varepsilon(r_2)$: according to theorem 2.4.1 on page 102 and the subsequent remark, the reduction w.r.t. $\underset{\mathbb{T}_e}{<}$ by an element with the minimal leading e-difference may sometimes require fewer steps.

As a last refinement of the priority, the reducer with a smaller degree of the main part of the leading term or a smaller sugar degree or a smaller number of elements in the support is better: r_1 is better than r_2 if, in the componentwise difference of tuples,

$$\Big(\deg\big(\mathrm{T}(\ell\mathrm{t}(r_2))\big), \mathrm{sdeg}(r_2), |\operatorname{supp}(r_2)|\Big) - \Big(\deg\big(\mathrm{T}(\ell\mathrm{t}(r_1))\big), \mathrm{sdeg}(r_1), |\operatorname{supp}(r_1)|\Big)$$

the left-most non-zero entry is positive.

The described priority check process for a pair of polynomials is summarized in the following procedure.

Require: s – a term, $r_1, r_2 \in R_s(F)$.
Ensure: **TRUE** if r_1 is a better reducer of s than r_2; **FALSE** otherwise.

```
if |supp(r1)|=1 and |supp(r2)|>1 then
  return TRUE {A monomial is a better reducer.}
end if
if |supp(r2)| = 1 and |supp(r1)|>1 then
  return FALSE
end if
if deg_e(ℓt(r1))<deg_e(ℓt(r2)) then
  return TRUE {The reducer with a smaller e-degree of the leading
  term is better.}
end if
if deg_e(ℓt(r2))<deg_e(ℓt(r1)) then
  return FALSE
end if
if |ℓc(r2)|<|ℓc(r1)| then
  return TRUE {The reducer with a greater absolute value of the lead-
  ing coefficient is better.}
end if
if |ℓc(r1)|<|ℓc(r2)| then
  return FALSE
end if
if hcd(r1)<hcd(r2) then
  return TRUE {The reducer with a lower heuristical coefficients dis-
  persion is better.}
end if
if hcd(r2)<hcd(r1) then
  return FALSE
end if
if the order < is <_{T_e} then
  if ℓΔ_ε(r1)<ℓΔ_ε(r2) then
```

return TRUE {If the order is $\underset{\mathbb{T}_e}{<}$ then the reducer with a smaller leading e-difference is preferable.}
end if
if $\ell\Delta_\varepsilon(r_2) < \ell\Delta_\varepsilon(r_1)$ **then**
return FALSE
end if
end if
if $\deg\big(\mathrm{T}(\ell\mathrm{t}(r_1))\big) < \deg\big(\mathrm{T}(\ell\mathrm{t}(r_2))\big)$ **then**
return TRUE {The reducer with a smaller degree of the main part of the leading term is better.}
end if
if $\deg\big(\mathrm{T}(\ell\mathrm{t}(r_2))\big) < \deg\big(\mathrm{T}(\ell\mathrm{t}(r_1))\big)$ **then**
return FALSE
end if
if $\mathrm{sdeg}(r_1) < \mathrm{sdeg}(r_2)$ **then**
return TRUE {The reducer with a smaller sugar degree is better.}
end if
if $\mathrm{sdeg}\big(\mathrm{T}(\ell\mathrm{t}(r_2))\big) < \mathrm{sdeg}\big(\mathrm{T}(\ell\mathrm{t}(r_1))\big)$ **then**
return FALSE
end if
if $|\,\mathrm{supp}\,(r_1)\,| < |\,\mathrm{supp}\,(r_2)\,|$ **then**
return TRUE {The reducer with a less number of terms in the support is better.}
end if
return FALSE

Note that the priority on reducers defined above is not a linear order in the sense that the priority between two different reducers can happen to be indiscernible. Our implementation resolves this issue by a quite random refinement of the priority by taking the chronologically first found best reducer. The selection of the best reducer is naturally generalized from the selection between a pair of polynomials to the selection among an arbitrary non-empty set as a kind of a maximal element function.

Require: s - the term to be reduced, R - a non-empty set of polynomials whose leading terms divide s.
Ensure: r - is the best reducer of s among the elements of R
$r :=$ any polynomial from R
$R := R \backslash \{r\}$
while $R \neq \emptyset$ **do**
$r' :=$ any polynomial from R
if r' is better than r **then**

$r := r'$

end if

$R := R\backslash\{r'\}$

end while

Let $P{\subseteq}\mathcal{P}(e)\backslash\{0\}$ be the set of polynomials to be symbolically reduced modulo $F{\subseteq}\mathcal{P}(e)\backslash\{0\}$. The symbolic reduction is, in a sense, a kind of the reducibility type translation process: for any term in $s{\in}\operatorname{supp}(P)$ which is reducible modulo F or properly reducible modulo P, the symbolic reduction creates a so-called *linear reducer* – a polynomial $h{=}\frac{s}{\ell\mathrm{t}(r)}{\cdot}r$ with $r{\in}\langle P{\cup}F\rangle_{\mathcal{P}(e)}$ such that $\ell\mathrm{t}(r)\,|\,s$ – which is added to the original set resulting in the change of the reducibility type of of the term: s becomes linearly reducible modulo $P{:=}P{\cup}\{h\}$. Indeed, $0{\neq}s{=}\frac{s}{\ell\mathrm{t}(r)}{\cdot}\ell\mathrm{t}(r)$ and hence, by part a) of corollary 2.3.6, $\ell\mathrm{t}(h){=}\frac{s}{\ell\mathrm{t}(r)}{\cdot}\ell\mathrm{t}(r){=}s$.

The following layout presents one of the possible ways to build up the symbolic reduction process which takes an arbitrary finite set P of non-zero polynomials in $\mathcal{P}(e)$ and extends P with certain shifts with a term of the elements $P{\cup}F$ in such a way that any term in the support of the initial P which is reducible modulo F or properly reducible modulo the initial P becomes linearly reducible modulo the outcome of the symbolic reduction.

Require: P – the set of non-zero polynomials to be symbolically reduced modulo F and symbolically properly auto-reduced.

Ensure: P is the symbolically reduced, RT is the set of reducible terms in the final set P.

$RT := \emptyset$

$Done := \emptyset$

while $\operatorname{supp}(P) \neq Done$ **do**

$s :=$ any term in $\operatorname{supp}(P)\backslash Done$

$R := R_s\Big(F{\cup}\big(P\backslash\{p{\in}P|\ell\mathrm{t}(p){=}s\}\big)\Big)$

if $R{\neq}\emptyset$ **then**

$r :=$ the best reducer of s in R

$P := P{\cup}\{\frac{s}{\ell\mathrm{t}(r)}{\cdot}r\}$

$RT := RT{\cup}\{s\}$

end if

$Done := Done{\cup}\{s\}$

end while

Despite the simplicity of the symbolic reduction algorithm, the termination and the correctness proof turn to be somewhat a bit tricky. The correctness is defined in a natural way: any term $s{\in}\operatorname{supp}(Q)$ which is properly reducible modulo $Q{\supseteq}P$ or reducible modulo F must admit a linear reduction modulo Q. Indeed, if it is true then the reduction of Q can be performed as the linear elimination of terms in Q and this is exactly what we need from the output of the symbolic reduction algorithm.

Lemma 3.4.1. For any finite sets P and F of non-zero polynomials.

- *Termination:* The symbolic reduction of P modulo F terminates in finitely many steps resulting in a finite set of non-zero polynomials.
- *Correctness:* Let $Q \supseteq P$ be the result of symbolic reduction of P modulo F provided by the previous statement and RT be the set of reducible terms generated in the flow of symbolic reduction. Then for any term $s \in \operatorname{supp}(Q)$: if s is reducible modulo F or properly reducible modulo Q then $s \in RT$ and s is linearly reducible modulo Q.

Proof. Let $\hat{P}=P=\{p_1, \ldots, p_k\}$ be the finite set of non-zero polynomials submitted as the input of symbolic reduction. For each term $s \in \mathbb{T}(e)^*$ selected in the iteration of the while-loop during the symbolic reduction of $\hat{P}$ denote the intermediate data of the iteration when s has been selected by

$P_s \supseteq \hat{P}$	–	the intermediate result of symbolic reduction i.e. the set of polynomials P constructed so far;
$L_s=\{p \in P_s \mid \ell\mathrm{t}(p)=s\}$	–	the set of linear reducers of s in P_s,
$R_s=R_s\big(F \cup (P_s \setminus L_s)\big)$	–	the set of reducers of s,
in the case $R_s \neq \emptyset$:		
$r_s \in R_s,\, h_s=\frac{s}{\ell\mathrm{t}(r_s)} \cdot r_s$	–	the reducer of s selected from R_s and the linear reducer added to P_s.

Note that trivially

$$0 \neq s=\frac{s}{\ell\mathrm{t}(r_s)} \cdot \ell\mathrm{t}(r_s)$$

and hence, by part a) of corollary 2.3.6,

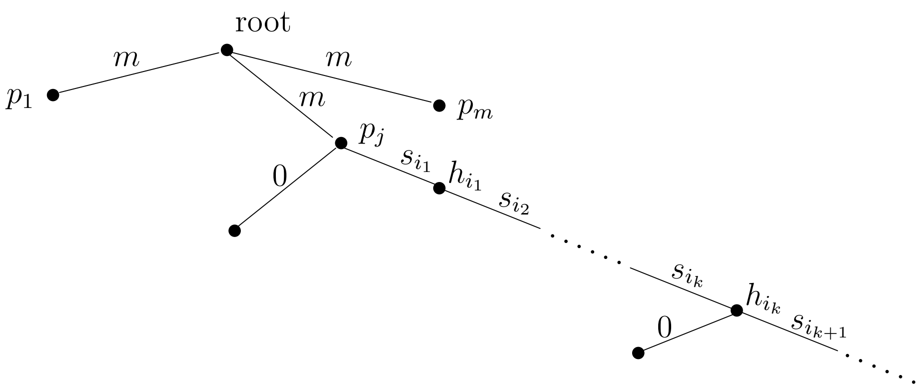
$$h_s \neq 0 \quad \text{and} \quad \ell\mathrm{t}(h_s)=\frac{s}{\ell\mathrm{t}(r_s)} \cdot \ell\mathrm{t}(r_s) \qquad (*)$$

or, in other words, h_s is indeed a linear reducer of s for each $s \in \operatorname{supp}(Q)$ such that $R_s \neq \emptyset$.

Termination: The data flow of the algorithm can be seen as the incremental construction of a tree structure where all edges are marked with terms and certain nodes are marked with polynomials such that before the execution of the while-loop and at each iteration of the while-loop for each polynomial in the intermediate result of symbolic reduction there exists a node in the tree marked with this polynomial. The tree is set up as follows. Choose $m \in \mathbb{T}(e)^*$ – a term such that $m > s$ for any $s \in \mathrm{supp}(\hat{P})$. The existence of such m is clear since $\mathrm{supp}(\hat{P})$ is a finite set. Before the execution of the while-loop, the tree consists of the nodes marked with polynomials from the initial set $\hat{P}$: for each $p \in \hat{P}$ there is one and only one node marked with p, all these nodes are connected to the unmarked root node of the tree by an edge marked with m. Consider an arbitrary iteration of the while-loop. Let $s \in \mathrm{supp}\,(P_s)$ be the term selected in this iteration. Assume that for each $p \in P_s$ there exists a node marked with p. Take any polynomial $q \in P_s$ such that $s \in \mathrm{supp}\,(q) \left(\subseteq \mathrm{supp}\,(P_s) \right)$. Then, by assumption, the tree contains a node marked with q. Consider two possibilities.

- The term s is reducible modulo F or properly reducible modulo P_s. In this case $R_s \neq \emptyset$, the algorithm selects a reducer $r_s \in R_s$ of s and extends the intermediate result P_s by the linear reducer $h_s = \frac{s}{\ell\mathrm{t}(r_s)} \cdot r_s$ (unless already $h_s \in P_s$ which is irrelevant for our construction) of s; the tree is extended by a new node marked with h_s connected to a node marked with q by an edge marked with s.

- The term s is minimal – the set R_s is empty. In this case the tree is extended by a new unmarked node connected to a node marked with q by an edge marked with 0.



Since each iteration of the while-loop results in a new node added to the tree, an infinite number of iterations of the loop must result in a tree with an infinite number of nodes. Then, by König's lemma, the tree must have an infinite path. Choose such a path starting from the root of the tree. Consider the sequence of terms

$$U = (m, s_{i_1}, \ldots, s_{i_k}, s_{i_{k+1}}, \ldots)$$

that mark the edges of the path. Note that U can not contain 0 because, by construction, any 0 marked edge is connected to a leaf. We claim that the sequence U is strictly descending. Indeed, $m{>}s_{i_1}$ by the choice of m since, by construction of the tree, $s_{i_1}{\in}\mathrm{supp}(\hat{P})$. For any $k{>}0$ the edges corresponding to s_{i_k} and $s_{i_{k+1}}$ share a common node marked with h_{i_k}; by $(*)$, $\ell\mathrm{t}(h_{i_k}){=}s_{i_k}$. On the other hand, again by construction of the tree, the edge s_{i_k} and the node h_{i_k} should have been already in the tree at the iteration when $s_{i_{k+1}}$ has been selected and $s_{i_{k+1}}{\in}\,\mathrm{supp}\,(h_{i_k})$, therefore, since no term can be selected twice in the while-loop, $\ell\mathrm{t}(h_{i_k}){=}s_{i_k} \neq s_{i_{k+1}}{\in}\,\mathrm{supp}\,(h_{i_k})$. Then $s_{i_{k+1}}{\in}\,\mathrm{supp}\,(h_{i_k})\setminus\{\ell\mathrm{t}(h_{i_k})\}$ and hence $s_{i_k} > s_{i_{k+1}}$.

The infinite strictly descending sequence U contradicts the well-ordering property of the extended term order.

Correctness: We would need a supplementary statement: for any non-zero term $s{\in}\mathbb{T}(e)^*$

s is reducible modulo F or properly reducible modulo Q $\quad(**)$
if and only if
s is reducible modulo F or properly reducible modulo $\hat{P}$.

Take an arbitrary $s{\in}\mathbb{T}(e)^*$. If s is reducible modulo F then $(**)$ is trivial since F is not changed in the symbolic reduction.
Assume s minimal modulo F. In this case the *"if"* direction is trivial: if s is properly reducible modulo $\hat{P}$ then there exists a proper reducer $h{\in}\hat{P}$ of s which is also a proper reducer of s in Q because clearly $\hat{P}{\subseteq}Q$ by construction and therefore s is properly reducible modulo Q. The *"only if"* direction is slightly more tricky: Let s be properly reducible modulo Q. Consider $h{\in}Q$ – a proper reducer of s: $\ell\mathrm{t}(h)\,|\,s$ and $\ell\mathrm{t}(h){\neq}s$. Assume s is not properly reducible modulo $\hat{P}$. Then surely $h{\in}Q\setminus\hat{P}$, hence h is a reducer added to P somewhere in the flow of symbolic reduction and it is easy to see by $(*)$ and e.g. induction on the depth of the node marked with h in the tree constructed in the termination proof [5] that $h{=}v{\cdot}g$ with $v{\in}\mathbb{T}(e)^*$, $g{\in}F{\cup}\hat{P}$ and $\ell\mathrm{t}(h){=}v{\cdot}\ell\mathrm{t}(g)$. But then $\ell\mathrm{t}(g)\,|\,\ell\mathrm{t}(h)\,|\,s$ implies that g is a proper reducer of s and hence $g{\in}\hat{P}$ because s is minimal modulo F by assumption. Therefore s is properly reducible modulo $\hat{P}$.[6]

[5]It is easy to show that there exists one and only one node marked with h in the constructed tree provided that $h{\notin}\hat{P}$. However this fact is irrelevant for our argumentation.

[6]The meaning of the *"only if"* direction of $(**)$ can be also informally stated as the fact that the reducers $Q\setminus\hat{P}$ added to $\hat{P}$ during symbolic reduction do not result in any new

Now we are ready to prove the correctness statement. It is clear from the layout on page 238 that each term in $s\in\operatorname{supp}(Q)$ has been selected once and only once in the while-loop of symbolic reduction. If the set R_s of reducers of s constructed at the iteration when s has been selected is not empty then, by inspection of the "if" condition in the while-loop body, s has been added to RT and the intermediate result P_s of symbolic reduction has been enlarged with $h_s{=}\frac{s}{\ell t(r_s)}{\cdot}r_s$ with $r_s{\in}R_s$, according to $(*)$, h_s is a linear reducer of s. Therefore, after the symbolic reduction, $s{\in}RT$ and s is linearly reducible modulo Q because no elements are ever removed from RT or P during symbolic reduction. Let $s\in\operatorname{supp}(Q)$ be reducible modulo F or properly reducible modulo Q. We would need the following two easily verifiable properties of the set of divisors of s with respect to any pair $A, B{\subseteq}\mathcal{P}(e)\setminus\{0\}$ of sets of non-zero polynomials:

$$A{\subseteq}B \quad\Longrightarrow\quad R_s(A){\subseteq}R_s(B), \tag{2}$$

$$R_s(A){\cup}R_s(B) = \{p{\in}A|\ell \mathrm{t}(p){=}s\}{\cup}\{p{\in}B|\ell \mathrm{t}(p){=}s\} = \\ \{p{\in}A{\cup}B|\ell \mathrm{t}(p){=}s\} = R_s(A{\cup}B). \tag{3}$$

Let $s\in\operatorname{supp}(Q)$ be reducible modulo F or properly reducible modulo Q. Due to the above observation it is sufficient to show that $R_s{\neq}\emptyset$. By $(**)$ s is reducible modulo F or properly reducible modulo $\hat{P}$. Consider two cases.

- If s is reducible modulo F then $R_s(F){\neq}\emptyset$ and therefore

$$R_s = R_s\big(F{\cup}(P_s{\setminus}L_s)\big) \overset{(3)}{=} \underbrace{R_s(F)}_{\substack{\not\Vert\\ \emptyset}}{\cup}R_s\big(P_s{\setminus}L_s\big) \neq \emptyset.$$

- If s is minimal modulo F and properly reducible modulo $\hat{P}$ then $R_s(F){=}\emptyset$, $R_s(\hat{P}{\setminus}L(s)){\neq}\emptyset$ where $L(s) = \big\{p{\in}\hat{P}|\ell \mathrm{t}(p){=}s\big\}$ – the set of linear reducers of s in $\hat{P}$. Furthermore, since no term can be considered more than once during symbolic reduction (as maintained by the use of the *Done* set) then, by $(*)$, no polynomial with the leading term s could have been added to the intermediate result P_s of symbolic reduction before the iteration when s is selected to be treated. On the

terms properly reducible modulo Q which are not properly reducible modulo $\hat{P}$. Indeed, this is also a "hands-on" reason why it is sufficient to check the reducibility of each term only once in the while-loop.

other hand, clearly $P_s \supseteq \hat{P}$ since no polynomials are ever removed from the intermediate result of symbolic reduction which gives a not crucial but convenient for the further argumentation fact:

$$L(s) = \{p \in \hat{P} | \ell t(p) = s\} = \{p \in P_s | \ell t(p) = s\} = L_s. \tag{4}$$

Hence

$$R_s(P_s \backslash L_s) \stackrel{(4)}{=} R_s(P_s \backslash L(s)) \overset{P_s \supseteq \hat{P},\ (2)}{\supseteq} R_s(\hat{P} \backslash L(s)) \neq \emptyset \tag{5}$$

and therefore

$$R_s = R_s\big(F \cup (P_s \backslash L_s)\big) \stackrel{(3)}{=} \underbrace{R_s(F)}_{\emptyset} \cup R_s\big(P_s \backslash L_s\big) = R_s\big(P_s \backslash L_s\big) \stackrel{(5)}{\neq} \emptyset.$$

In either case $R_s \neq \emptyset$. $\square$

An extra property of the output of the symbolic reduction algorithm is formulated in the following lemma. It is not particularly important for the algorithm itself but convenient for the argumentation in the subsequent discussion of the linear elimination of the symbolically reduced set of polynomials in the next section

Lemma 3.4.2. Let $Q \subseteq \mathcal{P}(e) \backslash \{0\}$ and $RT \subseteq \operatorname{supp}(Q)$ be the outcome of the symbolic reduction of $P \subseteq \mathcal{P}(e) \backslash \{0\}$. Consider $H \subseteq \mathcal{P}(e) \backslash \{0\}$ be the set of *linear reducers* added to the initial set $P \subseteq \mathcal{P}(e) \backslash \{0\}$, $Q = P \cup H$. Then there is a one-to-one correspondence between the terms RT and the polynomials H:

- for any $s \in RT$ there exists one and only one $h \in H$ such that $\ell t(h) = s$,
- $RT = \ell t(H)$.

Proof. Clear by inspection of the symbolic reduction algorithm layout on page 238 and the stability of the leading term function within the computation of any element of H during symbolic reduction. $\square$

3.4.2 Row echelon elimination

The linear row echelon elimination is invoked as a part of the reduction process to perform the elimination of terms in the result of the preceding symbolic reduction. To describe the elimination as it is implemented in the present algorithm we would need some linear algebra machinery. Though the required linear algebra over $\mathbb{k}$ towards a particular form of a generating set of a finite dimensional $\mathbb{k}$-linear vector space in the extended polynomials $\mathcal{P}(e)$ usually referred as the *row echelon form* is pretty elementary and well-known it is usually described in the linear algebra textbooks as a part of matrix theory while we prefer to formulate things in the language of polynomials. Surely the language as well as the notation are absolutely irrelevant for the essence of linear algebra due to the (canonical) isomorphism between a $\mathbb{k}$-linear space and the direct sum of the sufficiently many instances of the field $\mathbb{k}$. However, we wish to provide a rather brief but full and stand-alone (up to a reasonable extent) derivation of the needed facts from linear algebra, mainly to demonstrate how the techniques typically used in polynomial computer algebra can be settled down in the linear algebra setup resulting in proofs which look pretty transparent and natural and at the same time do not look like the typical argumentation for linear algebra related facts[7].

On the other hand, if we do not see a natural way to employ the polynomial algebra technique for the argumentation of a fact then we restrict the presentation to a reference to the respective result from linear algebra.

All linear spaces in this section are assumed to be finite dimensional unless explicitly specified.

Since we use a customary representation of polynomials in $\mathcal{P}(e)$ as (the uniquely defined) linear combinations of extended terms $\mathbb{T}(e)$, it is convenient to define the set of extended terms which is sufficient to represent the result as well as any intermediate polynomials in a linear elimination process provided by the following definition.

Definition 3.4.2. Let $K\subseteq\mathcal{P}(e)$ be a $\mathbb{k}$-vector space. The **support** of K is

$$\operatorname{supp}(K) = \bigcup_{h\in K} \operatorname{supp}(h).$$

The support of a linear space is very easy to compute - it can be readily read off an arbitrary basis of the space according to the following lemma.

[7]Aside from aesthetical reasons, such an approach can be useful for a later search for a numerically advantageous modification or even an alternative to the Gröbner bases theory as an elimination technique in polynomial systems solving. A brilliant example of such an alternative is an approach presented in [Mou99] for computing the multiplicative structure of a zero-dimensional algebra based on a criterion which generalizes Buchberger's S-polynomial criterion but is formulated in the intrinsic terms of operations on polynomial vector spaces instead of pairs of polynomials.

Lemma 3.4.3. Let $K \subseteq \mathcal{P}(e)$ be a $\Bbbk$-vector space. Then for any $\Bbbk$-linear generating set $F \subseteq K \setminus \{0\}$ of K

$$\Big(\bigcup_{f \in F} \operatorname{supp}(f) = \Big) \operatorname{supp}(F) = \operatorname{supp}(K).$$

Proof. The inclusion $\operatorname{supp}(F) \subseteq \operatorname{supp}(K)$ is trivial due to $F \subseteq K$. Conversely, consider any $h = \sum_{f \in F} \alpha_f \cdot f \in K$, $\alpha_f \in \Bbbk$ for all $f \in F$. By the linear independence of extended terms (or, more rigidly, by inductive argument based on part b) of lemma 2.3.3)

$$\operatorname{supp}(h) \subseteq \bigcup_{f \in F} \operatorname{supp}(f) = \operatorname{supp}(F).$$

Therefore

$$\operatorname{supp}(K) = \bigcup_{h \in H} \operatorname{supp}(h) \subseteq \operatorname{supp}(F).$$

□

The important particular case of the lemma is the following.

Corollary 3.4.1. For any $\Bbbk$-linear basis (i.e. a linear independent generating set without 0 according to a common convention in linear algebra) $F \subseteq K \setminus \{0\}$ of a $\Bbbk$-linear vector space K,

$$\operatorname{supp}(F) = \operatorname{supp}(K).$$

Proof. Trivially by the lemma. □

A clear but somewhat required to note meaning of the latter corollary is the fact that the terms in the support of some given system of polynomials are sufficient to express (as linear combinations) the result of any linear operations on polynomials of the system. This fact is tacitly used in the sequel.

Definition 3.4.3. Let $K \subseteq \mathcal{P}(e)$ be a $\Bbbk$-linear vector space, $N = \operatorname{supp}(K)$. Fix an *arbitrary linear order* $<_N$ on N. Then, for any non-zero polynomial

$$h = \sum_{s \in \operatorname{supp}(h)} \alpha_s \cdot s \in K \setminus \{0\} \quad \text{with} \quad \alpha_s \in \Bbbk^* \quad \text{for all} \quad s \in \operatorname{supp}(p)$$

the $<_N$-leading term, the $<_N$-leading coefficient and the $<_N$-leading monomial are defined in the natural way as the $<_N$-maximal element in the support of h, the coefficient at the $<_N$-leading term and their product

$$\begin{aligned} \ell\mathrm{t}_{<_N}(h) &= \max_{<_N}\big(\operatorname{supp}(h)\big), \\ \ell\mathrm{c}_{<_N}(h) &= \alpha_s \quad \text{with} \quad s = \ell\mathrm{t}_{<_N}(h), \\ \ell\mathrm{m}_{<_N}(h) &= \ell\mathrm{c}_{<_N}(h) \cdot \ell\mathrm{t}_{<_N}(h), \end{aligned}$$

all well-defined by the uniqueness of the support of extended polynomials provided by part e) of theorem 2.3.1: $\emptyset \neq \mathrm{supp}\,(h) \subseteq N$. It is convenient to use the same notation as a shortcut for the set of $<_N$-leading terms, $<_N$-leading coefficients and $<_N$-leading monomials of an arbitrary set of non-zero polynomials $F \subseteq K \setminus \{0\}$:

$$\begin{aligned}
\ell\mathrm{t}_{<_N}(F) &= \Big\{ \ell\mathrm{t}_{<_N}(h) \;\Big|\; h \in F \Big\}, \\
\ell\mathrm{c}_{<_N}(F) &= \Big\{ \ell\mathrm{c}_{<_N}(h) \;\Big|\; h \in F \Big\}, \\
\ell\mathrm{m}_{<_N}(F) &= \Big\{ \ell\mathrm{m}_{<_N}(h) \;\Big|\; h \in F \Big\}.
\end{aligned}$$

Definition 3.4.4. Consider a finite set of polynomials $G \subseteq \mathcal{P}(e) \setminus \{0\}$. Fix an arbitrary linear order $<_N$ on $N = \mathrm{supp}\,(G)$. Then G is called the **row echelon basis** (of $\langle G \rangle_{\Bbbk}$ – the $\Bbbk$-linear vector span of G) w.r.t. $<_N$, iff no $<_N$-leading term of a polynomial P is in the support of any other member of P:

$$\forall g \in G \quad \ell\mathrm{t}_{<_N}(g) \notin \mathrm{supp}\,(G \setminus \{g\})\,.$$

Some immediate properties of the row echelon basis are collected in the following lemma.

Lemma 3.4.4. Let $G \subseteq \mathcal{P}(e) \setminus \{0\}$ be a row echelon basis w.r.t. some linear order $<_N$ on $N = \mathrm{supp}\,(G)$. Denote $K = \langle G \rangle_{\Bbbk}$. Then

a) the leading terms of G are pairwise different: for any $g_1, g_2 \in G$

$$\text{if} \;\; g_1 \neq g_2 \;\; \text{then} \;\; \ell\mathrm{t}_{<_N}(g_1) \neq \ell\mathrm{t}_{<_N}(g_2);$$

b) The "inner" terms of the polynomials in G are all different from the leading terms of G:

$$\Big(\bigcup_{g \in G} \big(\mathrm{supp}\,(g) \setminus \{ \ell\mathrm{t}_{<_N}(g) \} \big) \Big) \bigcap \ell\mathrm{t}_{<_N}(G) = \emptyset;$$

c) G is a $\Bbbk$-basis of K: the polynomials G are $\Bbbk$-linear independent;

d) $\Big(\big\{ \ell\mathrm{t}_{<_N}(h) \,|\, h \in K \setminus \{0\} \big\} = \Big) \; \ell\mathrm{t}_{<_N}(K) = \ell\mathrm{t}_{<_N}(G) \; \Big(= \big\{ \ell\mathrm{t}_{<_N}(g) \,|\, g \in G \big\} \Big)$;

e) Any row echelon basis of K is equivalent to G up to multiplication of the basis polynomials with non-zero scalars.

Proof.

a) Let $g_1, g_2 \in G$. If $g_1 \neq g_2$ then, by definition 3.4.4,

$$\ell\mathrm{t}_{<_N}(g_1) \notin \operatorname{supp}(G \backslash \{g_1\}) \supseteq \operatorname{supp}(g_2) \ni \ell\mathrm{t}_{<_N}(g_2)$$

and hence $\ell\mathrm{t}_{<_N}(g_1) \neq \ell\mathrm{t}_{<_N}(g_2)$.

b) Let $s \in \Big(\bigcup_{g\in G} \big(\operatorname{supp}(g) \backslash \{\ell\mathrm{t}_{<_N}(g)\}\big)\Big) \bigcap \ell\mathrm{t}_{<_N}(G)$. Then there exist $g_1, g_2 \in G$ such that $\ell\mathrm{t}_{<_N}(g_1) = s$ and $s \in \operatorname{supp}(g_2) \backslash \{\ell\mathrm{t}_{<_N}(g_2)\}$. It is easy to see that $\ell\mathrm{t}_{<_N}(g_1) \neq \ell\mathrm{t}_{<_N}(g_2)$ and hence $g_1 \neq g_2$ implies $g_2 \in G \backslash \{g_1\}$; therefore $\operatorname{supp}(g_2) \subseteq \operatorname{supp}(G \backslash \{g_1\})$ which contradicts $s = \ell\mathrm{t}_{<_N}(g_1) \notin \operatorname{supp}(G \backslash \{g_1\})$ according to definition 3.4.4:

$$s = \ell\mathrm{t}_{<_N}(g_1) \notin \operatorname{supp}(G \backslash \{g_1\}) \supseteq \operatorname{supp}(g_2) \supseteq \operatorname{supp}(g_2) \backslash \{\ell\mathrm{t}_{<_N}(g_2)\} \ni s.$$

c) Let

$$0 = \sum_{g\in G} \alpha_g \cdot g = \overbrace{\sum_{g\in G} \alpha_g \cdot \ell\mathrm{m}_{<_N}(g)}^{\displaystyle h_1 \atop \displaystyle \|} + \overbrace{\sum_{g\in G} \alpha_g \cdot \big(g - \ell\mathrm{m}_{<_N}(g)\big)}^{\displaystyle h_2 \atop \displaystyle \|}$$

$$\text{with } \ \alpha_g \in \Bbbk \ \text{ for all } \ g \in G.$$

Then, by part *b*), $\operatorname{supp}(h_1) \cap \operatorname{supp}(h_2) = \emptyset$ and hence, by the linear independence of extended terms, $h_1 = h_2 = 0$. In particular, by part *a*) and again the linear independence of extended terms,

$$h_1 = \sum_{g\in G} \alpha_g \cdot \ell\mathrm{m}_{<_N}(g) = 0 \ \text{ implies } \ \alpha_g \cdot \ell\mathrm{c}_{<_N}(g) = 0 \ \text{ and hence } \ \alpha_g = 0 \ \text{ for all } \ g \in G.$$

d) The inclusion $\ell\mathrm{t}_{<_N}(G) \subseteq \ell\mathrm{t}_{<_N}(K)$ is trivial. For the converse take any $h \in K \backslash \{0\}$. Then

$$h = \sum_{g\in G} \alpha_g \cdot g = \sum_{g\in G_h} \alpha_g \cdot g \ \text{ with } \ \alpha_g \in \Bbbk \ \text{ for all } \ g \in G, \quad G_h = \{g \in G \mid \alpha_g \neq 0\} \neq \emptyset.$$

Take

$$m = \max_{<_N} \big(\ell\mathrm{t}_{<_N}(G_h)\big).$$

By part *a*) there exists the unique $g_m \in G_h \subseteq G$ such that $\ell\mathrm{t}_{<_N}(g_m) = m$. Clearly

$$h = \alpha_{g_m} \cdot \ell\mathrm{m}_{<_N}(g_m) + \underbrace{\alpha_{g_m} \cdot (g_m - \ell\mathrm{m}_{<_N}(g_m)) + \sum_{g\in G_h \backslash \{g_m\}} \alpha_g \cdot g}_{\displaystyle \| \atop \displaystyle h'}$$

Note that $m \notin \operatorname{supp}(G \backslash \{g_m\}) \supseteq \operatorname{supp}(G_h \backslash \{g_m\})$ by definition 3.4.4 of the row echelon basis; then it is easy to see by the choice of m that h' is a linear combination of extended terms strictly $<_N$-smaller than m. Therefore, by the linear independence of extended terms and definition 3.4.3 of the $<_N$-leading term (the strict argument should use a proper analogue of part *b*) of lemma 2.3.10), $\ell t_{<_N}(h) = \ell t_{<_N}(g_m) \in \ell t_{<_N}(G)$.

e) Let $H \subseteq K \backslash \{0\}$ be another row echelon basis of K. By part *d*)

$$\ell t_{<_N}(H) = \ell t_{<_N}(K) = \ell t_{<_N}(G). \tag{$*$}$$

Take any $g \in G$. By $(*)$ and part *a*) there exists the unique $h \in H$ such that $\ell t_{<_N}(g) = \ell t_{<_N}(h)$. Consider

$$q = g - \frac{\ell c_{>_N}(g)}{\ell c_{>_N}(h)} \cdot h.$$

Assume $q \neq 0$. It is immediate by the linear independence of extended terms that $\operatorname{supp}(q) \subseteq \big(\operatorname{supp}(g) \cup \operatorname{supp}(h)\big) \backslash \{\ell t_{<_N}(g)\}$. W.l.o.g. assume

$$\ell t_{<_N}(q) \in \operatorname{supp}(g) \backslash \{\ell t_{<_N}(g)\}. \tag{1}$$

Note that $q \in K$ and hence, by $(*)$, there exists $g' \in G$ such that

$$\ell t_{<_N}(q) = \ell t_{<_N}(g'). \tag{2}$$

On the other hand it is easy to deduce from part *b*) that

$$\Big(\operatorname{supp}(g) \backslash \{\ell t_{<_N}(g)\}\Big) \cap \{\ell t_{<_N}(g')\} = \emptyset$$

which is a contradiction to (1) and (2). Then $q=0$ and hence $g = \frac{\ell c_{>_N}(g)}{\ell c_{>_N}(h)} \cdot h$. The presented reasoning gives an injective correspondence $G \to H$ where each $g \in G$ is associated with $h_g \in H$ such that $g = \alpha_g \cdot h_g$ with $0 \neq \alpha_g \in \Bbbk$. On the other hand, by part *c*) and the well-known invariance of the number of basis elements of a finite dimensional $\Bbbk$-linear space (alternatively by an easy combination of part *a*) and $(*)$), $|G| = |H|$ and therefore the aforesaid injection is bijective. □

An important issue not mentioned so far is the existence statement of a row echelon basis. Indeed the existence is well-known from linear algebra provided by the so-called Jordan-Gauss elimination process (or just the row echelon algorithm); it is also presented in the next lemma in the polynomial language which is indeed irrelevant due to the canonical isomorphism of the respective linear spaces. The essential part of the lemma is the correctness and termination proof of the row echelon algorithm which looks highly unusual compared with how it is typically done in linear algebra textbooks. Indeed the idea comes from the Gröbner bases theory: it is a "downcast" for the linear algebra setup of the correctness and termination argumentation for the auto-reduction algorithm of a polynomial set (cf. [BW93] proposition 5.30 on page 203).

Lemma 3.4.5. Let $F \subseteq \mathcal{P}(e)$ be a finite set of polynomials. Fix an arbitrary linear order $<_N$ on $N = \operatorname{supp}(F)$. Denote $K = \langle F \rangle_{\Bbbk}$ – the $\Bbbk$-linear vector span of F. Then the following **row echelon algorithm** computes the row echelon basis G of K in finitely many steps.

Require: $F \subseteq \mathcal{P}(e)$ – the finite input
Ensure: G – the row echelon basis of $K = \langle F \rangle_{\Bbbk}$

$G := F \setminus \{0\}$
while $\exists q \in G$ such that $\ell\mathrm{t}_{<_N}(q) \in \operatorname{supp}(G) \setminus \{q\}$ **do**
  choose the *pivot* $q \in G$ such that $\ell\mathrm{t}_{<_N}(q) \in \operatorname{supp}(G) \setminus \{q\}$
  $R := \{ g \in G \setminus \{q\} \mid \ell\mathrm{t}_{<_N}(q) \in \operatorname{supp}(g) \}$ $(\neq \emptyset)$
  $G := G \setminus R$
  for $r \in R$ **do**
    $\alpha := \mathfrak{C}_r(\ell\mathrm{t}_{<_N}(q))$ $(\neq 0)$ { the coefficient at $\ell\mathrm{t}_{<_N}(q)$ in r}
    $h := r - \frac{\alpha}{\ell\mathrm{c}_{<_N}(q)} \cdot q$
    if $h \neq 0$ **then**
      $G := G \cup \{h\}$
    end if
  end for
end while

Proof. The condition

$$(K=)\ \langle F \rangle_{\Bbbk} = \langle G \rangle_{\Bbbk} \tag{$*$}$$

is an invariant of the outer-while-loop. Indeed, it clearly holds before the loop starts and each run of the inner-while-loop results in a sequence of *non-degenerated elementary linear transformations* of the elements of G that does not change the span $\langle G \rangle_{\Bbbk}$ as it is well-known from linear algebra. If the algorithm terminates, then the resulting G is the $<_N$-row echelon basis of K according to $(*)$ and the clause of the outer-while-loop. To prove termination let $F = \{f_1, \ldots, f_m\}$ be the input set. Consider an (ordered) tuple $\Gamma = (f_1, \ldots, f_m)$. Each transformation of the basis in the flow of the algorithm can be associated with a respective transformation of Γ: every time some f_i is selected in the inner-while-loop, it is *always* replaced by the result of its non-degenerated elementary linear transformation h.

We need one more supplementary construction: define an order $\prec$ on the set 2^N of subsets of N as a lexicographical order where the subsets of N are regarded as $<_N$-descending tuples such that $\emptyset$ is the smallest and for any $t_1, t_2 \in 2^N \setminus \{\emptyset\}$

$$
\begin{aligned}
t_1 \prec t_2 \quad & \text{if and only if} \\
& s_1 < s_2 \text{ or} \\
& s_1 = s_2 \text{ and } t_1 \setminus \{s_1\} \prec t_2 \setminus \{s_2\} \text{ where } s_1 = \max_{<_N}(t_1),\ s_2 = \max_{<_N}(t_2).
\end{aligned}
$$

It is straightforward to show that $\prec$ is a linear order on 2^N.

By inspection of the algorithm it is clear that each iteration of the outer-while-loop results in at least one iteration of the inner-while-loop and hence in at least one elementary linear transformation. Then, by the pigeonhole argument, the infinitely many iterations of the outer-while-loop result in infinitely many changes of some entry f_k in Γ. On the other hand each change of f_k presumes $f_k \neq 0$ as it is an elementary linear transformation $f_k \to h$ that eliminates some term in $\operatorname{supp}(f_k)$ replacing it with a linear combination of strictly $<_N$-smaller terms and hence

$$2^N \in \operatorname{supp}(h) \prec \operatorname{supp}(f_k) \ni 2^N$$

as it is easy to see by the construction of $\prec$. But then infinitely many changes of f_k are impossible since 2^N is clearly a finite set: $\left|2^N\right| = 2^{|N|}$. $\square$

An important difference of the presented version of the row echelon algorithm from its conventional linear algebra counterpart is an arbitrary order of elimination. Indeed, the well-known row echelon elimination in a matrix traverses the columns from left to right; this provides an lucid visualization of the flow of computation as well as an easy to *see* correctness and termination arguments as an incremental construction of the row echelon form, such that the elimination in each column does not ruin the row echelon form of the already processed left part of the matrix and extends the eliminated part by the column. In our language, for a fixed order $<_N$ on the support N of the input of the row echelon algorithm, this order prescribes the elimination of the $<_N$-greatest leading term among all possible candidates at each iteration of the outer-while-loop; this might appear quite restrictive in the numerical setup to maintain the stability. A well-known recipe in numerical linear algebra is the elimination with *complete pivoting* where the predefined order of elimination is compensated by swapping the columns of the matrix during elimination. Unfortunately it is not always possible to use the analogue of complete pivoting in our application to the numerical computation of Gröbner bases because the swap of columns clearly corresponds to a modification of the order of terms $<_N$, which is not always possible due to restrictions imposed by the *term order* chosen for a Gröbner basis computation. The linear elimination implemented in our numerical Gröbner bases algorithm is generally able to eliminate a *good* number of terms using the following straightforward analogue of complete pivoting.

Require: $F \subseteq \mathcal{P}(e) \setminus \{0\}$ – the finite input
Require: $<_N$ – an arbitrary linear order on $\mathrm{supp}\,(F)$
Ensure: G – the row echelon basis of $K = \langle F \rangle_{\Bbbk}$

$N := \mathrm{supp}\,(F)$
$Pend := N$ { the pending set of terms to be eliminated}
$G := F$ { the initial basis}
while $Pend \neq \emptyset$ **do**
  take $m = \max\limits_{<_N}(Pend)$
  if $m \in \ell\mathrm{t}_{<_N}(G)$ **then**
    take any $f \in G$ such that $m = \ell\mathrm{t}_{<_N}(f)$
    if $m \in G \setminus \{f\}$ **then**
      choose $s \in Pend$ and $q \in G$ such that $m \geq \ell\mathrm{t}_{<_N}(q)$ and $s \in \mathrm{supp}\,(q)$ {e.g. $s = m$ and $q = f$ or using some selection strategy}
      define a new linear order $<'_N$ on N such that for any $s_1, s_2 \in N$
        $s_1 <'_N s_2$ iff $\sigma(s_1) <_N \sigma(s_2)$ with $\sigma = (m, s)$ – a permutation on N.
      $<_N \; := <'_N$ { swap m and s in the order $<_N$; now $\ell\mathrm{t}_{<_N}(q) = s = \max\limits_{<_N}(Pend)$}
      $R := \{r \in G \setminus \{q\} \mid s \in \mathrm{supp}\,(r)\}$
      $G := G \setminus R$
      for $r \in R$ **do**
        $\alpha := \mathfrak{C}_r(s)$ {the coefficient in r at s}
        $h := r - \frac{\alpha}{\ell\mathrm{c}_{<_N}(q)} \cdot q$ {eliminate s from h}
        if $h \neq 0$ **then**
          $G := G \cup \{h\}$
        end if
      end for
      $m := s$
    end if
  end if
  $Pend := Pend \setminus \{m\}$
end while

However, there are situations when the order of certain terms must be preserved: that is exactly the case when the extra freedom provided by a flexible order of elimination available in the row echelon algorithm as presented in lemma 3.4.5 is exploited to maintain the numerical stability by the selection of the next term among all candidates for elimination. A natural question is the efficiency of elimination in such an arbitrary order: is it possible that the already eliminated term reappears again during the subsequent elimination and hence has to be eliminated again? Such a situation is indeed possible in the usual total reduction of polynomials e.g. during a Gröbner basis construction unless the utilized reducible term selection strategy always prescribes the elimination of the maximal (w.r.t. the term order) term among all possible candidates. Fortunately, this is not the case for the row echelon algorithm according to the following lemma.

Lemma 3.4.6. In terms and conditions of lemma 3.4.5, no term in N can be selected more than once in the outer-while-loop.

Proof. Let $H \subseteq K \setminus \{0\}$ be the intermediate generating set of K attained at some point of the row echelon elimination w.r.t. some linear order $<_N$ on the terms N. For each term $s \in N$, the *row echelon property* w.r.t. H is defined as an abbreviation for the condition

$$\text{if } \exists h \in H \text{ such that } \ell\mathrm{t}_{<_N}(h) = s \text{ then } s \notin \mathrm{supp}\,(H \setminus \{h\}) \qquad (*)$$

It is clear (trivially by definition 3.4.4) that H is the row echelon basis of K if and only if the row echelon property $(*)$ w.r.t. H holds for all terms $s \in N$.

Consider an arbitrary iteration of the outer-while-loop. Let $\Sigma \subseteq N$ be the terms selected for elimination in the previous iterations. Assume no term has been selected for elimination more than once so far and the row echelon property $(*)$ w.r.t. the current basis G holds for all elements of Σ i.e. for each $s \in \Sigma$ there exists $p_s \in G$ such that

$$\ell\mathrm{t}_{<_N}(p_s) = s \tag{1}$$

and

$$s \notin G \setminus \{p_s\}. \tag{2}$$

Denote $p_\Sigma = \{p_s \mid s \in \Sigma\}$.

Let $q \in G$ be the pivot selected at the current iteration to eliminate the term $m = \ell\mathrm{t}_{<_N}(q)$. Then, according to the outer-while-loop clause, m must have failed the row echelon property and therefore $m \notin \Sigma$. Furthermore, according to (1), $\Sigma = \ell\mathrm{t}_{<_N}(p_\Sigma)$ and hence $q \notin p_\Sigma$. But then, by (2), none of the terms in Σ can occur in the support of q:

$$\begin{aligned} \mathrm{supp}\,(q) \bigcap \Sigma &= \mathrm{supp}\,(q) \bigcap \left(\mathop{\cup}_{s \in \Sigma} \{s\} \right) \\ &= \bigcup_{s \in \Sigma} \Big(\underbrace{\mathrm{supp}\,(q) \cap \{s\}} \Big) = \emptyset. \end{aligned} \tag{3}$$

$$\| \; s \in \Sigma \Rightarrow q \neq p_s \Rightarrow q \in G \setminus \{p_s\} \overset{(2)}{\Rightarrow} s \notin \mathrm{supp}\,(q)$$

$$\emptyset$$

The elimination of m does not break the row echelon property of any term in Σ. Indeed, let $R{=}\big\{r{\in}G\setminus\{q\}\,|\, m{\in}\operatorname{supp}(r)\big\}$ be the polynomials in $G\setminus\{q\}$ where m has to be eliminated. Denote

$$\text{for each } r{\in}R \quad h_r{=}r{-}\alpha_r{\cdot}q \quad \text{with} \quad \alpha_r = \frac{\mathfrak{C}_r(m)}{\ell\mathrm{c}_{<_N}(q)}{\in}\Bbbk^*,$$

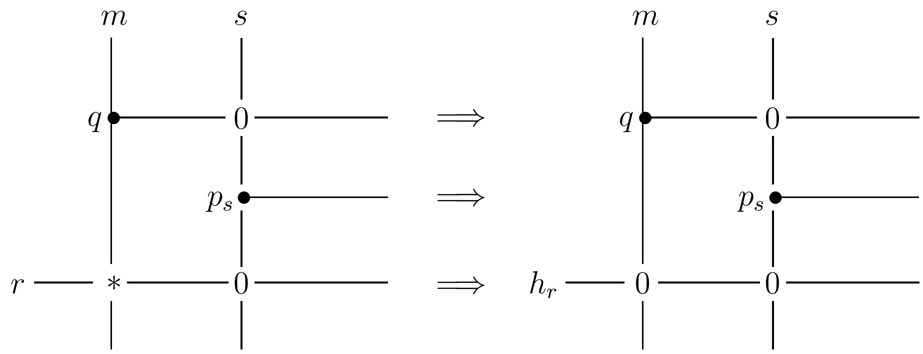
$$H{=}\{h_r\,|\, r{\in}R\}\setminus\{0\} \text{ – the result of elimination of } m,$$

$$\widehat{G}{=}(G\setminus R){\cup}H \text{ – the updated basis after the elimination of } m.$$

Take any $s{\in}\Sigma$. Consider two cases which are also visualized below (excluding trivial situations) in the matrix form, where horizontal lines represent polynomials, with their terms in the $<_N$-descending order aligned in columns represented by vertical lines, $\bullet$ indicates the $<_N$-leading term, $*$ is a term which is possibly in the support, 0 is a term definitely not in the support.

$m{>}s$: If $m{>}s{=}\ell\mathrm{t}_{<_N}(p_s)$ then $m{\notin}\operatorname{supp}(p_s)$ follows $p_s{\notin}R$, $R{\subseteq}G\setminus\{p_s\}$ and, by (2), $s{\notin}\operatorname{supp}(R)$. Therefore, for any $r{\in}R$, by (3) and the linear independence of N (or alternatively by part *b*) of lemma 2.3.3), the result h_r of elimination of m from r does not contain s in its support:

$$s{\notin}\operatorname{supp}(h_s)\,. \tag{4}$$



Therefore s satisfies the row echelon property $(*)$ w.r.t. $\widehat{G}$. Indeed, for any $g{\in}\widehat{G}$

if $g{\in}G\setminus R$ then trivially

if $g{=}p_s$ $({\in}G\setminus R)$ then, by (1), $s{=}\ell\mathrm{t}_{<_N}(p_s){=}\ell\mathrm{t}_{<_N}(g)$

otherwise $g{\in}G\setminus\{p_s\}$ and, by (2), $s{\notin}\operatorname{supp}(g)$;

otherwise $g{\in}H$ and, by (4), $s{\notin}\operatorname{supp}(g)$

which is clearly equivalent to $(*)$ for s w.r.t. $\widehat{G}$:

$$p_s{\in}\widehat{G},\, s{=}\ell\mathrm{t}_{<_N}(p_s),\, s{\notin}\widehat{G}\setminus\{p_s\}.$$

$s{>}m$: Take any $r{\in}R$. If $r{=}p_s$ then it is easy to see that the result h_{p_s} of elimination satisfies

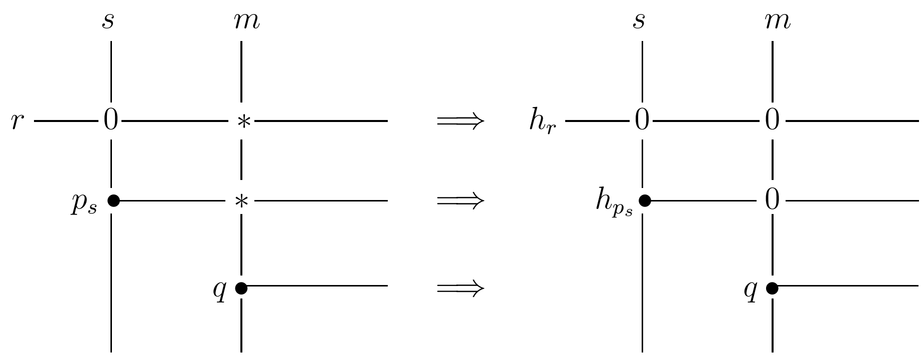
$$h_{p_s}{\neq}0 \quad\text{and}\quad \ell\mathrm{t}_{<_N}(h_{p_s}){=}\ell\mathrm{t}_{<_N}(p_s){=}s{>_N}\, m{=}\ell\mathrm{t}_{<_N}(q) \tag{5}$$

(a rigid argument is a proper version of part b) of lemma 2.3.10). Otherwise $r{\in}G\setminus\{p_s\}$ and then, by (2), $s{\notin}\,\mathrm{supp}\,(r)\subseteq\mathrm{supp}\,(G\setminus\{p_s\})$. Thus, by (3) and the linear independence of N,

$$s{\notin}\,\mathrm{supp}\,(r)\cup\mathrm{supp}\,(q)\supseteq\mathrm{supp}\,(r{-}\alpha_r{\cdot}q)=\mathrm{supp}\,(h_r)\,,$$

and hence

$$s{\notin}\,\mathrm{supp}\,(h_r)\,. \tag{6}$$



Therefore s satisfies the row echelon property $(*)$ w.r.t. $\widehat{G}$. Indeed, consider the two situations.

$p_s{\in}R$: In this case, for any $g{\in}\widehat{G}$

if $g{\in}H$ then

if $g{=}h_{p_s}$ then, by (5), $s{=}\ell\mathrm{t}_{<_N}(h_{p_s}){=}\ell\mathrm{t}_{<_N}(g)$

otherwise, by (6), $s{\notin}\,\mathrm{supp}\,(g)$

otherwise $g{\in}G\setminus R{\not\ni}p_s$ and then, trivially by (2), $s{\notin}\,\mathrm{supp}\,(g)$

which is clearly equivalent to $(*)$ for s w.r.t. $\widehat{G}$:

$$h_{p_s}{\in}\widehat{G},\, s{=}\ell\mathrm{t}_{<_N}(h_{p_s}),\, s{\notin}\,\mathrm{supp}\left(\widehat{G}\setminus\{h_{p_s}\}\right).$$

$p_s \notin R$: Then, for any $g \in \widehat{G}$

if $g \in H$ then, by (6), $s \notin \operatorname{supp}(g)$

otherwise $g \in G \setminus R$ and

if $g = p_s$ then, by (1), $s = \ell t_{<_N}(p_s) = \ell t_{<_N}(g)$

otherwise, trivially by (2), $s \notin \operatorname{supp}(g)$

which is again clearly equivalent to $(*)$ for s w.r.t. $\widehat{G}$:

$$p_s \in \widehat{G}, s = \ell t_{<_N}(p_s), s \notin \operatorname{supp}\left(\widehat{G} \setminus \{p_s\}\right).$$

We have shown that the elimination of m does not break the row echelon property of any term in Σ. Furthermore, as discussed in the proof of lemma 3.4.5, m satisfies the row echelon property w.r.t. $\widehat{G}$ – the new generating set of K attained after the elimination of m. Also, the initial assumption on Σ is turned into the same assumption for $\Sigma := \Sigma \cup \{m\}$. Then, by an inductive argument based on the presented reasoning, the row echelon elimination algorithm finishes after finitely many steps provided by the finiteness of N arriving at Σ and G such that none of the terms Σ have been selected more than once in the flow of the algorithm and the row echelon property w.r.t. G holds for any $s \in \Sigma$. On the other hand, according to the outer-while-loop clause, row echelon property w.r.t. G also holds for any $s \in N \setminus \Sigma$. Hence G is the row echelon basis of K. Indeed we have also attained an alternative to the correctness and termination proof of the row echelon algorithm used in lemma 3.4.5. □

Remark 3.4.1. The result of lemma 3.4.6 does not generally hold for a more "moderated" version of linear elimination than the row echelon elimination. Indeed consider the following example, where the elimination is performed for duplicated leading terms only and does not touch the "inner" terms in the polynomials in analogy with the usual Gaussian linear elimination of a matrix to a triangular form. Let $F = \{x+y, x+2y, y-1, y\}$ with $x > y > 1$. If we eliminate x after y then there appears an extra polynomial with the leading term y and hence y has to be eliminated once again:

$$\begin{pmatrix} x & + & y & & \\ x & + & 2y & & \\ & & y & - & 1 \\ & & y & & \end{pmatrix} \overset{y}{\Longrightarrow} \begin{pmatrix} x & + & y & & \\ x & + & 2y & & \\ & & y & - & 1 \\ & & & & 1 \end{pmatrix} \overset{x}{\Longrightarrow}$$

$$\begin{pmatrix} x & + & y & & \\ & & y & & \\ & & y & - & 1 \\ & & & & 1 \end{pmatrix} \overset{y}{\Longrightarrow} \begin{pmatrix} x & + & y & & \\ & & & & 1 \\ & & y & - & 1 \\ & & & & 1 \end{pmatrix} \overset{1}{\Longrightarrow} \begin{pmatrix} x & + & y & & \\ & & & & \\ & & y & - & 1 \\ & & & & 1 \end{pmatrix}.$$

Remark 3.4.2. Each elementary linear transformation in the flow of the row echelon algorithm – both the basic version on page 249 and the one with complete pivoting on page 251 – can be accompanied by multiplication of the result with an arbitrary non-zero constant. Indeed this gives a well-known variation of the row echelon elimination widely used in the numerical context where the multiplication with non-zero scalars commonly known as *scaling* is a trick to keep a similar order of magnitude in the coefficients of the elimination matrix to maintain numerical stability: it has been observed empirically that if the coefficients vary greatly in size then the loss of significance errors are introduced and the rounding error propagation is worse. In our approach it is straightforward to check that the argumentation in lemma 3.4.5 as well as in lemma 3.4.6 is applied verbatim to the variation of the row echelon elimination with scaling due to the fact that multiplication with a non-zero constant

- *does not* change the $\Bbbk$-linear vector span of a set of polynomials as it is well-known from linear algebra (indeed it is clear),
- *does not* change the support of a polynomial (immediate by the linear independence of terms).

Now we turn towards utilization of the row echelon elimination as a part of the reduction process in our numerical Gröbner bases algorithm. Recall that our implementation of the reduction process in the frame of a Gröbner basis computation always performs total reduction of a block of polynomials modulo the current basis as well as the total auto-reduction of the block. Since the block of polynomials to be reduced is typically a block of critical elements whose non-zero reduced forms are then added to the current basis, the total auto-reduction of the block can be considered as a compromise between two extreme approaches in a Gröbner basis computation:

- either to keep the current basis totally reduced at any point of a Gröbner basis computation
- or no polynomial in the current basis is ever reduced modulo any newer introduced in the basis reduced forms while the total auto-reduction of the attained Gröbner basis is performed at the very end of computation.

Further details and considerations concerning the auto-reduction compromise can be found in the discussion on page 229.

To reiterate the setup let $F \subseteq \mathcal{P}(e) \setminus \{0\}$ be the current basis attained at some point of a Gröbner basis computation w.r.t. some extended term order $<$, when the reduction of a block of polynomials $\widehat{P} \subseteq \mathcal{P}(e) \setminus \{0\}$ modulo F as well as the auto-reduction (also called inter-reduction in the literature) has been requested. Then the algorithm invokes the symbolic reduction procedure discussed in section 3.4.1. The outcome of symbolic reduction is the set $P = \widehat{P} \cup H$ attained as an extension of $\widehat{P}$ with the *linear reducers* H and the set $RT \subseteq \operatorname{supp}(P)$ of all terms $s \in \operatorname{supp}(P)$ such that s is either reducible modulo the current basis F or properly reducible modulo P; furthermore, $\ell\mathrm{t}(H) = RT$ and for each term $s \in RT$ there exists one and only one $h \in H$ with $\ell\mathrm{t}(h) = s$.

Indeed the symbolic reduction is the first phase of the reduction process where the computed linear reducers H are stored for the future use in the linear elimination process instead of being immediately used for elimination of the respective terms as it would have been the case in the "usual" arrangement of the reduction process. Therefore what we expect from the linear elimination is the execution of the postponed linear elimination operations (no surprise here!) resulting in a block $R \subseteq \mathcal{P}(e) \setminus \{0\}$ of reduced forms of $\widehat{P}$ such that no polynomial in R contains any term from RT in its support and R is also totally linearly auto-reduced i.e. it is the row echelon basis (of the $\Bbbk$-space $\langle R \rangle_{\Bbbk}$) or, for short, R is in the row echelon form. The existence and uniqueness of this informally introduced reduced form in $\langle P \rangle_{\Bbbk}$ for each element $\widehat{P}$ is provided as a particular case of the following lemma.

Lemma 3.4.7. Let $P = \widehat{P} \cup H \subseteq \mathcal{P}(e) \setminus \{0\}$ and $RT \subseteq \operatorname{supp}(P)$ be the result of the symbolic reduction of $\widehat{P}$ which extended the initial $\widehat{P}$ with linear reducers H and $RT = \ell\mathrm{t}(H)$, where the leading terms are considered w.r.t. the original term order $<$ on $\mathbb{T}(e) \subseteq \mathcal{P}(e)$. Denote $N = \operatorname{supp}(P)$, $IT = N \setminus RT$. The following holds.

a) $\langle H \rangle_{\Bbbk} \cap \langle IT \rangle_{\Bbbk} = \{0\}$.

b) For any $f\in\langle N\rangle_{\Bbbk}$ there exists the unique decomposition

$$f = \eta_f + \mathrm{NF}_H(f) \quad \text{with} \quad \eta_f\in\langle H\rangle_{\Bbbk} \quad \text{and}$$
$$\mathrm{NF}_H(f)\in\langle IT\rangle_{\Bbbk} \ \text{ – the } H\text{-normal form of } f;$$

or, in other words,

$$\langle N\rangle_{\Bbbk}=\langle H\rangle_{\Bbbk}\oplus\langle IT\rangle_{\Bbbk}.$$

Proof. Fix the order on N as the restriction of the original term order $<$.

a) Take any $0\neq f\in\langle H\rangle_{\Bbbk}\cap\langle IT\rangle_{\Bbbk}$. Then

$$f = \sum_{h\in H}\alpha_h\cdot h = \sum_{s\in IT}\beta_s\cdot s \quad \text{with} \quad \alpha_h,\beta_s\in\Bbbk \quad \text{for all} \quad h\in H, s\in IT.$$

Denote

$$B = \{h\in H|\, \alpha_h\neq 0\} \ (\overset{f\neq 0}{\neq} \emptyset).$$

Let $m=\max\big(\ell\mathrm{t}(B)\big)$. Then $m\in RT$ and by lemma 3.4.2 there exists one and only one $b_m\in B$ with $\ell\mathrm{t}(b_m)=m$. W.l.o.g. assume $\ell\mathrm{c}(b_m) = \alpha_{b_m}=1$. Then it is easy to see by the choice of m that

$$f' = (b_m - m) + \sum_{h\in B\setminus\{b_m\}}\alpha_h\cdot h$$

is a linear combination of terms strictly smaller than m and hence $m\notin \mathrm{supp}\,(f')$. On the other hand

$$f = m + f' = \sum_{s\in IT}\beta_s\cdot s$$

and hence, by the $\Bbbk$-linear independence of N, $m\in IT$ which is impossible due to $RT\cap IT=\emptyset$ by construction.

b) *Existence:* Note that $\langle N\rangle_{\Bbbk}=\langle RT\rangle_{\Bbbk}\oplus\langle IT\rangle_{\Bbbk}$ by construction. Therefore, for each $g\in\langle N\rangle_{\Bbbk}$ there exists the unique decomposition

$$g = RT(g) + IT(g) \quad \text{with} \quad RT(g)\in\langle RT\rangle_{\Bbbk}, \ IT(g)\in\langle IT\rangle_{\Bbbk},$$

in particular $RT(g)$ is well-defined and $RT(g)\neq 0$ if and only if $g\notin\langle IT\rangle_{\Bbbk}$.

Take any polynomial $f\in\langle N\rangle_{\Bbbk}$. If $f\in\langle IT\rangle_{\Bbbk}$ then $\eta_f=0$ and $\mathrm{NF}_H(f)=f$ clearly provide the desired representation.

Otherwise let $m=\ell\mathrm{t}\big(RT(f)\big)\in RT$. Assume that the desired representation exists for any $g\in\langle N\rangle_{\Bbbk}\setminus\langle IT\rangle_{\Bbbk}$ such that $\ell\mathrm{t}\big(RT(g)\big)<m$. By lemma 3.4.2 there exists $h\in H$ such that $\ell\mathrm{t}(h)=m$. W.l.o.g. assume $\ell\mathrm{c}\big(RT(f)\big)=\ell\mathrm{c}(h)=1$. Let $g=\big(\ell\mathrm{t}(h)-h\big)+\big(f-\ell\mathrm{t}\big(RT(f)\big)\big)$. Then trivially $f=h+g$. If $g\in\langle IT\rangle_{\Bbbk}$ then

the desired representation of f is trivially attained with $\eta_f{=}h$, $\mathrm{NF}_H(f)=g$. Otherwise it is easy to see that $\ell\mathrm{t}\big(RT(g)\big){<}m$ and hence $g{=}\eta_g{+}\mathrm{NF}_H(g)$ with $\eta_g{\in}\langle H\rangle_{\Bbbk}$, $\mathrm{NF}_H(g)\in\langle IT\rangle_{\Bbbk}$ by assumption. But then $f{=}\eta_f{+}\mathrm{NF}_H(f)$, with $\eta_f{=}\eta_g{+}h$, $\mathrm{NF}_H(f)=\mathrm{NF}_H(g)$, is clearly the desired representation of f.
Uniqueness: By part a). ☐

The next lemma presents an appropriate form of the row echelon algorithm to attain the discussed desired outcome of the linear elimination process. The statement of the lemma might seem pretty technical while the idea behind is indeed transparent: we compute the row echelon basis of P w.r.t. some linear order $<_N$ on $N{=}\operatorname{supp}(P)$ which coincides with the *term order* on the terms $IT{=}\operatorname{supp}(P)\setminus RT$ and which is arbitrary defined on RT such that any term in RT is greater than any in IT; the desired block R of reduced forms in the row echelon form is attained as a part of the row echelon basis of $\langle P\rangle_{\Bbbk}$ whose polynomials are linear combinations of the terms IT.

Furthermore, the polynomials R appear as the row echelon basis of the intersection of the $\Bbbk$-linear spaces $\langle P\rangle_{\Bbbk}$ and $\langle IT\rangle_{\Bbbk}$[8]. That is a direct analogue to the well-known elimination property of Gröbner bases and can also be proved using the machinery of the Gröbner bases theory as a combination of

- the interpretation of the row echelon basis of a homogeneous linear system as the totally reduced Gröbner basis w.r.t. the lexicographical order induced by an arbitrary linear order on the variables of the system (see e.g. section 10.5 in [BW93]),
- the elimination property of Gröbner bases w.r.t. a lexicographical term order (see e.g. theorem 2 on page 113 in [CLO97]) and
- a trick based on the direct decomposition of a ring of polynomials into the sum of degree-homogeneous $\Bbbk$-vector spaces together with the observation that a linear combination is a polynomial combination with constants (polynomials of degree zero) to translate the result back from polynomial ideals to the $\Bbbk$-linear spaces.

Lemma 3.4.8. Let $P{=}\widehat{P}{\cap}H{\subseteq}\mathcal{P}(e)\setminus\{0\}$, $RT{\subseteq}\operatorname{supp}(P)$ be the result of the symbolic reduction of $\widehat{P}$ which extended the initial $\widehat{P}$ with linear reducers H. Denote $N{=}\operatorname{supp}(P)$, $IT{=}N{\setminus}RT$. Fix an arbitrary linear order $<_N$ on N such that

[8] In view of this fact, our approach to the elimination of the terms RT by the use of the row echelon algorithm is a particular way to find the intersection $\langle P\rangle_{\Bbbk}{\cap}\langle IT\rangle_{\Bbbk}$. This observation provides an interesting possibility to use a different method to compute the aforementioned intersection in the linear algebra part of the reduction process instead of the currently utilized direct elimination in our numerical Gröbner bases algorithm.

- $<_N$ coincides with the original term order $<$ on IT: $<_N|_{IT} = <$, (O_1)
- any term in IT is $<_N$-less than any in RT. (O_2)

Let Q be the row echelon basis of $\langle P\rangle_{\Bbbk}$ w.r.t. $<_N$. Denote

$$R{=}Q{\cap}\langle IT\rangle_{\Bbbk}.$$

For each element $g{\in}\widehat{P}$ consider the *H-normal form* $\mathrm{NF}_H(g)$ provided by the unique decomposition

$$g = \eta_g + \mathrm{NF}_H(g) \quad \text{with} \quad \eta_g{\in}\langle H\rangle_{\Bbbk},\ \mathrm{NF}_H(g)\,{\in}\langle IT\rangle_{\Bbbk}$$

according to lemma 3.4.7. Denote

$\overline{P} \;\; = \left\{\mathrm{NF}_H(g)\;\middle|\; g{\in}\widehat{P}\right\}$ – the H-normal forms of the elements of $\widehat{P}$,

$\widetilde{P}$ – the row echelon basis of $\left\langle\overline{P}\right\rangle_{\Bbbk}$ w.r.t. $<_N$

Then

a) $\langle R\rangle_{\Bbbk} = \left\langle\widetilde{P}\right\rangle_{\Bbbk}$,

b) The polynomial sets R and $\widetilde{P}$ are equivalent up to multiplication of polynomials with non-zero scalars.

Proof. First of all note the following particular property of the order $<_N$ frequently used in the proof: for any $g{\in}\langle N\rangle_{\Bbbk}\backslash\{0\}$

$$g{\in}\langle IT\rangle_{\Bbbk} \;\text{ if and only if }\; \ell\mathrm{t}_{<_N}(g){\in}IT. \qquad (*)$$

The "only if" direction is trivial. Conversely, if $g{\in}\langle N\rangle_{\Bbbk}\backslash\{0\}$, $\ell\mathrm{t}_{<_N}(g){\in}IT$ and there exists $RT{\ni}s{\in}\,\mathrm{supp}\,(g)$ then, by (O_2), $\ell\mathrm{t}_{<_N}(g){<_N}s$ which is absurd.

a) Denote

$$K = \langle P\rangle_{\Bbbk}{\cap}\langle IT\rangle_{\Bbbk}.$$

We claim that

$$\langle R\rangle_{\Bbbk} = K = \left\langle\overline{P}\right\rangle_{\Bbbk} = \left\langle\widetilde{P}\right\rangle_{\Bbbk}. \qquad (\dagger)$$

which clearly implies the desired statement. Hence, all we need is to show all three equalities of $\Bbbk$-linear spaces in $(\dagger)$.

$\boxed{\langle R\rangle_{\Bbbk}{=}K{:}}$ The inclusion $\langle R\rangle_{\Bbbk}{\subseteq}K$ is clear due to $R{\subseteq}K$ by construction. To show the converse take any $f{\in}K$. Then, in particular, $f{\in}\langle P\rangle_{\Bbbk}$. On the other hand, Q is the row echelon basis of $\langle P\rangle_{\Bbbk}$, hence $\langle Q\rangle_{\Bbbk}{=}\langle P\rangle_{\Bbbk}$ and therefore there exists the unique representation

$$f = \sum_{q\in Q\setminus R} \alpha_q{\cdot}q + \sum_{q\in R} \alpha_q{\cdot}q \quad \text{with} \quad \alpha_q{\in}\Bbbk \quad \text{for each} \quad q{\in}Q.$$

Consider the set

$$B = \Big\{q{\in}Q\setminus R \;\Big|\; \alpha_q{\neq}0\Big\}.$$

Assume $B{\neq}\emptyset$ (otherwise there is nothing to prove). Take

$$m = \max_{<_N}\big(\ell\mathrm{t}_{<_N}(B)\big).$$

By part $a)$ of lemma 3.4.4, the $<_N$-leading terms of Q are pairwise different and hence there exists one and only one $b_m{\in}B$ such that $\ell\mathrm{t}_{<_N}(b_m){=}m$. Furthermore $Q{\ni}b_m{\notin}R{=}Q{\cap}\langle IT\rangle_{\Bbbk}$, then $b_m{\notin}\langle IT\rangle_{\Bbbk}$ and hence, according to $(*)$, $m{=}\ell\mathrm{t}_{<_N}(b_m){\in}RT$. Take

$$f' = \alpha_{b_m}{\cdot}\big(b_m{-}\ell\mathrm{m}_{<_N}(b_m)\big) + \sum_{q\in Q\setminus(R\cup\{b_m\})} \alpha_q{\cdot}q + \sum_{q\in R} \alpha_q{\cdot}q.$$

Then trivially $f{=}\alpha_{b_m}{\cdot}\ell\mathrm{m}_{<_N}(b_m){+}f'$. Recall that $b_m{\in}B{\subseteq}Q$ is the unique element of Q with the $<_N$-leading term m; then it is easy to see by the choice of m and (O_2) that f' is a linear combination of terms strictly $<_N$-smaller than m. Then clearly (a rigid argument is by a proper modification of part $b)$ of lemma 2.3.10) $\ell\mathrm{t}_{<_N}(f){=}m{\in}RT$ On the other hand, $f{\in}K{\subseteq}\langle IT\rangle_{\Bbbk}$ and hence $\ell\mathrm{t}_{<_N}(f){\in}IT$ which contradicts $RT{\cap}IT{=}\emptyset$.

$\boxed{K{=}\big\langle\overline{P}\big\rangle_{\Bbbk}:}$ The inclusion $\big\langle\overline{P}\big\rangle_{\Bbbk}{\subseteq}K$ is clear due to $\overline{P}{\subseteq}K$ by construction. To show the converse take any $f{\in}K$. Then trivially $f{\in}\langle P\rangle_{\Bbbk}{=}\Big\langle\widehat{P}{\cup}H\Big\rangle_{\Bbbk}$ and hence there exists (not generally unique) representation

$$f = \sum_{g\in\widehat{P}} \alpha_g{\cdot}g + \sum_{h\in H} \beta_h{\cdot}h \quad \text{with} \quad \alpha_g,\beta_h{\in}\Bbbk \quad \text{for each} \quad g{\in}\widehat{P},\ h{\in}H.$$

The latter representation of f can be rewritten using the unique decomposition $g{=}\mathrm{NF}_H(g)+\eta_g$ with $\mathrm{NF}_H(g)\in\langle IT\rangle_{\Bbbk}$ and $\eta_g{\in}\langle H\rangle_{\Bbbk}$ for each $g{\in}\widehat{P}$ provided by lemma 3.4.7

$$f = \overbrace{\sum_{g\in\widehat{P}} \alpha_g{\cdot}\mathrm{NF}_H(g)}^{\substack{\mathrm{NF}_H(f)\,\in\langle IT\rangle_{\Bbbk}\\ \|}} + \overbrace{\sum_{g\in\widehat{P}} \alpha_g{\cdot}\eta_g + \sum_{h\in H} \beta_h{\cdot}h}^{\substack{\eta_f{\in}\langle H\rangle_{\Bbbk}\\ \|}}\,.$$

On the other hand $f{\in}K{\subseteq}\langle IT\rangle_{\Bbbk}$ and then, using lemma 3.4.7 once again, $\eta_f{=}0$ and

$$f{=}\mathrm{NF}_H(f)=\sum_{g\in\widehat{P}}\alpha_g{\cdot}\mathrm{NF}_H(g)\in\left\langle \mathrm{NF}_H(g)\mid g{\in}\widehat{P}\right\rangle_{\Bbbk}=\langle\overline{P}\rangle_{\Bbbk}.$$

$\boxed{\langle\overline{P}\rangle_{\Bbbk}=\left\langle\widetilde{P}\right\rangle_{\Bbbk}:}$ Immediate by lemma 3.4.5 since $\widetilde{P}$ is the row echelon basis of $\langle\overline{P}\rangle_{\Bbbk}$ by construction.

b) In part *a*) we have shown that R and $\widetilde{P}$ are the row echelon bases of the same $\Bbbk$-vector space and hence, by part *e*) of lemma 3.4.4, R and P are equivalent up to multiplication of polynomials with non-zero scalars. □

Remark 3.4.3.

- By the uniqueness of the row echelon form (up to non-zero scaling factors), $\widetilde{P}$ is essentially the set of H-normal forms $\overline{P}$ reduced to the row echelon form w.r.t. $<_N$ e.g. using the algorithm from lemma 3.4.5.

- The condition that the order $<_N$ coincides with the original term order $<$ on terms IT is not important (indeed it is not used) in the proof of the lemma but essential for the row echelon elimination being considered as a part of the reduction process in our numerical Gröbner bases algorithm: it guarantees that the leading term of any element R is the same w.r.t. $<_N$ and $<$ and hence R is precisely what we wish to achieve from the linear elimination part of the reduction process: the elimination of all terms RT that have been recognized as reducible at the stage of symbolic reduction and possibly of other terms in $IT{\subseteq}N$ linearly reducible w.r.t. $<\ \big(=<_N|_{IT}\big)$.

- On the other hand, there are no restrictions imposed on the $<_N$-order on the terms in RT. Indeed, the output R read off the result of the elimination consists precisely of the polynomials of the row echelon basis which do not contain any terms RT; therefore the order on RT is irrelevant for maintaining the equivalence of the leading terms of the resulting polynomials R w.r.t. the original term order $<$ and the order $<_N$ used in the linear elimination. Furthermore, the order on RT can be arbitrarily modified during the elimination e.g. by the use of the row echelon elimination with complete pivoting.

The latter remark is particularly important for our implementation of linear elimination being combined with one more fact: It is clear that the row echelon elimination algorithm, both the basic version from lemma 3.4.5 on page 249 and the modification on page 251 with complete pivoting, can be restricted to perform elimination only on certain selected terms in the support of the input polynomials. This simple observation is crucial for our arrangement of the elimination in two phases:

First we perform the elimination restricted to the terms RT; these have been recognized as reducible during of symbolic reduction which also provided the possibility for the *linear* elimination of every reducible term. For this purpose we use the row echelon elimination with *complete pivoting* as presented on page 251.

Then we perform the elimination restricted to the terms IT which have not been recognized as reducible during symbolic reduction but appear linearly reducible. This also includes the elimination of terms which have originally been linearly reducible (before the run of symbolic reduction) and hence have not been treated in the symbolic reduction being completely delegated to the linear elimination part of the reduction process. At this stage, we use the version of the row echelon elimination with *partial pivoting* described on page 249, but looking for the next term to be eliminated among all terms which have not been treated so far following the statement of lemma 3.4.6.

Both phases of linear elimination use exactly the same pivot selection strategy which incorporates the selection of the next term to be eliminated as well as the selection the pivot polynomial to eliminate the selected term. The only difference is that it is forbidden to modify the order of terms during the elimination with partial pivoting as it is done for the terms IT and in this case the pivot polynomial must have the term to be eliminated as the leading term while the complete pivoting on RT is free of the latter restriction since the modification of the order of terms is exactly the trick to put the term to be eliminated into the leading position of the pivot polynomial.

The utilized pivot selection is a simplified reducer selection strategy used in the symbolic reduction with an emphasis towards selection of a pivot polynomial with a greater absolute value of the leading coefficient. Let $<_N$ be the linear order on N attained the some point of linear elimination when $s \in \operatorname{supp}(N)$ is selected to be eliminated. Take a pair p_1, p_2 of polynomials in the current elimination set such that $\ell\mathrm{t}_{<_N}(p_1) = \ell\mathrm{t}_{<_N}(p_2) = s$ (in the case of complete pivoting we assume, to simplify the notation, that $<_N$ is already modified to make s the leading term of p_1 and p_2). Then p_1 is an unconditionally better pivot than p_2 if p_1 is a monomial. Otherwise p_1 is a better pivot if $|\ell\mathrm{c}_{<_N}(p_1)| > |\ell\mathrm{c}_{<_N}(p_2)|$ or $|\ell\mathrm{c}_{<_N}(p_1)| = |\ell\mathrm{c}_{<_N}(p_2)|$ and $\ell\mathrm{t}_{<_N}(p_1) < \ell\mathrm{t}_{<_N}(p_2)$. If the priority of p_1 against p_2 is still indistinguishable according to the rules above then it is decided by a smaller number of terms in the support or a lower sugar degree: p_1 is better than p_2 if in the componentwise difference of tuples

$$\Big(|\operatorname{supp}(p_2)|, \operatorname{sdeg}(p_2)\Big) - \Big(|\operatorname{supp}(p_1)|, \operatorname{sdeg}(p_1)\Big)$$

the left-most non-zero entry is positive.

The following layout presents the linear elimination part of the reduction process as it is implemented in our numerical Gröbner bases algorithm.

Require: $P \supseteq \widehat{P}$, $RT \subseteq \operatorname{supp}(P)$ the output of symbolic reduction of the initial block of polynomials $\widehat{P}$.

Ensure: R – the result of elimination as described in lemma 3.4.8.

Ensure: $Simp$ – the set of simplifiers such that $\ell\mathrm{t}(Simp) \subseteq RT$ eventually used at the next call of the symbolic reduction.

$N := \operatorname{supp}(P)$

$IT := N \setminus RT$

• Set the initial linear order $<_N$ on N such that $<_N$ coincides with the original term order $<$ for any pair of terms from RT as well as for any pair from IT and any element from RT is $<_N$-greater than any in IT.

• Perform the linear elimination with complete pivoting restricted to the terms RT in P.

• Perform the linear elimination with partial pivoting restricted to the terms IT in P.

$R := \{p \in P \mid \ell\mathrm{t}(p) \in IT\}$ { the row echelon basis of $\langle P\rangle_{\Bbbk} \cap \langle IT\rangle_{\Bbbk}$ by lemma 3.4.8, for any $p \in P$ the condition $\ell\mathrm{t}(p) \in IT$ is equivalent to $p \in \langle IT\rangle_{\Bbbk}$ by construction of the order $<_N$}

$Simp := \{p \in P \mid \ell\mathrm{t}(p) \in RT\}$

The important issues concerning the linear elimination not present in this deliberately simplified layout are the following.

1. The result R of the linear elimination can generally acquire new leading terms which have not been present in the original system $\widehat{P}$ as it has been submitted to the preceding symbolic reduction. Therefore, to have the result of the whole reduction process totally auto-reduced the symbolic reduction followed by linear elimination has to be invoked iteratively until no terms can be recognized as reducible in the symbolic reduction.

2. If the coefficients is a floating-point domain then two extra operations are applied on the result of each elementary elimination as follows. Given polynomials $p, q \in \mathcal{P}(e) \setminus \{0\}$ such that $\ell\mathrm{t}(q) \in \operatorname{supp}(p)$ the elementary linear elimination eliminates $\ell\mathrm{t}(q)$ from p by subtracting the appropriate scalar multiple of q from p

$$r := p - \frac{\beta}{\ell\mathrm{c}(q)} \cdot q \quad \text{with} \quad \beta = \mathfrak{C}_p\big(\ell\mathrm{t}(q)\big) \text{ – the coefficient at } \ell\mathrm{t}(q) \text{ in } p.$$

In our implementation, no zero tolerance thresholds are applied to the coefficients in the computation of r and hence no coefficients originating from p and q generally cancel except of course the coefficient at $\ell\mathrm{t}(q)$ which is even not computed being assumed to be zero by construction following a common numerical computing practice[9]. If the result of elimination

$$r = \sum_{s \in \operatorname{supp}(r)} \alpha_s \cdot s \quad \text{with} \quad \alpha_s \in \Bbbk^* \quad \text{for} \quad s \in \operatorname{supp}(r)$$

is not zero, then the following two extra actions are applied to it.

- *Normalization:* That is scaling using a binary exponent as it avoids the nonsensical rounding during scaling on a binary machine (where the algorithm is executed). The performed normalization is almost identical to the normalization of the initial polynomials in the preprocessing step of the algorithm as described on page 206, the only difference is the utilization of a zero threshold to annihilate a polynomial with very small coefficients. Let m be the maximal binary exponent in the normalized internal floating-point representation of the coefficients of r, i.e. m is the smallest integer power of 2 such that

  $$2^m > |\alpha_s| \quad \text{for all} \quad s \in \operatorname{supp}(r) \,.$$

  If $m \leq 0$ and the absolute value of each coefficient in r is smaller than some zero tolerance threshold $1 \gg \zeta > 0$ given to the algorithm as a part of the input data

  $$|a_s| < \zeta \quad \text{for all} \quad s \in \operatorname{supp}(r)$$

[9] We have experimented with different schemes which treat a very small coefficient created in the elimination process as zero and have found such mechanism potentially dangerous for the numerical stability of the algorithm. This is due to the fact that both the final (extended) Gröbner basis as well as the intermediate basis at some point of computation are typically over-determined systems that is roughly speaking having more polynomials than variables. An over-determined system almost certainly looses solutions under an arbitrary small perturbation of coefficients. On the other hand a mechanism that excludes considerably small coefficients in an intermediate result of the computation provides effectively such a kind of perturbation unless the mechanism uses a (apparently) sophisticated technique to reliably detect the tiny numerical values which are purely the result of numerical roundoff. The search for such a (likely to be sophisticated) mechanism is an open challenge; we did not succeed in establishing it so far. Instead we currently use a very simple scheme to annihilate the whole polynomial if all coefficients in the polynomial are tiny after normalization and also ro remove the terms whose e-degrees become greater or equal to the e-bound $\mathcal{B}$ after trimming of e-degrees at the postprocessing of the result of an elementary linear elimination; such terms indeed represent the zero term in $\mathcal{P}(e)$.

then r is treated as zero $r{:=}\,0$ and then naturally excluded from any further consideration in the reduction process.

If $m{>}0$ then r is multiplied with 2^{-m}

$$r := 2^{-m} \cdot r;$$

no rounding is involved in the scaling which only affects the exponent in the floating-point representation of the coefficients α_s, $s{\in}\,\mathrm{supp}\,(r)$. Also no numerical zero tolerance schemes are involved in this case: the cancellation of selected extra small coefficients in a polynomial is not exploited in the current implementation of normalization and the annihilation of the whole polynomial, as discussed in the treatment of the case $m{\leq}0$, is not considered as an option due to the common sense reasons since it would require a ridiculously high value for the zero tolerance threshold to be the case.

The described normalization follows an empirically observed in numerical Gaussian elimination phenomena that if the coefficients of a linear system have greatly different size then the elimination is likely to introduce the loss of significance errors and also the rounding error propagation gets worse. However the subject of scaling is currently not well understood in numerical analysis especially the guarantee that the loss of significance as well as the rounding errors become smaller by scaling or even get minimized.

- *Trimming of e-degrees:* It is very similar to the initial assignment of e-degrees at the preprocessing stage before the actual algorithm starts as described in section 3.1. In particular, we use exactly the same definition of the infinitesimal order of a coefficient $\mathcal{O}(\cdot)$ (page 205) although a better implementation could e.g. dynamically adjust $\mathcal{O}(\cdot)$ observing the order of magnitude of coefficients as they appear in the intermediate polynomials in the flow of computation.

  The *result* of the normalization

$$r = \sum_{s\in\,\mathrm{supp}(r)} \alpha_s \cdot s \quad \text{with} \quad \alpha_s{\in}\mathbb{k}^* \quad \text{for} \quad s{\in}\,\mathrm{supp}\,(r)$$

  is the subject for the trimming of e-degrees performed for each term $\mathrm{supp}\,(r) \cap IT$ as follows. For any term $s{\in}\,\mathrm{supp}\,(r)$ such that $s{\notin}RT$ (equivalent to $s{\in}IT$ by construction – s has not been recognized as reducible at the preceding call to symbolic reduction)

denote

$\alpha = \alpha_s$ – the coefficient at s in r,
$t = \mathrm{T}(s)$ – the main part of s (without the power of e),
$d = \deg_e(s)$ – the e-degree of s so that $s{=}t{\cdot}e^d$.

If d is smaller than the infinitesimal order (page 205) $\mathcal{O}(\alpha)$ then s in r *"is assigned"* the higher e-degree $\mathcal{O}(\alpha)$:

$$\text{if } \ d{<}\mathcal{O}(\alpha) \ \text{ then } \ r := (r - \alpha{\cdot}s) + \alpha{\cdot}t{\cdot}e^{\mathcal{O}(\alpha)}.$$

Note that if $\mathcal{O}(\alpha){\geq}\mathcal{B}$ then the term $t{\cdot}e^{\mathcal{O}(\alpha)}$ represents the zero term in $\mathcal{P}(e)$; in this case the trimming of the e-degree indeed removes s from r which is perfectly compatible with our interpretation of the infinitesimal order of a coefficient: if $\mathcal{O}(\alpha){\geq}\mathcal{B}$ then α is indistinguishable from zero in the given precision.

Recall that in our setup e is always a local variable i.e. e has a negative weight which is equivalent to $e{<}1$ in the extended term order. Then it is easy to see by the quasi-monotonicity of the extended term order (definition 2.3.9 on page 37) that the described trimming replaces s with a strictly smaller term w.r.t. the extended term order. This trick reduces the chance to have a tiny value as the leading coefficient of a polynomial. To see why it is so, assume $r{\neq}0$ since otherwise there is nothing to do. Let $s{\in}\operatorname{supp}(r)$ be a term in r.

If s is reducible i.e. $s{\in}RT$ then we do not bother about any extra actions to be done with s since the term is either to be linear eliminated later on in the flow of the current call to the linear elimination provided that the input of the linear elimination is generated by the symbolic reduction process or the term has been already eliminated and hence nothing will be done with it any more in the current call to the linear elimination and the unique polynomial where s occurs (as the leading term) will be omitted in the final set of reduced forms (since its leading term is reducible) attained as the result of the elimination process.

On the other hand, if $s{\in}IT$ then there might be no way to reduce it. Assume the coefficient α_s at s in r is very small: small enough to have $\deg_e(s){<}\mathcal{O}(\alpha_s)$. If s is the leading term of r then there is the danger that the tiny leading coefficient will persist in the result of the linear elimination and create a potential numerical stability problem if the polynomial with the tiny leading coefficient is added to the current basis and used as a reducer in the subsequent elimination. Increasing the e-degree of s makes the term smaller and possibly removes it from the leading position. If s with the higher e-degree assigned to it is still the leading term of r, then it will be at least impossible to use r to eliminate a term $s' \in \operatorname{supp}(p')$ in a polynomial $p' \in \mathcal{P}(e) \setminus \{0\}$ if s' has a smaller e-degree (due to the reducibility relation formulated in terms of divisibility of terms cf. definition 3.0.6 on page 203); this is fine because the magnitude of the coefficient at s' in p' is likely to be considerably higher (due to the smaller e-degree of s') than the magnitude of the coefficient α_s at s in r. Moreover, even if the term s is not the leading term of r but has a very small coefficient, its replacement with a smaller term reduces the dangerous possibility that s would become the leading term somewhere later in the course of the algorithm if all terms higher than s get eliminated from r.

If the used term order extension is ${}_e\mathbb{T}$ (a term with a a higher e-degree is always smaller) then the trimming of e-degrees efficiently eradicates the currently irreducible leading terms with tiny coefficients. On the other hand, the trimming of e-degrees is less efficient for $\mathbb{T}_e$ used as the term order extension(a term with a smaller main part w.r.t. the original term order is smaller); this is a natural price for computing a basis w.r.t. the order $\mathbb{T}_e$ that is indeed the computation of the genuine Gröbner basis but generally with a more stable in the numerical respect flow of reduction.

Note that the described trimming of e*-degrees is the sole mechanism in the whole implementation of the numerical Gröbner bases algorithm which essentially deviates from the path of computation prescribed by the theory of Gröbner bases in the algebra $\mathcal{P}(e)$ of extended polynomials. The deviation has to be introduced in order to maintain the numerical stability even at the cost of the theoretical background used to develop and describe the algorithm.*

3. It is clear that the trimming of e-degrees described above may result in new terms not originally present in $\operatorname{supp}(P)$ and hence not treated in the preceding call to symbolic reduction. The elimination of any of such newly introduced due to the trimming of e-degrees terms is not performed in the current call of the row echelon elimination even if there exists such a possibility due to the following reasons: Let p be a polynomial in the intermediate elimination set and $s \in \operatorname{supp}(p)$ be such a newly introduced term. Then clearly s is not in the set RT of terms which have been recognized as reducible by the preceding

call of symbolic reduction: $s \notin RT \subseteq \operatorname{supp}(P)$. In our implementation the linear elimination of any term which is not in RT is performed without complete but with a partial pivoting: the pivot to eliminate s is searched among polynomials $\{p \in P \mid \ell\mathrm{t}(p) = s\}$ of the current elimination set P whose leading term is s. However it is preferable to perform elimination using complete pivoting for numerical stability reasons.

On the other hand if we postpone the elimination of s in the current call to the row echelon elimination then s might appear and be recognized as reducible by the successive call of symbolic reduction in the symbolic reduction and row echelon elimination loop, e.g. due to a higher e-degree of s or the new leading terms possibly introduced in the linear elimination; in this case s will be eliminated using complete pivoting by the successive call to the row echelon elimination. It is even more desirable to postpone the elimination of s at the current row echelon elimination due to the fact that the trimming of e-degrees never decreases the e-degree of a term and hence s must have been created by increasing the e-degree of some already existing term in $\operatorname{supp}(P)$ which means $\deg_e(s) > 0$. But s is also the leading term of a linear reducer $q \neq p$ of s (we assume that such q exists since otherwise no linear elimination can be performed anyway); therefore the leading coefficient of q is considered as numerically dangerous according to our numerical interpretation of the e-degree of a term and hence the elimination modulo q should be avoided if possible. On the other hand if there is a possibility to reduce s modulo a polynomial q' with a smaller e-degree of the leading term (i.e. with a more reliable in the numerical respect leading coefficient) of the leading term then the successive call of symbolic reduction will create a linear reducer of s as an appropriate shift of q' since, according to the utilized reducer selection strategy, the symbolic reduction looks for a reducer with a minimal e-degree of the leading term.

Chapter 4

Computational experiments

In this chapter we present a series of numerical experiments with the implementation of the algorithm described in chapter 3. The current implementation is capable to deal with quite complex systems: For example it can compute the famous polynomial computer algebra systems test problems `cyclic5` and `cyclic6`. For `cyclic5` it is sufficient to use the hardware double precision floating-point. The more complicated `cyclic6` requires software emulated floating-point with at least 96 bits in mantissa.

The experiments presented here are demonstrative examples which allow a geometrical interpretation and an easy analysis of the flow of computation.

Our algorithm for the numerical construction of Gröbner bases and its implementation do not require the input to be a zero-dimensional system. However, the numerical computation of a (generally extended) Gröbner basis has not been considered as a goal in its own right in our work but as a major component in the numerical determination of zeros of a polynomial system. For this reason, all presented examples are zero-dimensional systems. For each example, the output of the algorithm is used for the computation of the zeros via a so-called eigenvector method implemented as Maple script and automatically generated as a part of the output; the attained approximations of the solutions are checked against the polynomials of the input system. The generated Maple scripts are not too sophisticated but provide a basic implementation for finding zeros by the eigenvector method which appears sufficient for testing our algorithm on a quite wide range of examples: currently the scripts do not handle the case of multiple zeros and do not implement any refinement of the attained approximations of the zeros.

We begin the chapter with a brief and rather superficial introduction to the eigenvector method for polynomial systems solving (section 1). Section 2 presents the postprocessing of a result of the algorithm described in chapter 3 and the implementation of the eigenvector method utilized in our experiments for finding solutions in more details. Section 3 presents a set of numerical examples which is indeed the main content of this chapter.

4.1 Solving polynomial systems via eigenvectors

Consider a zero-dimensional system $\mathcal{F}$ of polynomials $\mathcal{P}{=}\Bbbk[X]$ in finitely many indeterminates X over an algebraically closed field $\Bbbk$ of characteristic zero, presumably the field $\mathbb{C}$ of complex numbers. Let $I{=}\langle\mathcal{F}\rangle_{\mathcal{P}}$ be the ideal generated by $\mathcal{F}$ in $\mathcal{P}$. It is well-known that $\mathcal{F}$ and I have the same zeros $Z{=}Z(\mathcal{F})$. Let m be the number of points in Z. The eigenvector approach to polynomial systems solving exploits the $\Bbbk$-linear algebra structure of the quotient algebra $Q{=}\mathcal{P}/I$ together with the differential structure in the case of multiple zeros. A detailed description of the method can be found e.g. in [Ste93], [Ste96b], the profound study of the multiple zeros case is presented in [MS], a nice overview of the method in a slightly more general context is contained in chapter 2, §4 of [CLO98].

We do not intend to duplicate the excellent treatment of the subject presented in the aforementioned sources. Here, we provide a short overview of the method with an emphasis towards the utilization of the eigenvector approach in our implementation. For simplicity we ignore the case of multiple zeros. This is not a burdensome restriction for the subsequent description of the computational experiments since our sample implementation of the eigenvector method does not handle the case of multiple zeros anyway.

It is well-known that the quotient Q is finite dimensional as a $\Bbbk$-vector space provided that I is zero-dimensional (see e.g. theorem 6 of chapter 5, §3 of [CLO97], theorem 2.2.7 of [AL94], or theorem 6.54 of [BW93]). Assume I is radical which is equivalent to the fact that it does not have multiple zeros (see e.g. corollary 2.6 on page 143 in [CLO98]). Then the $\Bbbk$-dimension of the quotient algebra Q is the number m of points in Z (see e.g. theorem 2.10 on page 41 in [CLO98]) and therefore an arbitrary set $N{=}\{t_1,\ldots,t_m\}{\subseteq}\mathcal{P}$ of m *linearly independent modulo* I polynomials, i.e. $\sum_{j=1}^{m}\alpha_j{\cdot}t_j{\notin}I$ for any $0{\neq}(\alpha_1,\ldots,\alpha_m){\in}\Bbbk^m$, *represents* a $\Bbbk$-linear basis in Q: for any $h{\in}\mathcal{P}$ there exists the unique representation

$$h{\equiv}\sum_{j=1}^{m}\alpha(h)_j{\cdot}t_j \mod I \quad \text{with} \quad \alpha(h)_1,\ldots,\alpha(h)_m{\in}\Bbbk. \tag{$\dagger$}$$

Consider a polynomial $f{\in}\mathcal{P}$. For each $i{=}1,\ldots,m$, let

$$r_i{=}\sum_{j=1}^{m}\alpha(f{\cdot}t_i)_j{\cdot}t_j{\in}\mathcal{P} \quad \text{with} \quad \alpha(f{\cdot}t_i)_1,\ldots,\alpha(f{\cdot}t_i)_m{\in}\Bbbk$$

be the uniquely defined polynomial such that

$$f{\cdot}t_i \equiv r_i \mod I$$

provided by (†). Take any $z{\in}Z$. Then

$$(r_i - f{\cdot}t_i)(z) = \sum_{j=1}^{m}\alpha(f{\cdot}t_i)_j{\cdot}t_j(z) - f(z){\cdot}t_i(z) = 0,\ i{=}1,\ldots,m$$

or, in the matrix notation,

$$A_f{\cdot}N(z) - f(z){\cdot}N(z) = \big(A_f - f(z){\cdot}E\big){\cdot}N(z) = 0, \quad \text{with} \tag{‡}$$

E – the *identity* $m{\times}m$ matrix,

$$A_f{=}\begin{pmatrix} a(f{\cdot}t_1)_1 & \cdots & a(f{\cdot}t_1)_m \\ \vdots & a(f{\cdot}t_i)_j & \vdots \\ a(f{\cdot}t_m)_1 & \cdots & a(f{\cdot}t_m)_m \end{pmatrix} \in \Bbbk^{m\times m},\ N(z){=}\begin{pmatrix} t_1(z) \\ \vdots \\ t_i(z) \\ \vdots \\ t_m(z) \end{pmatrix} \in \Bbbk^{m}.$$

Remark 4.1.1. The constructed correspondence

$$A : \begin{cases} \mathcal{P} \longrightarrow \Bbbk^{m\times m} \\ f \longrightarrow A_f \end{cases}$$

is a linear map as it is easy to conclude according to (†):

$$A_{p+c\cdot q} = A_p{+}c{\cdot}A_q \ \text{ for any } \ p,q{\in}\mathcal{P},\ c{\in}\Bbbk.$$

The identity (‡) is at the heart of solving zero-dimensional polynomial systems as matrix eigenvector problems. For the algorithmic construction of the solutions as well as for an easier argumentation of the validity of the construction, we follow a conventional choice of the set N as a *closed set* of terms $t_1,\ldots,t_m$ where the closedness denotes the condition that any divisor of an element of N is in N. The existence of such a basis is known from the theory of Gröbner bases: the set of all terms minimal modulo a Gröbner basis of I w.r.t. some term order is closed and represents a basis of the $\Bbbk$-linear space Q (see e.g. [Buc83]). Take a polynomial $f{\in}\mathcal{P}$ such that the values $f(z)$ are all distinct for $z{\in}Z$.[1] Assume that the system F is consistent: $Z{\neq}\emptyset$. Then $1{\notin}I$ and it is immediate by the closedness condition that $1{\in}N$. W.l.o.g. assume $t_1{=}1$. Then $N(z){\neq}0$ due to $t_1(z){=}1(z){=}1{\neq}0$ and, according to (‡), $N(z)$ is an eigenvector of A_f associated with the eigenvalue $f(z)$, $z{\in}Z$. The values of f on Z are all distinct by assumption and therefore, as well-known from linear algebra, the quotient algebra Q considered as a $\Bbbk$-linear vector space is isomorphic to the direct sum of 1-dimensional eigenspaces of A_f spanned by the linearly independent vectors $N(z)$, $z{\in}Z$, and then it is easy to see that there is a one to one correspondence between the zeros Z and a set $V{\subseteq}\Bbbk^{\times m}$ of m linearly independent eigenvectors of A_f; in particular, for any $v{=}(\alpha(v)_1,\ldots,\alpha(v)_m){\in}V$, there is one and only one $z{\in}Z$ such that $v{=}\alpha(v)_1{\cdot}N(z)$ (note that $t_1(z){=}1$ and hence $a(v)_1{\neq}0$ since $v{\neq}0$):

$$W := \left\{ \frac{v}{\alpha(v)_1} \mid v{\in}V \right\} = \big\{ N(z) \mid z{\in}Z \big\}$$

i.e. the m linearly independent eigenvectors of A_f normalized to make their first components equal to 1 are precisely the results of evaluation $N(z)$ of the basis terms at the solutions $z{\in}Z$; the correspondence between the elements of W and Z is one to one.

For each $w{\in}W$, the coordinates in the variables $X{\cap}N$ of the corresponding zero $z{\in}Z$ can readily be read off the respective components of w. The remaining coordinates can be attained as follows. Let $x{\in}X{\setminus}N$. Consider the (uniquely defined) polynomial $r_x{\in}\mathcal{P}$ provided by (†) such that r_x is a linear combination of the terms N and $x = r_x{+}h_x$ with $h_x{\in}I$. The value $x(z)$ of the x-coordinate of the solution z can be attained by the evaluation

$$x(z) = (r_x{+}h_x)(z) = r_x(z){+}h_x(z) = r_x(z){+}0 = r_x(z).$$

Furthermore, it is immediate by the closedness condition that N consists of terms in $X{\cap}N$: if there exists $s{\in}N$ such that $x \,|\, s$ with $x{\in}X{\setminus}N$ then, by the closedness of N, $x{\in}N$. Consequently $r_x{\in}\Bbbk[X{\cap}N]$ and hence the evaluation

[1] A typical choice is

$$f{=}\sum_{x\in X} \alpha_x{\cdot}x \quad \text{with randomly selected} \quad 0{\neq}\alpha_x{\in}\Bbbk,\ x{\in}X$$

which ensures with a small probability of failure that the values $f(z)$ are all distinct for $z{\in}Z$.

of r_x requires only the $X{\cap}N$-coordinates of z which are already known from w.

The outlined eigenvector approach for finding all m solutions Z of $\mathcal{F}$ can be summarized as follows.

- Choose N – a closed set of terms which represent a $\Bbbk$-linear space basis of the quotient $\mathcal{P}/\langle\mathcal{F}\rangle_{\mathcal{P}}$.
- Choose a polynomial $f{\in}\mathcal{P}$ such that the values $f(z)$ are all distinct, $z{\in}Z$.
- Construct the $m{\times}m$ matrix A_f as described above.
- For each $x{\in}X\backslash N$ take the (uniquely defined) polynomial $r_x{\in}\Bbbk[X{\cap}N]$ such that $x \equiv r_x \mod \langle F\rangle_{\mathcal{P}}$.
- Compute the set W of m linearly independent eigenvectors of A_f with the value of the component corresponding to $1{\in}N$ normalized to $1{\in}\Bbbk$.

for $w{\in}W$ **do**

- the $X{\cap}N$-coordinates of the solution $z{\in}Z$ corresponding to w are read off the respective components of w;
- for each $x{\in}X\backslash N$ the value $x(z)$ of the x-coordinate of the solution z is computed as $r_x(z)$: evaluating the respective r_x by substitution of the (already known) $X{\cap}N$-coordinates of the solution into it.

end for

4.2 Solving polynomial systems in our implementation

As described in chapter 3, a major step in our approach to the numerical solution of a system $\mathcal{F}$ of polynomials $P{=}\Bbbk[X]$ equipped with an admissible term order $<_T$ on the terms $\mathbb{T}{=}X^*$ is the translation of the system $\mathcal{F}$ into the system F in the algebra

$$\mathcal{P}(e) = \Bbbk[X,e] \cong {}^{\Bbbk[X,y]}\big/_{\left\langle y^{\mathcal{B}}\right\rangle}, \quad \text{with some } \ \mathcal{B}{>}0$$

of extended polynomials (defined in section 2.3 on page 25) with the specified extension of the term order $<_T$ on $\mathbb{T}$ to the extended term order $<$ (definition 2.3.9 on page 37) on the extended terms $\mathbb{T}(e)$ and the computation of a Gröbner basis G of F in $\mathcal{P}(e)$. Throughout this section we fix the extended term order $<$ which is assumed to be either $\underset{\mathbb{T}_e}{<}$ (page 90) or $\underset{e\mathbb{T}}{<}$ (page 168). Let G be a Gröbner basis in $\mathcal{P}(e)$ w.r.t. $<$ such that $G{\neq}\emptyset$ and $1{\notin}\ell\mathrm{t}(G)$[2] presumably attained by the algorithm described in chapter 3 with the input

[2]This convenient for the subsequent reasoning and not burdensome restriction excludes the case of a trivial ideal $I{=}\langle G\rangle_{\mathcal{P}(e)}$. The condition $\emptyset{\neq}G({\subseteq}\ I\backslash\{0\})$ is clearly equivalent to

system $\mathcal{F}\in\mathcal{P}$. Also assume that $\deg_e\bigl(\ell\mathrm{t}(g)\bigr)=0$ for any $g\in G$ (i.e. the leading terms of G are essentially the terms $\mathbb{T}$ in the main variables X) and $\ell\mathrm{t}(G)$ contains a pure power x^{d_x}, $d_x\in\mathbb{N}$, for each $x\in X$. Let $I=\langle G\rangle_{\mathcal{P}(e)}$ be the ideal generated by G in $\mathcal{P}(e)$. These restrictions reflect the extent of applicability provided in our current implementation.

The Gröbner basis in $\mathcal{P}(e)$ is the subject for the final postprocessing translating the result back to the algebra $\mathcal{P}$ as described below with the subsequent application of the eigenvector method to compute the solutions as described in section 4.1.
Denote by

$$\begin{aligned} N(e) &= \bigl\{s\in\mathbb{T}(e) \bigm| \ell\mathrm{t}(g)\nmid s \ \text{ for any } \ g\in G\bigr\} \subseteq \mathbb{T}(e) - \text{the minimal extended} \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{terms modulo } G, \\ N &= N(e)\cap\mathbb{T} \subseteq \mathbb{T} - \text{the minimal terms in the main variables } X, \\ NX &= \bigl\{t\cdot x \bigm| t\in N,\ x\in X\bigr\}\cup\{1\} \subseteq \mathbb{T}, \\ B &= NX\backslash N \subseteq \mathbb{T}. \end{aligned}$$

It is immediate by construction that $N(e)$ is a closed set in $\mathbb{T}(e)$, N is a closed set in $\mathbb{T}$, $1\in N$ due to the assumption $1\notin\ell\mathrm{t}(G)$, $B\cap N=\emptyset$ and $B\sqcup N=NX$. Furthermore, the assumption

$$x^{d_x}\in\ell\mathrm{t}(G),\ d_x\in\mathbb{N}, \ \text{ for each } \ x\in X$$

implies that N is a finite set (cf. theorem 6 of chapter 5, §3 in [CLO97], theorem 2.2.7 in [AL94], or theorem 6.54 in [BW93]). Denote by

$$m - \text{the number of elements in } N.$$

For each term $b\in B$, consider the polynomial

$$g_b = b - r_b \in I$$

where r_b is the *canonical form* [3] of b modulo G – the (uniquely defined) linear combination of terms $N(e)$ such that $b\equiv r_b \mod I$. Note that r_b, $b\in B$, is a

$I\neq 0$. It is also easy to see that $1\notin\ell\mathrm{t}(G)$ is equivalent to $I\neq\mathcal{P}(e)$. Indeed, if $1\notin\ell\mathrm{t}(G)$ then 1 is minimal modulo G and hence $1\notin I$ by part e) of theorem 2.3.2. Conversely, let $1=\ell\mathrm{t}(g)$ with $g\in G$. Then $\mathrm{supp}\,(g)\setminus\{1\}\subseteq\mathbb{T}(e)\backslash\mathbb{T}$: the restriction of (the extended term order) $<$ on $\mathbb{T}$ coincides with the admissible term order $<_T$, therefore $s\in\bigl(\mathrm{supp}\,(g)\setminus\{1\}\bigr)\cap\mathbb{T}$ implies $1<s$ and, in this case, 1 can not be the leading term of g. The conclusion $I=\mathcal{P}(e)$ is now immediate by the observation that any polynomial of the form $1-e\cdot h$, $h\in\mathcal{P}(e)$, is invertible in $\mathcal{P}(e)$: the inverse is given by $\sum_{k=0}^{\mathcal{B}-1}(e\cdot h)^k$.

[3] Given an ideal I in $\mathcal{P}(e)$ w.r.t. some extended term order, the canonical form $r_f\in\mathcal{P}(e)$ of a polynomial $f\in\mathcal{P}(e)$ is the uniquely defined linear combination of terms minimal mod-

linear combination of extended terms $N(e)$ strictly $<$-smaller than b provided that r_b can be computed from b and G by the total reduction algorithm (page 204) and therefore, by the linear independence of extended terms,

$$g_b{\neq}0, \quad \ell\mathrm{t}(g_b){=}b, \quad \ell\mathrm{c}(g_b){=}1.$$

Denote

$$\Gamma = \left\{g_b \;\middle|\; b{\in}B\right\}.$$

We claim that Γ is a Gröbner basis of I in $\mathcal{P}(e)$. Indeed, consider the set $G'{\subseteq}G$ attained from G by excluding all elements $g{\in}G$ such that $\ell\mathrm{t}(g')\,|\,\ell\mathrm{t}(g)$ for some $g'{\in}G\setminus\{g\}$. According to lemma 2.3.39 on page 85, G' is a so-called minimal Gröbner basis of I, i.e. G' is a Gröbner basis of I and the leading terms of different elements of G' do not divide each other (cf. definition 2.3.26 on page 84). As discussed above, $\Gamma{\subseteq}I\setminus\{0\}$; therefore, by definition 2.3.17 on page 50 of a Gröbner basis in $\mathcal{P}(e)$, Γ is Gröbner basis of I if $\ell\mathrm{t}(G'){\subseteq}\ell\mathrm{t}(\Gamma)$ since G' is a Gröbner basis of I: In this case

$$\left\langle \ell\mathrm{t}(I)\right\rangle = \left\langle \ell\mathrm{t}(G')\right\rangle \subseteq \left\langle \ell\mathrm{t}(\Gamma)\right\rangle \subseteq \left\langle \ell\mathrm{t}(I)\right\rangle \text{ implies } \left\langle \ell\mathrm{t}(\Gamma)\right\rangle = \left\langle \ell\mathrm{t}(I)\right\rangle.$$

To show the inclusion $\ell\mathrm{t}(G'){\subseteq}\ell\mathrm{t}(\Gamma)$ note that $\ell\mathrm{t}(\Gamma){=}B$. Consider $v{=}\ell\mathrm{t}(g)$ with some $g{\in}G'$. Then $v{\in}\mathbb{T}\setminus\{1\}$ because $\ell\mathrm{t}(G'){\subseteq}\ell\mathrm{t}(G)$ by construction and $\ell\mathrm{t}(G){\subseteq}\mathbb{T}\setminus\{1\}$ by assumption. Hence there exists $x{\in}X$ such that $x\,|\,v$. Let $t{=}v/x$. If t is reducible modulo G then t is reducible modulo G' i.e. $\ell\mathrm{t}(g')\,|\,t$ with $g'{\in}G'$ and in this case $\ell\mathrm{t}(g')$ is a proper divisor of $\ell\mathrm{t}(g){=}v$ which contradicts the fact that G' is a minimal Gröbner basis by construction. Therefore t is minimal modulo G i.e. $t{\in}N$ and hence $v{=}t{\cdot}x{\in}B$ by construction.

Let $a{\in}\Bbbk$. Consider the partial evaluation E_a at $e{=}a$ defined as

$$E_a : \left\{ \begin{array}{rcl} \mathcal{P}(e) & \longrightarrow & \mathcal{P} \\ f(X,e) & \longrightarrow & f(X,a), \end{array} \right.$$

denote

$$\begin{aligned} &g_b^{(a)} = E_a(g_b),\ b{\in}B, \\ &\Gamma^{(a)} = E_a(\Gamma)\ \big(= \{g_b^{(a)} \mid b{\in}B\}\big). \end{aligned}$$

ulo I (i.e. the terms not contained in the ideal of leading terms of I), cf. lemma 1 in [MT], it can be computed e.g. by the total reduction algorithm modulo a Gröbner basis of I as described on page 204 which also provides the mathematical existence statement. The precise existence and uniqueness statements are easy to attain by complete analogue with the classical theory of Gröbner bases (see e.g. chapter 2, §6 in [CLO97], chapter 1, §6 in [AL94], or chapter 5, §2 in [BW93]).

Note that E_a is a $\Bbbk$-linear map but generally not a homomorphism of rings unless $a{=}0$. The important properties of the map E_a on Γ are summarized in the following lemma.

Lemma 4.2.1.

a) For any $b{\in}B$ the image $g_b^{(a)}$ of g_b under E_a has the form

$$g_b^{(a)} = b{-}r_b^{(a)} \quad \text{where} \quad r_b^{(a)}{\in}\mathcal{P} \quad \text{is a linear combination of } N,$$
$$g_b^{(a)}{\neq}0, \quad b{\in}\operatorname{supp}\left(g_b^{(a)}\right),$$
$$\text{the coefficient } \mathfrak{C}_b\left(g_b^{(a)}\right) \text{ at } b \text{ in } g_b^{(a)} \text{ is } 1.$$

b) There is a one to one correspondence between B and $\Gamma^{(a)}$: for each $b{\in}B$ there is one and only one $g{\in}\Gamma^{(a)}$ such that $b{\in}\operatorname{supp}(g)$.

Proof.

a) Take any $b{\in}B({\subseteq}\ \mathbb{T})$. Note that $E_a(b){=}b$ by definition of E_a due to $b{\in}\mathbb{T}$. Then

$$g_b^{(a)} = E_a(g_b) = E_a(b{-}r_b) = E_a(b){-}E_a(r_b) = b{-}r_b^{(a)} \quad \text{with} \quad r_b^{(a)} = E_a(r_b).$$

We claim that $r_b^{(a)}$ is a linear combination of the terms N. Indeed, by construction, $\operatorname{supp}(r_b){\subseteq}N(e)$ and therefore, by definition of E_a and the linear independence of $\mathbb{T}$, $\operatorname{supp}\left(r_b^{(a)}\right){\subseteq}\mathrm{T}\big(N(e)\big)$ (for $s{\in}\mathbb{T}(e)$, $\mathrm{T}(s){=}t$ denotes the *main part* of s in the unique decomposition $s{=}t{\cdot}e^d$ with $t{\in}\mathbb{T}$ and $0{\le}d{<}\mathcal{B}$, see lemma 2.3.1 on page 29). Furthermore, it is easy to see that $\mathrm{T}\big(N(e)\big){=}N$ – the inclusion $N{\subseteq}\mathrm{T}\big(N(e)\big)$ is trivial due to $N{\subseteq}N(e)$ by construction and $\mathrm{T}(N){=}N$ provided that $N{\subseteq}\mathbb{T}$, the converse inclusion is clear by the transitivity of divisibility of extended terms – and therefore $\operatorname{supp}\big(E_a(r_b)\big){\subseteq}N$. Hence, by the linear independence of $\mathbb{T}$ and $N{\cap}B{=}\emptyset$ by construction,

$$g_b^{(a)}{\neq}0, \quad b{\in}\operatorname{supp}\left(g_b^{(a)}\right), \quad \mathfrak{C}_b\left(g_b^{(a)}\right){=}1.$$

b) The desired correspondence is given by

$$\varphi : \left\{ \begin{array}{ccc} B & \longrightarrow & \Gamma^{(a)} \\ b & \longrightarrow & g_b^{(a)}. \end{array} \right.$$

Indeed, φ is surjective by construction and, by part a), $b \in \operatorname{supp}\left(g_b^{(a)}\right)$, $b \in B$. The injectivity of φ is also easy to see as follows. Let $g_{b_1}^{(a)} = g_{b_2}^{(a)}$ for some $b_1, b_2 \in B$. By part a)

$$b_1 - r_{b_1}^{(a)} = g_{b_1}^{(a)} = g_{b_2}^{(a)} = b_2 - r_{b_2}^{(a)}$$

where $r_{b_1}^{(a)}, r_{b_2}^{(a)}$ are linear combinations of N. Then

$$b_2 - b_1 = r_{b_2}^{(a)} - r_{b_1}^{(a)} \text{ implies } b_1 = b_2$$

by the linear independence of terms $\mathbb{T}$ and $N \cap B = \emptyset$ by construction.
$\square$

The particular structure of the elements of $\Gamma^{(a)}$ revealed in the lemma above is the key to find the solutions $Z^{(a)}$ of the ideal $I^{(a)} = \langle \Gamma^{(a)} \rangle_{\mathcal{P}}$ using the eigenvector approach, described in section 4.1, under the *formal assumption* that N represents a $\Bbbk$−linear basis of $Q^{(a)} = \mathcal{P}/I^{(a)}$. We also assume that $I^{(a)}$ does not have multiple solutions and then, according to section 4.1, $Z^{(a)}$ consists of m (the number of elements in N) points. Recall that N is a closed set in $\mathbb{T}$ with $1 \in N$ by construction and hence all m solutions can be attained from the eigenvectors of the matrix A_f constructed as described on page 272, where $f \in \mathcal{P}$ is an arbitrary polynomial such that the values $f(z)$ are all distinct, $z \in Z^{(a)}$. We take

$$f = \sum_{x \in X} \alpha_x \cdot x \ \text{ with randomly selected } \ 0 \neq \alpha_x \in \Bbbk, \ x \in X$$

which ensures that the values $f(z)$ are all distinct for $z \in Z$ with a small probability of failure. According to remark 4.1.1 on page 272

$$A_f = \sum_{x \in X} \alpha_x \cdot A_x$$

where the matrices A_x, $x \in X$, usually referred as *multiplication matrices* of $Q^{(a)}$, are formed of the rows of coefficients $\big(\alpha(v)_t \mid t \in N\big) \in \Bbbk^m$ such that (cf. equation (†) on page 271)

$$v \equiv \sum_{t \in N} \alpha(v)_t \cdot t \mod I^{(a)}, \ v \in NX \ (= B \sqcup N).$$

For each $v \in B$ the coefficients are taken of the respective polynomial $g_b^{(a)}$ provided by the latter lemma and the *formal assumption* that N is a $\Bbbk$-linear space of $Q^{(a)}$. For each $v \in N$ naturally $\alpha(v)_v = 1$ and $\alpha(v)_t = 0$ for all $v \neq t \in N$.

To compute the actual solutions from the normalized linear independent eigenvectors (page 273) W of the matrix A_f we slightly refine the general construction presented at the end of section 4.1. Recall (see page 273) that A_f must have precisely m eigenspaces of dimension 1 provided by the assumption that all solutions $I^{(a)}$ are simple[4]; in particular, W consists of m eigenvectors. Consider $w \in W$. The coordinates in $X \cap N$ of the solution z corresponding to w are read from the respective coordinates of w. According to the discussion on page 273 the closedness of N implies that N consists of terms in the variables $X \cap N$ only. Furthermore, it is immediate by construction of B that $X \setminus N \subseteq B$ and hence, by lemma 4.2.1, for each $x \in X \setminus N$ there exists the unique polynomial $x - r_x \in \Gamma^{(a)}$ where r_x is a linear combination of N and therefore r_x is a polynomial in the variables $X \cap N$. Consequently the missing coordinates in $X \setminus N$ of the solution z can be computed by substituting the already known coordinates in $X \cap N$ to r_x for each $x \in X \setminus N$.

The described transformation of the Gröbner basis $G \subseteq \mathcal{P}(e)$ to $\Gamma^{(a)} \subseteq \mathcal{P}$ with the subsequent application of the eigenvector method to compute the solutions of $I^{(a)}$ can be summarized as follows.

if

$G = \emptyset$ or

$1 \in \ell\mathrm{t}(G)$ or

$\deg_e\big(\ell\mathrm{t}(g)\big) > 0$ for some $g \in G$ or

$\ell\mathrm{t}(G)$ does not contain a pure power of x for some $x \in X$

then

report failure

end if

$N := \big\{ s \in \mathbb{T} \;\big|\; \ell\mathrm{t}(g) \nmid s \;\text{ for any }\; g \in G \big\}$

$NX := \big\{ t \cdot x \;\big|\; t \in N,\; x \in X \big\} \cup \{1\}$

$B := NX \setminus N$

$\Gamma := \emptyset$

for $b \in B$ **do**

$r :=$ the canonical form of b modulo G – the uniquely defined linear combination of terms $\mathbb{T}(e)$ minimal modulo G such that $b \equiv r \bmod \langle G \rangle_{\mathcal{P}(e)}$ – computed by the total reduction algorithm described on page 204.

$\Gamma := \Gamma \cup \{r\}$

end for

[4] If however the utilized matrix eigenvector solver (our current implementation uses the Maple `eigenvector` routine) reports the existence of an eignespace with a higher dimension, this indicates a multiple solution (see [MS], [Moe93]) which is not currently handled in our implementation and this case is not considered.

Choose $a \in \Bbbk$ {a=1 in the current implementation}
$\Gamma^{(a)} := \{g(X,a) \mid g(X,e) \in \Gamma\}$
for $x \in X$ **do**
construct the multiplication matrix A_x from the coefficients of the polynomials in $\Gamma^{(a)}$.
end for
$A := \sum_{x \in X} \alpha_x \cdot A_x$ with a randomly selected $0 \neq \alpha_x \in \Bbbk$ for each $x \in X$
if there is an eigenspace of A with dimension >1 **then**
report failure {multiple zero has been detected}
end if
$W :=$ the normalized linearly independent eigenvectors of A as described on page 273.
$Z := \emptyset$
for $w \in W$ **do**
read the $X \cap N$-coordinates of the solution z of $\Gamma^{(a)}$ corresponding to w from the respective components of w
for $x \in X \setminus N$ **do**
take (the uniquely defined) polynomial $x - r \in \Gamma^{(a)}$ where r is a polynomial in the variables $X \cap N$ and compute the value $x(z)$ of the x-coordinate of the solution z evaluating r at the already known $X \cap N$-coordinates of the solution.
end for
$Z := Z \cup \{z\}$
end for

The listed procedure contains however a severe pitfall which is totally unacceptable from the abstract and computer algebra point of view but somewhat admissible in the numerical context from a practical point of view. Namely, the assumption that N represents a $\Bbbk$-linear space basis in the quotient $\mathcal{P}/\langle \Gamma^{(a)} \rangle$ is indeed unrealistic: typically the system $\Gamma^{(a)}$ is *inconsistent* even if we started with a consistent system $\mathcal{F}$ doing all computations in the exact arithmetic (unless a=0 as discussed below)[5]! It appears natural to interpret the attained m solutions as the exact solutions of a system in some close neighborhood of $\Gamma^{(a)}$ (see [Ste99]) following a common practice regarding a result of numerical computation utilized for an algebraic problem (see

[5]This remark appear to project a pessimistic picture of the computational value of our algorithm but the situation is not so dramatic from a more practical point of view: the examples of computation described in the remaining part of the chapter show a reasonable behavior of our implementation concerning polynomial systems solving: the attained solutions produce small residuals being substituted into the polynomials of the initial system. Moreover, if the accuracy of the attained solutions is found insufficient it can always be improved by a sequence of Newton steps.

e.g. [KL96], [HKL99], [HWSZ00]).

The existence of such a system with m solutions can be argued as follows. The system G is a Gröbner basis in $\mathcal{P}(e)$ w.r.t. the extended term order $<$ on $\mathbb{T}(e)$ by assumption. Then, according to the results of chapter 2, the result $\Gamma^{(0)}$ of partial evaluation of the Gröbner basis $\Gamma \supseteq G$ constructed as described above, is a Gröbner basis in $\mathcal{P}$ w.r.t. the admissible term order $<_T$ on $\mathbb{T}$ (theorem 2.5.6 on page 166 when the order $<$ is $\underset{\mathbb{T}_e}{<}$, theorem 2.6.2 on page 186 for the case when $<$ is $\underset{e\mathbb{T}}{<}$). Also it is easy to see by lemma 4.2.1 that N is the set of all terms $\mathbb{T}$ minimal modulo $\Gamma^{(0)}$. Then, as it is well-known (see e.g. [Buc83]), N represents a $\Bbbk$-linear space basis in the quotient $Q^{(0)}{=}\mathcal{P}/\left\langle \Gamma^{(0)} \right\rangle$ and therefore, as we have seen in section 4.1 for a particular case of simple zeros and as it is stated e.g. in [Ste96b] or [MS] in a more general setup which also covers the case of multiple zeros, $\Gamma^{(0)}$ has precisely $\dim_C Q^{(0)}{=}m$ (the number of elements in N) solutions counting multiplicities[6]. As it is pointed out in [Ste97], the coefficients of a polynomial system are smooth functions on its zero set. Therefore, N also represents a $\Bbbk$-linear space basis in the quotient $\mathcal{P}/\left\langle \widehat{\Gamma^{(0)}} \right\rangle$ for

$$\widehat{\Gamma^{(0)}} = \left\{ g + \sum_{t \in N} \beta(g)_t \mid g{\in}\Gamma^{(0)},\ \beta(g)_t{\in}\Bbbk \text{ are small for all } t{\in}N \right\}$$

which can be seen as a small perturbation of $\Gamma^{(0)}$ *corresponding* to a small change in the zero set of $\Gamma^{(0)}$ according to theorem 4.1 in [Ste97] based on the interpretation of $\Gamma^{(0)}$ as the interpolation on its zero set. Hence all m solutions of $\widehat{\Gamma^{(0)}}$ are close to the solutions of $\Gamma^{(0)}$ and can be computed via the multiplication matrices constructed from the elements of $\widehat{\Gamma^{(0)}}$. The latter consideration is the key for the analysis of the situation when $a{\neq}0$. Take

$$g = \sum_{s \in \operatorname{supp}(g)} \gamma_s{\cdot}s \in \Gamma,\ \ \gamma_s{\in}\Bbbk \quad \text{for} \quad s{\in}\operatorname{supp}(g)\,.$$

[6] In view of this fact we could use the system $\Gamma^{(0)}$ for the computation of zeros. However, to avoid an artificial extra deviation of the computed solutions from the solutions of the initial system $\mathcal{F}$ it seems more natural to evaluate at $a{=}1$ if $G{\subseteq}\Gamma$ is computed by the algorithm presented in chapter 3 as it is always the case in our implementation: a positive e-degree of a term in a polynomial in G is indeed a *tag* assigned at some point of execution of the algorithm from chapter 3 (as an indication that the coefficient at the term in the polynomial is small), this tags also propagate to $\Gamma{\supseteq}G$ and the evaluation at $a{=}1$ can be considered as just removing the tags resulting in a system in $\mathcal{P}$.

Denote

$$S(g)_0 = \mathrm{supp}\,(g) \cap \mathbb{T} \quad \text{– the terms in } g \text{ with a zero } e\text{-degree,}$$
$$S(g)_+ = \mathrm{supp}\,(g) \setminus \mathbb{T} \quad \text{– the terms in } g \text{ with a positive } e\text{-degree.}$$

Let $h_g = \sum_{s \in S(g)_+} \gamma_s \cdot s$. Consider $g^{(0)} = E_0(g)$, $g^{(a)} = E_a(g)$ – the images of g under the partial evaluation at $e=0, a$. Then it is immediate that

$$g^{(0)} = \sum_{s \in S(g)_0} \gamma_s \cdot s,$$
$$g^{(a)} = g^{(0)} + h_g^{(a)} \quad \text{with} \quad h_g^{(a)} = E_a(h_g) = \sum_{s \in S(g)_+} \Big(\gamma_s \cdot a^{\deg_e(s)} \Big) \cdot \mathrm{T}(s),$$

where the coefficients of $h_g^{(a)}$ are all small if a is chosen sufficiently small. In this case,

$$\Gamma^{(a)} = \{ g^{(a)} \mid g \in \Gamma \}$$

can be considered as a system in a close neighborhood of $\Gamma^{(0)}$[7]. However the chance that $\Gamma^{(a)}$ is a system of the form $\widehat{\Gamma^{(0)}}$ with m solutions is very small – the system $\Gamma^{(0)}$ is typically *overdetermined* and hence an arbitrary perturbation of $\Gamma^{(0)}$ would almost inevitably result in an inconsistent system.

From a practical point of view of numerical computing of solutions from the multiplication matrices A_x, $x \in X$, formed from the elements of $\Gamma^{(a)}$, this is, however incidental according to our experiments and our current implementation does not include any mechanism to handle the latter issue. A better implementation could employ certain mechanism based e.g. on the results of [Mou99] where it is shown that N represents a $\Bbbk$-linear space basis in $\mathcal{P}/\langle \Gamma^{(a)} \rangle$ (and hence $\Gamma^{(a)}$ has m – the number of elements in N – solutions) if and only if the multiplication matrices A_x, $x \in X$, formed from the elements of $\Gamma^{(a)}$ commute. Consequently a possible improvement of our implementation could introduce certain commutativity threshold and employ e.g. a combination of a local linearization of the commutativity conditions on the multiplication matrices (for example, expressed in terms of their coefficients) with perhaps several Newton steps to refine the commutativity if

[7] Furthermore, in our implementation the value $a=1$ can generally be regarded as sufficiently small! Indeed, for any $g \in \Gamma$, all terms of h_g have a positive e-degree and therefore the coefficients of h_g are likely to be small if the Gröbner basis $G \subset \Gamma$ is computed by the algorithm described in chapter 3 – a positive e-degree of $s \in \mathrm{supp}\,(h_g)$ is assigned during the computation of G as an indicator that the coefficient γ_s at s in h_g is small. We use this fact as a shortcut for the selection of a: in the current implementation the partial evaluation $e=a$ is always performed at $a=1$ (a more advanced implementation could employ certain play around with the value of a based e.g. on some consistency measure of $\Gamma^{(a)}$).

its deficiency exceeds the threshold. It would be also desirable to have the results towards the sensitivity of the eigenvector problem for the multiplication matrices to the deficiency of their commutativity as well as a relation between the solutions attained according to our approach and the solutions of the initial system $\mathcal{F}$ as it could provide some a-priory information on the approximation quality to be expected but we can not provide such an analysis at present time inasmuch as it requires further investigation.

4.3 Numerical examples

In this section we present discuss and analyze a series of numerical examples carried with our implementation of the algorithm for the numerical computation of Gröbner bases (described in chapter 3) combined with a simple implementation of the eigenvector method for solving a system of polynomial equations presented as a potentially extended Gröbner basis as discussed in the previous two sections. The implementation of the algorithm is done in `C++` (≈250Kb of sources) with the `STL` library for container classes and specific shared data storage objects with reference counting to suppress redundant data copying.

Each example below is constructed in such a way that the genuine Gröbner basis w.r.t. the specified term order is not appropriate for the numerical determination of the solutions because it provides an extremely ill-conditioned representation of the ideal of the original polynomial system; therefore one needs an extended Gröbner basis – a slightly perturbed genuine Gröbner basis of a system in a close neighborhood (w.r.t. a norm on coefficients) of the original system – for the numerical determination of the solutions.

The description of each example generally involves a discussion of the genuine Gröbner basis and an analysis of the flow of computation of an extended Gröbner basis with our implementation, illustrated with an automatically generated report which describes intermediate steps taken by the algorithm. To follow the description it might be helpful to consult the introductory part of chapter 3 for a general overview of the algorithm.

The first example emphasizes the mathematical origin of the phenomena which appears when a classical Gröbner basis is not an appropriate tool for the numerical computation of solutions. In the next examples, we gradually restrict the presentation of the mathematical aspects but concentrate on the flow of computation.

4.3.1 Hyperbola and ellipse

Consider the system $F{=}\{F_1, F_2\}$ of two conics

$$\begin{aligned} F_1 &= 0.1078210{-}0.1141510\,x + 0.1266110\,y \\ &\qquad\qquad + 3.589658\,xy + 0.9643230\,y^2 + 2.267448\,x^2, \\ F_2 &= -4.657648{+}3.310486\,x + 2.271471\,y \\ &\qquad\qquad + 2.688612\,xy + 3.356959\,y^2 + 1.698297\,x^2 \end{aligned}$$

as a system in $\mathbb{C}[x, y]$ equipped with the *total degree (reverse) lexicographical order* induced by $y{<}x$. The genuine totally reduced Gröbner basis of F (displayed to the decimal approximation with 9 digits) is

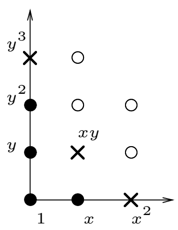
$$A = \{A_1, A_2, A_3\}$$

with

$$\begin{aligned} A_1 &= -0.00000096530604\,\underline{xy} + 0.32933628\,y^2 \\ &\qquad\qquad + 0.424498002\,x + 0.272080071\,y - 0.592300613, \\ A_2 &= 0.00000054060173\,\underline{x^2} + 0.291990057\,y^2 \\ &\qquad\qquad + 0.376360264\,x + 0.241226453\,y - 0.525134204, \\ A_3 &= 0.000000748907419\,\underline{y^3} + 0.32933537\,y^2 \\ &\qquad\qquad + 0.424496032\,x + 0.272077461\,y - 0.592297865 \end{aligned}$$

and the leading terms $\{xy,\ x^2,\ y^3\}$ underlined in the polynomials above. The leading coefficients of the elements of A are all small, with the relative order of magnitude 10^{-6}, and therefore A is poorly suited for the numerical computation of zeros of F by whatever method. Below we show that A is indeed not an appropriate representation of the ideal $\langle F\rangle$ for any numerical computation due to an excessive sensitivity of the above Gröbner basis to small changes in the coefficients of F.

Note that the system F is a complete intersection case; therefore, as it is well-known, the solutions are smooth functions of the coefficients of the system and the number of solutions (counting multiplicities) is a characteristic property of the ideal $\langle F\rangle$ which remains invariant under a small perturbation of F (a minute change in the coefficients). The normal set



$$N(A) = \{1, y, x, y^2\}$$

associated with A contains four terms and therefore F has four – the number of elements in the normal set – solutions. The geometric interpretation of the system F is presented on figure 4.1: The hyperbola F_1 intersects the ellipse F_2 in the four points (shown to the decimal approximation with 9 digits)

$$Z = \big\{(-0.575266032, 1.23331186),\ (-1.92671388, 1.69702747),\\ (1.28342857, -0.974159952),\ (0.493007573, -1.56790352)\big\}$$

which represent all four solutions of F. Thus the computation of the zeros of F is a perfectly well-conditioned problem: all solutions are simple, well-separated, and a small change in the coefficients of F results in a small change in the solutions (and vice versa) due to a rather acute angles of all intersections as can be seen on figure 4.1.

Following the standard approach in computer algebra to compute the solutions of a polynomial system as the solutions of a Gröbner basis of the system, we would solve F by computing the solutions of the genuine Gröbner basis A. However, despite the perfect well-conditioning of the zeros of F, the numerical computation of the zeros of A is an ill-conditioned problem since A is an ill-conditioned representation of the ideal of F: a small change in the coefficients of F and hence the corresponding small change in the zero set results in at least a gross change in the coefficients of the genuine Gröbner basis A or even in a structural change of the genuine Gröbner basis and the associated normal set. The origin of this phenomenon lies in a particular configuration of the zero set Z: as can be seen on figure 4.1, all elements of Z are very close to the parabola $P{=}0$ defined by the polynomial

$$P = 0.32933537\, y^2 + 0.424496032\, x + 0.272077461\, y - 0.592297865$$

which is a linear combination of the terms of the normal set $N(A)$: The parabola P is attained from A_3 by removing the leading monomial. This particular choice of P is not really important. The important observation is the fact that all points Z are very close to solutions of a polynomial which is a linear combination of the normal set elements.

On the other hand, as it is well-known from the theory of Gröbner bases, the normal set $N(A)$ represents a linear space basis of the quotient $\mathbb{C}[x,y]/\langle F\rangle$. Hence there exists a unique representation

$$f \equiv \sum_{t\in N(A)} \alpha_t(f)\cdot t \ \bmod \langle F\rangle \quad\text{with}\quad \alpha_t(f){\in}\mathbb{C} \ \text{ for all } \ t{\in}N. \tag{†}$$

for any polynomial $f{\in}\mathbb{C}[x,y]$, cf. equation (†) on page 271. In particular, there exists one and only one polynomial $g{\in}\langle F\rangle$ of the form

$$g = y^3 - \sum_{t\in N(A)} \alpha_t(y^3)\cdot t$$

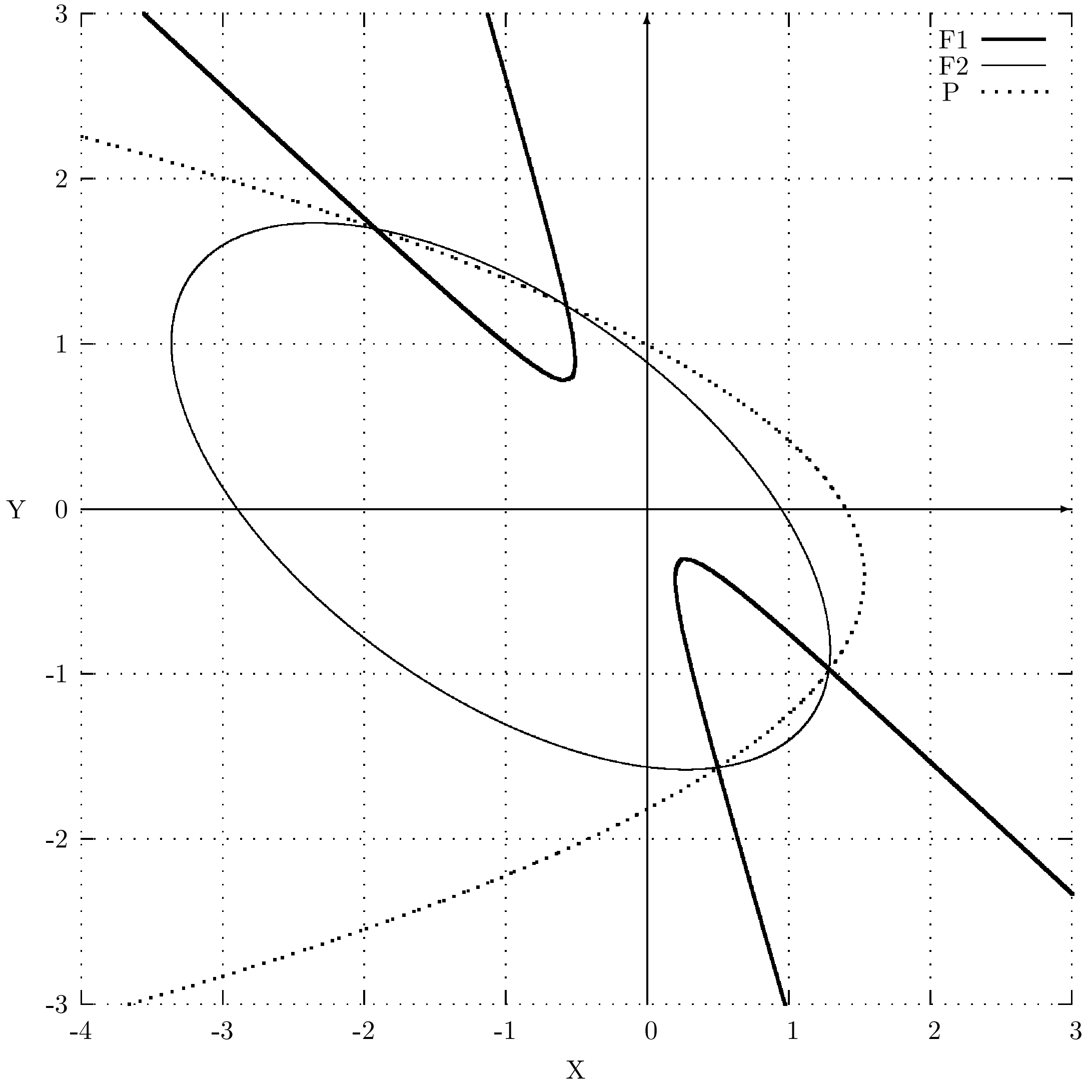


Figure 4.1: The system F from page 284.

and trivially $A_3{=}\ell c(A_3){\cdot}g$. The multivariate polynomial interpolation point of view on g is the key to reveal a high sensitivity of the coefficients $\alpha_t(y^3)$, $t{\in}N(A)$, and therefore high sensitivity of A_3 to small perturbations of F. Consider the system of linear equations

$$y^3(z) = \sum_{t\in N(A)(z)} \alpha_t{\cdot}t(z),\ \ z{\in}Z,$$

with indeterminates $\alpha_t{\in}\mathbb{C}$ attained by the evaluation of y^3 and the normal set terms at the solutions of F. The $4{\times}4$ matrix of the system is

$$N(A)(Z) = \Big[t(z)\Big] \ \text{ with } \ t{\in}N(A),\ z{\in}Z. \tag{$*$}$$

All solutions of F are simple. Therefore the ideal $\langle F\rangle$ is radical and hence, by Hilbert's Nullstellensatz, a polynomial $f{\in}\mathbb{C}[x,y]$ is in the ideal $\langle F\rangle$ if and only if $f(z){=}0$ for all $z{\in}Z$. Then it is easy to see that

$$\det\Big(N(A)(Z)\Big) \neq 0$$

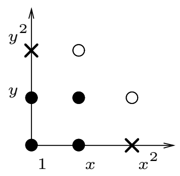
since $N(A)$ represents a linear space basis in $\mathbb{C}[x,y]/\langle F\rangle$. Hence the system $(*)$ has a unique solution which, by the uniqueness of $(\dagger)$, gives the coefficients of g as well as A_3.

However all points Z are close to the solutions of P and therefore, due to the continuous dependence of $\det\big(N(A)(Z)\big)$ on Z, the absolute value of $\det\big(N(A)(Z)\big)$ must be small which means that the vectors $N(A)(z)$, $z{\in}Z$, are *"almost linearly dependent"*; hence the solutions of the linear system $(*)$ – the coefficients $\alpha_t(y^3)$, $t{\in}N(A)$ – are very sensitive to small changes in Z and therefore to small changes in F. Consequently any numerical computation which involves $A_3{\in}A$ is better to be avoided or at least should be carried with extra care. Our argument is easily verified by direct computation:

$$N(A)(Z) = \begin{bmatrix} 1.0 & 1.23331186 & -0.575266032 & 1.52105815 \\ 1.0 & 1.69702747 & -1.92671388 & 2.87990225 \\ 1.0 & -0.974159952 & 1.28342857 & 0.948987612 \\ 1.0 & -1.56790352 & 0.493007573 & 2.45832144 \end{bmatrix},$$

$$\det\Big(N(A)(Z)\Big) = -\,0.0000261954081.$$

Furthermore there exists a small change in Z which moves the zero set on the solutions of P. Denote by $\widetilde{F}$ the perturbed F corresponding to this change in Z. Note that all points Z are well-separated, hence the ideal $\langle \widetilde{F} \rangle$ is radical. Therefore, the polynomial P lies in the ideal $\langle \widetilde{F} \rangle$ and gives a linear dependence of the terms $N(A)$ modulo $\langle \widetilde{F} \rangle$. In this case $N(A)$ can not be a normal set of $\langle \widetilde{F} \rangle$ and this is a reason why the (totally reduced) Gröbner basis of $\langle \widetilde{F} \rangle$ is structurally different from A: the basis contains only 2 polynomials, has another set $\{x^2, y^2\}$ of leading terms, and the normal set



$$N = \{1, y, x, xy\}.$$

The set N also represents a linear space basis in the quotient modulo the ideal of the original system F as can be established using the multivariate interpolation point of view; see [Ste97] for details and the general treatment of the problem. Therefore the system

$$G = \{x^2 - \mathrm{NF}_N\big(x^2\big)\,,\; y^2 - \mathrm{NF}_N\big(y^2\big)\},$$

where $\mathrm{NF}_N(x^2)$ and $\mathrm{NF}_N(y^2)$ are the uniquely defined representations of x^2 and y^2 respectively modulo $\langle F \rangle$ as linear combinations of N, is an extended Gröbner basis of F in the sense of definition 4.1 in [Ste97]: it is the genuine Gröbner basis of $\widetilde{F}$, with slightly different coefficients at the terms N. Furthermore, it is easy to show using the multivariate interpolation point of view in complete analogy with the argumentation for A_3 above that G is a well-conditioned representation of F:

$$N(Z) = \begin{bmatrix} 1.0 & 1.23331186 & -0.575266032 & -0.709482421 \\ 1.0 & 1.69702747 & -1.92671388 & -3.26968639 \\ 1.0 & -0.974159952 & 1.28342857 & -1.25026472 \\ 1.0 & -1.56790352 & 0.493007573 & -0.772988308 \end{bmatrix},$$

$$\det\Big(N(Z)\Big) = -\,8.93716387$$

Consequently, G is preferable for the numerical determination of the zero set of F: The initially well-conditioned problem should not be transformed to a representation A where a small uncertainty in the initial data (e.g. due to a limited meaningful accuracy) results in large uncertainties in the data of its representation.

Now we turn to a detailed analysis of the computation of a numerical Gröbner basis of F by our algorithm. We use the term order extension $\underset{e\mathbb{T}}{<}$, where a higher degree of e results in a smaller term, and the hardware single precision floating-point arithmetic. The result of our computation will be the extended Gröbner basis G enlarged to the full border basis for the normal set N. The precise input for the computation is the following.

```
alphabet = x,y;
termorder = tdeg;
extorder = eT;
domain = float;
ebound=2;
ideal = 0.1078210-0.1141510*x+0.1266110*y+3.589658*x*y+0.9643230*y^2+2.267448*x^2,% F1
        -4.657648+3.310486*x+2.271471*y+2.688612*x*y+3.356959*y^2+1.698297*x^2;   % F2
```

The computation begins with the normalization and translation of the input polynomials $F{=}\{F_1, F_2\}$ to the algebra of extended polynomials. The translation in this case is trivial since all coefficients of F have order one.

```
Alphabet:            x>y
Extended term order: eT(Total degree)
Coefficients domain: float
Zero test:           rel<0.0004882812
E-bound:             2
The normalized and translated input basis contains 2 polynomials:
F1=0.566862*x^2+0.8974145*x*y+0.2410807*y^2-0.02853775*x+0.03165275*y+0.02695525
F2=0.2122871*x^2+0.3360765*x*y+0.4196199*y^2+0.4138108*x+0.2839339*y-0.582206
```

The initial auto-reduction amounts to the elimination of the term x^2 from F_2 with subsequent elimination of y^2 from F_1. The elimination of x^2 creates a tiny leading coefficient -0.0000009834766 at the leading term xy in F_2; hence the term is assigned e-degree one

$$xy \to xy{\cdot}e$$

by the mechanism for trimming of e-degrees (see page 266). This results in a new leading term of F_2

$$\ell\mathrm{t}(F_2) = \max_{\substack{<\\ e\mathbb{T}}}\bigl(\mathrm{supp}\,(F_2)\bigr) = \max_{\substack{<\\ e\mathbb{T}}}\Bigl(\{xy{\cdot}e, 1, y, x, y^2\}\Bigr) = y^2$$

with order 1 coefficient 0.3293363. At this point, our algorithm essentially deviates from the normal flow of a Gröbner basis computation in order to maintain numerical stability.

```
Entering auto-reduction of the basis.
Entering symbolic auto-reduction.
The set of polynomials for symbolic reduction is
F1=0.566862*x^2+0.8974145*x*y+0.2410807*y^2-0.02853775*x+0.03165275*y+0.02695525
F2=0.2122871*x^2+0.3360765*x*y+0.4196199*y^2+0.4138108*x+0.2839339*y-0.582206
reducing supp(F1)=[1,y,x,y^2,x*y,x^2]
with F1=0.566862*x^2+0.8974145*x*y+0.2410807*y^2-0.02853775*x+0.03165275*y+0.02695525
found an auto-reducer F2 for x^2
```

```
with F2=0.2122871*x^2+0.3360765*x*y+0.4196199*y^2+0.4138108*x+0.2839339*y-0.582206
x^2 is added to the set of terms to keep
reducing supp(F2)=[1,y,x,y^2,x*y,x^2]
with F2=0.2122871*x^2+0.3360765*x*y+0.4196199*y^2+0.4138108*x+0.2839339*y-0.582206
The symbolically reduced set of polynomials is
F1=0.566862*x^2+0.8974145*x*y+0.2410807*y^2-0.02853775*x+0.03165275*y+0.02695525
F2=0.2122871*x^2+0.3360765*x*y+0.4196199*y^2+0.4138108*x+0.2839339*y-0.582206
The set of reducible terms is [<empty>]
The set of terms to keep is [x^2]
Symbolic reduction finished with 0 reducers added to the basis
Entering elimination.
The initial polynomials
F1=0.566862*x^2+0.8974145*x*y+0.2410807*y^2-0.02853775*x+0.03165275*y+0.02695525
F2=0.2122871*x^2+0.3360765*x*y+0.4196199*y^2+0.4138108*x+0.2839339*y-0.582206
The initial matrix
   |      x^2      x*y      y^2          x         y         1
---+----------------------------------------------------------
 F2| 0.212287 0.336076  0.41962   0.413811  0.283934 -0.582206
 F1| 0.566862 0.897415 0.241081 -0.0285378 0.0316527 0.0269553
Eliminating syzygy term x^2 by F1
with F1=0.566862*x^2+0.8974145*x*y+0.2410807*y^2-0.02853775*x+0.03165275*y+0.02695525
F2 := F2 + F1 * -0.3744953 --->
F2 := 0.2122871*x^2+0.3360765*x*y+0.4196199*y^2+0.4138108*x+0.2839339*y-0.582206 +
      0.566862*x^2+0.8974145*x*y+0.2410807*y^2-0.02853775*x+0.03165275*y+0.02695525 *
      -0.3744953 --->
F2=-9.834766e-07*x*y+0.3293363*y^2+0.424498*x+0.2720801*y-0.5923007 -n->
F2=-9.834766e-07*x*y+0.3293363*y^2+0.424498*x+0.2720801*y-0.5923007 -e->
F2=0.3293363*y^2+0.424498*x+0.2720801*y-0.5923007-9.834766e-07*x*y*[E]
   |      x^2      x*y      y^2          x         y         1      x*y*[E]
---+-----------------------------------------------------------------------
 F1| 0.566862 0.897415 0.241081 -0.0285378 0.0316527 0.0269553            .
 F2|        .        . 0.329336   0.424498   0.27208 -0.592301 -9.83477e-07
The matrix after elimination
   |      x^2      x*y      y^2          x         y         1      x*y*[E]
---+-----------------------------------------------------------------------
 F1| 0.566862 0.897415 0.241081 -0.0285378 0.0316527 0.0269553            .
 F2|        .        . 0.329336   0.424498   0.27208 -0.592301 -9.83477e-07
Leaving with 1 eliminations done.
Entering symbolic auto-reduction.
The set of polynomials for symbolic reduction is
F1=0.566862*x^2+0.8974145*x*y+0.2410807*y^2-0.02853775*x+0.03165275*y+0.02695525
F2=0.3293363*y^2+0.424498*x+0.2720801*y-0.5923007-9.834766e-07*x*y*[E]
reducing supp(F1)=[1,y,x,y^2,x*y,x^2]
with F1=0.566862*x^2+0.8974145*x*y+0.2410807*y^2-0.02853775*x+0.03165275*y+0.02695525
found an auto-reducer F2 for y^2
with F2=0.3293363*y^2+0.424498*x+0.2720801*y-0.5923007-9.834766e-07*x*y*[E]
y^2 is added to the set of terms to keep
reducing supp(F2)=[x*y*[E],1,y,x,y^2]
with F2=0.3293363*y^2+0.424498*x+0.2720801*y-0.5923007-9.834766e-07*x*y*[E]
The symbolically reduced set of polynomials is
F1=0.566862*x^2+0.8974145*x*y+0.2410807*y^2-0.02853775*x+0.03165275*y+0.02695525
F2=0.3293363*y^2+0.424498*x+0.2720801*y-0.5923007-9.834766e-07*x*y*[E]
The set of reducible terms is [<empty>]
The set of terms to keep is [y^2]
Symbolic reduction finished with 0 reducers added to the basis
Entering elimination.
The initial polynomials
F1=0.566862*x^2+0.8974145*x*y+0.2410807*y^2-0.02853775*x+0.03165275*y+0.02695525
F2=0.3293363*y^2+0.424498*x+0.2720801*y-0.5923007-9.834766e-07*x*y*[E]
The initial matrix
   |      x^2      x*y      y^2          x         y         1      x*y*[E]
---+-----------------------------------------------------------------------
 F1| 0.566862 0.897415 0.241081 -0.0285378 0.0316527 0.0269553            .
 F2|        .        . 0.329336   0.424498   0.27208 -0.592301 -9.83477e-07
Eliminating syzygy term y^2 by F2
```

```
with F2=0.3293363*y^2+0.424498*x+0.2720801*y-0.5923007-9.834766e-07*x*y*[E]
F1 := F1 + F2 * -0.73202 --->
F1 := 0.566862*x^2+0.8974145*x*y+0.2410807*y^2-0.02853775*x+0.03165275*y+0.02695525 +
      0.3293363*y^2+0.424498*x+0.2720801*y-0.5923007-9.834766e-07*x*y*[E] *
      -0.73202 --->
F1=0.566862*x^2+0.8974145*x*y-0.3392788*x-0.1675153*y+0.4605312+7.199246e-07*x*y*[E] -n->
F1=0.566862*x^2+0.8974145*x*y-0.3392788*x-0.1675153*y+0.4605312+7.199246e-07*x*y*[E] -e->
F1=0.566862*x^2+0.8974145*x*y-0.3392788*x-0.1675153*y+0.4605312+7.199246e-07*x*y*[E]
   |      x^2      x*y      y^2         x         y         1      x*y*[E]
---+-------------------------------------------------------------------------
 F1| 0.566862 0.897415        . -0.339279 -0.167515  0.460531  7.19925e-07
 F2|        .        . 0.329336  0.424498   0.27208 -0.592301 -9.83477e-07
The matrix after elimination
   |      x^2      x*y      y^2         x         y         1      x*y*[E]
---+-------------------------------------------------------------------------
 F1| 0.566862 0.897415        . -0.339279 -0.167515  0.460531  7.19925e-07
 F2|        .        . 0.329336  0.424498   0.27208 -0.592301 -9.83477e-07
Leaving with 1 eliminations done.
Entering symbolic auto-reduction.
The set of polynomials for symbolic reduction is
F2=0.3293363*y^2+0.424498*x+0.2720801*y-0.5923007-9.834766e-07*x*y*[E]
F1=0.566862*x^2+0.8974145*x*y-0.3392788*x-0.1675153*y+0.4605312+7.199246e-07*x*y*[E]
reducing supp(F2)=[x*y*[E],1,y,x,y^2]
with F2=0.3293363*y^2+0.424498*x+0.2720801*y-0.5923007-9.834766e-07*x*y*[E]
reducing supp(F1)=[x*y*[E],1,y,x,x*y,x^2]
with F1=0.566862*x^2+0.8974145*x*y-0.3392788*x-0.1675153*y+0.4605312+7.199246e-07*x*y*[E]
The symbolically reduced set of polynomials is
F2=0.3293363*y^2+0.424498*x+0.2720801*y-0.5923007-9.834766e-07*x*y*[E]
F1=0.566862*x^2+0.8974145*x*y-0.3392788*x-0.1675153*y+0.4605312+7.199246e-07*x*y*[E]
The set of reducible terms is [<empty>]
The set of terms to keep is [<empty>]
Symbolic reduction finished with 0 reducers added to the basis
No reducible terms found - exit reduction.
The auto-reduced set of polynomials is
F2=0.3293363*y^2+0.424498*x+0.2720801*y-0.5923007-9.834766e-07*x*y*[E]
F1=0.566862*x^2+0.8974145*x*y-0.3392788*x-0.1675153*y+0.4605312+7.199246e-07*x*y*[E]
```

The result of auto-reduction appears to be a Gröbner basis in the algebra of extended polynomials since all critical elements formed from the auto-reduced basis are detected as redundant by the co-prime criteria.

```
The current critical set:
S-pairs: []
E-orbit: []

Forming critical elements from
F2=0.3293363*y^2+0.424498*x+0.2720801*y-0.5923007-9.834766e-07*x*y*[E]
F1=0.566862*x^2+0.8974145*x*y-0.3392788*x-0.1675153*y+0.4605312+7.199246e-07*x*y*[E]

The e-shift of F2 is excluded by co-prime.
The e-shift of F1 is excluded by co-prime.
The s-pair (F2,F1) is excluded by co-prime.
2 e-shifts and 1 s-pairs excluded by co-prime.
The critical set before chain:
S-pairs: []
E-orbit: []
The critical set after chain:
S-pairs: []
E-orbit: []
```

The computation of a Gröbner basis in the algebra of extended polynomials is now finished. The result of auto-reduction is already totally reduced and therefore no more actions are applied to it.

```
The minimal Groebner basis with 2 polynomials is
F2=0.3293363*y^2+0.424498*x+0.2720801*y-0.5923007-9.834766e-07*x*y*[E]
F1=0.566862*x^2+0.8974145*x*y-0.3392788*x-0.1675153*y+0.4605312+7.199246e-07*x*y*[E]

Statistics:
   E-shifts excluded by co-prime: 2
    S-pairs excluded by co-prime: 1
      E-shifts excluded by chain: 0
       S-pairs excluded by chain: 0
      Number of row eliminations: 2
E-shifts delivered for reduction: 0
 S-pairs delivered for reduction: 0
             symbolic reductions: 0
The reduced Groebner basis contains 2 polynomials:
F2=0.3293363*y^2+0.424498*x+0.2720801*y-0.5923007-9.834766e-07*x*y*[E]
F1=0.566862*x^2+0.8974145*x*y-0.3392788*x-0.1675153*y+0.4605312+7.199246e-07*x*y*[E]
```

The leading terms of the attained (totally) reduced Gröbner basis in the algebra of extended polynomials are terms in the main variables and furthermore they are pure powers of all main variables x and y. Therefore the basis is suitable for translation into an extended Gröbner basis of F in the original algebra $\mathbb{C}[x, y]$, expansion to a border basis of $\langle F \rangle$ and the subsequent construction of multiplication matrices.

```
The Groebner basis is stable
The ideal is zero-dimensional
The normal set is [1,y,x,x*y]
The system has 4 solutions
The border set contains 4 terms: [y^2,x^2,x*y^2,x^2*y]
The border basis with 4 polynomials is
  y^2: F2=-2.986238e-06*x*y+y^2+1.28895*x+0.8261467*y-1.798468
  x^2: F1=x^2+1.583128*x*y-0.598521*x-0.2955134*y+0.812422
x*y^2: B1=x*y^2-1.214423*x*y-1.026998*x+0.3809033*y-1.047168
x^2*y: B2=x^2*y+1.324065*x*y+2.006772*x+0.4535406*y+1.12633
The multiplication with 'x' matrix M_x is
       |        1          y         x       x*y
-------+----------------------------------------
  1 * x|        0          0         1         0
  y * x|        0          0         0         1
  x * x|-0.812422  0.2955134  0.598521 -1.583128
x*y * x| -1.12633 -0.4535406 -2.006772 -1.324065

The multiplication with 'y' matrix M_y is
       |       1          y        x          x*y
-------+-----------------------------------------
  1 * y|       0          1        0            0
  y * y|1.798468 -0.8261467 -1.28895 2.986238e-06
  x * y|       0          0        0            1
x*y * y|1.047168 -0.3809033 1.026998     1.214423

The matrix for eigenvector computation is (0.5645571,0.8566645) * [M_x,M_y] =
         |          1          y          x         x*y
---------+---------------------------------------------
  1 * ...|          0  0.8566645  0.5645571           0
  y * ...|   1.540683 -0.7077305  -1.104198   0.5645596
  x * ...| -0.4586586  0.1668342  0.3378993 -0.03710181
x*y * ...|  0.2611941 -0.5823559 -0.2531443   0.2928426
```

The latter matrix is plugged into the automatically generated script to be loaded and executed in Maple for computing the solutions of F by the eigenvector method.

```
Digits:=7:
with(linalg):
F:=[2.267448*x^2+3.589658*x*y+0.964323*y^2-0.114151*x+0.126611*y+0.107821,
    1.698297*x^2+2.688612*x*y+3.356959*y^2+3.310486*x+2.271471*y-4.657648]:
M:=[[0,0.8566645,0.5645571,0],
    [1.540683,-0.7077305,-1.104198,0.5645596],
    [-0.4586586,0.1668342,0.3378993,-0.03710181],
    [0.2611941,-0.5823559,-0.2531443,0.2928426]]:
E:=eigenvectors(M):
Z:=[]:
for i in E do
  val_x:=op(i[3])[3]/op(i[3])[1]:
  val_y:=op(i[3])[2]/op(i[3])[1]:
  Z:=[op(Z),[x=val_x,y=val_y]]:
od:
for sol in Z do subs(sol,F) od;
```

The execution of the script in Maple gives the solutions of F. For each solution we also provide an estimation of the sensitivity of the root-finding problem for the root of F: it is an absolute condition number $\|J_F^{-1}(x,y)\|$ computed as the maximum matrix norm of the inverse Jacobian

$$J_F^{-1} = \begin{pmatrix} \frac{\partial F_1}{\partial x} & \frac{\partial F_1}{\partial y} \\ \\ \frac{\partial F_2}{\partial x} & \frac{\partial F_2}{\partial y} \end{pmatrix}^{-1}$$

of F at the solution (x, y).

x	y	$\|J_F^{-1}(x,y)\|$
-0.5752696	1.233315	0.7107091
-1.926692	1.696989	0.6402831
1.283420	-0.9741578	0.4474002
0.4930342	-1.567946	0.3201388

The quality of the attained solutions is checked by substitution of the attained points into the original system F in the loop at the last line of the script.

F_1	F_2
-0.0000042	0.000011
0.0000713	-0.000296
0.0000713	-0.000046
-0.0000456	0.000317

The attained residuals are all at most of order of magnitude 10^{-4} which can be expected for a numerical computation performed with 7 decimal digits which is the accuracy of coefficients in the input F and also roughly the accuracy provided by the single precision floating-point arithmetic used in the computation: according to the order of residuals and the order of the absolute condition numbers, the coordinates of each solution are valid up to at least 3 decimals; this statement is also easy to check by a straightforward comparison with the "exact" solutions listed on page 285.

4.3.2 Ellipse and 3 lines

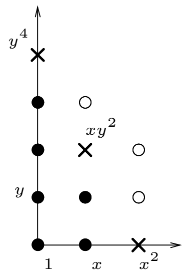
Consider the system $F=\{F_1, F_2\}$ with

$$\begin{aligned} F_1 =& x^2 + 0.000001\, xy + y^2 - 1 \\ F_2 =& y^3 - 3\, x^2 y \end{aligned}$$

in $\mathbb{C}[x, y]$ equipped with the *total degree (reverse) lexicographical order* induced by $y<x$. The genuine totally reduced Gröbner basis of F displayed to the decimal approximation with 7 digits is

$$\begin{aligned} A = \{A_1,\ A_2,\ A_3\} \quad &\text{with} \\ A_1 &= 0.5\, \underline{x^2} + 0.0000005\, xy + 0.5\, y^2 - 0.5, \\ A_2 &= 0.0000005\, \underline{xy^2} + 0.6666667\, y^3 - 0.5\, y, \\ A_3 &= 0.6666667\, \underline{y^4} + 0.000000375\, xy - 0.5\, y^2. \end{aligned}$$

The leading terms of A are $\{x^2,\ xy^2,\ y^4\}$ and the leading coefficient at the leading term xy^2 of A_2 is very small. The associated normal set is



$$N(A) = \{1, y, x, y^2, xy, y^3\}.$$

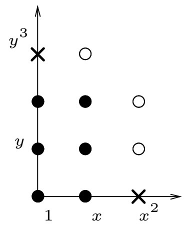
The basis A is an ill-conditioned representation of the ideal $\langle F \rangle$. Indeed, a small change in F attained by replacing the tiny coefficient at xy in F_1 results in the system

$$\widetilde{F} = \{x^2 + y^2 - 1,\ y^3 - 3\, x^2 y\}$$

with a structurally different genuine Gröbner basis

$$\{x^2 + y^2 - 1,\ 4\, y^3 - 3\, y\}$$

and the normal set



$$N = \{1, y, x, y^2, xy, xy^2\}.$$

The second polynomial

$$P = 4\,y^3 - 3\,y$$

of the Gröbner basis of $\widetilde{F}$ is a linear combination of the terms $N(A)$. Therefore the set $N(A)$ is linearly dependent modulo $\langle \widetilde{F}\,\rangle$ and hence can not represent a linear space basis in the quotient $\mathbb{C}[x,y]/\langle \widetilde{F}\,\rangle$. In particular, $N(A)$ can not be the normal set of any genuine Gröbner basis of $\widetilde{F}$ which explains the "jump" in the normal set from $N(A)$ for F to N for $\widetilde{F}$ although the systems F and $\widetilde{F}$ are very close to each other from a geometric point of view. It is straightforward to check by direct computation that the tiny leading coefficient of A_2 originates from the tiny coefficient at the term xy in F_1 and vanishes under the transition $\widetilde{F} \to F$ and results in the correspondence $A_2 \to P$ computing the respective Gröbner bases.[8]

The geometric interpretation is given on figure 4.3.2. A slightly perturbed circle $F_1{=}0$ is intersected with three lines $F_2{=}0$. The intersection points give all six solutions of F. All the solutions are almost but not precisely on the three horizontal lines defined by the equation $P{=}0$. The solutions of a slightly different system $\widetilde{F}$ are very close to the solutions of F but they are precisely on the three parallel lines $P{=}0$.

The observed jump in the normal sets from $N(A)$ to N together with the presence of a small leading coefficient at the polynomial A_3 in the genuine Gröbner basis A is an indication that A might be poorly-suited for numerical determination of zeros of F as well as for any numerical computation. It is easy to check, by complete analogy with the analysis provided in the discussion for the previous example with an ellipse intersected with hyperbola, that $N(A)$ represents an *"almost linearly dependent"* basis in the quotient $\mathbb{C}[x,y]/\langle F\rangle$ and therefore A is an ill-conditioned representation of $\langle F\rangle$ while the set N represents an appropriate from a numerical point of view linear basis of the quotient. Hence an extended Gröbner basis associated with N – the genuine (totally) reduced Gröbner basis of $\widetilde{F}$ with slightly different

[8] An appropriate context where the latter statement gains a mathematically strict meaning is given e.g. by the concept of a comprehensive Gröbner basis [Wei92].

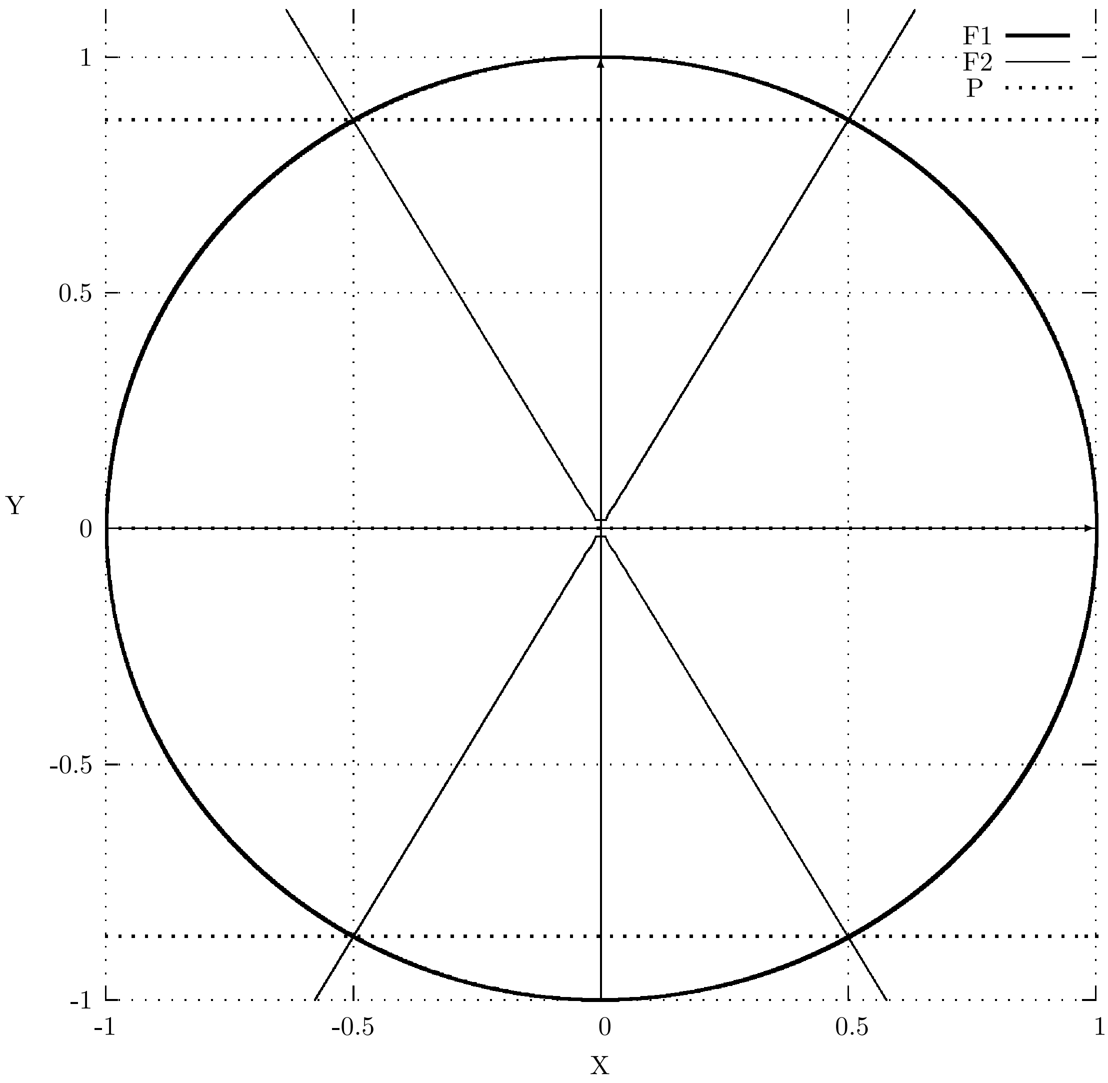


Figure 4.2: The system F from page 294

coefficients at the terms N – is a preferable representation of $\langle F \rangle$ for the numerical determination of zeros.

The desired extended Gröbner basis is computed by our algorithm using the $\underset{e\mathbb{T}}{<}$ extension of the term order (the priority is given to the degrees of e and a higher degree results in a smaller term) of the term order and the hardware single precision floating-point arithmetic.

```
alphabet = x,y;
termorder = tdeg;
extorder = eT;
domain = float;
ebound = 2;

ideal = x^2+1e-6*x*y+y^2-1,y^3-3*x^2*y;
```

At first the algorithm performs normalization and translation of the input polynomials F=$\{F_1, F_2\}$ to the algebra of extended polynomials.

```
Alphabet:            x>y
Extended term order: eT(Total degree)
Coefficients domain: float
Zero test:           rel<0.0004882812
E-bound:             2
The normalized and translated input basis contains 2 polynomials:
F1=0.5*x^2+0.5*y^2-0.5+5e-07*x*y*[E]
F2=-0.75*x^2*y+0.25*y^3
```

The initial auto-reduction amounts to elimination of the leading term x^2y from F_2.

```
Entering auto-reduction of the basis.
Entering symbolic auto-reduction.
The set of polynomials for symbolic reduction is
F1=0.5*x^2+0.5*y^2-0.5+5e-07*x*y*[E]
F2=-0.75*x^2*y+0.25*y^3
reducing supp(F1)=[x*y*[E],1,y^2,x^2]
with F1=0.5*x^2+0.5*y^2-0.5+5e-07*x*y*[E]
reducing supp(F2)=[y^3,x^2*y]
with F2=-0.75*x^2*y+0.25*y^3
found an auto-reducer F1 for x^2*y
with F1=0.5*x^2+0.5*y^2-0.5+5e-07*x*y*[E]
x^2*y is added to the set of reducible terms
Added the reducer for x^2*y : y * F1 = R1=0.5*x^2*y+0.5*y^3-0.5*y+5e-07*x*y^2*[E]
reducing supp(R1)=[x*y^2*[E],y,y^3,x^2*y]
with R1=0.5*x^2*y+0.5*y^3-0.5*y+5e-07*x*y^2*[E]
The symbolically reduced set of polynomials is
F1=0.5*x^2+0.5*y^2-0.5+5e-07*x*y*[E]
F2=-0.75*x^2*y+0.25*y^3
R1=0.5*x^2*y+0.5*y^3-0.5*y+5e-07*x*y^2*[E]
The set of reducible terms is [x^2*y]
The set of terms to keep is [<empty>]
Symbolic reduction finished with 1 reducers added to the basis
Entering elimination.
The initial polynomials
F1=0.5*x^2+0.5*y^2-0.5+5e-07*x*y*[E]
F2=-0.75*x^2*y+0.25*y^3
R1=0.5*x^2*y+0.5*y^3-0.5*y+5e-07*x*y^2*[E]
```

```
The initial matrix
   | x^2*y |  y^3 x^2 y^2    y    1 x*y^2*[E] x*y*[E]
---+----------------------------------------------------
 F2| -0.75 | 0.25    .   .    .    .         .       .
 R1|   0.5 |  0.5    .   . -0.5    .     5e-07       .
 F1|     . |    . 0.5 0.5    . -0.5         .   5e-07
Eliminating reducible x^2*y by F2
with F2=-0.75*x^2*y+0.25*y^3
R1 := R1 + F2 * 0.6666667 --->
R1 := 0.5*x^2*y+0.5*y^3-0.5*y+5e-07*x*y^2*[E] +
      -0.75*x^2*y+0.25*y^3 *
      0.6666667 --->
R1=0.6666667*y^3-0.5*y+5e-07*x*y^2*[E] -n->
R1=0.6666667*y^3-0.5*y+5e-07*x*y^2*[E] -e->
R1=0.6666667*y^3-0.5*y+5e-07*x*y^2*[E]
   | x^2*y |      y^3 x^2 y^2    y    1 x*y^2*[E] x*y*[E]
---+--------------------------------------------------------
 F2| -0.75 |     0.25    .   .    .    .         .       .
 R1|     . | 0.666667    .   . -0.5    .     5e-07       .
 F1|     . |        . 0.5 0.5    . -0.5         .   5e-07
removing pivot at x^2*y
with F2=-0.75*x^2*y+0.25*y^3
   | |      y^3 x^2 y^2    y    1 x*y^2*[E] x*y*[E]
---+------------------------------------------------
 R1| | 0.666667    .   . -0.5    .     5e-07       .
 F1| |        . 0.5 0.5    . -0.5         .   5e-07
The matrix after elimination
   | |      y^3 x^2 y^2    y    1 x*y^2*[E] x*y*[E]
---+------------------------------------------------
 R1| | 0.666667    .   . -0.5    .     5e-07       .
 F1| |        . 0.5 0.5    . -0.5         .   5e-07
Leaving with 1 eliminations done.
Entering symbolic auto-reduction.
The set of polynomials for symbolic reduction is
F1=0.5*x^2+0.5*y^2-0.5+5e-07*x*y*[E]
R1=0.6666667*y^3-0.5*y+5e-07*x*y^2*[E]
reducing supp(F1)=[x*y*[E],1,y^2,x^2]
with F1=0.5*x^2+0.5*y^2-0.5+5e-07*x*y*[E]
reducing supp(R1)=[x*y^2*[E],y,y^3]
with R1=0.6666667*y^3-0.5*y+5e-07*x*y^2*[E]
The symbolically reduced set of polynomials is
F1=0.5*x^2+0.5*y^2-0.5+5e-07*x*y*[E]
R1=0.6666667*y^3-0.5*y+5e-07*x*y^2*[E]
The set of reducible terms is [<empty>]
The set of terms to keep is [<empty>]
Symbolic reduction finished with 0 reducers added to the basis
No reducible terms found - exit reduction.
The auto-reduced set of polynomials is
F1=0.5*x^2+0.5*y^2-0.5+5e-07*x*y*[E]
R1=0.6666667*y^3-0.5*y+5e-07*x*y^2*[E]
```

The result of auto-reduction appears to be a Gröbner basis in the algebra of extended polynomials since all critical elements formed from the auto-reduced basis are detected as redundant by the co-prime criteria.

```
The current critical set:
S-pairs: []
E-orbit: []

Forming critical elements from
F1=0.5*x^2+0.5*y^2-0.5+5e-07*x*y*[E]
R1=0.6666667*y^3-0.5*y+5e-07*x*y^2*[E]
```

```
The e-shift of F1 is excluded by co-prime.
The e-shift of R1 is excluded by co-prime.
The s-pair (F1,R1) is excluded by co-prime.
2 e-shifts and 1 s-pairs excluded by co-prime.
The critical set before chain:
S-pairs: []
E-orbit: []
The critical set after chain:
S-pairs: []
E-orbit: []
```

The computation of a Gröbner basis in the algebra of extended polynomials is now finished. The result of auto-reduction is already totally reduced and therefore no more actions are applied to it.

```
The minimal Groebner basis with 2 polynomials is
F1=0.5*x^2+0.5*y^2-0.5+5e-07*x*y*[E]
R1=0.6666667*y^3-0.5*y+5e-07*x*y^2*[E]

Statistics:
   E-shifts excluded by co-prime: 2
    S-pairs excluded by co-prime: 1
      E-shifts excluded by chain: 0
       S-pairs excluded by chain: 0
      Number of row eliminations: 1
E-shifts delivered for reduction: 0
 S-pairs delivered for reduction: 0
             symbolic reductions: 1
The reduced Groebner basis contains 2 polynomials:
F1=0.5*x^2+0.5*y^2-0.5+5e-07*x*y*[E]
R1=0.6666667*y^3-0.5*y+5e-07*x*y^2*[E]
```

The leading terms of the attained (totally) reduced Gröbner basis in the algebra of extended polynomials are terms in the main variables and furthermore they are the pure powers of all main variables x and y. Therefore the basis is suitable for the translation into the extended Gröbner basis of F in the original algebra $\mathbb{C}[x, y]$, expansion to the border basis of $\langle F \rangle$ and the subsequent construction of multiplication matrices.

```
The Groebner basis is stable
The ideal is zero-dimensional
The normal set is [1,y,x,y^2,x*y,x*y^2]
The system has 6 solutions
The border set contains 5 terms: [x^2,y^3,x^2*y,x*y^3,x^2*y^2]
The border basis with 5 polynomials is
    x^2: F1=x^2+1e-06*x*y+y^2-1
    y^3: R1=7.5e-07*x*y^2+y^3-0.75*y
  x^2*y: B1=x^2*y+2.5e-07*x*y^2-0.25*y
  x*y^3: B2=x*y^3-0.75*x*y+1.875e-07*y^2
x^2*y^2: B3=x^2*y^2+1.875e-07*x*y-0.25*y^2

The multiplication with 'x' matrix M_x is
         |1    y x  y^2        x*y    x*y^2
---------+---------------------------------
    1 * x|0    0 1    0          0        0
    y * x|0    0 0    0          1        0
    x * x|1    0 0   -1     -1e-06        0
  y^2 * x|0    0 0    0          0        1
```

```
  x*y * x|0 0.25 0    0          0 -2.5e-07
x*y^2 * x|0    0 0 0.25 -1.875e-07        0

The multiplication with 'y' matrix M_y is
         |1    y x        y^2  x*y    x*y^2
---------+---------------------------------
    1 * y|0    1 0          0    0        0
    y * y|0    0 0          1    0        0
    x * y|0    0 0          0    1        0
  y^2 * y|0 0.75 0          0    0 -7.5e-07
  x*y * y|0    0 0          0    0        1
x*y^2 * y|0    0 0 -1.875e-07 0.75        0

The matrix for eigenvector computation is (0.9893925,0.639063) * [M_x,M_y] =
          |        1         y         x        y^2       x*y     x*y^2
----------+-------------------------------------------------------------
    1 * ...|        0  0.639063 0.9893925          0         0         0
    y * ...|        0         0         0   0.639063 0.9893925         0
    x * ...|0.9893925         0         0 -0.9893925  0.639062         0
  y^2 * ...|        0 0.4792973         0          0         0  0.989392
  x*y * ...|        0 0.2473481         0          0         0 0.6390628
x*y^2 * ...|        0         0         0   0.247348 0.4792971         0
```

The latter matrix is plugged into the automatically generated script to be loaded and executed in Maple for computing the solutions of F by the eigenvector method.

```
Digits:=7:
with(linalg):
F:=[x^2+1e-06*x*y+y^2-1,-3*x^2*y+y^3]:
M:=[[0,0.639063,0.9893925,0,0,0],
    [0,0,0,0.639063,0.9893925,0],
    [0.9893925,0,0,-0.9893925,0.639062,0],
    [0,0.4792973,0,0,0,0.989392],
    [0,0.2473481,0,0,0,0.6390628],
    [0,0,0,0.247348,0.4792971,0]]:
E:=eigenvectors(M):
Z:=[]:
for i in E do
  val_x:=op(i[3])[3]/op(i[3])[1]:
  val_y:=op(i[3])[2]/op(i[3])[1]:
  Z:=[op(Z),[x=val_x,y=val_y]]:
od:
for sol in Z do subs(sol,F) od;
```

The execution of the script in Maple gives the solutions (x, y) of F. In the table below we provide an estimation of the sensitivity of the root-finding problem for each solution of F: it is an absolute condition number $\|J_F^{-1}(x, y)\|$ computed as the maximum matrix norm of the inverse Jacobian J_F^{-1} of F at the solution (x, y). For comparison we also list the "exact" solutions (x^*, y^*) of F computed independently with an extra high precision using `solve`(F)

in Maple.

x	y	$\|J_F^{-1}(x,y)\|$	x^*	y^*
-1.0	-0.0	0.5000002	-1.0	0.0
0.5000001	0.8660261	0.5996783	0.499999892	0.866025216
-0.5000001	-0.8660258	0.5996786	-0.499999892	-0.866025216
1.0	0.0	0.5000002	1.0	0.0
-0.5000001	0.8660262	0.5996789	-0.500000108	0.866025591
0.5000001	-0.8660261	0.5996789	0.500000108	-0.866025591

The quality of the attained solutions is checked by the substitution of the attained points into the original system F in the loop at the last line of the script.

F_1	F_2
0.0	0.0
0.000002	0.0000008
0.000001	-0.0000004
0.0	0.0
0.000001	0.0000010
0.000001	-0.0000008

The attained residuals are all at most of order 10^{-6} which is even more than can usually be expected for a numerical computation carried with 7 decimal digits accuracy of coefficients in the input F and roughly the precision provided by the single floating-point arithmetic used in the computation: The order of residuals and absolute condition numbers indicate that both coordinates of each solution are valid up to at least 5 decimal digits. This statement can easily be verified by comparison of the approximate and "exact" solutions in a table above.

4.3.3 Ellipse and an intricate curve

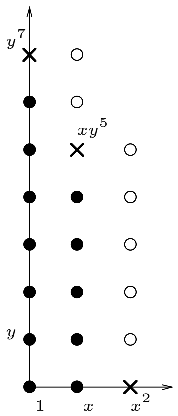
The system under consideration is F=$\{F_1, F_2\}$ with

$$F_1 = \left(x^2 + y^2\right)^3 - 4\,x^2y^2,$$
$$F_2 = x^2 + 0.000002\,xy + 2\,y^2 - 1$$

in $\mathbb{C}[x,y]$ equipped with the *total degree (reverse) lexicographical order* induced by $y<x$. This example is of the same kind as the two previous but also slightly more complicated. The system has 12 solutions: 8 real and 4 complex. The real solutions can be seen on figure 4.3.3: The "intricate curve" $F_1{=}0$ intersects the slightly distorted ellipse $F_2{=}0$ at eight points. It is clear from figure 4.3.3 that F is perfectly well-conditioned for the computation of the real zeros. However F is extremely ill-conditioned w.r.t. the determination of its complex solutions which are very sensitive to small changes in the coefficients of F; hence it is even more important that the situation for the determination of the zeros is not aggravated by the choice of an inappropriate representation of the ideal $\langle F\rangle$! The genuine (totally) reduced Gröbner basis of F is

$$
\begin{aligned}
A = \{A_1, A_2, A_3\} \quad &\text{with} \\
A_1 &= 0.25\,\underline{x^2} + 0.0000005\,xy + 0.5\,y^2 - 0.25, \\
A_2 &= -0.0000003\,\underline{xy^5} - 0.05\,y^6 + 0.000001\,xy^3 \\
&\qquad + 0.55\,y^4 - 0.0000003\,xy - 0.35\,y^2 + 0.05, \\
A_3 &= -0.05\,\underline{y^7} - 0.0000023\,xy^4 + 0.55\,y^5 \\
&\qquad + 0.0000018\,xy^2 - 0.35\,y^3 - 0.0000003\,x + 0.05\,y.
\end{aligned}
$$

The leading terms of A are $\{x^2,\ xy^5,\ y^7\}$. The associated normal set is



$$\{1, y, x, y^2, xy, y^3, xy^2, y^4, xy^3, y^5, xy^4, xy^5\}.$$

The Gröbner basis A is an ill-conditioned representation of the ideal $\langle F\rangle$ due to its high sensitivity to small changes in the coefficients of F: the small coefficient at xy^5 in A_2 indicates that. A stable (under small changes in the coefficients of F) representation is provided by the extended Gröbner basis

$$
\begin{aligned}
\{\ &\underline{x^2} + 0.000002\,xy + 2\,y^2 - 1, \\
&0.000006\,xy^5 + \underline{y^6} - 0.00002\,xy^3 - 11\,y^4 + 0.00000575\,xy + 7\,y^2 - 1\ \}
\end{aligned}
$$

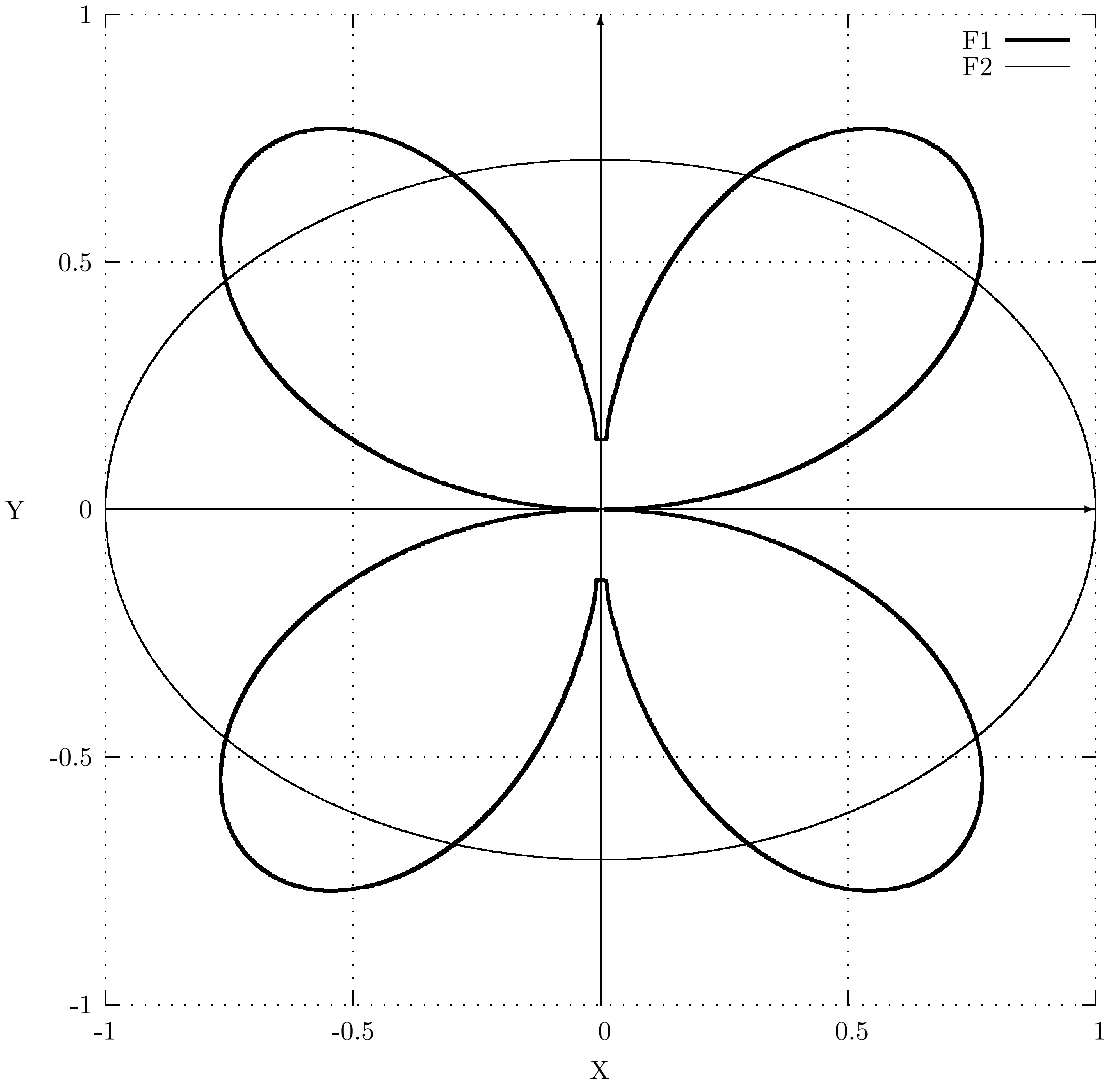
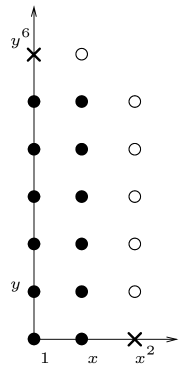


Figure 4.3: The system F from page 301

of F which is a slightly perturbed genuine (totally) reduced Gröbner basis

$$\{\ \underline{x^2} + 2\,y^2 - 1,\ \underline{y^6} - 11\,y^4 + 7\,y^2 - 1\ \}$$

with the leading terms $\{x^2,\ y^6\}$ and the associated normal set



$$\{1, y, x, y^2, xy, y^3,$$
$$xy^2, y^4, xy^3, y^5, xy^4, xy^5\}$$

of the system

$$\left\{\left(x^2 + y^2\right)^3 - 4\,x^2y^2,\ x^2 + 2\,y^2 - 1\right\}$$

attained as a slight perturbation of F by removing the term xy with a tiny coefficient from F_2.

Now we compute this extended Gröbner basis by our algorithm using the $\underset{e\mathbb{T}}{<}$ extension of the term order (the priority is given to the degrees of e and a higher degree results in a smaller term) of the term order and the hardware single precision floating-point arithmetic.

```
alphabet = x,y;
termorder = tdeg;
domain = float;
extorder = eT;
ebound = 2;
Ideal = (x^2+y^2)^3-4*x^2*y^2, x^2+2e-6*x*y+2*y^2-1;
```

The initial normalization and translation of the input polynomials $F=\{F_1, F_2\}$ to the algebra of extended polynomials assigns e-degree one to the term xy in F_2 as it has a tiny coefficient.

```
Alphabet:            x>y
Extended term order: eT(Total degree)
Coefficients domain: float
Zero test:           rel<0.0004882812
E-bound:             2
The normalized and translated input basis contains 2 polynomials:
F1=0.125*x^6+0.375*x^4*y^2+0.375*x^2*y^4+0.125*y^6-0.5*x^2*y^2
F2=0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E]
```

The initial auto-reduction essentially amounts to the total reduction of F_1 modulo F_2 by eliminating all multiples of x^2 from F_1 using the appropriate shifts of F_1. The tiny coefficient as well as the positive e-degree of xy of F_2 are inherited in all the created in the flow of reduction shifts of F_2 and remain in the totally reduced modulo F_2 form R_4 of F_1 as the monomials

$$-0.0000003\, xy^5{\cdot}e,\ 0.000001\, xy^3{\cdot}e,\ -0.0000002875\, xy{\cdot}e.$$

The flow of reduction does not noticeably deviate from the usual path that it would have taken being performed by a certain instantiation of the standard total reduction algorithm known from the classical theory of Gröbner bases except of the sole occasion at the near end of the reduction process where an extra small coefficient 0.0000000625 created at the term $xy{\cdot}e$ in a partially reduced modulo F_2 form R_3 of F_1 triggers the trimming of e-degrees mechanism; the term is assigned the e-degree two

$$xy{\cdot}e \ \longrightarrow\ xy{\cdot}e^2$$

as the reflection of the extra small coefficient, it is therefore removed from R_3 being considered as indistinguishable from zero in the algebra of extended polynomials due to the chosen e-bound $\mathcal{B}$=2. The term $xy{\cdot}e$ is however minimal modulo F_2 as it is not divisible by $\ell\mathrm{t}(F_2)$=x^2 and it would not have had a chance to be the leading term of R_4 after the reduction is finished. Therefore, this occasion can be considered as irrelevant.

```
Entering auto-reduction of the basis.
Entering symbolic auto-reduction.
The set of polynomials for symbolic reduction is
F1=0.125*x^6+0.375*x^4*y^2+0.375*x^2*y^4+0.125*y^6-0.5*x^2*y^2
F2=0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E]
reducing supp(F1)=[x^2*y^2,y^6,x^2*y^4,x^4*y^2,x^6]
with F1=0.125*x^6+0.375*x^4*y^2+0.375*x^2*y^4+0.125*y^6-0.5*x^2*y^2
found an auto-reducer F2 for x^6
with F2=0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E]
x^6 is added to the set of reducible terms
Added the reducer for x^6 : x^4 * F2 = R1=0.25*x^6+0.5*x^4*y^2-0.25*x^4+5e-07*x^5*y*[E]
found an auto-reducer F2 for x^4*y^2
with F2=0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E]
x^4*y^2 is added to the set of reducible terms
Added the reducer for x^4*y^2 :
                 x^2*y^2 * F2 = R2=0.25*x^4*y^2+0.5*x^2*y^4-0.25*x^2*y^2+5e-07*x^3*y^3*[E]
found an auto-reducer F2 for x^2*y^4
with F2=0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E]
x^2*y^4 is added to the set of reducible terms
Added the reducer for x^2*y^4 : y^4 * F2 = R3=0.25*x^2*y^4+0.5*y^6-0.25*y^4+5e-07*x*y^5*[E]
found an auto-reducer F2 for x^2*y^2
with F2=0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E]
x^2*y^2 is added to the set of reducible terms
Added the reducer for x^2*y^2 : y^2 * F2 = R4=0.25*x^2*y^2+0.5*y^4-0.25*y^2+5e-07*x*y^3*[E]
reducing supp(F2)=[x*y*[E],1,y^2,x^2]
with F2=0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E]
reducing supp(R1)=[x^5*y*[E],x^4,x^4*y^2,x^6]
with R1=0.25*x^6+0.5*x^4*y^2-0.25*x^4+5e-07*x^5*y*[E]
```

```
found an auto-reducer F2 for x^4
with F2=0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E]
x^4 is added to the set of reducible terms
Added the reducer for x^4 : x^2 * F2 = R5=0.25*x^4+0.5*x^2*y^2-0.25*x^2+5e-07*x^3*y*[E]
found an auto-reducer F2 for x^5*y*[E]
with F2=0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E]
R5 is a better reducer because it has a higher degree of the main part
of the leading term : R5=0.25*x^4+0.5*x^2*y^2-0.25*x^2+5e-07*x^3*y*[E]
x^5*y*[E] is added to the set of reducible terms
Added the reducer for x^5*y*[E] :
                  x*y*[E] * R5 = R6=0.25*x^5*y*[E]+0.5*x^3*y^3*[E]-0.25*x^3*y*[E]
reducing supp(R2)=[x^3*y^3*[E],x^2*y^2,x^2*y^4,x^4*y^2]
with R2=0.25*x^4*y^2+0.5*x^2*y^4-0.25*x^2*y^2+5e-07*x^3*y^3*[E]
found an auto-reducer F2 for x^3*y^3*[E]
with F2=0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E]
R4 is a better reducer because it has a higher degree of the main part
of the leading term : R4=0.25*x^2*y^2+0.5*y^4-0.25*y^2+5e-07*x*y^3*[E]
x^3*y^3*[E] is added to the set of reducible terms
Added the reducer for x^3*y^3*[E] :
                  x*y*[E] * R4 = R7=0.25*x^3*y^3*[E]+0.5*x*y^5*[E]-0.25*x*y^3*[E]
reducing supp(R3)=[x*y^5*[E],y^4,y^6,x^2*y^4]
with R3=0.25*x^2*y^4+0.5*y^6-0.25*y^4+5e-07*x*y^5*[E]
reducing supp(R4)=[x*y^3*[E],y^2,y^4,x^2*y^2]
with R4=0.25*x^2*y^2+0.5*y^4-0.25*y^2+5e-07*x*y^3*[E]
reducing supp(R5)=[x^3*y*[E],x^2,x^2*y^2,x^4]
with R5=0.25*x^4+0.5*x^2*y^2-0.25*x^2+5e-07*x^3*y*[E]
found an auto-reducer F2 for x^2
with F2=0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E]
x^2 is added to the set of terms to keep
found an auto-reducer F2 for x^3*y*[E]
with F2=0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E]
x^3*y*[E] is added to the set of reducible terms
Added the reducer for x^3*y*[E] :
                  x*y*[E] * F2 = R8=0.25*x^3*y*[E]+0.5*x*y^3*[E]-0.25*x*y*[E]
reducing supp(R6)=[x^3*y*[E],x^3*y^3*[E],x^5*y*[E]]
with R6=0.25*x^5*y*[E]+0.5*x^3*y^3*[E]-0.25*x^3*y*[E]
reducing supp(R7)=[x*y^3*[E],x*y^5*[E],x^3*y^3*[E]]
with R7=0.25*x^3*y^3*[E]+0.5*x*y^5*[E]-0.25*x*y^3*[E]
reducing supp(R8)=[x*y*[E],x*y^3*[E],x^3*y*[E]]
with R8=0.25*x^3*y*[E]+0.5*x*y^3*[E]-0.25*x*y*[E]
The symbolically reduced set of polynomials is
F1=0.125*x^6+0.375*x^4*y^2+0.375*x^2*y^4+0.125*y^6-0.5*x^2*y^2
F2=0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E]
R1=0.25*x^6+0.5*x^4*y^2-0.25*x^4+5e-07*x^5*y*[E]
R2=0.25*x^4*y^2+0.5*x^2*y^4-0.25*x^2*y^2+5e-07*x^3*y^3*[E]
R3=0.25*x^2*y^4+0.5*y^6-0.25*y^4+5e-07*x*y^5*[E]
R4=0.25*x^2*y^2+0.5*y^4-0.25*y^2+5e-07*x*y^3*[E]
R5=0.25*x^4+0.5*x^2*y^2-0.25*x^2+5e-07*x^3*y*[E]
R6=0.25*x^5*y*[E]+0.5*x^3*y^3*[E]-0.25*x^3*y*[E]
R7=0.25*x^3*y^3*[E]+0.5*x*y^5*[E]-0.25*x*y^3*[E]
R8=0.25*x^3*y*[E]+0.5*x*y^3*[E]-0.25*x*y*[E]
The set of reducible terms is
                [x^3*y*[E],x^3*y^3*[E],x^5*y*[E],x^2*y^2,x^4,x^2*y^4,x^4*y^2,x^6]
The set of terms to keep is [x^2]
Symbolic reduction finished with 8 reducers added to the basis
Entering elimination.
The initial polynomials
F1=0.125*x^6+0.375*x^4*y^2+0.375*x^2*y^4+0.125*y^6-0.5*x^2*y^2
F2=0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E]
R1=0.25*x^6+0.5*x^4*y^2-0.25*x^4+5e-07*x^5*y*[E]
R2=0.25*x^4*y^2+0.5*x^2*y^4-0.25*x^2*y^2+5e-07*x^3*y^3*[E]
R3=0.25*x^2*y^4+0.5*y^6-0.25*y^4+5e-07*x*y^5*[E]
R4=0.25*x^2*y^2+0.5*y^4-0.25*y^2+5e-07*x*y^3*[E]
R5=0.25*x^4+0.5*x^2*y^2-0.25*x^2+5e-07*x^3*y*[E]
R6=0.25*x^5*y*[E]+0.5*x^3*y^3*[E]-0.25*x^3*y*[E]
```

```
R7=0.25*x^3*y^3*[E]+0.5*x*y^5*[E]-0.25*x*y^3*[E]
R8=0.25*x^3*y*[E]+0.5*x*y^3*[E]-0.25*x*y*[E]
The initial matrix
   |   x^6 x^4*y^2 x^2*y^4   x^4 x^2*y^2 x^5*y*[E] x^3*y^3*[E] x^3*y*[E] |   y^6
---+----------------------------------------------------------------------------
 F1| 0.125   0.375   0.375     .    -0.5         .           .         . | 0.125
 R1|  0.25     0.5       . -0.25       .     5e-07           .         . |     .
 R2|     .    0.25     0.5     .   -0.25         .       5e-07         . |     .
 R3|     .       .    0.25     .       .         .           .         . |   0.5
 R5|     .       .       .  0.25     0.5         .           .     5e-07 |     .
 R4|     .       .       .     .    0.25         .           .         . |     .
 R6|     .       .       .     .       .      0.25         0.5     -0.25 |     .
 R7|     .       .       .     .       .         .        0.25         . |     .
 R8|     .       .       .     .       .         .           .      0.25 |     .
 F2|     .       .       .     .       .         .           .         . |     .
   |   y^4   x^2   y^2     1 x*y^5*[E] x*y^3*[E] x*y*[E]
---+----------------------------------------------------
 F1|     .     .     .     .         .         .       .
 R1|     .     .     .     .         .         .       .
 R2|     .     .     .     .         .         .       .
 R3| -0.25     .     .     .     5e-07         .       .
 R5|     . -0.25     .     .         .         .       .
 R4|   0.5     . -0.25     .         .     5e-07       .
 R6|     .     .     .     .         .         .       .
 R7|     .     .     .     .       0.5     -0.25       .
 R8|     .     .     .     .         .       0.5   -0.25
 F2|     .  0.25   0.5 -0.25         .         .   5e-07
Eliminating reducible x^3*y^3*[E] by R6
with R6=0.25*x^5*y*[E]+0.5*x^3*y^3*[E]-0.25*x^3*y*[E]
R2 := R2 + R6 * -1e-06 --->
R2 := 0.25*x^4*y^2+0.5*x^2*y^4-0.25*x^2*y^2+5e-07*x^3*y^3*[E] +
      0.25*x^5*y*[E]+0.5*x^3*y^3*[E]-0.25*x^3*y*[E] *
      -1e-06 --->
R2=0.25*x^4*y^2+0.5*x^2*y^4-0.25*x^2*y^2-2.5e-07*x^5*y*[E]+2.5e-07*x^3*y*[E] -n->
R2=0.25*x^4*y^2+0.5*x^2*y^4-0.25*x^2*y^2-2.5e-07*x^5*y*[E]+2.5e-07*x^3*y*[E] -e->
R2=0.25*x^4*y^2+0.5*x^2*y^4-0.25*x^2*y^2-2.5e-07*x^5*y*[E]+2.5e-07*x^3*y*[E]
   | x^3*y^3*[E] x^4*y^2 x^2*y^4   x^4 x^2*y^2 x^5*y*[E]   x^6 x^3*y*[E] |   y^6
---+----------------------------------------------------------------------------
 R7|        0.25       .       .     .       .         .     .         . |     .
 R6|         0.5       .       .     .       .      0.25     .     -0.25 |     .
 R2|           .    0.25     0.5     .   -0.25  -2.5e-07     .   2.5e-07 |     .
 F1|           .   0.375   0.375     .    -0.5         . 0.125         . | 0.125
 R1|           .     0.5       . -0.25       .     5e-07  0.25         . |     .
 R3|           .       .    0.25     .       .         .     .         . |   0.5
 R5|           .       .       .  0.25     0.5         .     .     5e-07 |     .
 R4|           .       .       .     .    0.25         .     .         . |     .
 R8|           .       .       .     .       .         .     .      0.25 |     .
 F2|           .       .       .     .       .         .     .         . |     .
   |   y^4   x^2   y^2     1 x*y^5*[E] x*y^3*[E] x*y*[E]
---+----------------------------------------------------
 R7|     .     .     .     .       0.5     -0.25       .
 R6|     .     .     .     .         .         .       .
 R2|     .     .     .     .         .         .       .
 F1|     .     .     .     .         .         .       .
 R1|     .     .     .     .         .         .       .
 R3| -0.25     .     .     .     5e-07         .       .
 R5|     . -0.25     .     .         .         .       .
 R4|   0.5     . -0.25     .         .     5e-07       .
 R8|     .     .     .     .         .       0.5   -0.25
 F2|     .  0.25   0.5 -0.25         .         .   5e-07
R7 := R7 + R6 * -0.5 --->
R7 := 0.25*x^3*y^3*[E]+0.5*x*y^5*[E]-0.25*x*y^3*[E] +
      0.25*x^5*y*[E]+0.5*x^3*y^3*[E]-0.25*x^3*y*[E] *
      -0.5 --->
R7=-0.125*x^5*y*[E]+0.5*x*y^5*[E]+0.125*x^3*y*[E]-0.25*x*y^3*[E] -n->
```

```
R7=-0.125*x^5*y*[E]+0.5*x*y^5*[E]+0.125*x^3*y*[E]-0.25*x*y^3*[E] -e->
R7=-0.125*x^5*y*[E]+0.5*x*y^5*[E]+0.125*x^3*y*[E]-0.25*x*y^3*[E]
   | x^3*y^3*[E] x^4*y^2 x^2*y^4   x^4 x^2*y^2 x^5*y*[E]   x^6 x^3*y*[E] |   y^6
---+-----------------------------------------------------------------------------
 R6|         0.5       .       .     .       .      0.25     .     -0.25 |     .
 R2|           .    0.25     0.5     .   -0.25  -2.5e-07     .   2.5e-07 |     .
 F1|           .   0.375   0.375     .    -0.5         . 0.125         . | 0.125
 R1|           .     0.5       . -0.25       .     5e-07  0.25         . |     .
 R3|           .       .    0.25     .       .         .     .         . |   0.5
 R5|           .       .       .  0.25     0.5         .     .     5e-07 |     .
 R4|           .       .       .     .    0.25         .     .         . |     .
 R7|           .       .       .     .       .    -0.125     .     0.125 |     .
 R8|           .       .       .     .       .         .     .      0.25 |     .
 F2|           .       .       .     .       .         .     .         . |     .
   |   y^4   x^2   y^2     1 x*y^5*[E] x*y^3*[E] x*y*[E]
---+-----------------------------------------------------
 R6|     .     .     .     .         .         .       .
 R2|     .     .     .     .         .         .       .
 F1|     .     .     .     .         .         .       .
 R1|     .     .     .     .         .         .       .
 R3| -0.25     .     .     .     5e-07         .       .
 R5|     . -0.25     .     .         .         .       .
 R4|   0.5     . -0.25     .         .     5e-07       .
 R7|     .     .     .     .       0.5     -0.25       .
 R8|     .     .     .     .         .       0.5   -0.25
 F2|     .  0.25   0.5 -0.25         .         .   5e-07
removing pivot at x^3*y^3*[E]
with R6=0.25*x^5*y*[E]+0.5*x^3*y^3*[E]-0.25*x^3*y*[E]
   | x^4*y^2 x^2*y^4   x^4 x^2*y^2 x^5*y*[E]   x^6 x^3*y*[E] |   y^6   y^4   x^2
---+-----------------------------------------------------------------------------
 R2|    0.25     0.5     .   -0.25  -2.5e-07     .   2.5e-07 |     .     .     .
 F1|   0.375   0.375     .    -0.5         . 0.125         . | 0.125     .     .
 R1|     0.5       . -0.25       .     5e-07  0.25         . |     .     .     .
 R3|       .    0.25     .       .         .     .         . |   0.5 -0.25     .
 R5|       .       .  0.25     0.5         .     .     5e-07 |     .     . -0.25
 R4|       .       .     .    0.25         .     .         . |     .   0.5     .
 R7|       .       .     .       .    -0.125     .     0.125 |     .     .     .
 R8|       .       .     .       .         .     .      0.25 |     .     .     .
 F2|       .       .     .       .         .     .         . |     .     .  0.25
   |   y^2     1 x*y^5*[E] x*y^3*[E] x*y*[E]
---+-----------------------------------------
 R2|     .     .         .         .       .
 F1|     .     .         .         .       .
 R1|     .     .         .         .       .
 R3|     .     .     5e-07         .       .
 R5|     .     .         .         .       .
 R4| -0.25     .         .     5e-07       .
 R7|     .     .       0.5     -0.25       .
 R8|     .     .         .       0.5   -0.25
 F2|   0.5 -0.25         .         .   5e-07
Eliminating reducible x^2*y^2 by R5
with R5=0.25*x^4+0.5*x^2*y^2-0.25*x^2+5e-07*x^3*y*[E]
F1 := F1 + R5 * 1 --->
F1 := 0.125*x^6+0.375*x^4*y^2+0.375*x^2*y^4+0.125*y^6-0.5*x^2*y^2 +
      0.25*x^4+0.5*x^2*y^2-0.25*x^2+5e-07*x^3*y*[E] *
      1 --->
F1=0.125*x^6+0.375*x^4*y^2+0.375*x^2*y^4+0.125*y^6+0.25*x^4-0.25*x^2+5e-07*x^3*y*[E] -n->
F1=0.125*x^6+0.375*x^4*y^2+0.375*x^2*y^4+0.125*y^6+0.25*x^4-0.25*x^2+5e-07*x^3*y*[E] -e->
F1=0.125*x^6+0.375*x^4*y^2+0.375*x^2*y^4+0.125*y^6+0.25*x^4-0.25*x^2+5e-07*x^3*y*[E]
   | x^4*y^2 x^2*y^4   x^4 x^2*y^2 x^5*y*[E]   x^6 x^3*y*[E] |   y^6   y^4   x^2
---+-----------------------------------------------------------------------------
 R2|    0.25     0.5     .   -0.25  -2.5e-07     .   2.5e-07 |     .     .     .
 F1|   0.375   0.375  0.25       .         . 0.125     5e-07 | 0.125     . -0.25
 R1|     0.5       . -0.25       .     5e-07  0.25         . |     .     .     .
 R3|       .    0.25     .       .         .     .         . |   0.5 -0.25     .
```

```
 R5|       .       .  0.25     0.5          .      .      5e-07 |      .     . -0.25
 R4|       .       .     .    0.25          .      .          . |      .   0.5     .
 R7|       .       .     .       .     -0.125      .      0.125 |      .     .     .
 R8|       .       .     .       .          .      .       0.25 |      .     .     .
 F2|       .       .     .       .          .      .          . |      .     .  0.25
   |   y^2     1 x*y^5*[E] x*y^3*[E] x*y*[E]
---+---------------------------------------
 R2|     .     .         .         .       .
 F1|     .     .         .         .       .
 R1|     .     .         .         .       .
 R3|     .     .     5e-07         .       .
 R5|     .     .         .         .       .
 R4| -0.25     .         .     5e-07       .
 R7|     .     .       0.5     -0.25       .
 R8|     .     .         .       0.5   -0.25
 F2|   0.5 -0.25         .         .   5e-07
R2 := R2 + R5 * 0.5 --->
R2 := 0.25*x^4*y^2+0.5*x^2*y^4-0.25*x^2*y^2-2.5e-07*x^5*y*[E]+2.5e-07*x^3*y*[E] +
      0.25*x^4+0.5*x^2*y^2-0.25*x^2+5e-07*x^3*y*[E] *
      0.5 --->
R2=0.25*x^4*y^2+0.5*x^2*y^4+0.125*x^4-0.125*x^2-2.5e-07*x^5*y*[E]+5e-07*x^3*y*[E] -n->
R2=0.25*x^4*y^2+0.5*x^2*y^4+0.125*x^4-0.125*x^2-2.5e-07*x^5*y*[E]+5e-07*x^3*y*[E] -e->
R2=0.25*x^4*y^2+0.5*x^2*y^4+0.125*x^4-0.125*x^2-2.5e-07*x^5*y*[E]+5e-07*x^3*y*[E]
   | x^4*y^2 x^2*y^4   x^4 x^2*y^2 x^5*y*[E]   x^6 x^3*y*[E] |   y^6   y^4    x^2
---+-------------------------------------------------------------------------------
 R2|    0.25     0.5 0.125       .  -2.5e-07     .      5e-07 |     .     . -0.125
 F1|   0.375   0.375  0.25       .         . 0.125      5e-07 | 0.125     .  -0.25
 R1|     0.5       . -0.25       .     5e-07  0.25          . |     .     .      .
 R3|       .    0.25     .       .         .     .          . |   0.5 -0.25      .
 R5|       .       .  0.25     0.5         .     .      5e-07 |     .     .  -0.25
 R4|       .       .     .    0.25         .     .          . |     .   0.5      .
 R7|       .       .     .       .    -0.125     .      0.125 |     .     .      .
 R8|       .       .     .       .         .     .       0.25 |     .     .      .
 F2|       .       .     .       .         .     .          . |     .     .   0.25
   |   y^2     1 x*y^5*[E] x*y^3*[E] x*y*[E]
---+---------------------------------------
 R2|     .     .         .         .       .
 F1|     .     .         .         .       .
 R1|     .     .         .         .       .
 R3|     .     .     5e-07         .       .
 R5|     .     .         .         .       .
 R4| -0.25     .         .     5e-07       .
 R7|     .     .       0.5     -0.25       .
 R8|     .     .         .       0.5   -0.25
 F2|   0.5 -0.25         .         .   5e-07
R4 := R4 + R5 * -0.5 --->
R4 := 0.25*x^2*y^2+0.5*y^4-0.25*y^2+5e-07*x*y^3*[E] +
      0.25*x^4+0.5*x^2*y^2-0.25*x^2+5e-07*x^3*y*[E] *
      -0.5 --->
R4=-0.125*x^4+0.5*y^4+0.125*x^2-0.25*y^2-2.5e-07*x^3*y*[E]+5e-07*x*y^3*[E] -n->
R4=-0.125*x^4+0.5*y^4+0.125*x^2-0.25*y^2-2.5e-07*x^3*y*[E]+5e-07*x*y^3*[E] -e->
R4=-0.125*x^4+0.5*y^4+0.125*x^2-0.25*y^2-2.5e-07*x^3*y*[E]+5e-07*x*y^3*[E]
   | x^4*y^2 x^2*y^4    x^4 x^2*y^2 x^5*y*[E]   x^6 x^3*y*[E] |   y^6   y^4    x^2
---+--------------------------------------------------------------------------------
 R2|    0.25     0.5  0.125       .  -2.5e-07     .      5e-07 |     .     . -0.125
 F1|   0.375   0.375   0.25       .         . 0.125      5e-07 | 0.125     .  -0.25
 R1|     0.5       .  -0.25       .     5e-07  0.25          . |     .     .      .
 R3|       .    0.25      .       .         .     .          . |   0.5 -0.25      .
 R4|       .       . -0.125       .         .     .   -2.5e-07 |     .   0.5  0.125
 R5|       .       .   0.25     0.5         .     .      5e-07 |     .     .  -0.25
 R7|       .       .      .       .    -0.125     .      0.125 |     .     .      .
 R8|       .       .      .       .         .     .       0.25 |     .     .      .
 F2|       .       .      .       .         .     .          . |     .     .   0.25
   |   y^2     1 x*y^5*[E] x*y^3*[E] x*y*[E]
---+---------------------------------------
```

```
 R2|     .     .         .         .       .
 F1|     .     .         .         .       .
 R1|     .     .         .         .       .
 R3|     .     .     5e-07         .       .
 R4| -0.25     .         .     5e-07       .
 R5|     .     .         .         .       .
 R7|     .     .       0.5     -0.25       .
 R8|     .     .         .       0.5   -0.25
 F2|   0.5 -0.25         .         .   5e-07
removing pivot at x^2*y^2
with R5=0.25*x^4+0.5*x^2*y^2-0.25*x^2+5e-07*x^3*y*[E]
   | x^4*y^2 x^2*y^4    x^4 x^5*y*[E]   x^6 x^3*y*[E] |   y^6   y^4    x^2   y^2
---+---------------------------------------------------------------------------------
 R2|    0.25     0.5  0.125  -2.5e-07     .     5e-07 |     .     . -0.125     .
 F1|   0.375   0.375   0.25         . 0.125     5e-07 | 0.125     .  -0.25     .
 R1|     0.5       .  -0.25     5e-07  0.25         . |     .     .      .     .
 R3|       .    0.25      .         .     .         . |   0.5 -0.25      .     .
 R4|       .       . -0.125         .     .  -2.5e-07 |     .   0.5  0.125 -0.25
 R7|       .       .      .    -0.125     .     0.125 |     .     .      .     .
 R8|       .       .      .         .     .      0.25 |     .     .      .     .
 F2|       .       .      .         .     .         . |     .     .   0.25   0.5
   |     1 x*y^5*[E] x*y^3*[E] x*y*[E]
---+----------------------------------
 R2|     .         .         .       .
 F1|     .         .         .       .
 R1|     .         .         .       .
 R3|     .     5e-07         .       .
 R4|     .         .     5e-07       .
 R7|     .       0.5     -0.25       .
 R8|     .         .       0.5   -0.25
 F2| -0.25         .         .   5e-07
Eliminating reducible x^2*y^4 by R2
with R2=0.25*x^4*y^2+0.5*x^2*y^4+0.125*x^4-0.125*x^2-2.5e-07*x^5*y*[E]+5e-07*x^3*y*[E]
F1 := F1 + R2 * -0.75 --->
F1 := 0.125*x^6+0.375*x^4*y^2+0.375*x^2*y^4+0.125*y^6+0.25*x^4-0.25*x^2+5e-07*x^3*y*[E] +
      0.25*x^4*y^2+0.5*x^2*y^4+0.125*x^4-0.125*x^2-2.5e-07*x^5*y*[E]+5e-07*x^3*y*[E] *
      -0.75 --->
F1=0.125*x^6+0.1875*x^4*y^2+0.125*y^6
     +0.15625*x^4-0.15625*x^2+1.875e-07*x^5*y*[E]+1.25e-07*x^3*y*[E] -n->
F1=0.125*x^6+0.1875*x^4*y^2+0.125*y^6
     +0.15625*x^4-0.15625*x^2+1.875e-07*x^5*y*[E]+1.25e-07*x^3*y*[E] -e->
F1=0.125*x^6+0.1875*x^4*y^2+0.125*y^6
     +0.15625*x^4-0.15625*x^2+1.875e-07*x^5*y*[E]+1.25e-07*x^3*y*[E]
   | x^4*y^2 x^2*y^4     x^4 x^5*y*[E]   x^6 x^3*y*[E] |   y^6   y^4      x^2   y^2
---+-----------------------------------------------------------------------------------
 F1|  0.1875       . 0.15625 1.875e-07 0.125  1.25e-07 | 0.125     . -0.15625     .
 R2|    0.25     0.5   0.125  -2.5e-07     .     5e-07 |     .     .   -0.125     .
 R1|     0.5       .   -0.25     5e-07  0.25         . |     .     .        .     .
 R3|       .    0.25       .         .     .         . |   0.5 -0.25        .     .
 R5*x^6+0.125*y^6+0.25*x^4-0.15625*x^2+1.25e-07*x^3*y*[E] -e->
F1=0.03125*x^6+0.125*y^6+0.25*x^4-0.15625*x^2+1.25e-07*x^3*y*[E]
   | x^4*y^2     x^4 x^5*y*[E]     x^6 x^3*y*[E] |   y^6   y^4      x^2   y^2     1
---+-----------------------------------------------------------------------------------
 R3|  -0.125 -0.0625  1.25e-07       .  -2.5e-07 |   0.5 -0.25   0.0625     .     .
 R1|     0.5   -0.25     5e-07    0.25         . |     .     .        .     .     .
 R4|       .  -0.125         .       .  -2.5e-07 |     .   0.5    0.125 -0.25     .
 F1|       .    0.25         . 0.03125  1.25e-07 | 0.125     . -0.15625     .     .
 R7|       .       .    -0.125       .     0.125 |     .     .        .     .     .
 R8|       .       .         .       .      0.25 |     .     .        .     .     .
 F2|       .       .         .       .         . |     .     .     0.25   0.5 -0.25
   | x*y^5*[E] x*y^3*[E] x*y*[E]
---+----------------------------
 R3|     5e-07         .       .
 R1|         .         .       .
 R4|         .     5e-07       .
```

```
 F1|        .         .        .
 R7|      0.5     -0.25        .
 R8|        .       0.5    -0.25
 F2|        .         .    5e-07
R3 := R3 + R1 * 0.25 --->
R3 := -0.125*x^4*y^2+0.5*y^6-0.0625*x^4-0.25*y^4
         +0.0625*x^2+1.25e-07*x^5*y*[E]+5e-07*x*y^5*[E]-2.5e-07*x^3*y*[E] +
      0.25*x^6+0.5*x^4*y^2-0.25*x^4+5e-07*x^5*y*[E] *
      0.25 --->
R3=0.0625*x^6+0.5*y^6-0.125*x^4-0.25*y^4
     +0.0625*x^2+2.5e-07*x^5*y*[E]+5e-07*x*y^5*[E]-2.5e-07*x^3*y*[E] -n->
R3=0.0625*x^6+0.5*y^6-0.125*x^4-0.25*y^4
     +0.0625*x^2+2.5e-07*x^5*y*[E]+5e-07*x*y^5*[E]-2.5e-07*x^3*y*[E] -e->
R3=0.0625*x^6+0.5*y^6-0.125*x^4-0.25*y^4
     +0.0625*x^2+2.5e-07*x^5*y*[E]+5e-07*x*y^5*[E]-2.5e-07*x^3*y*[E]
   | x^4*y^2    x^4 x^5*y*[E]     x^6 x^3*y*[E] |   y^6   y^4      x^2   y^2     1
---+-----------------------------------------------------------------------------------
 R1|     0.5  -0.25     5e-07    0.25         . |     .     .        .     .     .
 R3|       . -0.125   2.5e-07  0.0625  -2.5e-07 |   0.5 -0.25   0.0625     .     .
 R4|       . -0.125         .       .  -2.5e-07 |     .   0.5    0.125 -0.25     .
 F1|       .   0.25         . 0.03125  1.25e-07 | 0.125     . -0.15625     .     .
 R7|       .      .    -0.125       .     0.125 |     .     .        .     .     .
 R8|       .      .         .       .      0.25 |     .     .        .     .     .
 F2|       .      .         .       .         . |     .     .     0.25   0.5 -0.25
   | x*y^5*[E] x*y^3*[E] x*y*[E]
---+---------------------------
 R1|        .         .        .
 R3|    5e-07         .        .
 R4|        .     5e-07        .
 F1|        .         .        .
 R7|      0.5     -0.25        .
 R8|        .       0.5    -0.25
 F2|        .         .    5e-07
removing pivot at x^4*y^2
with R1=0.25*x^6+0.5*x^4*y^2-0.25*x^4+5e-07*x^5*y*[E]
   |    x^4 x^5*y*[E]     x^6 x^3*y*[E] |   y^6   y^4      x^2   y^2     1 x*y^5*[E]
---+-------------------------------------------------------------------------------------
 R3| -0.125   2.5e-07  0.0625  -2.5e-07 |   0.5 -0.25   0.0625     .     .     5e-07
 R4| -0.125         .       .  -2.5e-07 |     .   0.5    0.125 -0.25     .         .
 F1|   0.25         . 0.03125  1.25e-07 | 0.125     . -0.15625     .     .         .
 R7|      .    -0.125       .     0.125 |     .     .        .     .     .       0.5
 R8|      .         .       .      0.25 |     .     .        .     .     .         .
 F2|      .         .       .         . |     .     .     0.25   0.5 -0.25         .
   | x*y^3*[E] x*y*[E]
---+------------------
 R3|        .       .
 R4|    5e-07       .
 F1|        .       .
 R7|    -0.25       .
 R8|      0.5   -0.25
 F2|        .   5e-07
Eliminating reducible x^3*y*[E] by R8
with R8=0.25*x^3*y*[E]+0.5*x*y^3*[E]-0.25*x*y*[E]
F1 := F1 + R8 * -4.999999e-07 --->
F1 := 0.03125*x^6+0.125*y^6+0.25*x^4-0.15625*x^2+1.25e-07*x^3*y*[E] +
      0.25*x^3*y*[E]+0.5*x*y^3*[E]-0.25*x*y*[E] *
      -4.999999e-07 --->
F1=0.03125*x^6+0.125*y^6+0.25*x^4-0.15625*x^2-2.5e-07*x*y^3*[E]+1.25e-07*x*y*[E] -n->
F1=0.03125*x^6+0.125*y^6+0.25*x^4-0.15625*x^2-2.5e-07*x*y^3*[E]+1.25e-07*x*y*[E] -e->
F1=0.03125*x^6+0.125*y^6+0.25*x^4-0.15625*x^2-2.5e-07*x*y^3*[E]+1.25e-07*x*y*[E]
   |    x^4 x^5*y*[E] x^3*y*[E]     x^6 |   y^6   y^4      x^2   y^2     1 x*y^5*[E]
---+-------------------------------------------------------------------------------------
 R3| -0.125   2.5e-07  -2.5e-07  0.0625 |   0.5 -0.25   0.0625     .     .     5e-07
 R4| -0.125         .  -2.5e-07       . |     .   0.5    0.125 -0.25     .         .
 F1|   0.25         .         . 0.03125 | 0.125     . -0.15625     .     .         .
```

```
 R7|       .    -0.125     0.125        . |      .     .         .     .     .       0.5
 R8|       .         .      0.25        . |      .     .         .     .     .         .
 F2|       .         .         .        . |      .     .      0.25   0.5 -0.25         .
   | x*y^3*[E]  x*y*[E]
---+-------------------
 R3|         .        .
 R4|     5e-07        .
 F1|  -2.5e-07 1.25e-07
 R7|     -0.25        .
 R8|       0.5    -0.25
 F2|         .    5e-07
R3 := R3 + R8 * 1e-06 --->
R3 := 0.0625*x^6+0.5*y^6-0.125*x^4-0.25*y^4
        +0.0625*x^2+2.5e-07*x^5*y*[E]+5e-07*x*y^5*[E]-2.5e-07*x^3*y*[E] +
      0.25*x^3*y*[E]+0.5*x*y^3*[E]-0.25*x*y*[E] *
      1e-06 --->
R3=0.0625*x^6+0.5*y^6-0.125*x^4-0.25*y^4
     +0.0625*x^2+2.5e-07*x^5*y*[E]+5e-07*x*y^5*[E]+5e-07*x*y^3*[E]-2.5e-07*x*y*[E] -n->
R3=0.0625*x^6+0.5*y^6-0.125*x^4-0.25*y^4
     +0.0625*x^2+2.5e-07*x^5*y*[E]+5e-07*x*y^5*[E]+5e-07*x*y^3*[E]-2.5e-07*x*y*[E] -e->
R3=0.0625*x^6+0.5*y^6-0.125*x^4-0.25*y^4
     +0.0625*x^2+2.5e-07*x^5*y*[E]+5e-07*x*y^5*[E]+5e-07*x*y^3*[E]-2.5e-07*x*y*[E]
   |     x^4 x^5*y*[E] x^3*y*[E]      x^6 |    y^6   y^4       x^2   y^2     1 x*y^5*[E]
---+-------------------------------------------------------------------------------------
 R3| -0.125   2.5e-07         .  0.0625 |    0.5 -0.25    0.0625     .     .     5e-07
 R4| -0.125         .  -2.5e-07        . |      .   0.5     0.125 -0.25     .         .
 F1|   0.25         .         . 0.03125 |  0.125     . -0.15625     .     .         .
 R7|       .    -0.125     0.125        . |      .     .         .     .     .       0.5
 R8|       .         .      0.25        . |      .     .         .     .     .         .
 F2|       .         .         .        . |      .     .      0.25   0.5 -0.25         .
   | x*y^3*[E]  x*y*[E]
---+-------------------
 R3|     5e-07 -2.5e-07
 R4|     5e-07        .
 F1|  -2.5e-07 1.25e-07
 R7|     -0.25        .
 R8|       0.5    -0.25
 F2|         .    5e-07
R4 := R4 + R8 * 1e-06 --->
R4 := -0.125*x^4+0.5*y^4+0.125*x^2-0.25*y^2-2.5e-07*x^3*y*[E]+5e-07*x*y^3*[E] +
      0.25*x^3*y*[E]+0.5*x*y^3*[E]-0.25*x*y*[E] *
      1e-06 --->
R4=-0.125*x^4+0.5*y^4+0.125*x^2-0.25*y^2+1e-06*x*y^3*[E]-2.5e-07*x*y*[E] -n->
R4=-0.125*x^4+0.5*y^4+0.125*x^2-0.25*y^2+1e-06*x*y^3*[E]-2.5e-07*x*y*[E] -e->
R4=-0.125*x^4+0.5*y^4+0.125*x^2-0.25*y^2+1e-06*x*y^3*[E]-2.5e-07*x*y*[E]
   |     x^4 x^5*y*[E] x^3*y*[E]      x^6 |    y^6   y^4       x^2   y^2     1 x*y^5*[E]
---+-------------------------------------------------------------------------------------
 R3| -0.125   2.5e-07         .  0.0625 |    0.5 -0.25    0.0625     .     .     5e-07
 R4| -0.125         .         .        . |      .   0.5     0.125 -0.25     .         .
 F1|   0.25         .         . 0.03125 |  0.125     . -0.15625     .     .         .
 R7|       .    -0.125     0.125        . |      .     .         .     .     .       0.5
 R8|       .         .      0.25        . |      .     .         .     .     .         .
 F2|       .         .         .        . |      .     .      0.25   0.5 -0.25         .
   | x*y^3*[E]  x*y*[E]
---+-------------------
 R3|     5e-07 -2.5e-07
 R4|     1e-06 -2.5e-07
 F1|  -2.5e-07 1.25e-07
 R7|     -0.25        .
 R8|       0.5    -0.25
 F2|         .    5e-07
R7 := R7 + R8 * -0.5 --->
R7 := -0.125*x^5*y*[E]+0.5*x*y^5*[E]+0.125*x^3*y*[E]-0.25*x*y^3*[E] +
      0.25*x^3*y*[E]+0.5*x*y^3*[E]-0.25*x*y*[E] *
      -0.5 --->
```

```
R7=-0.125*x^5*y*[E]+0.5*x*y^5*[E]-0.5*x*y^3*[E]+0.125*x*y*[E] -n->
R7=-0.125*x^5*y*[E]+0.5*x*y^5*[E]-0.5*x*y^3*[E]+0.125*x*y*[E] -e->
R7=-0.125*x^5*y*[E]+0.5*x*y^5*[E]-0.5*x*y^3*[E]+0.125*x*y*[E]
   |    x^4 x^5*y*[E] x^3*y*[E]     x^6 |   y^6   y^4      x^2   y^2     1 x*y^5*[E]
---+--------------------------------------------------------------------------------
 R3| -0.125   2.5e-07         .  0.0625 |   0.5 -0.25   0.0625     .     .     5e-07
 R4| -0.125         .         .       . |     .   0.5    0.125 -0.25     .         .
 F1|   0.25         .         . 0.03125 | 0.125     . -0.15625     .     .         .
 R7|      .    -0.125         .       . |     .     .        .     .     .       0.5
 R8|      .         .      0.25       . |     .     .        .     .     .         .
 F2|      .         .         .       . |     .     .     0.25   0.5 -0.25         .
   | x*y^3*[E]  x*y*[E]
---+-------------------
 R3|     5e-07 -2.5e-07
 R4|     1e-06 -2.5e-07
 F1|  -2.5e-07 1.25e-07
 R7|      -0.5    0.125
 R8|       0.5    -0.25
 F2|         .    5e-07
removing pivot at x^3*y*[E]
with R8=0.25*x^3*y*[E]+0.5*x*y^3*[E]-0.25*x*y*[E]
   |    x^4 x^5*y*[E]     x^6 |   y^6   y^4      x^2   y^2     1 x*y^5*[E] x*y^3*[E]
---+------------------------------------------------------------------------------
 R3| -0.125   2.5e-07  0.0625 |   0.5 -0.25   0.0625     .     .     5e-07     5e-07
 R4| -0.125         .       . |     .   0.5    0.125 -0.25     .         .     1e-06
 F1|   0.25         . 0.03125 | 0.125     . -0.15625     .     .         .  -2.5e-07
 R7|      .    -0.125       . |     .     .        .     .     .       0.5      -0.5
 F2|      .         .       . |     .     .     0.25   0.5 -0.25         .         .
   |  x*y*[E]
---+---------
 R3| -2.5e-07
 R4| -2.5e-07
 F1| 1.25e-07
 R7|    0.125
 F2|    5e-07
Eliminating reducible x^4 by F1
with F1=0.03125*x^6+0.125*y^6+0.25*x^4-0.15625*x^2-2.5e-07*x*y^3*[E]+1.25e-07*x*y*[E]
R3 := R3 + F1 * 0.5 --->
R3 := 0.0625*x^6+0.5*y^6-0.125*x^4-0.25*y^4
        +0.0625*x^2+2.5e-07*x^5*y*[E]+5e-07*x*y^5*[E]+5e-07*x*y^3*[E]-2.5e-07*x*y*[E] +
      0.03125*x^6+0.125*y^6+0.25*x^4-0.15625*x^2-2.5e-07*x*y^3*[E]+1.25e-07*x*y*[E] *
      0.5 --->
R3=0.078125*x^6+0.5625*y^6-0.25*y^4-0.015625*x^2
     +2.5e-07*x^5*y*[E]+5e-07*x*y^5*[E]+3.75e-07*x*y^3*[E]-1.875e-07*x*y*[E] -n->
R3=0.078125*x^6+0.5625*y^6-0.25*y^4-0.015625*x^2
     +2.5e-07*x^5*y*[E]+5e-07*x*y^5*[E]+3.75e-07*x*y^3*[E]-1.875e-07*x*y*[E] -e->
R3=0.078125*x^6+0.5625*y^6-0.25*y^4-0.015625*x^2
     +2.5e-07*x^5*y*[E]+5e-07*x*y^5*[E]+3.75e-07*x*y^3*[E]-1.875e-07*x*y*[E]
   |    x^4 x^5*y*[E]      x^6 |    y^6   y^4       x^2   y^2     1 x*y^5*[E] x*y^3*[E]
---+----------------------------------------------------------------------------------
 R4| -0.125         .        . |      .   0.5     0.125 -0.25     .         .     1e-06
 F1|   0.25         .  0.03125 |  0.125     .  -0.15625     .     .         .  -2.5e-07
 R7|      .    -0.125        . |      .     .         .     .     .       0.5      -0.5
 R3|      .   2.5e-07 0.078125 | 0.5625 -0.25 -0.015625     .     .     5e-07  3.75e-07
 F2|      .         .        . |      .     .      0.25   0.5 -0.25         .         .
   |    x*y*[E]
---+-----------
 R4|   -2.5e-07
 F1|   1.25e-07
 R7|      0.125
 R3| -1.875e-07
 F2|      5e-07
R4 := R4 + F1 * 0.5 --->
R4 := -0.125*x^4+0.5*y^4+0.125*x^2-0.25*y^2+1e-06*x*y^3*[E]-2.5e-07*x*y*[E] +
      0.03125*x^6+0.125*y^6+0.25*x^4-0.15625*x^2-2.5e-07*x*y^3*[E]+1.25e-07*x*y*[E] *
```

```
      0.5 --->
R4=0.015625*x^6+0.0625*y^6+0.5*y^4
     +0.046875*x^2-0.25*y^2+8.75e-07*x*y^3*[E]-1.875e-07*x*y*[E] -n->
R4=0.015625*x^6+0.0625*y^6+0.5*y^4
     +0.046875*x^2-0.25*y^2+8.75e-07*x*y^3*[E]-1.875e-07*x*y*[E] -e->
R4=0.015625*x^6+0.0625*y^6+0.5*y^4
     +0.046875*x^2-0.25*y^2+8.75e-07*x*y^3*[E]-1.875e-07*x*y*[E]
   |  x^4 x^5*y*[E]      x^6 |    y^6   y^4       x^2   y^2     1 x*y^5*[E] x*y^3*[E]
---+--------------------------------------------------------------------------------
 F1| 0.25         .  0.03125 |  0.125     .  -0.15625     .     .         .  -2.5e-07
 R7|    .    -0.125        . |      .     .         .     .     .       0.5      -0.5
 R3|    .   2.5e-07 0.078125 | 0.5625 -0.25 -0.015625     .     .     5e-07  3.75e-07
 R4|    .         . 0.015625 | 0.0625   0.5  0.046875 -0.25     .         .  8.75e-07
 F2|    .         .        . |      .     .      0.25   0.5 -0.25         .         .
   |    x*y*[E]
---+-----------
 F1|   1.25e-07
 R7|      0.125
 R3| -1.875e-07
 R4| -1.875e-07
 F2|      5e-07
removing pivot at x^4
with F1=0.03125*x^6+0.125*y^6+0.25*x^4-0.15625*x^2-2.5e-07*x*y^3*[E]+1.25e-07*x*y*[E]
   | x^5*y*[E]      x^6 |    y^6   y^4       x^2   y^2     1 x*y^5*[E] x*y^3*[E]
---+---------------------------------------------------------------------------
 R7|    -0.125        . |      .     .         .     .     .       0.5      -0.5
 R3|   2.5e-07 0.078125 | 0.5625 -0.25 -0.015625     .     .     5e-07  3.75e-07
 R4|         . 0.015625 | 0.0625   0.5  0.046875 -0.25     .         .  8.75e-07
 F2|         .        . |      .     .      0.25   0.5 -0.25         .         .
   |    x*y*[E]
---+-----------
 R7|      0.125
 R3| -1.875e-07
 R4| -1.875e-07
 F2|      5e-07
Eliminating reducible x^5*y*[E] by R7
with R7=-0.125*x^5*y*[E]+0.5*x*y^5*[E]-0.5*x*y^3*[E]+0.125*x*y*[E]
R3 := R3 + R7 * 2e-06 --->
R3 := 0.078125*x^6+0.5625*y^6-0.25*y^4-0.015625*x^2
        +2.5e-07*x^5*y*[E]+5e-07*x*y^5*[E]+3.75e-07*x*y^3*[E]-1.875e-07*x*y*[E] +
      -0.125*x^5*y*[E]+0.5*x*y^5*[E]-0.5*x*y^3*[E]+0.125*x*y*[E] *
      2e-06 --->
R3=0.078125*x^6+0.5625*y^6-0.25*y^4-0.015625*x^2
     +1.5e-06*x*y^5*[E]-6.25e-07*x*y^3*[E]+6.249999e-08*x*y*[E] -n->
R3=0.078125*x^6+0.5625*y^6-0.25*y^4-0.015625*x^2
     +1.5e-06*x*y^5*[E]-6.25e-07*x*y^3*[E]+6.249999e-08*x*y*[E] -e->
R3=0.078125*x^6+0.5625*y^6-0.25*y^4-0.015625*x^2
     +1.5e-06*x*y^5*[E]-6.25e-07*x*y^3*[E]
   | x^5*y*[E]      x^6 |    y^6   y^4       x^2   y^2     1 x*y^5*[E] x*y^3*[E]
---+---------------------------------------------------------------------------
 R7|    -0.125        . |      .     .         .     .     .       0.5      -0.5
 R4|         . 0.015625 | 0.0625   0.5  0.046875 -0.25     .         .  8.75e-07
 R3|         . 0.078125 | 0.5625 -0.25 -0.015625     .     .   1.5e-06 -6.25e-07
 F2|         .        . |      .     .      0.25   0.5 -0.25         .         .
   |    x*y*[E]
---+-----------
 R7|      0.125
 R4| -1.875e-07
 R3|          .
 F2|      5e-07
removing pivot at x^5*y*[E]
with R7=-0.125*x^5*y*[E]+0.5*x*y^5*[E]-0.5*x*y^3*[E]+0.125*x*y*[E]
   |      x^6 |    y^6   y^4       x^2   y^2     1 x*y^5*[E] x*y^3*[E]    x*y*[E]
---+----------------------------------------------------------------------------
 R4| 0.015625 | 0.0625   0.5  0.046875 -0.25     .         .  8.75e-07 -1.875e-07
```

```
 R3| 0.078125 | 0.5625 -0.25 -0.015625      .      .   1.5e-06 -6.25e-07          .
 F2|        . |      .     .      0.25   0.5 -0.25         .         .      5e-07
Eliminating reducible x^6 by R3
with R3=0.078125*x^6+0.5625*y^6-0.25*y^4
          -0.015625*x^2+1.5e-06*x*y^5*[E]-6.25e-07*x*y^3*[E]
R4 := R4 + R3 * -0.2 --->
R4 := 0.015625*x^6+0.0625*y^6+0.5*y^4
        +0.046875*x^2-0.25*y^2+8.75e-07*x*y^3*[E]-1.875e-07*x*y*[E] +
      0.078125*x^6+0.5625*y^6-0.25*y^4
         -0.015625*x^2+1.5e-06*x*y^5*[E]-6.25e-07*x*y^3*[E] *
      -0.2 --->
R4=-0.05*y^6+0.55*y^4+0.05*x^2-0.25*y^2
      -3e-07*x*y^5*[E]+1e-06*x*y^3*[E]-1.875e-07*x*y*[E] -n->
R4=-0.05*y^6+0.55*y^4+0.05*x^2-0.25*y^2
      -3e-07*x*y^5*[E]+1e-06*x*y^3*[E]-1.875e-07*x*y*[E] -e->
R4=-0.05*y^6+0.55*y^4+0.05*x^2-0.25*y^2
      -3e-07*x*y^5*[E]+1e-06*x*y^3*[E]-1.875e-07*x*y*[E]
   |      x^6 |    y^6   y^4       x^2   y^2     1 x*y^5*[E] x*y^3*[E]    x*y*[E]
---+------------------------------------------------------------------------------
 R3| 0.078125 | 0.5625 -0.25 -0.015625      .      .   1.5e-06 -6.25e-07          .
 R4|        . |  -0.05  0.55      0.05 -0.25      .    -3e-07     1e-06 -1.875e-07
 F2|        . |      .     .      0.25   0.5 -0.25         .         .      5e-07
removing pivot at x^6
with R3=0.078125*x^6+0.5625*y^6-0.25*y^4-0.015625*x^2+1.5e-06*x*y^5*[E]-6.25e-07*x*y^3*[E]
   | |   y^6  y^4  x^2   y^2     1 x*y^5*[E] x*y^3*[E]    x*y*[E]
---+--------------------------------------------------------------
 R4| | -0.05 0.55 0.05 -0.25     .    -3e-07     1e-06 -1.875e-07
 F2| |     .    . 0.25   0.5 -0.25         .         .      5e-07
Eliminating syzygy term x^2 by F2
with F2=0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E]
R4 := R4 + F2 * -0.2 --->
R4 := -0.05*y^6+0.55*y^4+0.05*x^2-0.25*y^2
         -3e-07*x*y^5*[E]+1e-06*x*y^3*[E]-1.875e-07*x*y*[E] +
      0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E] *
      -0.2 --->
R4=-0.05*y^6+0.55*y^4-0.35*y^2
      +0.05-3e-07*x*y^5*[E]+1e-06*x*y^3*[E]-2.875e-07*x*y*[E] -n->
R4=-0.05*y^6+0.55*y^4-0.35*y^2
      +0.05-3e-07*x*y^5*[E]+1e-06*x*y^3*[E]-2.875e-07*x*y*[E] -e->
R4=-0.05*y^6+0.55*y^4-0.35*y^2
      +0.05-3e-07*x*y^5*[E]+1e-06*x*y^3*[E]-2.875e-07*x*y*[E]
   | |   y^6  y^4  x^2   y^2     1 x*y^5*[E] x*y^3*[E]    x*y*[E]
---+--------------------------------------------------------------
 R4| | -0.05 0.55    . -0.35  0.05    -3e-07     1e-06 -2.875e-07
 F2| |     .    . 0.25   0.5 -0.25         .         .      5e-07
The matrix after elimination
   | |   y^6  y^4  x^2   y^2     1 x*y^5*[E] x*y^3*[E]    x*y*[E]
---+--------------------------------------------------------------
 R4| | -0.05 0.55    . -0.35  0.05    -3e-07     1e-06 -2.875e-07
 F2| |     .    . 0.25   0.5 -0.25         .         .      5e-07
Leaving with 18 eliminations done.
Entering symbolic auto-reduction.
The set of polynomials for symbolic reduction is
F2=0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E]
R4=-0.05*y^6+0.55*y^4-0.35*y^2+0.05-3e-07*x*y^5*[E]+1e-06*x*y^3*[E]-2.875e-07*x*y*[E]
reducing supp(F2)=[x*y*[E],1,y^2,x^2]
with F2=0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E]
reducing supp(R4)=[x*y*[E],x*y^3*[E],x*y^5*[E],1,y^2,y^4,y^6]
with R4=-0.05*y^6+0.55*y^4-0.35*y^2+0.05-3e-07*x*y^5*[E]+1e-06*x*y^3*[E]-2.875e-07*x*y*[E]
The symbolically reduced set of polynomials is
F2=0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E]
R4=-0.05*y^6+0.55*y^4-0.35*y^2+0.05-3e-07*x*y^5*[E]+1e-06*x*y^3*[E]-2.875e-07*x*y*[E]
The set of reducible terms is [<empty>]
The set of terms to keep is [<empty>]
Symbolic reduction finished with 0 reducers added to the basis
```

```
No reducible terms found - exit reduction.
The auto-reduced set of polynomials is
F2=0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E]
R4=-0.05*y^6+0.55*y^4-0.35*y^2+0.05-3e-07*x*y^5*[E]+1e-06*x*y^3*[E]-2.875e-07*x*y*[E]
```

The result of auto-reduction appears to be a Gröbner basis in the algebra of extended polynomials since all critical elements formed from the auto-reduced basis are detected as redundant by the co-prime criteria.

```
The current critical set:
S-pairs: []
E-orbit: []

Forming critical elements from
F2=0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E]
R4=-0.05*y^6+0.55*y^4-0.35*y^2+0.05-3e-07*x*y^5*[E]+1e-06*x*y^3*[E]-2.875e-07*x*y*[E]

The e-shift of F2 is excluded by co-prime.
The e-shift of R4 is excluded by co-prime.
The s-pair (F2,R4) is excluded by co-prime.
2 e-shifts and 1 s-pairs excluded by co-prime.
The critical set before chain:
S-pairs: []
E-orbit: []
The critical set after chain:
S-pairs: []
E-orbit: []
```

The computation of a Gröbner basis in the algebra of extended polynomials is now finished. The result of auto-reduction is already totally reduced and therefore no more actions are applied to it. The minimal Groebner basis with 2 polynomials is

```
F2=0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E]
R4=-0.05*y^6+0.55*y^4-0.35*y^2+0.05-3e-07*x*y^5*[E]+1e-06*x*y^3*[E]-2.875e-07*x*y*[E]

Statistics:
   E-shifts excluded by co-prime: 2
    S-pairs excluded by co-prime: 1
      E-shifts excluded by chain: 0
       S-pairs excluded by chain: 0
      Number of row eliminations: 18
E-shifts delivered for reduction: 0
 S-pairs delivered for reduction: 0
             symbolic reductions: 8
The reduced Groebner basis contains 2 polynomials:
F2=0.25*x^2+0.5*y^2-0.25+5e-07*x*y*[E]
R4=-0.05*y^6+0.55*y^4-0.35*y^2+0.05-3e-07*x*y^5*[E]+1e-06*x*y^3*[E]-2.875e-07*x*y*[E]
```

The leading terms of the attained (totally) reduced Gröbner basis in the algebra of extended polynomials are terms in the main variables and furthermore they are pure powers of all main variables x and y. Therefore the basis is suitable for translation into the extended Gröbner basis of F in the original algebra $\mathbb{C}[x, y]$, expansion to the border basis of $\langle F \rangle$ and the subsequent construction of multiplication matrices.

```
The Groebner basis is stable
The ideal is zero-dimensional
The normal set is [1,y,x,y^2,x*y,y^3,x*y^2,y^4,x*y^3,y^5,x*y^4,x*y^5]
The system has 12 solutions
The border set contains 8 terms: [x^2,x^2*y,x^2*y^2,x^2*y^3,y^6,x^2*y^4,x*y^6,x^2*y^5]
The border basis with 8 polynomials is
    x^2: F2=x^2+2e-06*x*y+2*y^2-1
  x^2*y: B1=x^2*y+2e-06*x*y^2+2*y^3-y
x^2*y^2: B2=x^2*y^2+2e-06*x*y^3+2*y^4-y^2
x^2*y^3: B3=x^2*y^3+2e-06*x*y^4+2*y^5-y^3
    y^6: R4=6e-06*x*y^5+y^6-2e-05*x*y^3-11*y^4+5.75e-06*x*y+6.999999*y^2-0.9999999
x^2*y^4: B4=x^2*y^4-1e-05*x*y^5+4e-05*x*y^3+21*y^4-1.15e-05*x*y-14*y^2+2
  x*y^6: B5=x*y^6-11*x*y^4-8.599999e-05*y^5
                 +6.999999*x*y^2+5.249999e-05*y^3-0.9999999*x-6.249999e-06*y
x^2*y^5: B6=x^2*y^5-6.999999e-05*x*y^4+21*y^5+5.849999e-05*x*y^2-14*y^3-9.999999e-06*x+2*y
The multiplication with 'x' matrix M_x is
         | 1  y           x y^2      x*y y^3
---------+----------------------------------
    1 * x| 0  0           1   0        0   0
    y * x| 0  0           0   0        1   0
    x * x| 1  0           0  -2   -2e-06   0
  y^2 * x| 0  0           0   0        0   0
  x*y * x| 0  1           0   0        0  -2
  y^3 * x| 0  0           0   0        0   0
x*y^2 * x| 0  0           0   1        0   0
  y^4 * x| 0  0           0   0        0   0
x*y^3 * x| 0  0           0   0        0   1
  y^5 * x| 0  0           0   0        0   0
x*y^4 * x|-2  0           0  14 1.15e-05   0
x*y^5 * x| 0 -2 9.999999e-06   0        0  14
         |         x*y^2 y^4  x*y^3 y^5        x*y^4 x*y^5
---------+------------------------------------------------
    1 * x|             0   0      0   0            0     0
    y * x|             0   0      0   0            0     0
    x * x|             0   0      0   0            0     0
  y^2 * x|             1   0      0   0            0     0
  x*y * x|        -2e-06   0      0   0            0     0
  y^3 * x|             0   0      1   0            0     0
x*y^2 * x|             0  -2 -2e-06   0            0     0
  y^4 * x|             0   0      0   0            1     0
x*y^3 * x|             0   0      0  -2       -2e-06     0
  y^5 * x|             0   0      0   0            0     1
x*y^4 * x|             0 -21 -4e-05   0            0 1e-05
x*y^5 * x| -5.849999e-05   0      0 -21 6.999999e-05     0

The multiplication with 'y' matrix M_y is
         |        1            y         x       y^2       x*y           y^3
---------+------------------------------------------------------------------
    1 * y|        0            1         0         0         0             0
    y * y|        0            0         0         1         0             0
    x * y|        0            0         0         0         1             0
  y^2 * y|        0            0         0         0         0             1
  x*y * y|        0            0         0         0         0             0
  y^3 * y|        0            0         0         0         0             0
x*y^2 * y|        0            0         0         0         0             0
  y^4 * y|        0            0         0         0         0             0
x*y^3 * y|        0            0         0         0         0             0
  y^5 * y|0.9999999            0         0 -6.999999 -5.75e-06             0
x*y^4 * y|        0            0         0         0         0             0
x*y^5 * y|        0 6.249999e-06 0.9999999         0         0 -5.249999e-05
         |     x*y^2 y^4 x*y^3         y^5 x*y^4  x*y^5
---------+---------------------------------------------
    1 * y|         0   0     0           0     0      0
    y * y|         0   0     0           0     0      0
    x * y|         0   0     0           0     0      0
```

```
  y^2 * y|        0   0     0            0     0      0
  x*y * y|        1   0     0            0     0      0
  y^3 * y|        0   1     0            0     0      0
x*y^2 * y|        0   0     1            0     0      0
  y^4 * y|        0   0     0            1     0      0
x*y^3 * y|        0   0     0            0     1      0
  y^5 * y|        0  11 2e-05            0     0 -6e-06
x*y^4 * y|        0   0     0            0     0      1
x*y^5 * y| -6.999999  0     0 8.599999e-05    11      0

The matrix for eigenvector computation is (0.9338036,0.9351656) * [M_x,M_y] =
          |         1         y         x       y^2          x*y        y^3
-----------+------------------------------------------------------------------
    1 * ...|         0 0.9351656 0.9338036         0            0          0
    y * ...|         0         0         0 0.9351656    0.9338036          0
    x * ...|0.9338036         0         0 -1.867607    0.9351638          0
  y^2 * ...|         0         0         0         0            0 0.9351656
  x*y * ...|         0 0.9338036         0         0            0 -1.867607
  y^3 * ...|         0         0         0         0            0          0
x*y^2 * ...|         0         0         0 0.9338036            0          0
  y^4 * ...|         0         0         0         0            0          0
x*y^3 * ...|         0         0         0         0            0 0.9338036
  y^5 * ...|0.9351656         0         0 -6.546159 -5.377202e-06          0
x*y^4 * ...|-1.867607         0         0  13.07325  1.073874e-05          0
x*y^5 * ...|         0 -1.867601 0.9351749         0            0    13.0732
          |     x*y^2       y^4        x*y^3       y^5     x*y^4    x*y^5
-----------+-------------------------------------------------------------------
    1 * ...|         0         0            0         0         0         0
    y * ...|         0         0            0         0         0         0
    x * ...|         0         0            0         0         0         0
  y^2 * ...| 0.9338036         0            0         0         0         0
  x*y * ...| 0.9351638         0            0         0         0         0
  y^3 * ...|         0 0.9351656    0.9338036         0         0         0
x*y^2 * ...|         0 -1.867607    0.9351638         0         0         0
  y^4 * ...|         0         0            0 0.9351656 0.9338036         0
x*y^3 * ...|         0         0            0 -1.867607 0.9351638         0
  y^5 * ...|         0  10.28682  1.870331e-05        0         0  0.933798
x*y^4 * ...|         0 -19.60987 -3.735214e-05        0         0  0.935175
x*y^5 * ...| -6.546214         0            0 -19.60979  10.28689         0
```

The latter matrix is plugged into the automatically generated script to be loaded and executed in Maple for computing the solutions of F by the eigenvector method.

```
Digits:=7:
with(linalg):
F:=[x^6+3*x^4*y^2+3*x^2*y^4+y^6-4*x^2*y^2,x^2+2e-06*x*y+2*y^2-1]:
M:=[[0,0.9351656,0.9338036,0,0,0,0,0,0,0,0,0],
    [0,0,0,0.9351656,0.9338036,0,0,0,0,0,0,0],
    [0.9338036,0,0,-1.867607,0.9351638,0,0,0,0,0,0,0],
    [0,0,0,0,0,0.9351656,0.9338036,0,0,0,0,0],
    [0,0.9338036,0,0,0,-1.867607,0.9351638,0,0,0,0,0],
    [0,0,0,0,0,0,0,0.9351656,0.9338036,0,0,0],
    [0,0,0,0.9338036,0,0,0,-1.867607,0.9351638,0,0,0],
    [0,0,0,0,0,0,0,0,0,0.9351656,0.9338036,0],
    [0,0,0,0,0,0.9338036,0,0,0,-1.867607,0.9351638,0],
    [0.9351656,0,0,-6.546159,-5.377202e-06,0,0,10.28682,1.870331e-05,0,0,0.933798],
    [-1.867607,0,0,13.07325,1.073874e-05,0,0,-19.60987,-3.735214e-05,0,0,0.935175],
    [0,-1.867601,0.9351749,0,0,13.0732,-6.546214,0,0,-19.60979,10.28689,0]]:
E:=eigenvectors(M):
Z:=[]:
for i in E do
  val_x:=op(i[3])[3]/op(i[3])[1]:
```

```
  val_y:=op(i[3])[2]/op(i[3])[1]:
  Z:=[op(Z),[x=val_x,y=val_y]]:
od:
for sol in Z do subs(sol,F) od;
```

The execution of the script in Maple gives the solutions (x, y) of F listed in the following table. We also provide an estimation of the sensitivity of the root-finding problem for each solution of F: it is an absolute condition number $\|J_F^{-1}(x, y)\|$ computed as the maximum matrix norm of the inverse Jacobian J_F^{-1} of F at the solution (x, y).

x	y	$\|J_F^{-1}(x,y)\|$
$-0.0005348599 + 4.436669\,i$	$-3.215999 - 0.0003984294\,i$	0.3848256
0.7584902	0.4608070	0.8860444
-0.2972799	-0.6751408	1.773081
$-0.0005348599 - 4.436669\,i$	$-3.215999 + 0.0003984294\,i$	0.3848256
0.2972352	-0.6750343	1.773397
$0.0005529034 + 4.436667\,i$	$3.215987 - 0.0004164831\,i$	0.3848113
$0.0005529034 - 4.436667\,i$	$3.215987 + 0.0004164831\,i$	0.3848113
-0.7584901	-0.4608068	0.8860444
0.2972791	0.6751418	1.773083
0.7584943	-0.4608175	0.8860334
-0.7584929	0.4608168	0.8860360
-0.2972355	0.6750300	1.773401

The quality of the attained solutions is checked by the substitution of the attained points into the original system F in the loop at the last line of the script.

F_1	F_2
$-0.8020 - 0.1734314\,i$	$0.00127 + 0.000350864\,i$
0.0000027	-0.0000057
0.0000271	0.000006
$-0.8020 + 0.1734314\,i$	$0.00127 - 0.000350864\,i$
-0.0000250	-0.0003090
$-0.8242 + 0.1691950\,i$	$0.00113 - 0.000422982\,i$
$-0.8242 - 0.1691950\,i$	$0.00113 + 0.000422982\,i$
0.0000028	-0.0000063
0.0000283	0.000008
0.0000047	0.000018
0.0000031	0.000015
-0.0000282	-0.0003205

The large residuals for complex solutions are no surprise since the determination of these zeros is intrinsically ill-conditioned in the original system F w.r.t. small changes in the system (not measured by the condition numbers in the table above!) and the poor conditioning of the problem is naturally inherited by the attained extended Gröbner basis but at least it does not make the situation worse as it well happens using conventional Gröbner bases. Indeed, the order of magnitude of at most 10^{-4} of the residuals for the real solutions is reasonably small for a result of numerical computation carried with single precision floating-point arithmetic: The order of residuals and absolute condition numbers indicate that both coordinates for each real solution are valid up to at least 3 decimal digits. This statement can easily be verified by comparison with the "exact" solutions (x^*, y^*) of F listed in the table below computed independently with an extra high precision using `solve`(F) in Maple.

x^*	y^*
$-0.0000109321183 + 4.43437739\,i$	$-3.21431974 - 0.00000975799701\,i$
0.758489344	0.460810816
-0.297315533	-0.675130755
$-0.0000109321183 - 4.43437739\,i$	$-3.21431974 + 0.00000975799701\,i$
0.297315833	-0.675130986
$0.0000109321183 + 4.43437739\,i$	$3.21431974 - 0.00000975799701\,i$
$0.0000109321183 - 4.43437739\,i$	$3.21431974 + 0.00000975799701\,i$
-0.758489344	-0.460810816
0.297315533	0.675130755
0.758489509	-0.460811439
-0.758489509	0.460811439
-0.297315833	0.675130986

For comparison we compute the zeros of F via the genuine Gröbner basis also attained by our algorithm using the same single precision floating-point arithmetic but with the *"fake" extension* $\underset{\mathbb{T}o}{<}$ of the term order – an option provided in our implementation for computing conventional Gröbner bases. The fake extension is indeed a switch which disables all e-degree mechanisms and hence turns our implementation into a variant of the usual Buchberger algorithm.

```
Alphabet:            x>y
Extended term order: To(Total degree)
Coefficients domain: float
The normalized and translated input basis contains 2 polynomials:
F1=0.125*x^6+0.375*x^4*y^2+0.375*x^2*y^4+0.125*y^6-0.5*x^2*y^2
F2=0.25*x^2+5e-07*x*y+0.5*y^2-0.25

................................................
<big skip>
................................................

The minimal Groebner basis with 3 polynomials is
F2=0.25*x^2+5e-07*x*y+0.5*y^2-0.25
R4=-3e-07*x*y^5-0.05*y^6+1e-06*x*y^3+0.55*y^4-3e-07*x*y-0.35*y^2+0.05
R9=-0.05*y^7-2.3e-06*x*y^4+0.55*y^5+1.8e-06*x*y^2-0.35*y^3-3e-07*x+0.05*y
```

```
Statistics:
   E-shifts excluded by co-prime: 3
    S-pairs excluded by co-prime: 1
      E-shifts excluded by chain: 0
       S-pairs excluded by chain: 0
      Number of row eliminations: 50
E-shifts delivered for reduction: 0
 S-pairs delivered for reduction: 2
             symbolic reductions: 16
The reduced Groebner basis contains 3 polynomials:
F2=0.25*x^2+5e-07*x*y+0.5*y^2-0.25
R4=-3e-07*x*y^5-0.05*y^6+1e-06*x*y^3+0.55*y^4-3e-07*x*y-0.35*y^2+0.05
R9=-0.05*y^7-2.3e-06*x*y^4+0.55*y^5+1.8e-06*x*y^2-0.35*y^3-3e-07*x+0.05*y
The Groebner basis is stable
The ideal is zero-dimensional
The normal set is [1,y,x,y^2,x*y,y^3,x*y^2,y^4,x*y^3,y^5,x*y^4,y^6]
The system has 12 solutions
The border set contains 8 terms: [x^2,x^2*y,x^2*y^2,x^2*y^3,x*y^5,x^2*y^4,y^7,x*y^6]
The border basis with 8 polynomials is
    x^2: F2=x^2+2e-06*x*y+2*y^2-1
  x^2*y: B1=x^2*y+2e-06*x*y^2+2*y^3-y
x^2*y^2: B2=x^2*y^2+2e-06*x*y^3+2*y^4-y^2
x^2*y^3: B3=x^2*y^3+2e-06*x*y^4+2*y^5-y^3
  x*y^5: R4=x*y^5+166666.7*y^6-3.333333*x*y^3-1833333*y^4+x*y+1166667*y^2-166666.7
x^2*y^4: B4=x^2*y^4+1.666667*y^6+6.666667e-06*x*y^3
              +2.666667*y^4-2e-06*x*y-2.333333*y^2+0.3333333
    y^7: R9=y^7+4.599999e-05*x*y^4-11*y^5
              -3.599999e-05*x*y^2+6.999999*y^3+5.999999e-06*x-0.9999999*y
  x*y^6: B5=x*y^6-11*x*y^4+7*x*y^2-0.9999999*x
The multiplication with 'x' matrix M_x is
         |         1 y         x      y^2    x*y y^3
---------+----------------------------------------
    1 * x|         0 0         1        0      0   0
    y * x|         0 0         0        0      1   0
    x * x|         1 0         0       -2 -2e-06   0
  y^2 * x|         0 0         0        0      0   0
  x*y * x|         0 1         0        0      0  -2
  y^3 * x|         0 0         0        0      0   0
x*y^2 * x|         0 0         0        1      0   0
  y^4 * x|         0 0         0        0      0   0
x*y^3 * x|         0 0         0        0      0   1
  y^5 * x|  166666.7 0         0 -1166667     -1   0
x*y^4 * x|-0.3333333 0         0 2.333333  2e-06   0
  y^6 * x|         0 0 0.9999999        0      0   0
         |  x*y^2       y^4        x*y^3 y^5  x*y^4       y^6
---------+-----------------------------------------------------
    1 * x|      0         0            0   0      0         0
    y * x|      0         0            0   0      0         0
    x * x|      0         0            0   0      0         0
  y^2 * x|      1         0            0   0      0         0
  x*y * x| -2e-06         0            0   0      0         0
  y^3 * x|      0         0            1   0      0         0
x*y^2 * x|      0        -2       -2e-06   0      0         0
  y^4 * x|      0         0            0   0      1         0
x*y^3 * x|      0         0            0  -2 -2e-06         0
  y^5 * x|      0   1833333     3.333333   0      0 -166666.7
x*y^4 * x|      0 -2.666667 -6.666667e-06   0      0 -1.666667
  y^6 * x|     -7         0            0   0     11         0


The multiplication with 'y' matrix M_y is
         |       1         y             x      y^2 x*y       y^3
---------+-------------------------------------------------------
    1 * y|       0         1             0        0   0         0
```

```
    y * y|      0         0             0        1   0         0
    x * y|      0         0             0        0   1         0
  y^2 * y|      0         0             0        0   0         1
  x*y * y|      0         0             0        0   0         0
  y^3 * y|      0         0             0        0   0         0
x*y^2 * y|      0         0             0        0   0         0
  y^4 * y|      0         0             0        0   0         0
x*y^3 * y|      0         0             0        0   0         0
  y^5 * y|      0         0             0        0   0         0
x*y^4 * y|166666.7        0             0 -1166667  -1         0
  y^6 * y|      0 0.9999999 -5.999999e-06        0   0 -6.999999
         |       x*y^2     y^4    x*y^3 y^5         x*y^4       y^6
---------+-----------------------------------------------------------
    1 * y|           0       0        0   0            0         0
    y * y|           0       0        0   0            0         0
    x * y|           0       0        0   0            0         0
  y^2 * y|           0       0        0   0            0         0
  x*y * y|           1       0        0   0            0         0
  y^3 * y|           0       1        0   0            0         0
x*y^2 * y|           0       0        1   0            0         0
  y^4 * y|           0       0        0   1            0         0
x*y^3 * y|           0       0        0   0            1         0
  y^5 * y|           0       0        0   0            0         1
x*y^4 * y|           0 1833333 3.333333   0            0 -166666.7
  y^6 * y| 3.599999e-05       0        0  11 -4.599999e-05         0
```

```
The matrix for eigenvector computation is (0.9561942,0.5756872) * [M_x,M_y] =
           |         1         y         x       y^2       x*y       y^3
-----------+------------------------------------------------------------
    1 * ...|         0 0.5756872 0.9561942         0         0         0
    y * ...|         0         0         0 0.5756872 0.9561942         0
    x * ...|0.9561942         0         0 -1.912388 0.5756853         0
  y^2 * ...|         0         0         0         0         0 0.5756872
  x*y * ...|         0 0.9561942         0         0         0 -1.912388
  y^3 * ...|         0         0         0         0         0         0
x*y^2 * ...|         0         0         0 0.9561942         0         0
  y^4 * ...|         0         0         0         0         0         0
x*y^3 * ...|         0         0         0         0         0 0.9561942
  y^5 * ...| 159365.7         0         0  -1115560 -0.9561942         0
x*y^4 * ...| 95947.55         0         0 -671632.8 -0.5756853         0
  y^6 * ...|         0 0.5756872 0.9561906         0         0  -4.02981
           |     x*y^2       y^4     x*y^3       y^5     x*y^4       y^6
-----------+-------------------------------------------------------------
    1 * ...|         0         0         0         0         0         0
    y * ...|         0         0         0         0         0         0
    x * ...|         0         0         0         0         0         0
  y^2 * ...| 0.9561942         0         0         0         0         0
  x*y * ...| 0.5756853         0         0         0         0         0
  y^3 * ...|         0 0.5756872 0.9561942         0         0         0
x*y^2 * ...|         0 -1.912388 0.5756853         0         0         0
  y^4 * ...|         0         0         0 0.5756872 0.9561942         0
x*y^3 * ...|         0         0         0 -1.912388 0.5756853         0
  y^5 * ...|         0   1753023  3.187314         0         0 -159365.1
x*y^4 * ...|         0   1055424  1.918951         0         0 -95949.47
  y^6 * ...| -6.693338         0         0  6.332559  10.51811         0
```

The generated Maple script for the actual determination of solutions is

```
Digits:=7:
with(linalg):
F:=[x^6+3*x^4*y^2+3*x^2*y^4+y^6-4*x^2*y^2,x^2+2e-06*x*y+2*y^2-1]:
M:=[[0,0.5756872,0.9561942,0,0,0,0,0,0,0,0,0],
    [0,0,0,0.5756872,0.9561942,0,0,0,0,0,0,0],
    [0.9561942,0,0,-1.912388,0.5756853,0,0,0,0,0,0,0],
```

```
    [0,0,0,0,0,0.5756872,0.9561942,0,0,0,0,0],
    [0,0.9561942,0,0,0,-1.912388,0.5756853,0,0,0,0,0],
    [0,0,0,0,0,0,0,0.5756872,0.9561942,0,0,0],
    [0,0,0,0.9561942,0,0,0,-1.912388,0.5756853,0,0,0],
    [0,0,0,0,0,0,0,0,0,0.5756872,0.9561942,0],
    [0,0,0,0,0,0.9561942,0,0,0,-1.912388,0.5756853,0],
    [159365.7,0,0,-1115560,-0.9561942,0,0,1753023,3.187314,0,0,-159365.1],
    [95947.55,0,0,-671632.8,-0.5756853,0,0,1055424,1.918951,0,0,-95949.47],
    [0,0.5756872,0.9561906,0,0,-4.02981,-6.693338,0,0,6.332559,10.51811,0]]:
E:=eigenvectors(M):
Z:=[]:
for i in E do
  val_x:=op(i[3])[3]/op(i[3])[1]:
  val_y:=op(i[3])[2]/op(i[3])[1]:
  Z:=[op(Z),[x=val_x,y=val_y]]:
od:
for sol in Z do subs(sol,F) od;
```

The execution of the script in Maple gives a greatly edifying result.

x	y
$0.7488262 + 3.265725\,i$	$1.811394 + 0.3990816\,i$
$-0.6816010 + 0.1531800\,i$	$-0.4552176 - 0.02168632\,i$
$-0.6816010 - 0.1531800\,i$	$-0.4552176 + 0.02168632\,i$
$0.6815958 + 0.1531741\,i$	$0.4552199 - 0.02168470\,i$
-0.4912186	0.07354324
$0.6815958 - 0.1531741\,i$	$0.4552199 + 0.02168470\,i$
$-0.000002541411 + 0.4926204\,i$	$-0.000001833985 - 0.3854314\,i$
$0.7488262 - 3.265725\,i$	$1.811394 - 0.3990816\,i$
$-0.000002541411 - 0.4926204\,i$	$-0.000001833985 + 0.3854314\,i$
$-0.7488762 + 3.265731\,i$	$-1.811406 + 0.3990857\,i$
0.4912072	-0.07353520
$-0.7488762 - 3.265731\,i$	$-1.811406 - 0.3990857\,i$

The "solutions" attained via the genuine Gröbner basis fail to give meaningful information towards the zero set of F as it is easy to **see** at least for the two real points according to the geometrical visualization of the real solutions on figure 4.3.3 (page303). The ridiculously high magnitudes of residuals computed at the last line of the script confirm this visual feeling.

F_1	F_2
$655.1551 + 669.7167\,i$	$-4.860456 + 7.782509\,i$
$-0.1788468 - 0.0934649\,i$	$-0.1453780 - 0.1693274\,i$
$-0.1788468 + 0.0934649\,i$	$0.1453780 + 0.1693274\,i$
$-0.1788417 + 0.0934564\,i$	$-0.1453791 + 0.1693205\,i$
0.009794884	-0.7478872
$-0.1788417 - 0.0934564\,i$	$-0.1453791 - 0.1693205\,i$
$-0.2040877 - 0.0000006161495\,i$	$-1.539790 + 0.000000323600\,i$
$655.1551 - 669.7167\,i$	$-4.860456 - 7.782509\,i$
$-0.2040877 + 0.0000006161495\,i$	$-1.539790 - 0.000000323600\,i$
$655.2564 - 669.7014\,i$	$-4.860336 - 7.782892\,i$
0.009793998	-0.7479008
$655.2564 + 669.7014\,i$	$-4.860336 + 7.782892\,i$

Despite the demonstrated disregarding results the genuine Gröbner basis **can** be used for the numerical determination of the solutions being attained either exactly or numerically with an extra high (greater than 12) number of digits carried in the computation. However the high sensitivity of the genuine Gröbner basis to small changes in the coefficients of the original system F clearly implies the intrinsic ill-conditioning of the zero determination problem using this basis no matter of the utilized numerical method which therefore must also be carried with extra high precision arithmetic including an unnecessary additional cost in the resolution process. Furthermore, the latter requirement for an extra high accuracy of the floating-point representation as well as possibly the computation of the genuine Gröbner basis with exact arithmetic are barely reasonable from the common sense point of view since the initial system is known with limited meaningful accuracy.

4.3.4 Singular system

The system $F{=}\{F_1, F_2\}$ in $\mathbb{C}[x, y]$ with

$$\begin{aligned} F_1 =& y^2 - x^3 - 0.0001\, yx - 1, \\ F_2 =& y^2 - (x + 1.1)\, x^2 \end{aligned}$$

is near singular: it has five solutions but one of them,

$$\left[\ x \approx 0.121 \times 10^9,\ y \approx 0.1331 \times 10^{13}\ \right]$$

with the order of magnitude 10^{13}, is extremely sensitive to small perturbations of F as it is *"close to infinity"* and disappears to infinity under a small change in the coefficients to a system like

$$\widetilde{F} = \left\{\ y^2 - x^3 - 1,\ y^2 - (x + 1.1)\, x^2\ \right\}$$

attained by excluding the monomial $0.0001\, yx$ from F_1: Therefore any representation of the ideal $\langle F \rangle$ and, in particular, any genuine or extended Gröbner basis for any term ordering is poorly conditioned w.r.t. the determination of the zeros. The remaining four solutions, all with order one, are rather well-conditioned as it is clearly seen on figure 4.3.4.

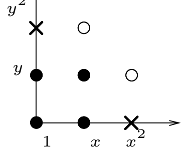
In actual fact, our algorithm is not intended for near singular systems. However it is interesting to see how our approach behaves on this example. Fix the *total degree (reverse) lexicographical order induced by $x{<}y$*. With the relatively inaccurate hardware single precision floating-point arithmetic, our implementation can handle this particular example and give a meaningful result

$$\begin{aligned} \big\{ -0.00009090909\, yx + \underline{x^2} - 0.9090909,& \\ \underline{y^2} - 0.0001\, yx - 0.00008264463\, y - 0.9090909\, x - 1 \big\}& \end{aligned}$$

which is a slightly perturbed version of the genuine Gröbner basis

$$\left\{ 11\, \underline{x^2} - 10,\ 11\, \underline{y^2} - 11 - 10\, x \right\}$$

of the system $\widetilde{F}$; the leading terms are $\{x^2, y^2\}$ and the associated normal set is



$$N(A) = \{1, x, y, x^2, y^2\}.$$

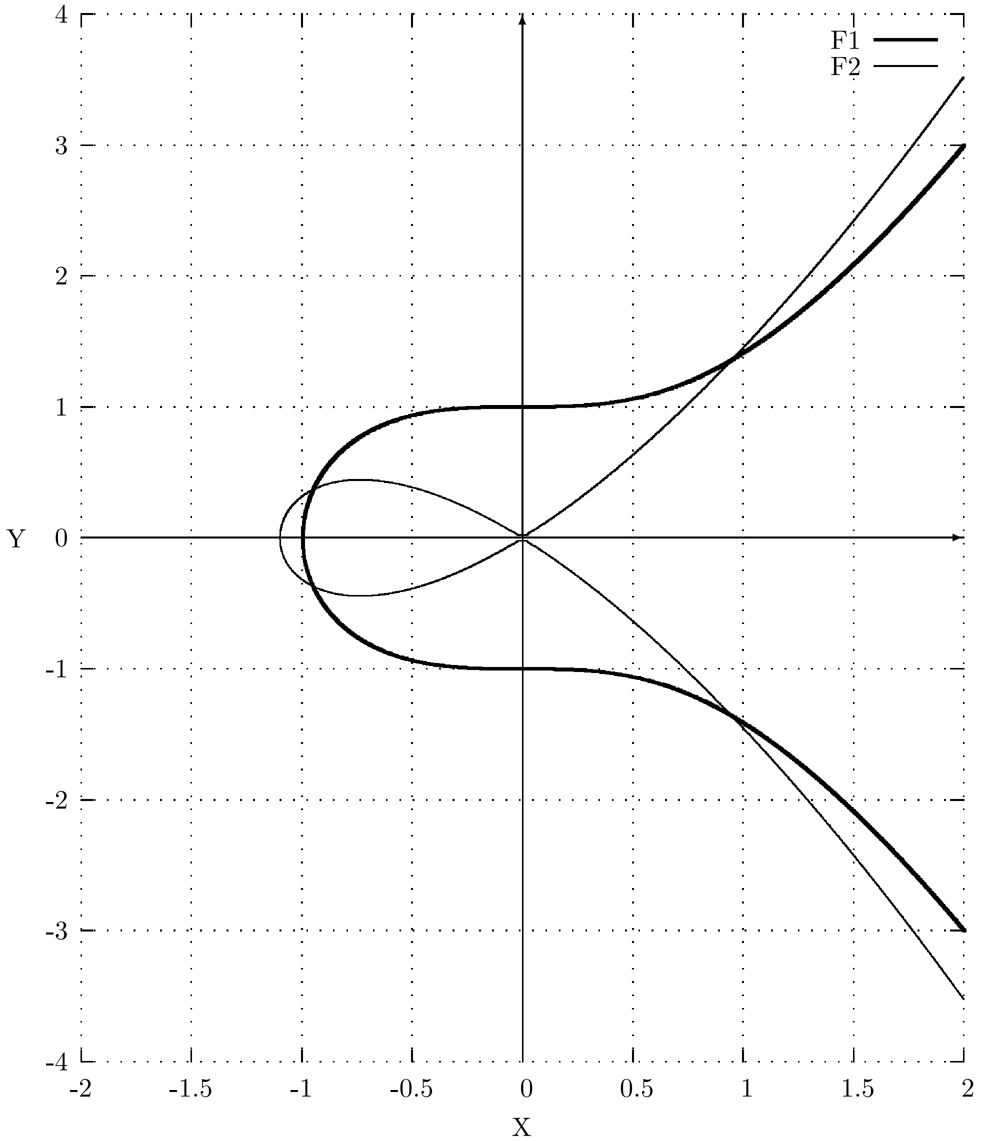
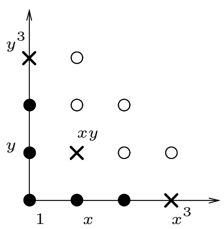


Figure 4.4: The system F from page 326

To understand the meaning and the numerical validity of this result let us first analyze the genuine Gröbner basis of F, computed by our algorithm with the fake extension of the same term order (an option to simulate a variant of the conventional Buchberger algorithm) with an extra high precision of the floating-point numbers representation (1024 bits in mantissa). Truncated to the 7 digits decimal approximation it is

$$\begin{aligned} A = \{A_1,\ A_2,\ A_3\} \quad \text{with} & \\ A_1 &= -\,0.5\,\underline{x^3} + 0.5\,y^2 - 0.55\,x^2 \\ A_2 &= -\,0.00005\,\underline{yx} + 0.55\,x^2 - 0.5 \\ A_3 &= -\,3.756574 \times 10^{-13}\,\underline{y^3} + 0.5\,y^2 \\ &\qquad - 0.55\,x^2 - 0.00004132231\,y - 0.4545455\,x \end{aligned}$$

with the leading terms $\{x^3,\ yx,\ y^3\}$ and the associated normal set



$$N(A) = \{1, x, y, x^2, y^2\}.$$

Naturally A is a poorly conditioned representation of $\langle F \rangle$, at least w.r.t. the determination of the solution near infinity, as this is intrinsic to F. As a confirmation of this fact, both multiplication matrices (constructed from A) of $\mathbb{C}[x,y]/\langle F \rangle$ in the linear basis represented by $N(A)$ have a row with extra large (relative) magnitude of coefficients.

```
The multiplication with 'y' matrix M_y is
       |     1          x         y          x^2       y^2
-------+-----------------------------------------------------
  1 * y|     0          0         1            0         0
  x * y|-10000          0         0        11000         0
  y * y|     0          0         0            0         1
x^2 * y|     0     -10000         0       -12100     11000
y^2 * y|     0 -1.21e+12 -1.1e+08 -1.4641e+12 1.331e+12
```

```
The multiplication with 'x' matrix M_x is
       |     1         x      y         x^2      y^2
-------+---------------------------------------------
  1 * x|     0         1      0           0        0
  x * x|     0         0      0           1        0
  y * x|-10000         0      0       11000        0
x^2 * x|     0         0      0        -1.1        1
y^2 * x|     0 -1.1e+08 -10000 -1.331e+08 1.21e+08
```

Less expected is the fact that A fails to give a meaningful approximation for any solution of F, including the four rather well-conditioned in the original representation of the problem provided by F, using the eigenvector method with the matrix

```
The matrix for eigenvector computation is (0.6539773,0.9206424)*[M_y,M_x]=
         |        1             x             y           x^2          y^2
---------+------------------------------------------------------------------
  1 * ...|        0     0.9206424     0.6539773             0            0
  x * ...|-6539.773             0             0      7194.671            0
  y * ...|-9206.424             0             0      10127.07    0.6539773
x^2 * ...|        0     -6539.773             0     -7914.138     7194.671
y^2 * ...|        0 -7.914138e+11 -7.194671e+07 -9.576107e+11 8.705551e+11
```

by execution of the following automatically generated script in Maple.

```
Digits:=7:
with(linalg):
F:=[-x^3+y^2-0.0001*y*x-1,-x^3+y^2-1.1*x^2]:
M:=[[0,0.9206424,0.6539773,0,0],
   [-6539.773,0,0,7194.671,0],
   [-9206.424,0,0,10127.07,0.6539773],
   [0,-6539.773,0,-7914.138,7194.671],
   [0,-7.914138e+11,-7.194671e+07,-9.576107e+11,8.705551e+11]]:
E:=eigenvectors(M):
Z:=[]:
for i in E do
  val_y:=op(i[3])[3]/op(i[3])[1]:
  val_x:=op(i[3])[2]/op(i[3])[1]:
  Z:=[op(Z),[y=val_y,x=val_x]]:
od:
for sol in Z do subs(sol,F) od;
```

Note that the utilized precision of the eigenvector problem computation – 7 decimal digits which approximately corresponds to the accuracy provided by the hardware single precision floating-point arithmetic – should be considered as sufficient for finding the four well-conditioned zeros of F by an appropriate numerical routine due to the simplicity of the system and the geometry of the solutions seen of figure 4.3.4. The attained approximations

x	y
-0.2738755	-0.3855014
-54309.90	-76455.25
-0.8259698	-1.162619
4756715.0	6685865.0
0	0

are however totally wrong for all 5 solutions including the four rather well-conditioned in the original representation of the problem provided by F and the execution of the loop in the last line of the script gives ridiculously high residuals

F_1	F_2
-0.8308565	0.08664553
1601964.0	1601932.0
0.915085	1.164732
-1.076270×10^{35}	-1.07627×10^{35}
-1.0	0.0

Now let us look at the aforementioned result attained by our algorithm using the hardware single precision floating-point arithmetic. The input for the computation is the following.

```
alphabet = y,x;
termorder = tdeg;
extorder = eT;
domain = float;
ebound=2;
ideal = y^2-x^3-1/10000*x*y-1,y^2-(x+11/10)*x^2;
```

The initial normalization and translation of the input polynomials $F = \{F_1, F_2\}$ to the algebra of extended polynomials assigned e-degree one to the term xy in F_1 as it has a small coefficient.

```
Alphabet:             y>x
Extended term order: eT(Total degree)
Coefficients domain: float
E-bound:              2
The normalized and translated input basis contains 2 polynomials:
F1=-0.5*x^3+0.5*y^2-0.5-5e-05*y*x*[E]
F2=-0.5*x^3+0.5*y^2-0.55*x^2
```

The subsequent auto-reduction is conveniently subdivided into two phases which also appear in the flow of computation as two iterations of symbolic reduction and the row echelon elimination.

- At the first phase, F_1 is reduced to

$$F_1 = -0.5\,x^3 + 0.5\,y^2 - 0.5 - 0.00005\,yx{\cdot}e \xrightarrow{F_2} \\ 0.55\,x^2 - 0.5 - 0.00005\,yx{\cdot}e =: F_1$$

  modulo F_2 by elimination of the leading term x^3. The leading term x^2 of the reduced F_1 is the direct consequence of the initial translation of the input system to the algebra of extended polynomials. Without this initial translation, the leading term of the reduced F_1 would have been xy; but the term xy, greater than x^2 in the utilized total degree (reverse) lexicographical order induced by $x<y$, has been transformed into $xy{\cdot}e$, less than x^2 in the utilized extension of the term order.

From the numerical point of view, this observed phenomena makes perfect sense since we would not like to have the term xy, with a small coefficient, as the leading term of the reduced form. From the comparison with a flow of computation of a genuine Gröbner basis point of view, this is a significant modification of the input system F: By neglecting the genuine leading term xy of the reduced F_1, we consider F_1 if it did not have the term xy at all, which forces the subsequent computation to to take a path which it would also take in the computation of the genuine Gröbner basis of the system $\widehat{F}$ (attained from F by removing xy with a small coefficient) by a version of the standard Buchberger algorithm. The rest of the computation can be considered as a simulation of the computation of the genuine Gröbner basis of $\widehat{F}$ on the system F. Note that $\widehat{F}$ has four solutions and a numerically well-conditioned genuine Gröbner basis. Therefore it is no surprise that the computation results in an extended Gröbner basis, of a system in a close neighborhood of F but with 4 solutions, which is a slightly perturbed genuine Gröbner basis of $\widehat{F}$.

- The second phase is the reduction of F_2 modulo F_1. It would not have taken place at all in the computation of the genuine Gröbner basis, i.e. without the initial translation to the algebra of extended polynomials.

The following is the detailed log of the flow of the initial auto-reduction.

```
Entering auto-reduction of the basis.
Entering symbolic auto-reduction.
The set of polynomials for symbolic reduction is
F1=-0.5*x^3+0.5*y^2-0.5-5e-05*y*x*[E]
F2=-0.5*x^3+0.5*y^2-0.55*x^2
reducing supp(F1)=[y*x*[E],1,y^2,x^3]
with F1=-0.5*x^3+0.5*y^2-0.5-5e-05*y*x*[E]
found an auto-reducer F2 for x^3
with F2=-0.5*x^3+0.5*y^2-0.55*x^2
x^3 is added to the set of terms to keep
reducing supp(F2)=[x^2,y^2,x^3]
with F2=-0.5*x^3+0.5*y^2-0.55*x^2
The symbolically reduced set of polynomials is
F1=-0.5*x^3+0.5*y^2-0.5-5e-05*y*x*[E]
F2=-0.5*x^3+0.5*y^2-0.55*x^2
The set of reducible terms is [<empty>]
```

```
The set of terms to keep is [x^3]
Symbolic reduction finished with 0 reducers added to the basis
Entering elimination.
The initial polynomials
F1=-0.5*x^3+0.5*y^2-0.5-5e-05*y*x*[E]
F2=-0.5*x^3+0.5*y^2-0.55*x^2
The initial matrix
   |  x^3 y^2   x^2    1 y*x*[E]
---+----------------------------
 F2| -0.5 0.5 -0.55    .       .
 F1| -0.5 0.5     . -0.5  -5e-05
Eliminating syzygy term x^3 by F2
with F2=-0.5*x^3+0.5*y^2-0.55*x^2
F1 := F1 + F2 * -1 --->
F1 := -0.5*x^3+0.5*y^2-0.5-5e-05*y*x*[E] +
      -0.5*x^3+0.5*y^2-0.55*x^2 *
      -1 --->
F1=0.55*x^2-0.5-5e-05*y*x*[E] -n->
F1=0.55*x^2-0.5-5e-05*y*x*[E] -e->
F1=0.55*x^2-0.5-5e-05*y*x*[E]
   |  x^3 y^2   x^2    1 y*x*[E]
---+----------------------------
 F2| -0.5 0.5 -0.55    .       .
 F1|    .   .  0.55 -0.5  -5e-05
The matrix after elimination
   |  x^3 y^2   x^2    1 y*x*[E]
---+----------------------------
 F2| -0.5 0.5 -0.55    .       .
 F1|    .   .  0.55 -0.5  -5e-05
Leaving with 1 eliminations done.
Entering symbolic auto-reduction.
The set of polynomials for symbolic reduction is
F2=-0.5*x^3+0.5*y^2-0.55*x^2
F1=0.55*x^2-0.5-5e-05*y*x*[E]
reducing supp(F2)=[x^2,y^2,x^3]
with F2=-0.5*x^3+0.5*y^2-0.55*x^2
found an auto-reducer F1 for x^3
with F1=0.55*x^2-0.5-5e-05*y*x*[E]
x^3 is added to the set of reducible terms
Added the reducer for x^3 : x * F1 = R1=0.55*x^3-0.5*x-5e-05*y*x^2*[E]
found an auto-reducer F1 for x^2
with F1=0.55*x^2-0.5-5e-05*y*x*[E]
x^2 is added to the set of terms to keep
reducing supp(F1)=[y*x*[E],1,x^2]
with F1=0.55*x^2-0.5-5e-05*y*x*[E]
reducing supp(R1)=[y*x^2*[E],x,x^3]
with R1=0.55*x^3-0.5*x-5e-05*y*x^2*[E]
found an auto-reducer F1 for y*x^2*[E]
with F1=0.55*x^2-0.5-5e-05*y*x*[E]
y*x^2*[E] is added to the set of reducible terms
Added the reducer for y*x^2*[E] : y*[E] * F1 = R2=0.55*y*x^2*[E]-0.5*y*[E]
reducing supp(R2)=[y*[E],y*x^2*[E]]
with R2=0.55*y*x^2*[E]-0.5*y*[E]
```

```
The symbolically reduced set of polynomials is
F2=-0.5*x^3+0.5*y^2-0.55*x^2
F1=0.55*x^2-0.5-5e-05*y*x*[E]
R1=0.55*x^3-0.5*x-5e-05*y*x^2*[E]
R2=0.55*y*x^2*[E]-0.5*y*[E]
The set of reducible terms is [y*x^2*[E],x^3]
The set of terms to keep is [x^2]
Symbolic reduction finished with 2 reducers added to the basis
Entering elimination.
The initial polynomials
F2=-0.5*x^3+0.5*y^2-0.55*x^2
F1=0.55*x^2-0.5-5e-05*y*x*[E]
R1=0.55*x^3-0.5*x-5e-05*y*x^2*[E]
R2=0.55*y*x^2*[E]-0.5*y*[E]
The initial matrix
   |  x^3 y*x^2*[E] | y^2   x^2    x    1 y*x*[E] y*[E]
---+---------------------------------------------------
 F2| -0.5         . | 0.5 -0.55    .    .       .     .
 R1| 0.55    -5e-05 |   .     .  -0.5   .       .     .
 R2|    .      0.55 |   .     .    .    .       .  -0.5
 F1|    .         . |   .  0.55    . -0.5  -5e-05     .
Eliminating reducible y*x^2*[E] by R2
with R2=0.55*y*x^2*[E]-0.5*y*[E]
R1 := R1 + R2 * 9.090909e-05 --->
R1 := 0.55*x^3-0.5*x-5e-05*y*x^2*[E] +
      0.55*y*x^2*[E]-0.5*y*[E] *
      9.090909e-05 --->
R1=0.55*x^3-0.5*x-4.545454e-05*y*[E] -n->
R1=0.55*x^3-0.5*x-4.545454e-05*y*[E] -e->
R1=0.55*x^3-0.5*x-4.545454e-05*y*[E]
   | y*x^2*[E]  x^3 | y^2   x^2    x    1 y*x*[E]        y*[E]
---+----------------------------------------------------------
 R2|      0.55    . |   .     .    .    .       .         -0.5
 F2|         . -0.5 | 0.5 -0.55    .    .       .            .
 R1|         . 0.55 |   .     .  -0.5   .       . -4.54545e-05
 F1|         .    . |   .  0.55    . -0.5  -5e-05            .
removing pivot at y*x^2*[E]
with R2=0.55*y*x^2*[E]-0.5*y*[E]
   |  x^3 | y^2   x^2    x    1 y*x*[E]        y*[E]
---+------------------------------------------------
 F2| -0.5 | 0.5 -0.55    .    .       .            .
 R1| 0.55 |   .     .  -0.5   .       . -4.54545e-05
 F1|    . |   .  0.55    . -0.5  -5e-05            .
Eliminating reducible x^3 by R1
with R1=0.55*x^3-0.5*x-4.545454e-05*y*[E]
F2 := F2 + R1 * 0.9090909 --->
F2 := -0.5*x^3+0.5*y^2-0.55*x^2 +
      0.55*x^3-0.5*x-4.545454e-05*y*[E] *
      0.9090909 --->
F2=0.5*y^2-0.55*x^2-0.4545454*x-4.132231e-05*y*[E] -n->
F2=0.5*y^2-0.55*x^2-0.4545454*x-4.132231e-05*y*[E] -e->
F2=0.5*y^2-0.55*x^2-0.4545454*x-4.132231e-05*y*[E]
   |  x^3 | y^2   x^2         x    1 y*x*[E]        y*[E]
```

```
---+---------------------------------------------------
 R1| 0.55 |    .     .      -0.5    .       . -4.54545e-05
 F2|    . | 0.5 -0.55 -0.454545    .       . -4.13223e-05
 F1|    . |    .  0.55         . -0.5  -5e-05           .
removing pivot at x^3
with R1=0.55*x^3-0.5*x-4.545454e-05*y*[E]
   | | y^2   x^2         x    1 y*x*[E]        y*[E]
---+---------------------------------------------
 F2| | 0.5 -0.55 -0.454545    .       . -4.13223e-05
 F1| |    .  0.55         . -0.5  -5e-05           .
Eliminating syzygy term x^2 by F1
with F1=0.55*x^2-0.5-5e-05*y*x*[E]
F2 := F2 + F1 * 1 --->
F2 := 0.5*y^2-0.55*x^2-0.4545454*x-4.132231e-05*y*[E] +
      0.55*x^2-0.5-5e-05*y*x*[E] *
      1 --->
F2=0.5*y^2-0.4545454*x-0.5-5e-05*y*x*[E]-4.132231e-05*y*[E] -n->
F2=0.5*y^2-0.4545454*x-0.5-5e-05*y*x*[E]-4.132231e-05*y*[E] -e->
F2=0.5*y^2-0.4545454*x-0.5-5e-05*y*x*[E]-4.132231e-05*y*[E]
   | | y^2  x^2         x    1 y*x*[E]        y*[E]
---+--------------------------------------------
 F2| | 0.5    . -0.454545 -0.5  -5e-05 -4.13223e-05
 F1| |    . 0.55         . -0.5  -5e-05           .
The matrix after elimination
   | | y^2  x^2         x    1 y*x*[E]        y*[E]
---+--------------------------------------------
 F2| | 0.5    . -0.454545 -0.5  -5e-05 -4.13223e-05
 F1| |    . 0.55         . -0.5  -5e-05           .
Leaving with 3 eliminations done.
Entering symbolic auto-reduction.
The set of polynomials for symbolic reduction is
F1=0.55*x^2-0.5-5e-05*y*x*[E]
F2=0.5*y^2-0.4545454*x-0.5-5e-05*y*x*[E]-4.132231e-05*y*[E]
reducing supp(F1)=[y*x*[E],1,x^2]
with F1=0.55*x^2-0.5-5e-05*y*x*[E]
reducing supp(F2)=[y*[E],y*x*[E],1,x,y^2]
with F2=0.5*y^2-0.4545454*x-0.5-5e-05*y*x*[E]-4.132231e-05*y*[E]
The symbolically reduced set of polynomials is
F1=0.55*x^2-0.5-5e-05*y*x*[E]
F2=0.5*y^2-0.4545454*x-0.5-5e-05*y*x*[E]-4.132231e-05*y*[E]
The set of reducible terms is [<empty>]
The set of terms to keep is [<empty>]
Symbolic reduction finished with 0 reducers added to the basis
No reducible terms found - exit reduction.
The auto-reduced set of polynomials is
F1=0.55*x^2-0.5-5e-05*y*x*[E]
F2=0.5*y^2-0.4545454*x-0.5-5e-05*y*x*[E]-4.132231e-05*y*[E]
```

The result of the auto-reduction with leading terms $\{x^2, y^2\}$ appears to be a Gröbner basis in the algebra of extended polynomials since the reduction of any possible critical element is detected to be redundant by the co-prime criteria.

```
The current critical set:
S-pairs: []
E-orbit: []

Forming critical elements from
F1=0.55*x^2-0.5-5e-05*y*x*[E]
F2=0.5*y^2-0.4545454*x-0.5-5e-05*y*x*[E]-4.132231e-05*y*[E]

The e-shift of F1 is excluded by co-prime.
The e-shift of F2 is excluded by co-prime.
The s-pair (F1,F2) is excluded by co-prime.
2 e-shifts and 1 s-pairs excluded by co-prime.
The critical set before chain:
S-pairs: []
E-orbit: []
The critical set after chain:
S-pairs: []
E-orbit: []
```

The attained Gröbner basis is indeed a minimal Gröbner basis and therefore no more actions are applied to it.

```
The minimal Groebner basis with 2 polynomials is
F1=0.55*x^2-0.5-5e-05*y*x*[E]
F2=0.5*y^2-0.4545454*x-0.5-5e-05*y*x*[E]-4.132231e-05*y*[E]

Statistics:
   E-shifts excluded by co-prime: 2
    S-pairs excluded by co-prime: 1
      E-shifts excluded by chain: 0
       S-pairs excluded by chain: 0
      Number of row eliminations: 4
E-shifts delivered for reduction: 0
 S-pairs delivered for reduction: 0
             symbolic reductions: 2
The reduced Groebner basis contains 2 polynomials:
F1=0.55*x^2-0.5-5e-05*y*x*[E]
F2=0.5*y^2-0.4545454*x-0.5-5e-05*y*x*[E]-4.132231e-05*y*[E]
```

The leading terms of the attained totally reduced Gröbner basis in the algebra of extended polynomials are terms in the main variables and furthermore they are pure powers of all main variables x and y. Therefore, the basis is suitable for the translation into an extended Gröbner basis in the original algebra $\mathbb{C}[x, y]$, expansion to the border basis and the subsequent construction of multiplication matrices. Note that the associated normal set contains four terms (in the main variables) and hence we have arrived to a system with four solutions.

```
The Groebner basis is stable
The ideal is zero-dimensional
```

```
The normal set is [1,x,y,y*x]
The system has 4 solutions
The border set contains 4 terms: [x^2,y^2,y*x^2,y^2*x]
The border basis with 4 polynomials is
  x^2: F1=-9.090909e-05*y*x+x^2-0.9090909
  y^2: F2=y^2-1e-04*y*x-8.264463e-05*y-0.9090909*x-1
y*x^2: B1=y*x^2-0.9090909*y-9.090909e-05*x-7.513148e-05
y^2*x: B2=y^2*x-0.0001652892*y*x-9.090909e-05*y-x-0.8264462
The multiplication with 'y' matrix M_y is
       |        1         x            y          y*x
-------+--------------------------------------------
  1 * y|        0         0            1            0
  x * y|        0         0            0            1
  y * y|        1 0.9090909 8.264463e-05        1e-04
y*x * y|0.8264462         1 9.090909e-05 0.0001652892

The multiplication with 'x' matrix M_x is
       |           1            x         y          y*x
-------+-----------------------------------------------
  1 * x|           0            1         0            0
  x * x|   0.9090909            0         0 9.090909e-05
  y * x|           0            0         0            1
y*x * x|7.513148e-05 9.090909e-05 0.9090909            0

The matrix for eigenvector computation is (0.6113505,0.5579273)*[M_y,M_x]=
         |        1         x            y          y*x
---------+--------------------------------------------
  1 * ...|        0 0.5579273    0.6113505            0
  x * ...|0.5072066         0            0    0.6114012
  y * ...|0.6113505 0.5557731 5.052483e-05    0.5579884
y*x * ...|0.5052902 0.6114012    0.5072621 0.0001010497
```

The latter matrix is plugged into the automatically generated script to be loaded and executed in Maple for computing the solutions of F by the eigenvector method.

```
Digits:=7:
with(linalg):
F:=[-x^3+y^2-1e-04*y*x-1,-x^3+y^2-1.1*x^2]:
M:=[[0,0.5579273,0.6113505,0],
    [0.5072066,0,0,0.6114012],
    [0.6113505,0.5557731,5.052483e-05,0.5579884],
    [0.5052902,0.6114012,0.5072621,0.0001010497]]:
E:=eigenvectors(M):
Z:=[]:
for i in E do
  val_y:=op(i[3])[3]/op(i[3])[1]:
  val_x:=op(i[3])[2]/op(i[3])[1]:
  Z:=[op(Z),[y=val_y,x=val_x]]:
od:
for sol in Z do subs(sol,F) od;
```

The execution of the script in Maple gives fairly good approximations (x, y) of all four well-conditioned solutions of F listed in the table below. We also give an estimation of the sensitivity of the root-finding problem for each computed approximation: it is an absolute condition number $\|J_F^{-1}(x, y)\|$ computed as the maximum matrix norm of the inverse Jacobian J_F^{-1} of F at (x, y). For comparison, the "exact" values (x^*, y^*) of the well-conditioned solutions of F computed independently by `solve`(F) in Maple with extra high precision are provided.

x	y	$\|J_F^{-1}(x, y)\|$	x^*	y^*
0.9533939	−1.366191	1.317376	0.953400492	−1.36619346
−0.9534774	−0.3649726	2.192412	−0.953479179	−0.364973124
−0.9534544	0.3650117	2.192234	−0.953445998	0.365001745
0.9535246	1.366411	1.317726	0.953524701	1.36641277

The quality of the attained approximation of solutions is verified by computing the residuals in the loop at the last line of the script.

F_1	F_2
0.000011	0.0000252
−0.0000048	−0.000001
0.000030	0.0000125
−0.0000043	−0.000004

The attained maximal order of magnitude of residuals is 10^{-5}. Such an accuracy can be expected using the hardware single precision floating-point arithmetic: The order of residuals and absolute condition numbers indicate that both coordinates for each real solution should be valid up to at least 4 decimal digits. This statement is easy to verify by a straightforward comparison of the approximate solutions (x, y) with the respective "exact" values (x^*, y^*).

4.3.5 A system with 3 variables

Consider the system $F=\{F_1,\ F_2,\ F_3\}$ with

$$\begin{aligned}
F_1 =& -1.2891205 - 1.1844526\,x - 3.9125691\,y \\
&+ 1.7231360\,z - 0.76916180\,x^2 + 0.36755920\,y^2 \\
&\quad - 0.92113870\,z^2 + 4.8567277\,xy + 3.1793749\,xz - 4.2271665\,yz, \\
F_2 =& 4.4301261 + 2.9870548\,x + 0.098400300\,y \\
&+ 4.4705923\,z + 1.4294897\,x^2 - 2.4863027\,y^2 \\
&\quad - 3.5137711\,z^2 - 4.9254763\,xy - 1.5589943\,xz - 4.6888701\,yz, \\
F_3 =& -2.2885406 - 2.8790072\,x - 3.4757485\,y \\
&- 2.5171945\,z + 0.42275070\,x^2 - 3.2018649\,y^2 \\
&\quad + 2.3570262\,z^2 + 4.1527797\,xy - 0.043121200\,xz + 3.7500442\,yz
\end{aligned}$$

in $\mathbb{C}[x,y,z]$ equipped with the *total degree reverse lexicographical order* induced by $z<y<x$. The system has eight zeros, all well-conditioned. Our goal is the numerical determination of the zeros to a meaningful approximation using the hardware double precision floating-point arithmetic which is standard for a nontrivial numerical computation, where one must expect some intermediate loss of accuracy even with a stabilized measure.

The genuine (totally) reduced Gröbner basis of F, attained numerically, with an extra high precision (1024 bits in mantissa) floating-point arithmetic, by our algorithm specialized to a version of a standard Buchberger algorithm for computing Gröbner bases by the use of the fake extension of the term order, is

$$
\begin{aligned}
A = \{A_i\}_{i=1}^{6} \quad &\text{with} \\
A_1 = &- 7.952219934 \times 10^{-10}\, \underline{y^2} + 0.1360085506\, xz \\
&- 0.548165136\, yz - 0.259230258\, z^2 + 0.1189209519\, x \\
&- 0.1550609403\, y + 0.352426534\, z + 0.1568840673, \\
A_2 = &3.328290069 \times 10^{-10}\, \underline{x^2} - 0.185255217\, xz \\
&+ 0.7466475501\, yz + 0.3530936652\, z^2 - 0.1619804537\, x \\
&+ 0.2112061918\, y - 0.4800349215\, z - 0.2136894462, \\
A_3 = &2.612088516 \times 10^{-9}\, \underline{xy} - 0.1964457669\, xz \\
&+ 0.7917496398\, yz + 0.3744226879\, z^2 - 0.1717650674\, x \\
&+ 0.2239643391\, y - 0.5090319744\, z - 0.2265976002, \\
A_4 = &1.943285645 \times 10^{-10}\, \underline{yz^2} - 3.43638153 \times 10^{-10}\, z^3 \\
&- 0.1156883108\, xz + 0.4662669955\, yz + 0.2205001846\, z^2 \\
&- 0.1011536697\, x + 0.1318941934\, y - 0.2997725499\, z \\
&- 0.1334449379, \\
A_5 = &5.642407493 \times 10^{-11}\, \underline{xz^2} - 5.096807599 \times 10^{-10}\, z^3 \\
&- 0.1353824852\, xz + 0.5456418561\, yz + 0.2580369858\, z^2 \\
&- 0.1183735427\, x + 0.1543471728\, y - 0.3508042647\, z \\
&- 0.1561619078, \\
A_6 = &- 1.384325212 \times 10^{-17}\, \underline{z^4} - 1.347074763 \times 10^{-16}\, z^3 \\
&- 3.73867356 \times 10^{-8}\, xz + 1.506824739 \times 10^{-7}\, yz \\
&+ 7.125855721 \times 10^{-8}\, z^2 - 3.26896078 \times 10^{-8}\, x \\
&+ 4.262395484 \times 10^{-8}\, y - 9.687683211 \times 10^{-8}\, z \\
&- 4.312510552 \times 10^{-8}
\end{aligned}
$$

in 10-decimal approximation of the coefficients. It has the leading terms

$$\{\, y^2,\, x^2,\, xy,\, yz^2,\, xz^2,\, z^4 \,\}$$

underlined in the equations above and the associated normal set

$$N(A) = \{1, z, y, x, z^2, yz, xz, z^3\}.$$

In actual fact A is a very ill-conditioned representation of the ideal $\langle F \rangle$. The indication of this fact are the very small (relative) orders of all (!) leading coefficients of A and hence the expected jump in the normal set $N(A)$

under a small change in the coefficients of F. A visually more convincing indication of the particular ill-conditioning of A w.r.t. the determination of zeros of $\langle F \rangle$ are the extra high orders of magnitude of coefficients in all (!) nontrivial rows of the multiplication matrices of the quotient $\mathbb{C}[x,y,z]/\langle F \rangle$ in the basis represented by $N(A)$. The matrices are displayed with 10-decimal approximation.

```
The border basis with 11 polynomials is
  y^2: F1=y^2-1.710321793e+08*x*z+6.893234098e+08*y*z+3.259847692e+08*z^2
                    -1.495443447e+08*x+1.949907593e+08*y-4.43180064e+08*z
                                                         -1.972833606e+08
  x*y: F3=x*y-7.520639735e+07*x*z+3.031098046e+08*y*z+1.433422664e+08*z^2
                   -6.575775143e+07*x+8.574148145e+07*y-1.948754689e+08*z
                                                         -8.674958707e+07
  x^2: F2=x^2-5.566077871e+08*x*z+2.243336772e+09*y*z+1.060886094e+09*z^2
                   -4.866776943e+08*x+6.345786796e+08*y-1.442286915e+09*z
                                                         -6.420397313e+08
y*z^2: R1=y*z^2-1.768335777*z^3-5.953232409e+08*x*z+2.399374466e+09*y*z
                 +1.134677165e+09*z^2-5.205290844e+08*x+6.787174792e+08*y
                                       -1.542606722e+09*z-6.86697492e+08
x*z^2: R3=x*z^2-9.033037061*z^3-2.399374476e+09*x*z+9.670373096e+09*y*z
                  +4.573171755e+09*z^2-2.097926158e+09*x+2.73548433e+09*y
                                      -6.217279861e+09*z-2.767646754e+09
y^2*z: B1=y^2*z+1.745695483*z^3+5.536126217e+08*x*z-2.231265132e+09*y*z
                 -1.055177351e+09*z^2+4.840588295e+08*x-6.311639424e+08*y
                                      +1.434525807e+09*z+6.385848444e+08
x*y*z: B2=x*y*z+3.665381773*z^3+9.673549635e+08*x*z-3.898800923e+09*y*z
                 -1.843764049e+09*z^2+8.458201515e+08*x-1.102864256e+09*y
                                      +2.506618536e+09*z+1.115831169e+09
x^2*z: B3=x^2*z-2.443978921*z^3-2.73434618e+08*x*z+1.102043378e+09*y*z
                +5.211622806e+08*z^2-2.390813287e+08*x+3.117379603e+08*y
                                      -7.085261394e+08*z-3.15403218e+08
  z^4: F2=z^4+9.730912585*z^3+2.700719114e+09*x*z-1.088490426e+10*y*z
               -5.147530118e+09*z^2+2.361410998e+09*x-3.079042011e+09*y
                                    +6.998126689e+09*z+3.115243813e+09
y*z^3: B4=y*z^3-7.521738192*z^3-1.819853085e+09*x*z+7.334685967e+09*y*z
                 +3.468612677e+09*z^2-1.591213638e+09*x+2.074782258e+09*y
                                      -4.715619024e+09*z-2.099176485e+09
x*z^3: B5=x*z^3-3.295113171*z^3-7.679472035e+08*x*z+3.095113334e+09*y*z
                 +1.463695835e+09*z^2-6.714652218e+08*x+8.755229976e+08*y
                                      -1.989911405e+09*z-8.858169423e+08

The multiplication with 'x' matrix M_x is
           |               1                 z                 y                 x
    -------+-----------------------------------------------------------------------
      1 * x|               0                 0                 0                 1
      z * x|               0                 0                 0                 0
      y * x| 8.674958707e+07   1.948754689e+08 -8.574148145e+07   6.575775143e+07
      x * x| 6.420397313e+08   1.442286915e+09 -6.345786796e+08   4.866776943e+08
    z^2 * x| 2.767646754e+09   6.217279861e+09  -2.73548433e+09   2.097926158e+09
    y*z * x|-1.115831169e+09  -2.506618536e+09  1.102864256e+09  -8.458201515e+08
    x*z * x|  3.15403218e+08   7.085261394e+08 -3.117379603e+08   2.390813287e+08
    z^3 * x| 8.858169423e+08   1.989911405e+09 -8.755229976e+08   6.714652218e+08
           |             z^2               y*z               x*z               z^3
    -------+-------------------------------------------------------------------------
      1 * x|               0                 0                 0                 0
      z * x|               0                 0                 1                 0
```

```
     y * x| -1.433422664e+08 -3.031098046e+08  7.520639735e+07                0
     x * x| -1.060886094e+09 -2.243336772e+09  5.566077871e+08                0
   z^2 * x| -4.573171755e+09 -9.670373096e+09  2.399374476e+09      9.033037061
   y*z * x|  1.843764049e+09  3.898800923e+09 -9.673549635e+08     -3.665381773
   x*z * x| -5.211622806e+08 -1.102043378e+09   2.73434618e+08      2.443978921
   z^3 * x| -1.463695835e+09 -3.095113334e+09  7.679472035e+08      3.295113171

The multiplication with 'y' matrix M_y is
       |                1                z                y                x
-------+-----------------------------------------------------------------------
  1 * y|                0                0                1                0
  z * y|                0                0                0                0
  y * y|  1.972833606e+08   4.43180064e+08 -1.949907593e+08  1.495443447e+08
  x * y|  8.674958707e+07  1.948754689e+08 -8.574148145e+07  6.575775143e+07
z^2 * y|   6.86697492e+08  1.542606722e+09 -6.787174792e+08  5.205290844e+08
y*z * y| -6.385848444e+08 -1.434525807e+09  6.311639424e+08 -4.840588295e+08
x*z * y| -1.115831169e+09 -2.506618536e+09  1.102864256e+09 -8.458201515e+08
z^3 * y|  2.099176485e+09  4.715619024e+09 -2.074782258e+09  1.591213638e+09
       |              z^2              y*z              x*z              z^3
-------+-----------------------------------------------------------------------
  1 * y|                0                0                0                0
  z * y|                0                1                0                0
  y * y| -3.259847692e+08 -6.893234098e+08  1.710321793e+08                0
  x * y| -1.433422664e+08 -3.031098046e+08  7.520639735e+07                0
z^2 * y| -1.134677165e+09 -2.399374466e+09  5.953232409e+08      1.768335777
y*z * y|  1.055177351e+09  2.231265132e+09 -5.536126217e+08     -1.745695483
x*z * y|  1.843764049e+09  3.898800923e+09 -9.673549635e+08     -3.665381773
z^3 * y| -3.468612677e+09 -7.334685967e+09  1.819853085e+09      7.521738192

The multiplication with 'z' matrix M_z is
       |                1                z                y                x
-------+-----------------------------------------------------------------------
  1 * z|                0                1                0                0
  z * z|                0                0                0                0
  y * z|                0                0                0                0
  x * z|                0                0                0                0
z^2 * z|                0                0                0                0
y*z * z|  6.86697492e+08  1.542606722e+09 -6.787174792e+08  5.205290844e+08
x*z * z| 2.767646754e+09  6.217279861e+09  -2.73548433e+09  2.097926158e+09
z^3 * z|-3.115243813e+09 -6.998126689e+09  3.079042011e+09 -2.361410998e+09
       |             z^2              y*z              x*z              z^3
-------+-----------------------------------------------------------------------
  1 * z|                0                0                0                0
  z * z|                1                0                0                0
  y * z|                0                1                0                0
  x * z|                0                0                1                0
z^2 * z|                0                0                0                1
y*z * z|-1.134677165e+09 -2.399374466e+09  5.953232409e+08      1.768335777
x*z * z|-4.573171755e+09 -9.670373096e+09  2.399374476e+09      9.033037061
z^3 * z| 5.147530118e+09  1.088490426e+10 -2.700719114e+09     -9.730912585
```

Indeed, the numerical computation in Maple of the solutions of F by the eigenvector method, performed with 16 decimal digits (roughly the precision provided by the hardware double precision floating-point arithmetic) from the following multiplication matrix displayed in 10-decimal approximation gives a very poor approximation for all solutions.

```
The matrix for eigenvector computation is
(0.9838438987,0.7990874089,0.6611809539) * [M_x,M_y,M_z] =
         |               1               z               y               x
---------+-----------------------------------------------------------------
  1 * ...|               0    0.6611809539    0.7990874089    0.9838438987
  z * ...|               0               0               0               0
  y * ...| 2.429947014e+08  5.458666501e+08  -2.40170894e+08  1.841943655e+08
  x * ...| 7.009873751e+08  1.574707715e+09 -6.928413004e+08  5.313610714e+08
z^2 * ...| 3.271663693e+09  7.349510466e+09 -3.233644159e+09  2.479980088e+09
y*z * ...|-1.154057494e+09 -2.592490679e+09  1.140646358e+09 -8.747963947e+08
x*z * ...| 1.248576215e+09  2.804818837e+09 -1.234066688e+09  9.464432915e+08
z^3 * ...| 4.891912165e+08  1.098925902e+09 -4.835064041e+08  3.708157686e+08
         |             z^2             y*z             x*z             z^3
---------+-----------------------------------------------------------------
  1 * ...|               0               0               0               0
  z * ...|    0.6611809539    0.7990874089    0.9838438987               0
  y * ...|-4.015167388e+08 -8.490423887e+08  2.106610162e+08               0
  x * ...|-1.158289311e+09 -2.449304424e+09  6.077116612e+08               0
z^2 * ...|-5.405993365e+09 -1.143144749e+10  2.836325245e+09     10.96133421
y*z * ...| 1.906928015e+09  4.032366675e+09 -1.000494766e+09    -3.831936838
x*z * ...|-2.063107757e+09 -4.362622434e+09  1.082436514e+09     5.448005387
z^3 * ...|-8.083240598e+08 -1.709272193e+09   4.24097817e+08     2.818509206
```

To compute the actual solutions the following Maple script is automatically generated. Naturally the matrix for the computation is now carried with 16-decimal approximation.

```
Digits:=16:
with(linalg):
F:=[-0.7691618*x^2+4.8567277*x*y+0.3675592*y^2+3.1793749*x*z-4.2271665*y*z
        -0.9211387*z^2-1.1844526*x-3.9125691*y+1.723136*z-1.2891205,
     1.4294897*x^2-4.9254763*x*y-2.4863027*y^2-1.5589943*x*z-4.6888701*y*z
        -3.5137711*z^2+2.9870548*x+0.0984003*y+4.4705923*z+4.4301261,
     0.4227507*x^2+4.1527797*x*y-3.2018649*y^2-0.0431212*x*z+3.7500442*y*z
        +2.3570262*z^2-2.8790072*x-3.4757485*y-2.5171945*z-2.2885406
   ]:
M:=[[0,0.661180953863612,0.7990874088588811,0.9838438986703313,0,0,0,0],
    [0,0,0,0,0.661180953863612,0.7990874088588811,0.9838438986703313,0],
    [2.429947013918121e+08,5.458666501066016e+08,-2.401708940038986e+08,
     1.841943654632582e+08,-4.015167388021537e+08,-8.490423886739839e+08,
     2.10661016200137e+08,0],
    [7.00987375079152e+08,1.574707714522487e+09,-6.928413003633732e+08,
     5.313610713875608e+08,-1.158289311077568e+09,-2.449304424299706e+09,
     6.077116611535125e+08,0],
    [3.271663692667255e+09,7.349510466240104e+09,-3.233644159357681e+09,
     2.479980087585034e+09,-5.405993364504865e+09,-1.143144749274683e+10,
     2.836325244742877e+09,10.96133420676665],
    [-1.154057493569214e+09,-2.59249067860893e+09,1.14064635823245e+09,
     -8.747963947106615e+08,1.90692801547865e+09,4.032366675473316e+09,
     -1.00049476582947e+09,-3.83193683827671],
    [1.248576215044787e+09,2.804818837360931e+09,-1.234066687961219e+09,
     9.464432914553375e+08,-2.063107756812799e+09,-4.362622433729164e+09,
     1.082436513970985e+09,5.448005386939956],
    [4.891912165476822e+08,1.098925902116375e+09,-4.83506404100088e+08,
     3.708157685739185e+08,-8.083240597947429e+08,-1.709272193123887e+09,
     4.240978170205644e+08,2.818509205549668]
   ]:
E:=eigenvectors(M):
```

```
Z:=[]:
for i in E do
  val_x:=op(i[3])[4]/op(i[3])[1]:
  val_y:=op(i[3])[3]/op(i[3])[1]:
  val_z:=op(i[3])[2]/op(i[3])[1]:
  Z:=[op(Z),[x=val_x,y=val_y,z=val_z]]:
od:
for sol in Z do subs(sol,F) od;
```

The execution of the script gives the following approximations of the solutions of F displayed with 5 decimals to save space.

x	y	z
-0.59614	0.74115	-0.57121
$2.2354+0.79877\,i$	$0.99215+0.57173\,i$	$0.53738-1.1097\,i$
$2.2354-0.79877\,i$	$0.99215-0.57173\,i$	$0.53738+1.1097\,i$
$-0.39461+0.72008\,i$	$0.14136-0.32706\,i$	$-0.39390-0.078167\,i$
$-0.39461-0.72008\,i$	$0.14136+0.32706\,i$	$-0.39390+0.078167\,i$
-2.2332	-0.030855	-0.76436
$1.2026+106.91\,i$	$-2.1442+36.911\,i$	$2.7812+0.17025\,i$
$1.2026-106.91\,i$	$-2.1442-36.911\,i$	$2.7812-0.17025\,i$

The attained approximations are however very poor as it is revealed by a large magnitude of the following residuals, displayed with 5 digits, computed at the execution of the last line of the script.

F_1	F_2	F_3
-4.1127	1.7949	-5.9870
$1.1198+0.56292\,i$	$4.1122+0.67743\,i$	$-4.4071-0.71677\,i$
$1.1198-0.56292\,i$	$4.1122-0.67743\,i$	$-4.4071+0.71677\,i$
$-0.054254+0.45804\,i$	$-0.15614-0.32480\,i$	$0.25680+0.87265\,i$
$-0.054254-0.45804\,i$	$-0.15614+0.32480\,i$	$0.25680-0.87265\,i$
1.4479	-3.6981	9.9553
$-10879.0-911.77\,i$	$6562.4+1048.5\,i$	$-16915.0-215.64\,i$
$-10879.0+911.77\,i$	$6562.4-1048.5\,i$	$-16915.0+215.64\,i$

As we have seen, the genuine Gröbner basis A is not an appropriate representation of the ideal $\langle F\rangle$ for the determination of the solutions by a numerical computation performed with a reasonably precise (double precision floating-point) arithmetic.

The extended Gröbner basis of F has seven polynomials (displayed to 10 decimals)

$$\begin{aligned}
\Big\{ &- 5.846852878 \times 10^{-9}\, y^2 + \underline{xz} - 4.030372603\, yz - 1.905985005\, z^2 \\
&\qquad + 0.874363791\, x - 1.140082294\, y + 2.591208659\, z + 1.153486796, \\
&\underline{xy} - 0.4397207452\, y^2 + 1.216011672\, yz + 0.7474463093\, z^2 - \\
&\qquad 0.7358708369\, x - 0.5398735303\, y - 0.8774970266\, z - 0.7274838955, \\
&\underline{x^2} - 3.254403874\, y^2 - 3.485692372\, yz - 1.961302538\, z^2 + \\
&\qquad 0.5076417124\, x - 3.034730349\, y + 2.929843441\, z + 1.850451077, \\
&\underline{yz^2} - 1.768335777\, z^3 - 3.480767439\, y^2 \\
&\qquad - 14.3159674\, yz - 5.49018173\, z^2 + 1.395957588\, x \\
&\qquad\qquad - 6.764149709\, y + 14.29839085\, z + 5.891392731, \\
&\underline{y^2z} + 1.745695483\, z^3 + 3.236891583\, y^2 \\
&\qquad + 11.04554965\, yz + 4.320611944\, z^2 - 1.213755921\, x \\
&\qquad\qquad + 5.194924958\, y - 12.31739281\, z - 5.063196751, \\
&\underline{y^3} - 0.2203136853\, z^3 - 0.5341612361\, y^2 \\
&\qquad - 0.2749328878\, yz - 0.388239965\, z^2 - 0.2011188989\, x \\
&\qquad\qquad - 1.198251315\, y + 1.325384159\, z + 0.5338500312, \\
&\underline{z^4} + 9.730912585\, z^3 + 15.79070748\, y^2 \\
&\qquad + 67.30204789\, yz + 17.46708982\, z^2 - 5.490633436\, x \\
&\qquad\qquad + 31.45148404\, y - 65.29455451\, z - 25.68378653 \quad \Big\}
\end{aligned}$$

with the leading terms

$$\{ xz,\, xy,\, x^2,\, yz^2,\, y^2z,\, y^3,\, z^4 \}$$

and the associated normal set

$$N = \{ 1,\, z,\, y,\, x,\, z^2,\, yz,\, y^2,\, z^3 \};$$

it is delivered by our algorithm using the $\underset{e\mathbb{T}}{<}$ extension of the term order and the somewhat reasonable hardware double precision floating-point arithmetic. It is a much more appropriate representation of the ideal $\langle F \rangle$ for the determination of the zero set numerically as it is shown below when we compute the solutions from this extended Gröbner basis. The following is the input for our algorithm.

```
alphabet = x,y,z;
termorder = revtdeg;
domain = double;
extorder = eT;
ebound=2;
ideal =
-1.2891205-1.1844526*x-3.9125691*y+1.7231360*z-.76916180*x^2+.36755920*y^2
                      -.92113870*z^2+4.8567277*x*y+3.1793749*x*z-4.2271665*y*z,
4.4301261+2.9870548*x+.98400300e-1*y+4.4705923*z+1.4294897*x^2-2.4863027*y^2
                        -3.5137711*z^2-4.9254763*x*y-1.5589943*x*z-4.6888701*y*z,
-2.2885406-2.8790072*x-3.4757485*y-2.5171945*z+.42275070*x^2-3.2018649*y^2
                   +2.3570262*z^2+4.1527797*x*y-.43121200e-1*x*z+3.7500442*y*z;
```

Due to the excessive length of the computation we do not provide the printout of the flow of the algorithm but turn directly to the results. To save space, all numbers are displayed in 10 decimals. All values in the computation are naturally carried in double precision floating-point which is roughly 16 decimals.

```
Alphabet:            x>y>z
Extended term order: eT(Reverse total degree)
Coefficients domain: double
Zero test:           rel<1.490116119e-08
E-bound:             2
The normalized and translated input basis contains 3 polynomials:
F1=-0.096145225*x^2+0.6070909625*x*y+0.0459449*y^2+0.3974218625*x*z-0.5283958125*y*z
                -0.1151423375*z^2-0.148056575*x-0.4890711375*y+0.215392*z-0.1611400625
F2=0.1786862125*x^2-0.6156845375*x*y-0.3107878375*y^2-0.1948742875*x*z-0.5861087625*y*z
                -0.4392213875*z^2+0.37338185*x+0.0123000375*y+0.5588240375*z+0.5537657625
F3=0.0528438375*x^2+0.5190974625*x*y-0.4002331125*y^2-0.00539015*x*z+0.468755525*y*z
                +0.294628275*z^2-0.3598759*x-0.4344685625*y-0.3146493125*z-0.286067575

..................................................
<big skip>
..................................................

Statistics:
   E-shifts excluded by co-prime: 7
    S-pairs excluded by co-prime: 7
      E-shifts excluded by chain: 0
       S-pairs excluded by chain: 4
      Number of row eliminations: 1113
E-shifts delivered for reduction: 0
 S-pairs delivered for reduction: 10
             symbolic reductions: 66
The reduced Groebner basis contains 7 polynomials:
F1=0.1360085506*x*z-0.548165136*y*z-0.259230258*z^2+0.1189209519*x-0.1550609403*y
                                +0.352426534*z+0.1568840673-7.952219855e-10*y^2*[E]
F2=0.1786862125*x^2-0.5815171014*y^2-0.6228451678*y*z-0.3504577221*z^2+0.0907085749*x
                     -0.542264472*y+0.5235226277*z+0.3306500943-8.711976022e-10*y^2*[E]
F3=0.7011771719*x*y-0.3083221489*y^2+0.8526396253*y*z+0.5240922894*z^2-0.5159758323*x
                    -0.3785469952*y-0.6152808835*z-0.5100951005+3.054458753e-10*y^2*[E]
R2=-0.04166411945*y*z^2+0.07367615285*z^3+0.1450231096*y^2+0.5964621731*y*z
              +0.2287435863*z^2-0.05816134318*x+0.2818223411*y-0.5957298621*z
                  -0.2454596895+1.719742599e-10*z^3*[E]+7.592625683e-10*y^2*[E]
                  +2.468611122e-09*y*z*[E]+1.05823543e-09*z^2*[E]-5.234303807e-10*x*[E]
                         +3.178419788e-10*y*[E]-2.132294565e-09*z*[E]-9.973418987e-10*[E]
S3=-0.5469673478*y^3+0.120504391*z^3+0.2921687519*y^2+0.1503793046*y*z
          +0.2123545807*z^2+0.1100054717*x+0.6554043404*y-0.7249418497*z
              -0.291998532+1.149108735e-09*z^3*[E]+2.692140881e-09*y^2*[E]
                +7.832297511e-09*y*z*[E]+3.262209302e-09*z^2*[E]-9.51066097e-10*x*[E]
```

```
                       +3.514933347e-09*y*[E]-8.714880021e-09*z*[E]-3.670826379e-09*[E]
S1=-0.06104491934*y^2*z-0.1065658397*z^3-0.1975957847*y^2-0.6742746844*y*z
             -0.2637514063*z^2+0.07409363164*x-0.3171237745*y+0.7519142478*z
                  +0.309082436-2.280548118e-10*z^3*[E]-9.297020143e-10*y^2*[E]
                                -3.090370135e-09*y*z*[E]-1.354714362e-09*z^2*[E]
                                      +6.378342031e-10*x*[E]-4.855452075e-10*y*[E]
                                              +2.756924e-09*z*[E]+1.286114577e-09*[E]
S1=-0.01016337876*z^4-0.09889894985*z^3-0.1604869399*y^2-0.6840161999*y*z
            -0.1775246481*z^2+0.05580338654*x-0.3196533438*y+0.6636132846*z
                +0.2610340489-4.582763575e-10*z^3*[E]-1.174805166e-09*y^2*[E]
                               -4.288254802e-09*y*z*[E]-1.626587062e-09*z^2*[E]
                                     +7.161633412e-10*x*[E]-1.079236881e-09*y*[E]
                                         +3.945386583e-09*z*[E]+1.730396525e-09*[E]
The Groebner basis is stable
The ideal is zero-dimensional
The normal set is [1,z,y,x,z^2,y*z,y^2,z^3]
The system has 8 solutions
The border set contains 12 terms: [x*z,x*y,x^2,y*z^2,x*z^2,y^2*z,
                                   x*y*z,y^3,x*y^2,z^4,y*z^3,x*z^3]
The border basis with 12 polynomials is
  x*z: F1=-5.846852878e-09*y^2+x*z-4.030372603*y*z-1.905985005*z^2+0.874363791*x
                                         -1.140082294*y+2.591208659*z+1.153486796
  x*y: F3=x*y-0.4397207452*y^2+1.216011672*y*z+0.7474463093*z^2-0.7358708369*x
                                     -0.5398735303*y-0.8774970266*z-0.7274838955
  x^2: F2=x^2-3.254403874*y^2-3.485692372*y*z-1.961302538*z^2+0.5076417124*x
                                     -3.034730349*y+2.929843441*z+1.850451077
y*z^2: R2=y*z^2-1.768335777*z^3-3.480767439*y^2-14.3159674*y*z-5.49018173*z^2
                         +1.395957588*x-6.764149709*y+14.29839085*z+5.891392731
x*z^2: B1=x*z^2-9.033037061*z^3-14.0287897*y^2-55.31475313*y*z-17.86974507*z^2
                          +4.861717172*x-26.26519696*y+56.51567044*z+22.73594074
y^2*z: S1=y^2*z+1.745695483*z^3+3.236891583*y^2+11.04554965*y*z+4.320611944*z^2
                           -1.213755921*x+5.194924958*y-12.31739281*z-5.063196751
x*y*z: B2=x*y*z+3.665381773*z^3+5.655982209*y^2+18.75963358*y*z+6.295931962*z^2
                             -1.587795564*x+9.67064796*y-21.62391232*z-8.54157768
  y^3: S3=y^3-0.2203136853*z^3-0.5341612361*y^2-0.2749328878*y*z-0.388239965*z^2
                          -0.2011188989*x-1.198251315*y+1.325384159*z+0.5338500312
x*y^2: B3=x*y^2-0.8979265317*z^3-2.43274415*y^2-2.8346635*y*z-0.7709917087*z^2
                         -0.1974040214*x-2.912907692*y+4.227888395*z+1.452817344
  z^4: S1=z^4+9.730912585*z^3+15.79070748*y^2+67.30204789*y*z+17.46708982*z^2
                         -5.490633436*x+31.45148404*y-65.29455451*z-25.68378653
y*z^3: B4=y*z^3-7.521738192*z^3-10.64041336*y^2-48.62523427*y*z-15.71147235*z^2
                           +4.829822921*x-21.54473006*y+48.63279597*z+19.68939861
x*z^3: B5=x*z^3-3.295113171*z^3-4.490074362*y^2-35.65728156*y*z-19.51219056*z^2
                           +6.341518045*x-11.63358292*y+38.44393299*z+17.23988991

The multiplication with 'x' matrix M_x is
       |           1           z               y            x
-------+-------------------------------------------------------
  1 * x|           0           0               0            1
  z * x| -1.153486796 -2.591208659     1.140082294  -0.874363791
  y * x| 0.7274838955 0.8774970266    0.5398735303  0.7358708369
  x * x| -1.850451077 -2.929843441     3.034730349 -0.5076417124
z^2 * x| -22.73594074 -56.51567044     26.26519696  -4.861717172
y*z * x|   8.54157768  21.62391232     -9.67064796   1.587795564
y^2 * x| -1.452817344 -4.227888395     2.912907692  0.1974040214
z^3 * x| -17.23988991 -38.44393299     11.63358292  -6.341518045
       |         z^2         y*z             y^2          z^3
-------+-------------------------------------------------------
  1 * x|           0           0               0            0
  z * x|  1.905985005  4.030372603 5.846852878e-09            0
  y * x|-0.7474463093 -1.216011672    0.4397207452            0
  x * x|  1.961302538  3.485692372     3.254403874            0
z^2 * x|  17.86974507  55.31475313      14.0287897  9.033037061
y*z * x| -6.295931962 -18.75963358    -5.655982209 -3.665381773
y^2 * x| 0.7709917087    2.8346635      2.43274415 0.8979265317
```

```
z^3 * x| 19.51219056  35.65728156     4.490074362   3.295113171

The multiplication with 'y' matrix M_y is
       |           1            z            y            x
-------+----------------------------------------------------
  1 * y|           0            0            1            0
  z * y|           0            0            0            0
  y * y|           0            0            0            0
  x * y| 0.7274838955 0.8774970266 0.5398735303 0.7358708369
z^2 * y| -5.891392731 -14.29839085  6.764149709 -1.395957588
y*z * y|  5.063196751  12.31739281 -5.194924958  1.213755921
y^2 * y|-0.5338500312 -1.325384159  1.198251315 0.2011188989
z^3 * y| -19.68939861 -48.63279597  21.54473006 -4.829822921
       |          z^2          y*z          y^2          z^3
-------+----------------------------------------------------
  1 * y|           0            0            0            0
  z * y|           0            1            0            0
  y * y|           0            0            1            0
  x * y|-0.7474463093 -1.216011672 0.4397207452            0
z^2 * y|   5.49018173   14.3159674  3.480767439  1.768335777
y*z * y| -4.320611944 -11.04554965 -3.236891583 -1.745695483
y^2 * y|  0.388239965 0.2749328878 0.5341612361 0.2203136853
z^3 * y|  15.71147235  48.62523427  10.64041336  7.521738192

The multiplication with 'z' matrix M_z is
       |          1            z               y            x
-------+------------------------------------------------------
  1 * z|          0            1               0            0
  z * z|          0            0               0            0
  y * z|          0            0               0            0
  x * z|-1.153486796 -2.591208659     1.140082294 -0.874363791
z^2 * z|          0            0               0            0
y*z * z|-5.891392731 -14.29839085     6.764149709 -1.395957588
y^2 * z| 5.063196751  12.31739281    -5.194924958  1.213755921
z^3 * z| 25.68378653  65.29455451    -31.45148404  5.490633436
       |         z^2          y*z             y^2          z^3
-------+------------------------------------------------------
  1 * z|          0            0               0            0
  z * z|          1            0               0            0
  y * z|          0            1               0            0
  x * z| 1.905985005  4.030372603 5.846852878e-09            0
z^2 * z|          0            0               0            1
y*z * z|  5.49018173   14.3159674     3.480767439  1.768335777
y^2 * z|-4.320611944 -11.04554965    -3.236891583 -1.745695483
z^3 * z|-17.46708982 -67.30204789    -15.79070748 -9.730912585

The matrix for eigenvector computation is
(0.9001147405,0.8538297014,0.7710213441) * [M_x,M_y,M_z] =
         |            1            z               y            x
---------+---------------------------------------------------------
  1 * ...|            0  0.7710213441    0.8538297014  0.9001147405
  z * ...| -1.038270468   -2.33238511     1.026204878 -0.7870277368
  y * ...| 0.6548189778  0.7898480084    0.4859481227  0.6623681874
  x * ...| -1.933833874  -3.885839427     4.071593358 -0.5027805567
z^2 * ...| -25.49520149  -63.07897882     29.41712287  -5.568013342
y*z * ...|  7.469118204   18.95659352    -7.924970204   1.389225952
y^2 * ...|  2.140313445   4.559735733   -0.3603443092    1.28523928
z^3 * ...| -12.52652476  -25.78458125     4.617324412   -5.59854456
         |          z^2           y*z             y^2           z^3
---------+---------------------------------------------------------
  1 * ...|            0             0               0             0
  z * ...|  2.486626543   4.481627491 5.262838461e-09             0
  y * ...|-0.6727874408 -0.3235286866     1.249628826             0
  x * ...|  2.596760587   5.206759503     3.304783536             0
z^2 * ...|  20.77250117   62.01302283     15.59950302   10.41164876
```

```
y*z * ...| -5.123080673  -15.27892465    -5.171041142  -3.426366189
y^2 * ...| -2.305812214  -5.730086272    0.1501190989 -0.3496212028
z^3 * ...|  17.51063301   21.72199857    0.9517105679    1.88552211
```

The actual computation of zeros of the system from the latter matrix is done in the following automatically generated Maple script. Note that the matrix is fed into Maple with 16 decimal digits to utilize completely the information in the coefficients (roughly 16 decimals provided by double precision floating-point) of the attained extended Gröbner basis.

```
Digits:=16:
with(linalg):
F:=[-0.7691618*x^2+4.8567277*x*y+0.3675592*y^2+3.1793749*x*z-4.2271665*y*z
                 -0.9211387*z^2-1.1844526*x-3.9125691*y+1.723136*z-1.2891205,
     1.4294897*x^2-4.9254763*x*y-2.4863027*y^2-1.5589943*x*z-4.6888701*y*z
                -3.5137711*z^2+2.9870548*x+0.0984003*y+4.4705923*z+4.4301261,
     0.4227507*x^2+4.1527797*x*y-3.2018649*y^2-0.0431212*x*z+3.7500442*y*z
                +2.3570262*z^2-2.8790072*x-3.4757485*y-2.5171945*z-2.2885406]:
M:=[[0,0.7710213441360505,0.8538297013599134,0.9001147405147112,0,0,0,0],
    [-1.038270468313438,-2.332385109509084,1.026204877926825,-0.7870277368454937,
      2.486626542555581,4.481627490769082,5.262838461375935e-09,0],
    [ 0.6548189778059842,0.7898480083707725,0.4859481226609593,0.662368187423571,
     -0.6727874407872535,-0.3235286866004636,1.249628825807781,0],
    [-1.933833873758256,-3.885839427489004,4.071593358155745,-0.502780556718639,
      2.59676058673525,5.206759503257073,3.30478353600223,0],
    [-25.49520149328264,-63.07897882288191,29.417122871492,-5.568013341654455,
      20.77250117309734,62.01302282775758,15.59950302321488,10.41164876263207],
    [ 7.46911820443428,18.95659352085575,-7.92497020380715,1.389225951732496,
     -5.123080673300609,-15.27892465106741,-5.171041142262112,-3.426366189058609],
    [ 2.140313445080858,4.559735733191697,-0.3603443092222598,1.285239280326299,
     -2.305812213501943,-5.730086272048754,0.1501190989079522,-0.3496212028022363],
    [-12.52652475676758,-25.78458124922453,4.617324411524294,-5.598544559594601,
      17.51063301306134,21.72199857045725,0.9517105678768463,1.885522109879311]
   ]:
E:=eigenvectors(M):
Z:=[]:
for i in E do
  val_x:=op(i[3])[4]/op(i[3])[1]:
  val_y:=op(i[3])[3]/op(i[3])[1]:
  val_z:=op(i[3])[2]/op(i[3])[1]:
  Z:=[op(Z),[x=val_x,y=val_y,z=val_z]]:
od:
for sol in Z do subs(sol,F) od;
```

The execution of the script gives the following approximations for the solutions of F (displayed to 11 decimals to save space). We also give an estimation of the sensitivity of the root-finding problem for each solution: it is an absolute condition number $\|J_F^{-1}(x, y, z)\|$ computed as the maximum matrix norm of the inverse Jacobian J_F^{-1} of F at (x, y, z).

x	y	z	$\|J_F^{-1}(x,y,z)\|$
$-0.38423241329 - 0.76025883539\,i$	$0.10138225615 + 0.24386154298\,i$	$-0.35992724322 + 0.099473771908\,i$	0.4832782918
$1.8239417240 + 1.2624628884\,i$	$0.40385622898 + 0.94834553745\,i$	$1.9340637897 - 1.0747794513\,i$	0.5994495382
3.2823157964	1.7197347046	-1.3541382190	0.2947189091
0.072843418071	-0.84799569617	-0.32582784216	0.3437115476
$1.8239417240 - 1.2624628884\,i$	$0.40385622898 - 0.94834553745\,i$	$1.9340637897 + 1.0747794513\,i$	0.5994495382
-4.5440031522	1.2112339726	-2.7982267555	0.4741171944
$-0.38423241329 + 0.76025883539\,i$	$0.10138225615 - 0.24386154298\,i$	$-0.35992724322 - 0.099473771908\,i$	0.4832782918
0.58841728224	0.14295239273	1.8037191602	0.7500351257

The computed approximations are checked against the polynomials F at the last line of the script. The low order of residuals

F_1	F_2	F_3
$2.655 \times 10^{-13} + 5.4 \times 10^{-15}\, i$	$-8.9 \times 10^{-14} - 1.724 \times 10^{-13}\, i$	$2.544 \times 10^{-13} + 2.674 \times 10^{-13}\, i$
$-6.79 \times 10^{-13} + 1.227 \times 10^{-12}\, i$	$-2.016 \times 10^{-12} - 1.677 \times 10^{-12}\, i$	$2.216 \times 10^{-12} - 1.29 \times 10^{-13}\, i$
6.81×10^{-13}	2.237×10^{-12}	-2.396×10^{-12}
-2.380×10^{-12}	2.688×10^{-12}	-5.704×10^{-12}
$-6.79 \times 10^{-13} - 1.227 \times 10^{-12}\, i$	$-2.016 \times 10^{-12} + 1.677 \times 10^{-12}\, i$	$2.216 \times 10^{-12} + 1.29 \times 10^{-13}\, i$
1.65×10^{-13}	-2.61×10^{-13}	-1.8×10^{-14}
$2.655 \times 10^{-13} - 5.4 \times 10^{-15}\, i$	$-8.9 \times 10^{-14} + 1.724 \times 10^{-13}\, i$	$2.544 \times 10^{-13} - 2.674 \times 10^{-13}\, i$
-1.700×10^{-12}	-2.157×10^{-12}	3.120×10^{-12}

shows that the attained extended Gröbner basis is preferable from the numerical point of view representation of the ideal $\langle F \rangle$ compared to the genuine Gröbner basis: The order of residuals and absolute condition numbers indicate that both coordinates for each real solution should be valid up to at least 11 decimal digits. Such an accuracy can be expected using the hardware double precision floating-point arithmetic. The claimed accuracy of the attained solutions is easy to verify by a straightforward comparison with the respective "exact" solutions (x^*, y^*, z^*) computed independently by `solve`(F) with extra high precision in Maple.

x^*	y^*	z^*
$-0.384232413291 - 0.760258835389\,i$	$0.101382256151 + 0.243861542984\,i$	$-0.359927243222 + 0.0994737719077\,i$
$1.82394172401 + 1.26246288843\,i$	$0.403856228979 + 0.948345537454\,i$	$1.93406378965 - 1.07477945133\,i$
3.28231579636	1.71973470458	-1.35413821900
0.0728434180703	-0.847995696174	-0.325827842163
$1.82394172401 - 1.26246288843\,i$	$0.403856228979 - 0.948345537454\,i$	$1.93406378965 + 1.07477945133\,i$
-4.54400315216	1.21123397263	-2.79822675551
$-0.384232413291 + 0.760258835389\,i$	$0.101382256151 - 0.243861542984\,i$	$-0.359927243222 - 0.0994737719077\,i$
0.588417282240	0.142952392732	1.80371916021

Bibliography

[Akr89] A. G. Akritas. *Elements of Computer Algebra with Applications.* John Wiley&Sons, New York, 1989.

[AL94] W. Adams and P. Loustaunau. *An Introduction to Grobner Bases.* 1994.

[AS88] W. Auzinger and H. J. Stetter. An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations. *International Series of Numerical Mathematics*, 80:11–30, 1988.

[Atk78] K. E. Atkinson. *An Introduction to Numerical Analysis.* John Wiley, New York, 1978.

[Buc65] B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.* Dissertation, Mathematisches Institut der Universit"at Innsbruck, Innsbruck, "Osterreich, 1965.

[Buc79] Bruno Buchberger. A criterion for detecting unnecessary reductions in the construction of Groebner bases. In *Proc. EUROSAM '79*, Lecture Notes in Computer Science, pages 3–21, Berlin, 1979. Springer-Verlag.

[Buc82] Bruno Buchberger. Miscellaneous results on gröbner-bases for polynomial ideals ii. Technical Report 82-23.0, CAMP-LINZ, 1982.

[Buc83] Bruno Buchberger. Gröbner bases: An algorithmic method in polynomial ideal theory. In N. K. Bose, editor, *Recent Trends in Multidimensional Systems Theory*, chapter 6, pages 184–232. Reidel, Dordrecht, The Netherlands, 1983.

[Buc84] Bruno Buchberger. A critical-pair/completion algorithm in reduction rings. Lecture Notes in Computer Science, pages 137–161, Berlin, 1984. Springer.

[Buc98] Bruno Buchberger. Introduction to Gröbner bases. In Bruno Buchberger and Franz Winkler, editors, *33 years of Gröbner bases: Gröbner bases and applications: Conference — February 1998, Linz, Austria*, London Mathematical Society Lecture Note Series, pages 3–31, Cambridge, UK, 1998. Cambridge University Press.

[Bün90] Reinhard Bündgen. Completion of multivariate polynomial ideals over commutative rings with the term completion procedure according to Peterson and Stickel. Universität Tübingen, 1990.

[BW93] Thomas Becker and Volker Weispfenning. *Gröbner Bases*. Graduate Texts in Mathematics, 141. Springer Verlag, 1993. In cooperation with Heinz Kredel.

[CLO97] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Undergraduate Texts in Mathematics. Springer Verlag, New York, 2nd edition, 1997.

[CLO98] D. Cox, J. Little, and D. O'Shea. *Using Alhebraic Geometry*. Graduate Texts in Mathematics. Springer Verlag, New York, 1998.

[Dub90] Thomas W. Dubé. The structure of polynomial ideals and gröbner bases. *SIAM Journal on Computing*, 19(4):750–775, August 1990.

[Fau] Jean-Charles Faugere. A new efficient algorithm for computing gröbner basis (f 4).

[GGM+96] H. Grassmann, G.-M. Greuel, B. Martin, W. Neumann, G. Pfister, W. Pohl, H. Schönemann, and T. Siebert. Standard bases, syzygies and their implementation in singular. In *Reports On Computer Algebra*, number 01. Centre for Computer Algebra, University of Kaiserslautern, July 1996. Online available at `http://www.mathematik.uni-kl.de/~zca`.

[Giu84] M. Giusti. Some effectivity problems in polynomial ideal theory. In John Fitch, editor, *Proceedings of the International Symposium on Symbolic and Algebraic Computation (EUROSAM 84)*, volume 174 of *LNCS*, pages 159–171, Cambridge, UK, July 1984. Springer.

[GM94] L. C. Grove and J. M. McShane. Computation in polynomial algebras and ideals. *Expositiones Mathematicae*, 12:289–304, 1994.

[GMN+91] Alessandro Giovini, Teo Mora, Gianfranco Niesi, Lorenzo Robbiano, and Carlo Traverso. 'one sugar cube, please' or selection strategies in the Buchberger algorithm. In Stephen M. Watt, editor, *ISSAC '91: proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation, July 15–17, 1991, Bonn, Germany*, pages 49–54, New York, NY 10036, USA, 1991. ACM Press.

[GP96] G. Greuel and G. Pfister. Advances and improvements in the theory of standard bases and syzygies, 1996.

[Gra] H. Grabe. The tangent cone algorithm and homogenization.

[Grä93] Hans-Gert Gräbe. On lucky primes. *Journal of Symbolic Computation*, 15(2):199–210 (or 199–209??), February 1993.

[HKL99] Markus A. Hitz, Erich Kaltofen, and Y. N. Lakshman. Efficient algorithms for computing the nearest polynomial with a real root and related problems. In *Proceedings of the 1999 international symposium on Symbolic and algebraic computation*, pages 205–212. ACM Press, 1999.

[HWSZ00] Yuzhen Huang, Wenda Wu, Hans J. Stetter, and Lihong Zhi. Pseudofactors of multivariate polynomials. In Carlo Traverso, editor, *Proceedings of the 2000 International Symposium on Symbolic and Algebraic Computation (ISSAC-00)*, pages 161–168, N. Y., August 7–9 2000. ACM Press.

[Kal97] Michael Kalkbrener. On the stability of Gröbner bases under specializations. *Journal of Symbolic Computation*, 24(1):51–58, July 1997.

[KGL92] S.R. Czapor K.O. Geddes and G. Labahn. *Algorithms for Computer Algebra*. Kluwer Academic Publishers, 1992.

[KL96] N. Karmarkar and Lakshman Y. N. Approximate polynomial greatest common divisors and nearest singular polynomials. In

Lakshman Y. N., editor, *ISSAC '96: Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation, July 24–26, 1996, Zurich, Switzerland*, pages 35–39, New York, NY 10036, USA, 1996. ACM Press.

[KM96] K. Kuehnle and E. W. Mayr. Exponential space computation of Gröbner bases. In Lakshman Y. N., editor, *ISSAC '96: Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation, July 24–26, 1996, Zurich, Switzerland*, pages 63–71, New York, NY 10036, USA, 1996. ACM Press.

[KRK88] A. Kandri-Rody and D. Kapur. Computing a Groebner Basis of a Polynomial Ideal over a Euclidean Domain. *JSC*, 6(1):37–58, August 1988.

[Lan92] Serge A. Lang. *Algebra*. Addison-Wesley, 3rd edition, 1992.

[Laz83] D. Lazard. Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations. In J. A. van Hulzen, editor, *Proc. European Computer Algebra Conference, EUROCAL '83, LNCS 162*, pages 146–156, London, 1983. Springer Verlag.

[MK00] Lorenzo Robbiano Martin Kreuzer. *Computational commutative algebra*, volume 1. Springer, Berlin, 2000.

[MM82] Ernst W. Mayr and Albert R. Meyer. The complexity of the word problems for commutative semigroups and polynomial ideals. *Advances in Math.*, 46:305–329, 1982.

[MM84] H. M. Möller and F. Mora. Upper and lower bounds for the degree of groebner bases. In John Fitch, editor, *Proceedings of the International Symposium on Symbolic and Algebraic Computation (EUROSAM 84)*, volume 174 of *LNCS*, pages 172–183, Cambridge, UK, July 1984. Springer.

[MMM95] Maria Grazia Marinari, Teo Mora, and Hans Michael Möller. Gröbner duality and multiplicities in polynomial system solving. In A. H. M. Levelt, editor, *ISSAC '95: Proceedings of the 1995 International Symposium on Symbolic and Algebraic Computation: July 10–12, 1995, Montreal, Canada*, ISSAC -PROCEEDINGS-1995, pages 167–179, New York, NY 10036, USA, 1995. ACM Press.

[MMT92] H. Michael Möller, Teo Mora, and Carlo Traverso. Gröbner bases computation using syzygies. In Paul S. Wang, editor, *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 320–328, Berkeley, CA, July 1992. ACM Press.

[Moe93] M. Moeller. Systems of algebraic equations solved by means of endomorphisms. In *Applied Algebra, Algebraic Algorithms and Error Correcting Codes, Proceedings of AAECC-10*, LNCS. Springer, 1993.

[Möl88] H. M. Möller. On the Construction of Gröbner Bases Using Syzygies. *JSC*, 6(2&3):345–360, October/December 1988.

[Mor82] T. Mora. An algorithm to compute the equations of tangent cones. In Jacques Calmet, J., editor, *Computer Algebra, EUROCAM'82, European Computer Algebra Conference, Marseille, France, 5–7 April 1982*, volume 144 of *Lecture Notes in Computer Science*, pages 158–165, Berlin-Heidelberg-New York, 1982. Springer-Verlag.

[Mou99] Bernard Mourrain. A new criterion for normal form algorithms. In *AAECC*, pages 430–443, 1999.

[MS] H. M. Möller and H. J. Stetter. Multivariate polynomial equations with multiple zeros solved by matrix eigenproblems. Technical report.

[MT] T. Mora and G. Pfister C. Traverso. An introduction to the tangent cone algorithm. *Issues in non-linear geometry and robotics, C.M. Hoffman ed., JAI Press.*

[Pan89] L. Pan. On the D-basis of polynomial ideals over principal ideal domains. *JSC*, 7(1):55–70, January 1989.

[Pau92] Franz Pauer. On lucky ideals for Gröbner basis computations. *Journal of Symbolic Computation*, 14(5):471–482, November 1992.

[Rob85] L. Robbiano. Term orderings on the polynomial ring. In Bob F. Caviness, editor, *Proceedings of the European Conference on Computer Algebra (EUROCAL '85): volume 2: research contributions*, volume 204 of *LNCS*, pages 513–517, Linz, Austria, April 1985. Springer.

[Rob86] Lorenzo Robbiano. On the theory of graded structures. *Journal of Symbolic Computation*, 2(2):139–170, June 1986.

[Sch96] H. Schönemann. Algorithms in singular. In *Reports On Computer Algebra*, number 02. Centre for Computer Algebra, University of Kaiserslautern, June 1996. Online available at `http://www.mathematik.uni-kl.de/~zca`.

[Ste93] H. J. Stetter. Multivariate polynomial equations as matrix eigenproblems. In *Proc. WSSIAA 2*, pages 355–371, 1993.

[Ste96a] Hans J. Stetter. Analysis of zero clusters in multivariate polynomial systems. Technical Report TR96-1567, Cornell University, Computer Science, February 11, 1996.

[Ste96b] Hans J. Stetter. Matrix eigenproblems are at the heart of polynomial system solving. *SIGSAM Bulletin (ACM Special Interest Group on Symbolic and Algebraic Manipulation)*, 30(4):22–25, December 1996.

[Ste97] Hans J. Stetter. Stabilization of polynomial systems solving with Gröbner bases. In Wolfgang W. Küchlin, editor, *ISSAC '97. Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation, July 21–23, 1997, Maui, Hawaii*, pages 117–124, New York, NY 10036, USA, 1997. ACM Press.

[Ste99] Hans J. Stetter. The nearest polynomial with a given zero, and similar problems. *ACM SIGSAM Bulletin*, 33(4):2–4, 1999.

[Sti87] Sabine Stifter. A generalization of reduction rings. *Journal of Symbolic Computation*, 4(3):351–364, December 1987.

[Sti91] Sabine Stifter. Gröbner Bases of Modules over Reduction Rings. Technical Report 91-48, RISC-Linz, Johannes Kepler University, Linz, Austria, 1991. Published in Journal of Algebra, vol. 159, no. 1, 1993, pp. 54 – 63.

[TV78] W. Trinks and U. Verfahren. Systeme algebraischer gleichungen zu losen. *J. Number Theory*, 10:475–488, 1978.

[Wei92] V. Weispfenning. Comprehensive Gröbner bases. *JSC*, 14(1):1–30, July 1992.

[Win88] Franz Winkler. A p-adic approach to the computation of Gröbner bases. *Journal of Symbolic Computation*, 6(2–3):287–304, October–December 1988. Computational aspects of commutative algebra.

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

[Zac78] G. Zacharias. Generalized Gröbner bases in commutative polynomial rings. Master's thesis, Lab. for Computer Science, MIT, 1978.

CURRICULUM VITAE

Aleksey Kondratyev
Research Institute for Symbolic Computation (RISC)
Johannes Kepler University, A−4040 Linz, Austria
tel: +43(0)732 2468-9981 fax: +43(0)732 2468-9930
Aleksey.Kondratyev@risc.uni-linz.ac.at
http://www.risc.uni-linz.ac.at/people/akondra/

Date and place of birth: September 30, 1973, Ulyanovsk, Russia

Nationality: Russian

Education:

1980−1990	School education, Ulyanovsk
1990−1995	Studies in Applied Mathematics at Ulyanovsk state University, Ulyanovsk
1995	Diploma degree (Dipl.-lng.)
1995−1998	Ph.D. study at Ulyanovsk state University, Ulyanovsk
1998−	Ph.D. study at Research Institute for Symbolic Computation, Johannes Kepler University, Linz

Grants:

FWF grant P 13266−MAT of the Austrian "Fonds zur Förderung der wissenschaftlichen Forschung" (November 1998 – October 2000)

Publications:

A. Kondratyev. On construction of an annihilator generating function. *Fundamental problems in Mathematics and Mechanics*, 1. 60−62, Ulyanovsk University Press, 1997.