JOHANNES KEPLER
UNIVERSITÄT LINZ
Netzwerk für Forschung, Lehre und Praxis

Involutive Bases for Polynomial Ideals

DISSERTATION

zur Erlangung des akademischen Grades

DOKTOR DER NATURWISSENSCHAFTEN

Angefertigt am *Institut für Symbolisches Rechnen*

Betreuung:

o.Univ.-Prof. Dr. Franz Winkler
Dr. rer. nat. habil. Joachim Apel

Eingereicht von:

Ralf Hemmecke

Linz, Februar 2003

Acknowledgements

First of all, I am grateful to Franz Winkler for giving me the opportunity to pursue this thesis at RISC-Linz and for his constant support.

Invaluable help I received from Joachim Apel. In particular during the time we spent on the treatment of criteria for the involutive basis algorithm I learned a lot from him. I would like to thank him for his continued support, his suggestions, and his many inspiring and clarifying comments.

Special thanks go to my brother, Raymond Hemmecke, for many fruitful discussions and his willingness to read several earlier drafts of my thesis, the care with which this was done, and all his valuable comments.

I thank Günter Landsmann for reading a draft of my thesis and pointing out an error in the notation.

Dearest thanks go to my wife, Jeannette Hemmecke, for her love and many deep discussions (of course, mostly not about mathematics). I am grateful for her support during every phase of my thesis. Especially in the final phase, she took it very generously upon herself to read my work and gave me valuable suggestions to round things up and clarify formulations. Thank you, Jeannette.

I would also like to thank all the people at RISC-Linz for creating a great scientific and personal environment.

Last but not least, I want to thank my parents Charlotte and Wolfgang Hemmecke who supported me during all the years of my studies.

Eidesstattliche Erklärung

Ich versichere, daß ich die Dissertation selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel verfaßt und mich auch sonst keiner unerlaubten Hilfe bedient habe.

Hagenberg, Februar 2003

Ralf Hemmecke

Abstract

This thesis contributes to the theory of polynomial involutive bases. Firstly, we present the two existing theories of involutive divisions, compare them, and come up with a generalised approach of *suitable partial divisions*. The thesis is built on this generalised approach. Secondly, we treat the question of choosing a "good" suitable partial division in each iteration of the involutive basis algorithm. We devise an efficient and flexible algorithm for this purpose, the *Sliced Division* algorithm. During the involutive basis algorithm, the Sliced Division algorithm contributes to an early detection of the involutive basis property and a minimisation of the number of critical elements. Thirdly, we give new criteria to avoid unnecessary reductions in an involutive basis algorithm. We show that the termination property of an involutive basis algorithm which applies our criteria is independent of the prolongation selection strategy used during its run. Finally, we present an implementation of the algorithms and results of this thesis in our software package Calix.

Zusammenfassung

Diese Dissertation beschäftigt sich mit involutiven Basen in einem polynomiellen Kontext. Zuerst stellen wir die beiden existierenden Theorien involutiver Divisionen vor, vergleichen sie miteinander und entwickeln daraus ein verallgemeinertes Konzept von *geeigneten partiellen Divisionen* (suitable partial divisions). Die gesamte Dissertation baut auf diesem verallgemeinerten Konzept auf. Als nächstes gehen wir der Frage nach, wie eine "gute" geeignete partielle Division während der Abarbeitung des Involutiven-Basen-Algorithmus gewählt werden kann. Zu diesem Zweck entwickeln wir einen effizienten und flexiblen Algorithmus, den wir *Sliced-Division*-Algorithmus nennen. Der Sliced-Division-Algorithmus kann als Unteralgorithmus im Involutiven-Basen-Algorithmus dazu verwendet werden, möglichst schnell zu erkennen, ob bereits eine involutive Basis vorliegt. Gleichfalls trägt er dazu bei, die Anzahl kritischer Elemente im Involutiven-Basen-Algorithmus und damit die Anzahl auszuführender Reduktionen möglichst klein zu halten. Des weiteren geben wir neue Kriterien an, um unnötige Reduktionen im Involutiven-Basen-Algorithmus zu vermeiden. Wir zeigen, daß die Terminationseigenschaft eines Involutiven-Basen-Algorithmus, der unsere Kriterien verwendet, nicht von der Auswahlstrategie kritischer Elemente während seiner Abarbeitung abhängt. Abschließend gehen wir auf die Implementation der Algorithmen und Ergebnisse dieser Dissertation ein und stellen das Softwarepaket CALIX vor.

Contents

Chapter 1

Introduction

Involutive bases have their origin in the Riquier-Janet Theory of partial differential equations at the beginning of the 20th century. Janet [Jan20], [Jan29] was the first to give an algorithm to transform a system of partial differential equations into a form (passive system) required for Riquier's Existence and Uniqueness Theorem. Recently, Zharkov and Blinkov [ZB93] applied Janet's algorithm in a polynomial context and thereby draw attention of other researchers to this alternative method of computing Gröbner bases. In this thesis we concentrate our work on several aspects of improving Janet's algorithm for polynomial ideals. An applicability to partial differential equations is only covered in so far as there is a simple formal translation of polynomials to linear partial differential equations in one unknown function and with constant coefficients.

In a polynomial context, Janet's algorithm is strikingly similar to Buchberger's well-known Gröbner basis algorithm [Buc85], but instead of S-polynomials it considers *non-multiplicative* prolongations as critical elements. The concept of *multiplicative* and *non-multiplicative* variables was introduced in [Jan20]. It is an essential ingredient in Janet's algorithm. Janet's particular separation of variables into *multiplicative* and *non-multiplicative* for each term was generalised by Gerdt and Blinkov [GB98a]. Another generalisation is due to Apel [Ape98a]. Both approaches result in a theory of *involutive divisions* which is the basis for Janet-like algorithms. Although the approaches of Gerdt/Blinkov and Apel are similar, neither of them covers the other. The existence of two partly incompatible theories for the same topic is insatisfactory, so the following problem arises.

Problem 1. *Find a compromise of the approaches of Gerdt/Blinkov and Apel which covers both theories of involutive bases reasonably well.*

Apel [Ape98a, Sect. 6.1] shows that a *maximal* (with respect to the refinement relation) admissible partial division would be advantageous with respect to a minimisation of the number of critical elements that must be considered during the involutive basis check and with respect to the chance that this check ends successfully. No efficient algorithm is known that computes such a *maximal* division. Apel gives, however, an algorithm to compute a *submaximal refinement* of a given admissible partial division and suggests to select the divisions during the run of the involutive basis algorithm only among *submaximal* divisions. Apel's theory of *admissible partial divisions* requires an order $\sqsubset$ on the terms.

The fact that such an order must be given before the *submaximal refinement* algorithm can be started is disadvantageous, because it can lead to submaximal (and even maximal) admissible partial divisions which are not “maximal” in an intuitive sense. Although any order $\sqsubset$ on the terms can be taken, there is not yet a satisfactory method to choose a “good” one. In the approach of Gerdt and Blinkov [GB98a] several concrete involutive divisions are given, but no efficient algorithm is known to produce a “good” decomposition of the term monoid.

The following problem should be solved.

Problem 2. *Give an efficient algorithm which produces a maximal admissible partial division that is also “maximal” in an intuitive sense.*

Since the involutive basis algorithm is similar to Buchberger’s algorithm, the question arises whether one can adapt the improvements known for Buchberger’s algorithm to the involutive basis algorithm. Concerning the criteria, Gerdt and Blinkov [GB98a] were the first to give a criterion (derived from Buchberger’s chain criterion) to avoid unnecessary reductions in the involutive basis algorithm. A natural problem is the following.

Problem 3. *Find additional criteria for the involutive basis algorithm.*

The question for a “good” selection strategy of critical elements in the involutive basis algorithm is of a difficult nature, since the proofs of termination of the involutive basis algorithm (in [Ape98a] as well as in [GB98a]) rely on a normal selection strategy. So the following problem arises.

Problem 4. *Which prolongation selection strategies different from a normal selection strategy can be used in the involutive basis algorithm?*

In order to solve the above mentioned problems, we proceed as follows.

We start with an overview of the notation in Chapter 2 and give basic concepts used in this thesis in Chapter 3.

In Chapter 4 we introduce partial divisions and recall the theories of involutive divisions by Gerdt and Blinkov [GB98a] and by Apel [Ape98a]. We compare their theories and build up a generalised theory of partial divisions which is appropriate for the involutive basis algorithm. Our theory is generalised in the sense that we take the main ingredients of both theories but remove certain restrictions. We introduce the new notion of a *suitable partial division* which combines notions of Apel and Gerdt/Blinkov, but also weakens their requirements in several places. This unifying approach answers Problem 1 and has not been published before.

Chapter 5 presents the theory of involutive bases in a polynomial context. It mainly contains known material, except that we generalise a characterisation theorem of Apel [Ape98a, Thm. 5.1] with respect to the requirements on the partial division. From this theorem and a termination proof of a completion procedure given in Chapter 4, we show termination and correctness of the involutive basis algorithm if the partial divisions selected during its run are *suitable partial divisions*.

Chapter 6 deals with an improvement of the involutive basis algorithm by giving an efficient sub-algorithm, the *Sliced Division algorithm*, for the computation of the intermediate partial divisions. The Sliced Division algorithm produces a “good” admissible partial division in the sense that this division is almost maximal (with respect to the refinement relation). Thus, it contributes to an early detection of the involutive basis property and a minimisation of the number of (non-multiplicative) prolongations that must be considered during the run of the involutive basis algorithm. The Sliced Division algorithm contributes to the solution of Problem 2. We additionally show a generalisation of the Sliced Division algorithm and present the classical partial divisions due to Janet, Thomas, and Pommaret as generalised Sliced Divisions. The Sliced Division algorithm has first been presented at the conference on Applications of Computer Algebra, Albuquerque, New Mexico, 2001, and most of the material of Sections 6.2–6.4 appears in [Hem03]. In Section 6.5 we present a method to construct a tree representation of a (directed) Sliced Division and give an algorithm which takes advantage of such a tree representation in order to find a simplifier for an involutive reduction in a time complexity independent of the number and degrees of the available simplifiers. The main ideas of Section 6.5 have been presented at the Workshop on Under- and Overdetermined Systems of Algebraic or Differential Equations, Karlsruhe, Germany, 2002, [Hem02b]. In Section 6.6 we present the *Sliced Dickson Lemma* which is an adaptation of the well-known Dickson Lemma [Dic13] to the case of Sliced Divisions. As a consequence, we show that under mild additional assumptions on the Sliced Divisions used during the involutive basis algorithm, a normal prolongation selection strategy is not necessary to ensure termination. The result of Section 6.6 contributes to a solution of Problem 4 and has not been published before.

In Chapter 7 new criteria for the involutive basis algorithm are given which are more powerful than the criterion of Gerdt and Blinkov [GB98a] in the sense that our criteria detect more prolongations that need not be reduced. Additionally, we show that the termination of an involutive basis algorithm which applies our criteria is independent of the prolongation selection strategy. Most part of Chapter 7 results from joint work with J. Apel and has been presented by the author at the conference on Applications of Computer Algebra, Volos, Greece, 2002, [Hem02a]. It has been published as a RISC-report [AH02], except for the more elaborated comparison of our criteria and Buchberger's criteria in the frame of the involutive basis algorithm in Section 7.6. In addition to the criteria given in [AH02], we give in Chapter 7 a new criterion, namely $\mathcal{C}_0$, in order to avoid the reduction of involutively top-reducible basis elements whose initial ancestor has changed. The content of Chapter 7 solves Problems 3 and 4.

In Chapter 8 we shortly describe our software package CALIX in which all the algorithms of this thesis are implemented. We suggest in this chapter a data structure for an actual implementation of the history management in the involutive basis algorithm. From this data structure we derive an efficient implementation of an additional criterion and thus we contribute again to the solution of Problem 3.

In summary, the main achievements of this thesis are:

- unifying the basic approaches of Gerdt/Blinkov and Apel,
- introducing the Sliced Division sub-algorithm for an early detection of the involutive basis property and a minimisation of the number of critical elements in the involutive basis algorithm,
- presenting new criteria for the involutive basis algorithm in order to avoid unnecessary reductions, and
- showing that the termination of an involutive basis algorithm which applies criteria is independent of the prolongation selection strategy.

Chapter 2

Notation

$\mathbb{N}$	$\mathbb{N} = \{0, 1, \dots\}$, natural numbers
$\mathbb{N}_k$	$\mathbb{N}_k = \{1, \dots, k\}$ for some $k \in \mathbb{N}$
$\mathbb{Q}$	rational numbers
$\mathbb{P}(T)$	set of all subsets of T
$\mathbb{P}_{\text{fin}}(T)$	set of all finite subsets of T
$\min_< S$	minimum of a set $S \neq \emptyset$ with respect to $<$
$\max_< S$	maximum of a set $S \neq \emptyset$ with respect to $<$
X	$X = \{x_1, \dots, x_n\}$, set of variables
$\mathbb{K}$	field
$\mathbb{K}[X]$	polynomial ring over $\mathbb{K}$ in the variables X
$\mathrm{Id}\,(G)$	ideal of $G \subseteq \mathbb{K}[X]$ in $\mathbb{K}[X]$
T	(commutative) monoid of all power products of $\mathbb{K}[X]$
$\deg_x t$	degree of t in the variable x
$\deg t$	total degree of t
$\prec$	admissible term order on T
$\mathrm{supp}\, f$	set of all terms of a polynomial $f \in \mathbb{K}[X]$
$\mathrm{lt}\, f$	leading term of f with respect to $\prec$
$\mathrm{lc}\, f$	leading coefficient of f with respect to $\prec$
$\mathrm{lm}\, f$	leading monomial of f with respect to $\prec$, $\mathrm{lm}\, f = \mathrm{lc}(f)\,\mathrm{lt}(f)$
$[U]$	monoid in T generated by $U \subseteq T$
$t[Y]$	$t[Y] = \{tu \mid u \in [Y]\}$, cone with vertex t
$\langle U \rangle$	$\langle U \rangle = UT$, monoid ideal of U in T, set of all multiples of elements of U
$u \trianglelefteq v$	u divides v
$u \triangleleft v$	u divides v properly
$\mathcal{M}$	$\mathcal{M} = (t[Y_t])_{t \in T}$, partial division
$\mathcal{M}_t$	$\mathcal{M}_t = t[Y_t]$, family member of $\mathcal{M}$ corresponding to t
$\mathcal{O}$	$\mathcal{O} = (t[X])_{t \in T}$, ordinary division
$\langle U \rangle_{\mathcal{M}}$	$\langle U \rangle_{\mathcal{M}} = \bigcup_{u \in U} \mathcal{M}_u$
$\mathcal{M} \geq \mathcal{N}$	$\mathcal{M}$ refines $\mathcal{N}$
$\mathcal{M} \equiv_U \mathcal{N}$	$\mathcal{M}$ and $\mathcal{N}$ are U-equivalent
$\mathfrak{M}$	family $\left(\mathcal{M}^U\right)_{U \in \mathbb{P}(T)}$ of partial divisions
$\mathfrak{L}$	involutive division in the sense of Gerdt and Blinkov
$\mathcal{J}$	Janet division
$\mathcal{T}$	Thomas division.
$\mathcal{P}$	Pommaret division
$\rightarrow_{G,\mathcal{M}}$	one-step $\mathcal{M}$-reduction relation modulo G
$\rightarrow^*_{G,\mathcal{M}}$	reflexive transitive closure of $\rightarrow_{G,\mathcal{M}}$
$\rightarrow_G$	ordinary one-step reduction relation modulo G

$\to_G^*$	reflexive transitive closure of $\to_G$
$\gcd(u,v)$	greatest common divisor of u and v
$\gcd(U)$	greatest common divisor of all elements of $U \neq \emptyset$
$\operatorname{lcm}(u,v)$	least common multiple of u and v
$\operatorname{lcm}(U)$	least common multiple of all elements of $U \neq \emptyset$
$[g_1, g_2]$	$[g_1, g_2] = \operatorname{lcm}(\operatorname{lt} g_1, \operatorname{lt} g_2)$
$\operatorname{spol}(f,g)$	S-polynomial of f and g
$\mathfrak{F}_{\preceq t}^G$	$\mathfrak{F}_{\preceq t}^G = \left\{ \sum_{g \in G} h_g g \,\middle|\, \forall_{g \in G}\, h_g \in \mathbb{K}[X] \wedge (h_g = 0 \vee \operatorname{lt}(h_g g) \preceq t) \right\}$
$\mathfrak{F}_{\prec t}^G$	$\mathfrak{F}_{\prec t}^G = \bigcup_{s \prec t} \mathfrak{F}_{\preceq s}^G$
$\left(\mathfrak{F}_{\preceq t}^G\right)_{t \in T}$	Gröbner filtration of $\operatorname{Id}(G)$
$\mathcal{S}(f,g)$	abbreviation for the relation $\operatorname{spol}(f,g) \in \mathfrak{F}_{\prec [f,g]}^G$
$>_{\mathcal{M},U}$	$\mathcal{M}$-sequence semi-order on U
$A[i]$	i-th entry of the array A
$a \trianglelefteq_G g$	a is an ancestor of g with respect to G
$\operatorname{anc} g$	ancestor polynomial a with $a \trianglelefteq_G g$
$\operatorname{wanc} g$	weak ancestor polynomial w with $\operatorname{lt} w \trianglelefteq \operatorname{lt} g$
$\operatorname{idx}(g)$	index of the polynomial g
$\mathcal{C}(t,g,G,\mathcal{M})$	(t,g) is redundant with respect to $(G,\mathcal{M})$
$\mathcal{R}(t,g,f)$	(t,g) is explicitly reduced and top-reduced using f

Chapter 3

Basic Concepts

In this thesis we build on concepts which are well-known in the theory of Gröbner bases. The theory of Gröbner bases is described in an number of standard textbooks, cf. [BW93], [AL94], [Frö97].

Let us fix some basic notions.

We denote by $\mathbb{N} = \{0, 1, 2, \dots\}$ the set of natural numbers and for $n \in \mathbb{N}$ we abbreviate the set $\{1, 2, \dots, n\}$ by $\mathbb{N}_n$. The rational numbers are denoted by $\mathbb{Q}$.

Definition 3.0.1. Let S be a set. The cardinality of S is denoted by $\operatorname{card} S$.

The set of all subsets of S is denoted by $\mathbb{P}(S)$, whereas $\mathbb{P}_{\text{fin}}(S)$ is the set of all finite subsets of S.

If $<$ is a linear order on S and S is non-empty, we denote by $\min_< S$ and $\max_< S$ the minimum and maximum of S, respectively.

For the whole thesis let $X = \{x_1, \dots, x_n\}$ be a set of indeterminates and $\mathbb{K}[X]$ be the (commutative) ring of polynomials in X over a field $\mathbb{K}$ of coefficients.

Definition 3.0.2. Let $G = \{g_1, \dots, g_r\} \subseteq \mathbb{K}[X]$. The **ideal** of G in $\mathbb{K}[X]$ is the set

$$\operatorname{Id}(G) := \left\{ h \in \mathbb{K}[X] \;\middle|\; \exists h_1, \dots, h_r \in \mathbb{K}[X] : h = \sum_{\varrho=1}^{r} h_\varrho g_\varrho \right\}.$$

Definition 3.0.3. By T we denote the (commutative) monoid of power products of $\mathbb{K}[X]$. An element of T is called a **term** or a **power product**.

For a term $t = x_1^{d_1} \cdots x_n^{d_n}$ we call d_i the **degree** of t in x_i or x_i**-degree** and denote it by $\deg_{x_i} t$; the sum $d_1 + \cdots + d_n$ is called the **total degree** of t and is written $\deg t$. If $s = x_1^{e_1} \dots x_n^{e_n}$ is another term, then the **least common multiple** $\operatorname{lcm}(s,t)$ (resp. the **greatest common divisor** $\gcd(s,t)$) is the term $u \in T$ such that $\deg_{x_i} u = \max(e_i, d_i)$ (resp. $\deg_{x_i} u = \min(e_i, d_i)$) for all $i \in \mathbb{N}_n$.

Definition 3.0.4. A **term order** or **admissible order** on T is a well-order $\prec$ on T that is compatible with the monoid structure of T, i. e.,

$$\forall t \in T \setminus \{1\} : 1 \prec t$$
$$\forall u, v, t \in T : u \prec v \implies tu \prec tv.$$

We use $\succ$, $\preceq$, and $\succeq$ with the obvious meaning.

Definition 3.0.5. The **lexicographical order** on T is defined by $u \prec_{\mathrm{lex}} v$ if and only if there exists an $i \in \mathbb{N}_n$ such that $\deg_{x_j} u = \deg_{x_j} v$ for all $1 \leq j < i$ and $\deg_{x_i} u < \deg_{x_i} v$.

A term $u \in T$ is said to be **inverse lexicographically** smaller than v (denoted $u \prec_{\mathrm{invlex}} v$) if and only if $v \prec_{\mathrm{lex}} u$, i. e., $\prec_{\mathrm{invlex}} = \prec_{\mathrm{lex}}^{-1}$.

We define the **reverse lexicographical order** on T by $u \prec_{\mathrm{revlex}} v$ if and only if there exists an $i \in \mathbb{N}_n$ such that $\deg_{x_j} u = \deg_{x_j} v$ for all $i < j \leq n$ and $\deg_{x_i} u > \deg_{x_i} v$.

A term u is said to be **degree lexicographically** (resp. **degree inverse lexicographically**, **degree reverse lexicographically**) smaller than a term v (denoted $u \prec_{\mathrm{deglex}} v$, $u \prec_{\mathrm{deginvlex}} v$, and $u \prec_{\mathrm{degrevlex}} v$, respectively) if $\deg u < \deg v$ or $\deg u = \deg v$ and $u \prec_{\mathrm{lex}} v$ (resp. $u \prec_{\mathrm{invlex}} v$, $u \prec_{\mathrm{revlex}} v$).

While $\prec_{\mathrm{lex}}$, $\prec_{\mathrm{deglex}}$, $\prec_{\mathrm{deginvlex}}$, and $\prec_{\mathrm{degrevlex}}$ are admissible; $\prec_{\mathrm{invlex}}$ and $\prec_{\mathrm{revlex}}$ are not.

Definition 3.0.6. Let $g \in \mathbb{K}[X] \setminus \{0\}$ and $\prec$ be an arbitrary order. The **support of** g (written: $\operatorname{supp} g$) is the set of terms of g that appear with a non-zero coefficient. The biggest term of $\operatorname{supp} g$ with respect to $\prec$ is denoted by $\operatorname{lt}_\prec g$ and is called the **leading term of** g **with respect to** $\prec$. The coefficient corresponding to $\operatorname{lt}_\prec(g)$ is called **leading coefficient of** g **with respect to** $\prec$ and is denoted by $\operatorname{lc}_\prec g$. The product $\operatorname{lc}_\prec(g) \operatorname{lt}_\prec(g)$ is called the **leading monomial of** g **with respect to** $\prec$ and is denoted by $\operatorname{lm}_\prec(g)$. The polynomial g is called **monic** if $\operatorname{lc}_\prec g = 1$. Furthermore, if $G \subseteq \mathbb{K}[X]$, let $\operatorname{lt}(G) := \{\operatorname{lt} g \mid 0 \neq g \in G\}$.

Since in this thesis we are not mainly interested in computations with respect to different term orders, we fix an arbitrary irreflexive term order $\prec$ on T and therefore omit the index $\prec$.

Chapter 4

Partial Divisions

Throughout this chapter we present the various available definitions in the context of partial divisions. The approach of Gerdt and Blinkov [GB98a] is compared to the approach of Apel [Ape98a, Ape98b]. We unify both approaches by taking ideas from both theories and give yet another approach which builds on weaker assumptions but retains the validity of the main results of [Ape98a] and [GB98a].

4.1 Preliminaries

Definition 4.1.1. For any set $U \subseteq T$ we define $[U]$ as the monoid generated by U, i. e.,

$$[U] := \{ t \in T \mid \exists u_1, \ldots, u_m \in U : t = u_1 u_2 \cdots u_m \} . \tag{4.1}$$

By definition, the product of $m = 0$ elements is 1, in particular, $[\emptyset] = \{1\}$. The **monoid ideal** $\langle U \rangle$ generated by U in T is defined by

$$\langle U \rangle := \{ w \in T \mid \exists u \in U \exists t \in T : w = ut \} . \tag{4.2}$$

Definition 4.1.2. Let $Y \subseteq X$, $t \in T$, $U \subseteq T$. By tU we denote the set $\{ tu \mid u \in U \}$. The set $t[Y]$ is called a **cone**, t its **vertex**, and the elements of Y its **multipliers**. The number of elements of Y is called the **dimension** of the cone.

The following definition recalls the divisibility relation on terms. We introduce here non-standard symbols in order to make the distinction between divisibility and proper divisibility more apparent.

Definition 4.1.3. Let $u, v \in T$. If there exists $t \in T$ such that $tu = v$, we say that u **divides** v (written: $u \trianglelefteq v$), if additionally $t \neq 1$, we say that u **properly divides** v (written: $u \triangleleft v$).

Let $u, v \in T$. The relation $u \trianglelefteq v$ can also be read as $v \in u[X]$. In fact, the whole divisibility relation on T can thus be represented by the family $(t[X])_{t \in T}$. By restricting the set X for each term t to a certain subset $Y_t \subseteq X$ one arrives at the notion of a partial division.

Definition 4.1.4. Let $(Y_t)_{t\in T}$ be a family of subsets of X. The family $\mathcal{M} = (t[Y_t])_{t\in T}$ is called a **partial division generated by** $(Y_t)_{t\in T}$ or simply a **partial division**. We write $\mathcal{M}_t$ for the family member $t[Y_t]$ corresponding to $t \in T$. If $u \in T$ and $v \in \mathcal{M}_u$ then v is called an $\mathcal{M}$**-multiple** of u, and u is an $\mathcal{M}$**-divisor** of v. The variables in Y_u are called **multiplicative for** u (with respect to $\mathcal{M}$), and variables in $X \setminus Y_u$ are called **non-multiplicative for** u (with respect to $\mathcal{M}$).

The **ordinary division** $(t[X])_{t\in T}$ is denoted by $\mathcal{O}$.

Definition 4.1.5. A pair of the form (x,t) with $x \in X$ and $t \in T$ is called a **prolongation**. Let $\mathcal{M}$ be a partial division. The prolongation (x,t) is called **multiplicative** (resp. **non-multiplicative**) with respect to $\mathcal{M}$ if x is multiplicative (resp. non-multiplicative) for t with respect to $\mathcal{M}$. If $x \in X$ and $t \in T$, we call the product xt an x**-prolongation of** t or simply a **prolongation of** t. An x-prolongation of t is called **multiplicative** if x is multiplicative for t with respect to $\mathcal{M}$ otherwise **non-multiplicative**.

The phrase "with respect to $\mathcal{M}$" is omitted if the division is clear from the context.

Definition 4.1.6. Let $U \subseteq T$. Each family $\mathcal{N} = (u[Y_u])_{u\in U}$ **induces** a partial division $\mathcal{M} = (t[Y_t])_{t\in T}$ by setting $Y_t := X$ for $t \in T \setminus U$. We also call $\mathcal{N}$ a partial division and mean its induced partial division. By $\langle U \rangle_{\mathcal{M}}$ we denote the set $\bigcup_{u\in U} \mathcal{M}_u$.

The latter notation is motivated by the notation of the *involutive span* as given in [HSS02].

The following definition introduces the *refinement* relation which is a partial order on the set of all partial divisions.

Definition 4.1.7. [Ape98a] Let $\mathcal{M}$ and $\mathcal{N}$ be two partial divisions. If $\mathcal{M}_t \supseteq \mathcal{N}_t$ for all $t \in T$ we say that $\mathcal{M}$ **refines** $\mathcal{N}$ (denoted $\mathcal{M} \geq \mathcal{N}$). Let $U \subseteq T$. $\mathcal{M}$ and $\mathcal{N}$ are U**-equivalent** if $\mathcal{M}_u = \mathcal{N}_u$ for all $u \in U$ (denoted $\mathcal{M} \equiv_U \mathcal{N}$).

Obviously, $\langle U \rangle = \langle U \rangle_{\mathcal{O}}$ and $\mathcal{O} \geq \mathcal{M}$ for the ordinary division $\mathcal{O}$ and any partial division $\mathcal{M}$.

Apel [Ape98a] and Gerdt and Blinkov [GB98a] introduced the concept of involutive division. Unfortunately, there are now two slightly differing notions. In order to distinguish them in the following, we take over the notion *admissible partial division* as defined in [Ape98b] for Apel's version and use *involutive division* exclusively in the sense as introduced in [GB98a].

Since we want to compare the two different approaches, we present the basic definitions below. In order to help the reader to catch the differences more easily, we build the definitions of both versions on the concept of a partial division.

4.2 Definitions of Gerdt and Blinkov

In [Jan20], Janet introduced a method to separate for a given set of terms $U \subset T$ the set X for each term into *multiplicative* and *non-multiplicative* variables. Janet used such an assignment in his algorithm to transform a given system of partial differential equations into a canonical form.

The idea of generalising the method of Janet in the context of polynomial systems goes back to Gerdt and Blinkov [GB98a]. A main feature of Janet's method to compute a canonical system is to add *non-multiplicative prolongations* to the given system and allow reduction only with respect to *multiplicative variables*. Gerdt and Blinkov generalised Janet's separation of variables in such a way that the main feature of Janet's method is retained. We repeat here the definitions stated in [GB98a] but use a different notation which better fits in our treatment of partial divisions.

Definition 4.2.1. [GB98a] A family $\mathfrak{L} = \left(\mathcal{L}^U\right)_{U\in\mathbb{P}_{\mathrm{fin}}(T)}$ of partial divisions is said to be an **involutive division** on T if for all finite sets $U, V \subset T$ and $u, v \in U$ with $u \neq v$ the following conditions are fulfilled.

$$\mathcal{L}_u^U \cap \mathcal{L}_v^U \neq \emptyset \implies \mathcal{L}_u^U \subset \mathcal{L}_v^U \vee \mathcal{L}_u^U \supset \mathcal{L}_v^U \tag{4.3}$$

$$U \subseteq V \implies \mathcal{L}^U \geq \mathcal{L}^V \tag{4.4}$$

Condition (4.3) deals only with one partial division $\mathcal{L}^U$ whereas condition (4.4) is a kind of filter axiom which connects the partial division for various sets of terms by saying that the "bigger" the set in question the "smaller" the corresponding cones.

Definition 4.2.2. [GB98a] An involutive division $\mathfrak{L} = \left(\mathcal{L}^U\right)_{U\in\mathbb{P}_{\mathrm{fin}}(T)}$ is called **continuous** if for any finite set $U \subset T$ and for any finite sequence $(u_i)_{1\leq i\leq k}$ of elements in U such that

$$\forall i < k \; \exists x \in X : xu_i \in \mathcal{L}_{u_{i+1}}^U \setminus \mathcal{L}_{u_i}^U$$

the inequality $u_i \neq u_j$ for $i \neq j$ holds.

In [CG01], Chen and Gao follow the approach of Gerdt and Blinkov but use another notation and say *Artinian* instead of continuous.

Definition 4.2.3. [GB98a] A continuous involutive division $\mathfrak{L} = \left(\mathcal{L}^U\right)_{U\in\mathbb{P}_{\mathrm{fin}}(T)}$ is **constructive** if for any finite set $U \subset T$ and any $u \in U$, $x \in X$ the conditions

$$xu \notin \langle U\rangle_{\mathcal{L}^U} \tag{4.5}$$

$$\forall v \in U \; \forall y \in X : yv \lhd xu \implies yv \in \langle U\rangle_{\mathcal{L}^U} \tag{4.6}$$

imply

$$\forall w \in \langle U\rangle_{\mathcal{L}^U} : xu \notin \mathcal{L}_w^{U\cup\{w\}}. \tag{4.7}$$

Definition 4.2.4. [GB98a] Given an involutive division $\mathfrak{L} = \left(\mathcal{L}^U\right)_{U\in\mathbb{P}_{\mathrm{fin}}(T)}$. A finite set $U \subseteq T$ is called **involutive with respect to $\mathfrak{L}$** or **$\mathfrak{L}$-involutive** if

$$\langle U\rangle = \langle U\rangle_{\mathcal{L}^U}. \tag{4.8}$$

A set $U \subseteq T$ is called **locally involutive with respect to $\mathfrak{L}$** or **locally $\mathfrak{L}$-involutive** if

$$\forall u \in U \; \forall x \in X : xu \in \langle U\rangle_{\mathcal{L}^U}. \tag{4.9}$$

The difference between involutive and locally involutive is that in an involutive set U each (ordinary) multiple of $u \in U$ must be an *involutive multiple* of some $v \in U$ (must lie in $\mathcal{L}_v^U$) whereas in a locally involutive set U, only products of $u \in U$ by one single (non-multiplicative) variable must lie in $\mathcal{L}_v^U$ for some $v \in U$.

An $\mathfrak{L}$-involutive set U is locally $\mathfrak{L}$-involutive with respect to any involutive division $\mathfrak{L}$. The opposite implication is not always true, but holds for continuous divisions $\mathfrak{L}$, cf. [GB98a].

Definition 4.2.5. [GB98a] Let $\mathfrak{L}$ be an involutive division. A finite $\mathfrak{L}$-involutive set $V \subset T$ is called **$\mathfrak{L}$-completion of a set** $U \subseteq V$ if $\langle V\rangle = \langle U\rangle$. The set U is called **finitely generated with respect to** $\mathfrak{L}$ if there exists an $\mathfrak{L}$-completion of U. An involutive division $\mathfrak{L}$ is called **noetherian** if every finite set $U \subseteq T$ is finitely generated with respect to $\mathfrak{L}$.

4.3 Definitions of Apel

Another approach of generalising the ideas of Janet in a polynomial context is due to Apel. Let us recall the definition introduced in [Ape98a] and [Ape98b].

Definition 4.3.1. [Ape98a] Let $U \subseteq T$ and let $\sqsubset$ be an irreflexive linear order[1] on U. Let $\mathcal{M}$ be a partial division generated by a family $(Y_t)_{t \in T}$ of subsets of X. $\mathcal{M}$ is called **admissible on** $(U, \sqsubset)$ if for all $u, v \in U$ with $u \sqsubset v$, one of the conditions (4.10) or (4.11) holds.

$$u[X] \cap v[Y_v] = \emptyset \tag{4.10}$$

$$u[Y_u] \subset v[Y_v] \tag{4.11}$$

$\mathcal{M}$ is **admissible on** U or **U-admissible** if there exists an irreflexive linear order $\sqsubset$ on U such that $\mathcal{M}$ is admissible on $(U, \sqsubset)$.

Remark 4.3.2. The first condition of admissibility involves the set X. So it would have been more precise to speak of X-admissibility. Because of the following observation, this additional qualification is not needed. Suppose a partial division $(u[Y_u])_{u \in U}$ is X-admissible on $(U, \sqsubset)$. Then it is also X'-admissible for all $X' \supseteq \bigcup_{u \in U} Y_u$.

Let $u, v \in U$ with $u \sqsubset v$ and suppose the relation

$$u[X] \cap v[Y_v] = \emptyset \tag{4.12}$$

holds. Then also

$$u[X'] \cap v[Y_v] = \emptyset \tag{4.13}$$

trivially holds for all $X' \subset X$. From (4.12) it follows that $\deg_x u > \deg_x v$ for some $x \in X \setminus Y_v$. Therefore, (4.13) holds for $X' \supset X$, as well.

Examples of admissible partial divisions are given in Section 4.4.

The Thomas division (cf. [Tho37] or Definition 4.4.4) is mainly of theoretical interest. Apel showed in [Ape98a] that the involutive basis algorithm terminates if one chooses an admissible partial division which refines the Thomas division.

[1] An irreflexive linear order $\sqsubset$ on U is an irreflexive and transitive relation on U such that for all $u, v \in U$ exactly one of the relations $u = v$, $u \sqsubset v$, and $v \sqsubset u$ holds.

In fact, also the concept of refinement, Definition 4.1.7, goes back to Apel, cf. [Ape98a]. He considers the whole lattice of partial divisions and argues to prefer *maximal* divisions for the computation of involutive bases. There is not yet an efficient algorithm to compute a *maximal* division which is admissible on a given set U. However, Apel gives an algorithm that computes a *submaximal* division. Let us recall the formal definitions from [Ape98a].

Definition 4.3.3. [Ape98a] Let $U \subseteq T$, let $\sqsubset$ be an irreflexive linear order on $U \subseteq T$ and let $\mathbb{M}_{(U,\sqsubset)}$ be the set of all partial divisions which are admissible on $(U, \sqsubset)$.

Let $\mathcal{M} \in \mathbb{M}_{(U,\sqsubset)}$ be a partial division generated by a family $(Y_t)_{t\in T}$ of subsets of X. $\mathcal{M}$ is called **maximal** if there is no $\mathcal{M}' \in \mathbb{M}_{(U,\sqsubset)} \setminus \{\mathcal{M}\}$ with $\mathcal{M}' \geq \mathcal{M}$.

For each $u \in U$ define the following sets.

$$A_u := \{x \in X \mid \exists v \in U : u \sqsubset v \wedge u \in \mathcal{M}_v \wedge x \notin Y_v\} \tag{4.14}$$

$$C_u := \{v \in U \mid v \sqsubset u \wedge \mathcal{M}_v \not\subseteq u[Y_u \cup \{x \in X \mid \deg_x u < \deg_x v\}]\} \tag{4.15}$$

$$I_u := \mathrm{Id}_{\mathbb{K}[X]}\left(A_u \cup \left\{t \in T \,\middle|\, \exists v \in C_u : t = \frac{\mathrm{lcm}(u,v)}{u}\right\}\right) \tag{4.16}$$

$\mathcal{M}$ is called **submaximal for** $(U, \sqsubset)$ if for all $u \in U$ the set Y_u is maximally independent for the monomial ideal I_u of $\mathbb{K}[X]$.

In fact, the conditions in the definition of a submaximal division are necessary for a partial division to be admissible and maximal, but they are not sufficient, cf. [Ape98a].

4.4 Classical Partial Divisions

The following three divisions, namely the Janet, Pommaret, and Thomas division, are called *classical* in the literature. We include them here as examples of partial divisions which fit into the definitions of both approaches. We reconsider these divisions in the context of generalised Sliced Divisions in Section 6.4.

Janet [Jan20] used a particular partial division which is defined as follows.

Definition 4.4.1 (Janet Division). Let $U \subseteq T$ be a finite set of power products and let $>$ be an order on the variables of X.[2] Define $Y_t = X$ for all $t \in T \setminus U$ and

$$Y_t := \{x \in X \mid \nexists u \in U : (\deg_x u > \deg_x t \wedge \forall y > x : \deg_y u = \deg_y t)\}$$

for all $t \in U$. The division $(t[Y_t])_{t\in T}$ is called **Janet division on** U.

Remark 4.4.2. Although Janet's original assignment of multiplicative variables to a term uses the order $x_n > \cdots > x_1$, the Janet division in [Ape98a] and [GB98a] is defined with respect to $x_1 > \cdots > x_n$. There are, in fact, $n!$ ways to determine a Janet division. We use the convention that, unless the order of the variables is explicitly mentioned, by a Janet division we mean a Janet division with respect to $x_1 > \cdots > x_n$.

[2] This is not to be confused with a term order on T.

A simple procedure to determine the Janet multiplicative variables for each term of a given set $U \subset T$ goes as follows. The set U is divided into disjoint subsets according to the degree in x_1. If $d_1, \ldots, d_m$ with $d_1 > \cdots > d_m$ are the different possible degrees, we denote the corresponding subsets by $U_{d_1}, \ldots, U_{d_m}$. The elements with the highest degree in x_1, namely those in U_{d_1}, get x_1 as multiplicative. For all other elements, x_1 is non-multiplicative. Now to each of the subsets separately a similar process is applied with respect to x_2. Again the terms with highest degree in x_2 (separately for each subset) get x_2 as multiplicative. Such a subdivision is continued until all variables are assigned as multiplicative or non-multiplicative for each term.

Let us illustrate this process by an example.

Example 4.4.3. *Let $X := \{x_1, x_2, x_3\}$ and*

$$U = \left\{x_1^7x_3^3, x_1^7x_3^2, x_1^7x_3, x_1^5x_2^3x_3^2, x_1^5x_2x_3, x_1^2x_2x_3^3, x_2^6x_3\right\}.$$

The first subdivision is $U_7 := \left\{x_1^7x_3^3, x_1^7x_3^2, x_1^7x_3\right\}$, $U_5 := \left\{x_1^5x_2^3x_3^2, x_1^5x_2x_3\right\}$, $U_2 := \left\{x_1^2x_2x_3^3\right\}$, $U_0 := \left\{x_2^6x_3\right\}$. x_1 is multiplicative only for elements in U_7. In order to decide for which elements x_2 is multiplicative, we must treat each of the above sets in a similar way. For U_7 we subdivide and obtain $U_{70} := U_7$. Hence, x_2 is multiplicative for all elements of U_{70}. For U_5, subdivision yields the sets $U_{53} := \left\{x_1^5x_2^3x_3^2\right\}$ and $U_{51} := \left\{x_1^5x_2x_3\right\}$. Hence x_2 is multiplicative for $x_1^5x_2^3x_3^2$ but not for $x_1^5x_2x_3$. Easily one derives that x_2 is also multiplicative for the elements in $U_{21} := U_2$ and $U_{06} := U_0$. In order to decide for which elements the variable x_3 is multiplicative, again we must consider each of the sets U_{70}, U_{53}, U_{51}, U_{21}, and U_{06} separately. Except for U_{70}, all these sets have just one element, their elements get x_3 as multiplicative. Subdivision of U_{70} with respect to x_3 yields $U_{703} := \left\{x_1^7x_3^3\right\}$, $U_{702} := \left\{x_1^7x_3^2\right\}$, and $U_{701} := \left\{x_1^7x_3\right\}$. Hence, only $x_1^7x_3^3$ gets x_3 as multiplicative.

The above process can be understood more easily by the following table. We mark multiplicative variables by $\surd$.

U		x_1		x_2		x_3
$x_1^7x_3^3$	U_7	$\surd$	U_{70}	$\surd$	U_{703}	$\surd$
$x_1^7x_3^2$		$\surd$		$\surd$	U_{702}	
$x_1^7x_3$		$\surd$		$\surd$	U_{701}	
$x_1^5x_2^3x_3^2$	U_5		U_{53}	$\surd$	U_{532}	$\surd$
$x_1^5x_2x_3$			U_{51}		U_{511}	$\surd$
$x_1^2x_2x_3^3$	U_2		U_{21}	$\surd$	U_{213}	$\surd$
$x_2^6x_3$	U_0		U_{06}	$\surd$	U_{061}	$\surd$

Thomas uses the Janet division in [Tho29] and [Tho34] for his treatise on Riquier's Existence Theorem, but introduces another division in [Tho37, p. 56].

Definition 4.4.4 (Thomas Division). Let $U \subseteq T$ be a set of power products. Define $Y_t = X$ for all $t \notin U$ and $Y_t := \{x \in X \mid \forall u \in U : \deg_x u \leq \deg_x t\}$ for all $t \in U$. The division $(t[Y_t])_{t \in T}$ is called **Thomas division on** U.

The Thomas division is important for the termination of the involutive basis algorithm.

Another division is due to Pommaret [Pom78].

Definition 4.4.5 (Pommaret Division). If $t \in [\{x_1, \dots, x_k\}]$ and $\deg_{x_k} t > 0$ for some $k \in \mathbb{N}_n$ then let $Y_t := \{x_k, \dots, x_n\}$. For $t = 1$ let $Y_t := X$. The division $(t[Y_t])_{t \in T}$ is called **Pommaret division**.

Remark 4.4.6. It is clear that one can define Pommaret divisions for any permutation of the variables. Nevertheless, if not specified otherwise, we use the notion of Pommaret division with the order of variables as defined above.

All of the above divisions are admissible with respect to the inverse lexicographical order on T, cf. [Ape98a]. Whereas the Janet division refines the Thomas division for any set $U \subseteq T$, the Pommaret division does not.

We have given the above divisions with respect to a fixed set $U \subseteq T$. By abuse of language, we call the corresponding families of divisions also Janet, Thomas, and Pommaret division, respectively. It will always become clear from the context, which point of view we take.

Example 4.4.7. *Let us take $X = \{x, y\}$ and $U = \{x^{10}, x^9, x^3y^3, xy^9\}$. The multiplicative variables corresponding to the Janet, Thomas, and Pommaret division on U are given by the following table.*

U	*Janet*	*Thomas*	*Pommaret*
x^{10}	x, y	x	x, y
x^9	y		x, y
x^3y^3	y		y
xy^9	y	y	y

The diagrams in Figure 4.1 are visualisations of the respective divisions. They depict the exponent vectors of the elements in U by black dots. Multiples of an element are shown by thick black lines or a grey area.

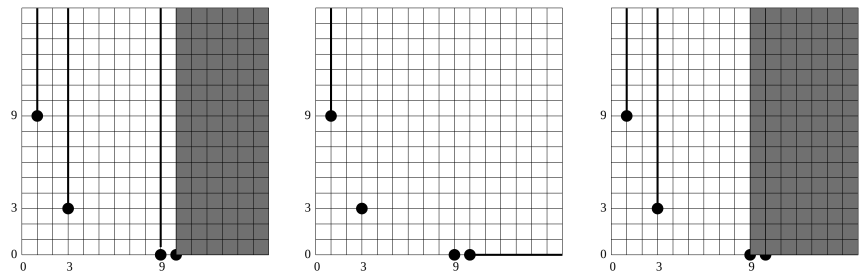


Figure 4.1: Janet, Thomas, and Pommaret division on U

Whereas the Pommaret division can contain embedded cones (the cone of x^{10} is contained in the cone of x^9), cones in a Janet or Thomas division are always disjoint.

Example 4.4.8. *Let us give another example with 3 variables x_1, x_2, and x_3. We set $U = \{m_0, m_1, m_2, m_3\}$ where*

$$m_0 = x_1x_2x_3, \qquad m_1 = x_1^7x_3, \qquad m_2 = x_2^7x_1, \qquad m_3 = x_3^7x_2.$$

The multiplicative variables are given through the following table.

U	*Janet*	*Pommaret*	*Thomas*	*SD*
$x_1x_2x_3$	x_3	x_3		x_1, x_2, x_3
$x_1^7x_3$	x_1, x_2, x_3	x_3	x_1	x_1, x_3
$x_2^7x_1$	x_2, x_3	x_2, x_3	x_2	x_1, x_2
$x_3^7x_2$	x_2, x_3	x_3	x_3	x_2, x_3

The last column represents the Sliced Division *which we introduce and describe in more detail in Chapter 6.*

We have visualised the above divisions in Figure 4.2.

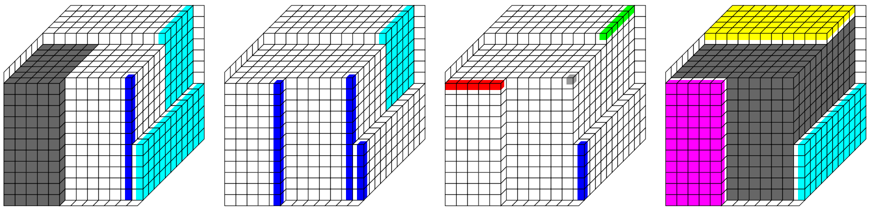
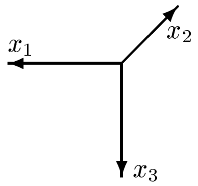


Figure 4.2: Janet, Pommaret, Thomas, and Sliced Division on U

In analogy to Figure 4.1, the pictures should be interpreted as the exponent vectors of the terms (and their multiples) according to the following 3-dimensional coordinate system.



So we look from "outside" onto a finite part of the term ideal $\langle U \rangle$.

The cones attached to the terms appear in various colours which correspond to the multiplicative variables of the respective vertex of the cone. They are assigned in the following way.

colour	*multiplicative*
dark grey	x_1, x_2, x_3
red	x_1
green	x_2
blue	x_3
yellow (red+green)	x_1, x_2
magenta (red+blue)	$x_1, \quad x_3$
cyan (green+blue)	x_2, x_3

If some element appears white then it is not covered by any cone of the decomposition. In fact, the coloured part represents $\langle U \rangle_{\mathcal{M}}$ for the corresponding partial division $\mathcal{M}$ on U.

Gerdt and Blinkov prove in [GB98a] that the Janet, Thomas, and Pommaret divisions are involutive, continuous, and constructive. They also show that the Janet and Thomas divisions are noetherian whereas the Pommaret division is not.

Gerdt and Blinkov [GB98a] give further explicit divisions whereas Apel [Ape98a] is not concerned with particular divisions but instead considers the whole lattice of partial divisions and argues to prefer a maximal (with respect to the refinement relation, cf. Definition 4.1.7) admissible division in each step of the involutive basis algorithm.

4.5 Matching the Different Approaches

There is a major difference between the approach of Gerdt and Blinkov and the approach of Apel. Gerdt and Blinkov always consider whole families $\mathfrak{L} = \left(\mathcal{L}^U\right)_{U\in\mathbb{P}_{\mathrm{fin}}(T)}$ of partial divisions where the divisions are connected by the filter condition (4.4). Apel defines properties of partial divisions only in connection with a given set of terms. For the involutive basis algorithm (cf. page 32) to be finite, he requires that the partial divisions used during each iteration of the involutive basis algorithm refine the Thomas division on the leading terms of the intermediate basis, cf. [Ape98a]. This is, of course, only a statement about the family of partial divisions with respect to the sets actually occurring during the involutive basis computation and not about properties of partial division with respect to any other set of terms. Apel's admissibility conditions (4.10) and (4.11) are, however, slightly stronger than the condition (4.3) of Gerdt and Blinkov.

In fact, in both approaches, one is interested in completing a given set U in a way given by Procedure 1. However, Procedure 1 is not a finite process in general. It may even happen that the step in line 3 is impossible because $u \in V$ and $x \in X$ with $xu \notin \langle V\rangle_{\mathcal{M}^V}$ do not exist.

Procedure 1 OneStepCompletion

Call: $V := \textsc{OneStepCompletion}(U, \mathfrak{M})$

Input: U is a finite set of terms.
$\mathfrak{M} = \left(\mathcal{M}^W\right)_{W\in\mathbb{P}(T)}$ is a family of partial divisions.

Output: V is a finite set of terms such that $U \subseteq V \subset T$ and $\langle V\rangle_{\mathcal{M}^V} = \langle U\rangle$.

1: $V := U$
2: <u>while</u> $\langle V\rangle_{\mathcal{M}^V} \neq \langle U\rangle$ <u>do</u>
3: Choose $u \in V$ and $x \in X$ with $xu \notin \langle V\rangle_{\mathcal{M}^V}$
4: $V := V \cup \{xu\}$
5: <u>return</u> V

Apel ensures correctness and termination by the requirement that for each intermediate set V (as defined in Procedure 1) the corresponding partial division is admissible on V and refines the Thomas division on V. Mainly for the justification of termination, the following Lemma is important.

Lemma 4.5.1. *[Ape98a, Lemma 4.3] Let $\mathcal{M} = (t[Y_t])_{t\in T}$ be a partial division on $U \subseteq T$ which refines the Thomas division on U. Then*

$$\forall u \in U\ \forall x \in X \setminus Y_u : xu \trianglelefteq \mathrm{lcm}(U). \tag{4.17}$$

Proof. Let $(u[Z_u])_{u\in U}$ be the Thomas division on U and let $u \in U$. Then $x \in X \setminus Y_u \subseteq X \setminus Z_u$ implies the existence of $v \in U$ such that $\deg_x u < \deg_x v$ and thus $xu \trianglelefteq \operatorname{lcm}(U)$ follows immediately. $\square$

Conditions (4.3) and (4.4) of Gerdt and Blinkov are not enough to entail correctness and termination of Procedure 1, but in [GB98a] it is shown that Procedure 1 is correct and terminates for constructive and noetherian involutive divisions if in line 3 always the smallest (with respect to some admissible term order) prolongation is chosen, i. e., a normal selection strategy is used.

For a better comparison let us consider families of partial divisions in the following, even though Apel never considers such objects in his theory.

It is obvious that the Gerdt/Blinkov approach does not cover all families $\mathfrak{M} = \left(\mathcal{M}^U\right)_{U\in\mathbb{P}(T)}$ where for each $U \subseteq T$ the partial division $\mathcal{M}^U$ is admissible on U, since none of the restrictions like constructivity or the filter condition (4.4) are apparent in Apel's theory. Even if we further require that each $\mathcal{M}^U$ refines the Thomas division on U, the family $\mathfrak{M}$ need not fulfil the filter condition. Note also that Gerdt and Blinkov define the family of partial divisions only over all finite subsets of T. Such a restriction is not necessary in Apel's theory.

In turn, it is also not the case that any involutive division in the sense of Gerdt and Blinkov is a family of admissible partial divisions. This is trivial since the admissibility conditions (4.10) and (4.11) are more restrictive than condition (4.3). Another reason is that Apel requires refinement of the Thomas division in order to make the Procedure 1 terminate. In the theory of Gerdt and Blinkov such a refinement is not necessary. It is also not true that all constructive noetherian involutive divisions $\mathfrak{L} = \left(\mathcal{L}^U\right)_{U\in\mathbb{P}_{\mathrm{fin}}(T)}$ are families of admissible partial divisions in the sense of Apel. We show this by a counter example.

Example 4.5.2. *Let for this example $X := \{x, y, z\}$ and $T := [X]$. Furthermore let $W = \{x^2, y^2, z^2\}$, $U \subseteq T$, and $\mathcal{T}^U = \left(u\left[Z_u^U\right]\right)_{u\in T}$ be the Thomas division on U. We define a division $\mathcal{L}^U = \left(t\left[Y_t^U\right]\right)_{t\in T}$ by*

$$Y_t^U := \begin{cases} X & \text{if } (U,t) \in \left\{\left(\left\{x^2,y^2\right\},x^2\right),\left(\left\{y^2,z^2\right\},y^2\right),\left(\left\{z^2,x^2\right\},z^2\right)\right\}, \\ \{x,y\} & \text{if } U = W \text{ and } t = x^2, \\ \{y,z\} & \text{if } U = W \text{ and } t = y^2, \\ \{z,x\} & \text{if } U = W \text{ and } t = z^2, \\ Z_t^U & \text{otherwise.} \end{cases}$$

Note that $\mathcal{L}_u^U \cap \mathcal{L}_v^U = \emptyset$ for any $u, v \in U$ with $u \neq v$.

We can define $\mathcal{L}^U$ for each $U \subseteq T$ and form the family of partial divisions $\mathfrak{L} = \left(\mathcal{L}^U\right)_{U\in\mathbb{P}_{\mathrm{fin}}(T)}$. The filter condition (4.4) is readily verified, so $\mathfrak{L}$ is an involutive division. As one easily recognises, there are exactly 4 sets, namely $\{x^2, y^2\}$, $\{y^2, z^2\}$, $\{z^2, x^2\}$, and W, on which the division $\mathfrak{L}$ differs from the Thomas division, but still refines the Thomas division.

$\mathfrak{L}$ is noetherian. *This follows from the fact that the Thomas division is noetherian. We need only consider sets $U \subseteq W$ where $\mathcal{L}^U$ differs from $\mathcal{T}^U$. However, if we add the monomial $x^2y^2z^2$ to U we are again in a Thomas situation, and therefore see that each set is finitely generated with respect to $\mathfrak{L}$.*

$\mathfrak{L}$ ***is continuous.*** *Continuity is, in fact, a statement that refers only to one fixed set U. We must only consider the cases $U \subseteq W$ where $\mathcal{L}^U$ differs from $\mathcal{T}^U$, since otherwise continuity follows from the continuity of the Thomas division. However, for all these sets it holds that if $u \in U$ and $\xi \in X \setminus Y_u^U$ there is no $v \in U$ such that $\xi u \in \mathcal{L}_v^U$. So any sequence that must be considered according to Definition 4.2.2 has length* 1*. Therefore, $\mathfrak{L}$ is continuous.*

$\mathfrak{L}$ ***is constructive.*** *We distinguish 4 cases.*

1. *Let us consider $U = W$ first. By symmetry it suffices to take $u = y^2$. The only non-multiplicative variable for u with respect to W is x. We must show* (4.7)*. Let us take $v = x^2$ or $v = z^2$. Then w does not divide xu for any $w \in \mathcal{L}_{x^2}^U \cup \mathcal{L}_{z^2}^U$. We conclude $xu \notin w[X] \supseteq \mathcal{L}_w^{U\cup\{w\}}$. Now consider the remaining case $v = y^2$ and let $w \in \mathcal{L}_v^U$. If $w = u$ then clearly $xu \notin \mathcal{L}_w^{U\cup\{w\}} = \mathcal{L}_u^U$. If $w \neq u$ then $x \notin Y_w^{U\cup\{w\}} = Z_w^{U\cup\{w\}}$ is non-multiplicative with respect to the Thomas division since $x \ntrianglelefteq w$ and $x^2 \in U$. Hence, $xu \notin \mathcal{L}_w^{U\cup\{w\}}$.*

2. *If U is one of the sets $\emptyset$, $\{x^2\}$, $\{y^2\}$, $\{z^2\}$, there is nothing to check since there are no non-multiplicative variables.*

3. *Let us consider the cases where U is one of the sets $\{x^2, y^2\}$, $\{y^2, z^2\}$, $\{z^2, x^2\}$. For symmetry reasons, it suffices to consider $U = \{x^2, y^2\}$. The only non-multiplicative variable is $x \in X \setminus Y_{y^2}^U$, so take $u = y^2$. Again, we must check* (4.7)*. First, take $v = x^2$. Any $w \in \mathcal{L}_v^U$ does not divide xu. We conclude $xu \notin w[X] \supseteq \mathcal{L}_w^{U\cup\{w\}}$. Now, take $v = y^2$ and let $w \in \mathcal{L}_v^U = y^2[\{y,z\}]$. If $w = u$ then clearly $xu \notin \mathcal{L}_w^{U\cup\{w\}} = \mathcal{L}_u^U$. If $w \neq u$ then $x \notin Y_w^{U\cup\{w\}} = Z_w^{U\cup\{w\}}$ since $x \ntrianglelefteq w$ and $x^2 \in U$. Hence, $xu \notin \mathcal{L}_w^{U\cup\{w\}}$.*

4. *For any other set $U \nsubseteq W$, we can take over the proof of the constructivity of the Thomas division since no other divisions than the Thomas division is involved in the proof, cf. [GB98a]. Indeed, take $u \in U$, $\xi \in X \setminus Y_u^U$ and assume there is $v \in U$, $w \in \mathcal{L}_v^U$ such that $\xi u \in \mathcal{L}_w^{U\cup\{w\}}$. By the properties of the Thomas division $\mathcal{L}^U = \mathcal{T}^U$, we have $\eta \in Y_v^{U\cup\{w\}}$ for all $\eta \in Y_w^{U\cup\{w\}}$ with* $\deg_\eta v = \deg_\eta w$. *Moreover $Y_v^U \supseteq Y_v^{U\cup\{w\}}$ by the filter property* (4.4)*. Hence, $\xi u \in \mathcal{L}_w^{U\cup\{w\}} \subseteq \mathcal{L}_v^U$.*

$\mathcal{L}^W$ ***is not admissible on*** W***.*** *Suppose for a contradiction that there is an irreflexive linear order $\sqsubset$ on W such that $\mathcal{L}^W$ is admissible on $(W, \sqsubset)$. Since $x^2[X] \cap y^2[\{y,z\}] = \emptyset$, but $y^2[X] \cap x^2[\{x,y\}] \neq \emptyset$, admissibility entails $x^2 \sqsubset y^2$. Analogously, we get $y^2 \sqsubset z^2$ and $z^2 \sqsubset x^2$, and thus the contradiction $x^2 \sqsubset x^2$.*

4.6 A Unifying Approach

The current situation is unsatisfactory. There are two theories for treating nearly the same thing. We would like to remedy this by concentrating on the most important points. One could say that Apel as well as Gerdt/Blinkov build their theories around Procedure 1 in the sense that they aim at making this procedure a finite process fulfilling its specifications. Of course, they go further than just considering terms, but a finite one-step completion process is a main feature of a Janet-like algorithm that they wanted to retain.

Our approach covers Apel's approach in the sense that every admissible partial division is a "good" division in our sense, and it covers the approach of Gerdt and Blinkov in the sense that every continuous involutive division is "good". Following Apel, we require the partial divisions to refine the Thomas division in order to ensure termination of Procedure 1. The filter condition (4.4) and the concept of a constructive involutive division are not needed in our approach.

When Janet introduced his kind of separation of variables into multiplicative and non-multiplicative ones for a finite set $U \subseteq T$, cf. Definition 4.4.1, in fact, he had in mind a disjoint decomposition of T, cf. [Jan20, Jan29]. Such a decomposition is of great importance in the Riquier-Janet Theory, cf. [Riq10] and [Tho29, Tho34]. By taking over the main algorithmic ideas from Janet, Apel [Ape98a] as well as Gerdt and Blinkov [GB98a] were not interested in a decomposition of T, but rather only of $\langle U\rangle$. They even weakened the requirement of a disjoint decomposition of $\langle U\rangle$ and allowed certain inclusions of cones.

If one speaks of a partial division $(t[Y_t])_{t\in T}$ on $U \subseteq T$, it is actually never of real interest what the values of Y_t for $t \in T \setminus U$ are. For a given set $U \subseteq T$, one can introduce equivalence classes of partial divisions with respect to the $\equiv_U$ relation as Apel showed in [Ape98a]. Since we already have the convenient notion of refinement of partial divisions available, it is unnecessary to introduce the concept of equivalence classes. In fact, Definition 4.1.6 can be considered as a tool to work with equivalence classes without ever defining them. We therefore develop our theory on partial divisions defined on the whole set T of terms, i. e., on families of cones indexed by all terms and not only the terms of U.

Like Apel and unlike Gerdt and Blinkov we state our definitions for a fixed set of terms instead of requiring properties on a whole family of partial divisions.

Let us start with the disjointness property.

Definition 4.6.1. Let $U \subseteq T$. A partial division $\mathcal{M}$ is called **disjoint on** U if for all $u, v \in U$ with $u \neq v$ the relation

$$\mathcal{M}_u \cap \mathcal{M}_v = \emptyset \tag{4.18}$$

holds. It is called **essentially disjoint on** U if for all $u, v \in U$ with $u \neq v$ the implication

$$\mathcal{M}_u \cap \mathcal{M}_v \neq \emptyset \implies \mathcal{M}_u \subset \mathcal{M}_v \vee \mathcal{M}_u \supset \mathcal{M}_v \tag{4.19}$$

holds.

Apart from admissibility, the concept of a *complete set* is an important notion. The basic idea is that the cones of a partial division should "cover" the whole monoid ideal generated by the given terms.

Definition 4.6.2. Let $\mathcal{M}$ be a partial division. The set $U \subseteq T$ is called **$\mathcal{M}$-complete** if $\mathcal{M}$ is essentially disjoint on U and

$$\langle U\rangle_{\mathcal{M}} = \langle U\rangle\,. \tag{4.20}$$

The set U is called **complete** if U is $\mathcal{M}$-complete for some partial division $\mathcal{M}$.

The previous definition corresponds to the concept of an involutive set U, but is not identical since in Definition 4.2.4 there is a reference to a whole family of partial divisions whereas a complete set is defined in terms of a single partial division. This difference is certainly of minor importance, but nevertheless we would like to call this property *complete*, since the concept of a complete set of terms (with respect to the Janet division) appears already in [Jan20]. Apel did not explicitly define the concept of a complete set in [Ape98a], but also uses the corresponding property.

As in the other two approaches, our goal is to give conditions such that Procedure 1 becomes a finite and correct algorithm. We start with an auxiliary notion which will be subsumed by another notion later, cf. Proposition 4.6.7.

Definition 4.6.3. Let $\mathcal{M}$ be an essentially disjoint partial division on $U \subseteq T$. The set U is called **$\mathcal{M}$-extensible** if

$$\exists u \in U \; \exists x \in X : xu \notin \langle U \rangle_{\mathcal{M}} . \tag{4.21}$$

The notion of an $\mathcal{M}$-extensible set U characterises the property that there is always a non-multiplicative prolongation which is not yet covered by the existing cones. A justification for the above definition is given by the following lemma.

Lemma 4.6.4. *Let U and $\mathfrak{M}$ be as in the specification of Procedure 1. If each set V that occurs during the run of the procedure is either $\mathcal{M}^V$-complete or $\mathcal{M}^V$-extensible and each such division $\mathcal{M}^V$ refines the Thomas division on V then Procedure 1 terminates and fulfils its specification.*

Proof. Obviously, the choice in line 3 of Procedure 1 is possible. Therefore, in each iteration the set V grows by adding the new element $xu \notin V$. By Lemma 4.5.1, we have $xu \trianglelefteq \operatorname{lcm} U$. Since there are only finitely many terms dividing $\operatorname{lcm} U$, the procedure must terminate with $\langle V \rangle_{\mathcal{M}^V} = \langle U \rangle$ as claimed in the specification. □

In view of the application of partial divisions in a polynomial context, another property becomes important, cf. Theorem 7.2.1. Consider the following example, which appears in [GB98a].

Example 4.6.5. *Let $X := \{x, y, z\}$, $U := \{x, y, z\}$ and let the partial division $\mathcal{M} = (u[Y_u])_{u \in U}$ be generated by the sets $Y_x = \{x, z\}$, $Y_y = \{x, y\}$, and $Y_z = \{y, z\}$.*

Here we find an infinite sequence $u_0, u_1, \ldots$ with the property that for each $i \in \mathbb{N}$ there exists $\xi \in X \setminus Y_{u_i}$ such that $\xi u \in \mathcal{M}_{u_{i+1}}$. Simply cycle *forever in $u_0 = x$, $u_1 = y$, $u_2 = z$, etc. Obviously, $\langle U \rangle_{\mathcal{M}} \neq \langle U \rangle$ and this division is thus not $\mathcal{M}$-extensible.*

Of course, one could remedy the problem in the previous example by adding a multiplicative prolongation, x^2, say. However, if one allows also multiplicative prolongations, one would sacrifice the whole theory, since it is build heavily on the method of *non-multiplicative prolongation* and *multiplicative reduction*. So we forbid such *cycles*.

Definition 4.6.6. Let $U \subseteq T$ be a finite set of terms and $\mathcal{M}$ be a partial division. Let $J = \{0, 1, \ldots, k-1\}$ and $I = J \cup \{k\}$ for some $k \in \mathbb{N}$ or let

$J = I = \mathbb{N}$. A finite or infinite sequence $C = (u_i)_{i \in I}$ of elements of U is called an **$\mathcal{M}$-sequence on U** if

$$\forall j \in J \; \exists x \in X : xu_j \in \mathcal{M}_{u_{j+1}} \setminus \mathcal{M}_{u_j}.$$

If every $\mathcal{M}$-sequence on U is finite then $\mathcal{M}$ is called **cycle-free on U**.

This new concept in connection with a refinement of the Thomas division is also sufficient for the termination of Procedure 1 as one easily concludes from the following proposition and Lemma 4.6.4. We have formulated the exact statement in Corollary 4.6.9 below.

Proposition 4.6.7. *Let $U \subset T$ be a finite set of terms and $\mathcal{M} = (t[Y_t])_{t \in T}$ be a partial division which is essentially disjoint and cycle-free on U. If U is not $\mathcal{M}$-complete then U is $\mathcal{M}$-extensible.*

Proof. Take $u_0 \in U$ and $t_0 \in T$ with $t_0 u_0 \notin \langle U \rangle_{\mathcal{M}}$. Then $t_0 u_0 = \frac{t_0}{y_0} \cdot y_0 u_0$ for some $y_0 \in X \setminus Y_{u_0}$ that divides t_0. If $y_0 u_0 \notin \langle U \rangle_{\mathcal{M}}$ then U is $\mathcal{M}$-extensible. Otherwise there exists $u_1 \in U$ and $t_1 \in [Y_{u_1}]$ such that $y_0 u_0 = t_1 u_1$. The relation $\frac{t_0}{y_0} \cdot t_1 u_1 \in \mathcal{M}_{u_1}$ is impossible, because $\frac{t_0}{y_0} \cdot t_1 u_1 = t_0 u_0 \notin \langle U \rangle_{\mathcal{M}}$. Hence, we find a divisor $y_1 \in X \setminus Y_{u_1}$ of $t_0 t_1$ such that $t_0 u_0 = \frac{t_0 t_1}{y_0 y_1} \cdot y_1 u_1$. If $y_1 u_1 \in \langle U \rangle_{\mathcal{M}}$, we can continue the above process and thus construct an $\mathcal{M}$-sequence $u_0, u_1, \ldots$ of terms of U. Since $\mathcal{M}$ is cycle-free, this sequence must be finite. Let u_k be the last element of this sequence. By construction there is $y_k \in X \setminus Y_{u_k}$ and $y_k u_k \notin \langle U \rangle_{\mathcal{M}}$, i. e., U is $\mathcal{M}$-extensible. $\square$

The most important properties that we use in the next chapters are collected in the following definition.

Definition 4.6.8. A partial division $\mathcal{M}$ is said to be a **suitable partial division on U** if and only if it is essentially disjoint on U, cycle-free on U, and refines the Thomas division on U.

Corollary 4.6.9. *Let U and $\mathfrak{M}$ be as in the specification of Procedure 1. If for each set V that occurs during the run of Procedure 1 the division $\mathcal{M}^V$ is a suitable partial division on V then Procedure 1 terminates and fulfils its specification.*

Proof. If in Procedure 1 the set V is not $\mathcal{M}^V$-complete then it is $\mathcal{M}$-extensible by the previous proposition. The statement follows then by Lemma 4.6.4. $\square$

Let us compare our concepts to that of Gerdt and Blinkov.

Proposition 4.6.10. *Let $\mathfrak{L} = \left(\mathcal{L}^U\right)_{U \in \mathbb{P}_{\mathrm{fin}}(T)}$ be an involutive division. Then for each finite set $U \subseteq T$ the partial division $\mathcal{L}^U$ is essentially disjoint.*

Furthermore, $\mathfrak{L}$ is continuous if and only if for each $U \subseteq T$ the partial division $\mathcal{L}^U$ is cycle-free on U.

Remark 4.6.11. From the previous proposition it follows immediately that in Example 4.5.2 for all $U \subseteq T$ the division $\mathcal{L}^U$ is a suitable partial division.

In fact, the notions of a locally involutive set and a set which is not $\mathcal{M}$-extensible are tightly connected.

Proposition 4.6.12. *Let $\mathfrak{L} = \left(\mathcal{L}^U\right)_{U \in \mathbb{P}_{\mathrm{fin}}(T)}$ be an involutive division. A set $V \subseteq T$ is locally involutive with respect to $\mathfrak{L}$ if and only if for every finite set $U \subseteq T$ the set V is not $\mathcal{L}^U$-extensible.*

Since we require a suitable partial division to refine the Thomas division, our notion cannot cover all involutive divisions in the sense of Gerdt and Blinkov. The reason for allowing only partial divisions which refine the Thomas division is two-fold. Firstly, it is very easy to check whether a partial division refines the Thomas division and it enforces termination of Procedure 1. Secondly, if an essentially disjoint, cycle-free partial division on some set U of terms does not refine the Thomas division then it would be possible to find a non-multiplicative prolongation $(x, u) \in X \times U$ with $xu \ntrianglelefteq \operatorname{lcm}(U)$. Such a term xu would be added to U in the completion process and thus increase the least common multiple. We want to exclude such a "bad" property, since it would give rise to an unnecessarily high number of elements in a complete set and thus increase the complexity of the completion process.

Here is the connection of our concept to the notion of an $\mathfrak{L}$-completion of Gerdt and Blinkov.

Proposition 4.6.13. *Let $\mathfrak{L} = \left(\mathcal{L}^U\right)_{U \in \mathbb{P}_{\mathrm{fin}}(T)}$ be an involutive division. A finite set $V \subset T$ is an $\mathfrak{L}$-completion of a set $U \subseteq V$ if $\langle V \rangle = \langle U \rangle$ and V is $\mathcal{L}^V$-complete.*

From Lemma 4.5.1 we easily get the following statement.

Corollary 4.6.14. *Let $\mathfrak{L} = \left(\mathcal{L}^U\right)_{U \in \mathbb{P}_{\mathrm{fin}}(T)}$ be an involutive division such that for each $U \subseteq T$ the partial division $\mathcal{L}^U$ refines the Thomas division on U. Then every finite set $U \subseteq T$ is finitely generated and thus $\mathfrak{L}$ is noetherian.*

There is no correspondence in our theory to the filter axiom (4.4) or a constructive involutive division since we deliberately wanted to exclude these additional restrictions. We have shown above that it is enough for the termination of Procedure 1 to replace these notions by the requirement that the divisions refine the Thomas division.

The connection of our notions to Apel's admissibility is as follows.

Proposition 4.6.15. *Let $U \subseteq T$ be a finite set of terms. If a partial division is U-admissible then it is essentially disjoint and cycle-free on U.*

Proof. Let $\sqsubset$ be an irreflexive linear order on U and let $\mathcal{M}$ be admissible on $(U, \sqsubset)$. That $\mathcal{M}$ is essentially disjoint follows easily from the definitions. Let $u_0, u_1, \ldots$ be any $\mathcal{M}$-sequence. Admissibility of $\mathcal{M}$ implies $u_{i+1} \sqsubset u_i$ for all i. Since U is finite, the $\mathcal{M}$-sequence must be finite. Therefore, $\mathcal{M}$ is cycle-free. □

Our concept of an essentially disjoint, cycle-free partial division is a slightly weaker condition than that of an admissible division. This can easily be seen from Example 4.5.2. As we have shown there, $\mathfrak{L}$ is a continuous involutive division, so in particular, $\mathcal{L}^W$ is an essentially disjoint and cycle-free partial division on W by Proposition 4.6.7. But $\mathcal{L}^W$ is not admissible on W.

In particular, it follows that the classical divisions of Janet, Thomas, and Pommaret are essentially disjoint and cycle-free, since they are admissible as is proven in [Ape98a].

The Janet and Thomas divisions are also suitable partial divisions. The Pommaret division is not a suitable partial division, since it does not always refine the Thomas division on the same corresponding set of terms. Take, for example, $X = \{x_1, x_2\}$ and $U = \{x_2^3\}$, then x_2^3 has x_1 and x_2 as Thomas-multiplicative, but only x_2 as Pommaret-multiplicative. In fact, Procedure 1 would never stop when started on the set U and when on all sets the Pommaret division is used.

Let us summarise this section. We have defined five notions, namely

- an essentially disjoint partial division,
- an $\mathcal{M}$-sequence,
- a cycle-free partial division,
- a complete set of terms, and
- a suitable partial division,

which we use in the following chapters.

The concept of an *essentially disjoint* partial division generalises Janet's initial idea of a disjoint decomposition of the set of all terms. It is the basic concept on which all the further theory is built. The concept of an essentially disjoint and *cycle-free* partial division basically ensures correctness of Procedure 1 by allowing to add one more element if completeness has not yet been reached. A *refinement of the Thomas division* can be seen as providing an upper bound on the number of elements that can be added and thus turns Procedure 1 into a (terminating) algorithm. These concepts are subsumed in Definition 4.6.8 under a *suitable partial division* which is sufficient to build a theory of (polynomial) involutive bases on it.

Our approach follows that of Apel in the sense that we only give definitions for a partial division with respect to a single set, but never connect partial divisions with respect to different sets as is done in the approach of Gerdt and Blinkov. We basically only take condition (4.3) of Gerdt and Blinkov and get rid of condition (4.4). This leads us to the concept of an essentially disjoint partial division, which is a weaker condition than that for an admissible partial division. Our concept of a cycle-free partial division borrows the idea from the concept of a continuous involutive division of Gerdt and Blinkov. We believe that *cycle-free* is more intuitive than *continuous*. Instead of taking the concepts of a noetherian or constructive involutive division, we rather follow Apel again and require a property that is easier to check, namely, the partial divisions should refine the Thomas division. We see this approach as a reasonable compromise of the approaches of Apel and Gerdt/Blinkov.

Chapter 5

Involutive Bases

In this chapter we recall the basic theory of involutive bases in a polynomial context. We present the Janet basis algorithm and its generalised form, the involutive basis algorithm. Additionally, we give the necessary proofs in order to employ our *suitable partial divisions* during the involutive basis algorithm.

Involutive bases have their origin in the Riquier-Janet Theory of partial differential equations at the beginning of the 20th century, [Riq10], [Jan20, Jan29], see also [Tho29, Tho34]. Janet was the first to give an algorithm to transform a system of partial differential equations (PDEs) into a form (passive system) required for Riquier's Theorem.

There is a simple translation from linear PDEs with constant coefficients in one dependent variable u and n independent variables $z_1, \dots, z_n$ to polynomials in n variables $x_1, \dots, x_n$. One simply rewrites a PDE by replacing any term of the form

$$\frac{\partial^{d_1+\cdots+d_n} u}{\partial z_1^{d_1} \dots \partial z_n^{d_n}}$$

by

$$x_1^{d_1} \dots x_n^{d_n}.$$

Zharkov and Blinkov [ZB93] employed such a translation in order to bring Janet's method into a polynomial context. They showed that it is an alternative method to compute Gröbner bases.

If the coefficients of the PDE are not constant but still polynomials in the independent variables, the same translation leads to elements in a Weyl algebra. Recall that a Weyl algebra is a (non-commutative) free associative $\mathbb{K}$-algebra

$$\mathbb{K}\langle z_1, \dots, z_n, x_1, \dots, x_n \rangle$$

modulo the commutation rules

$$z_i z_j = z_j z_i, \quad x_i x_j = x_j x_i, \quad x_i z_j = z_j x_i, \quad x_i z_i = z_i x_i + 1$$

for all $1 \leq i, j \leq n$ with $i \neq j$. We refer to [SST99] and [HSS02] for more details. The latter article explicitly treats involutive bases in the context of Weyl algebras.

This thesis considers mainly the case of polynomials, i. e., linear PDEs with constant coefficients by the translation above, but most results equally apply in a more general setting of algebras of solvable type (cf. [KRW90], [Kre93], [HSS02]), so in particular the Weyl algebra is covered.

5.1 Preliminaries

Let us start with a formal definition in the context of polynomial systems which describes an involutive basis without any relation to an algorithm to compute it. This definition builds on the notion of a Gröbner basis as invented in Buchberger's PhD thesis [Buc65]. A modern treatment of the Gröbner theory can be found, for example, in [BW93], [AL94], [Frö97].

Several definitions of the concept of an involutive basis are given in the literature. We repeat here the definition given in [AH02] which builds on a characterisation theorem of Apel [Ape98a]. See also Theorem 5.3.3.

Definition 5.1.1. Let $G \subseteq \mathbb{K}[X]$ be a set of polynomials and let I be an ideal of $\mathbb{K}[X]$. G is called an **involutive basis** if G is a Gröbner basis and $\operatorname{lt} G$ is complete. G is called an **involutive basis of** I if it is an involutive basis and $\operatorname{Id}(G) = I$. An involutive basis G is an $\mathcal{M}$**-involutive basis** if $\operatorname{lt} G$ is $\mathcal{M}$-complete for some partial division $\mathcal{M}$.

From Definition 5.1.1 it is relatively easy to derive an algorithm to compute an involutive basis: one first determines a Gröbner basis of the given set of polynomials and afterwards completes this basis by a polynomial analogue of Procedure 1 to an involutive basis. Such a process can, of course, not be more efficient than Buchberger's algorithm to compute Gröbner bases (cf. [Buc65], [Buc85]), but it is also not the algorithm that Janet gave at the beginning of the 20th century. The main features of Janet's algorithm are *non-multiplicative prolongation* and *multiplicative reduction*. Let us clarify these concepts.

As an extension of Definition 4.1.5, we define the notion of prolongation for polynomials.

Definition 5.1.2. A pair of the form (x, g) with $x \in X$ and $g \in \mathbb{K}[X]$ is called a **prolongation**. If $x \in X$ and $g \in \mathbb{K}[X]$, we call the product xg an x**-prolongation of** g or simply a **prolongation of** g.

A prolongation (x, g) or x-prolongation of g is called **multiplicative** (resp. **non-multiplicative**) if the same holds with g replaced by its leading term, cf. Definition 4.1.5.

It is important to note that a prolongation corresponds to a multiplication by one variable only. It should be considered as one step of a multiplication by a whole term.

Definition 5.1.3. [AH02] Let $\mathcal{M} = (t[Y_t])_{t \in T}$ be a partial division, $G \subseteq \mathbb{K}[X] \setminus \{0\}$, and $h, h' \in \mathbb{K}[X]$. We define the involutive reduction relation $\to_{G,\mathcal{M}}$ by

$$h \to_{G,\mathcal{M}} h'$$

if and only if

$$\exists g \in G, t \in [Y_{\operatorname{lt} g}], c \in \mathbb{K} \setminus \{0\} : h' = h + ctg \wedge \operatorname{lt}(tg) \in \operatorname{supp} h \setminus \operatorname{supp} h'.$$

The symbol $\to^*_{G,\mathcal{M}}$ denotes the reflexive, transitive closure of $\to_{G,\mathcal{M}}$ where $0 \to^*_{G,\mathcal{M}} 0$ by definition.

If there exists h' such that $h \to_{G,\mathcal{M}} h'$, we say that h is **$\mathcal{M}$-reducible modulo** G. If we have $h \to^*_{G,\mathcal{M}} h'$ and $\operatorname{lt} h \in \mathcal{M}_{\operatorname{lt} g} \setminus \operatorname{supp} h'$ for $h \in \mathbb{K}[X] \setminus \{0\}$ and $g \in G$ then we say that h is **$\mathcal{M}$-top-reducible modulo** G, h is **$\mathcal{M}$-top-reducible by** g, and h is **$\mathcal{M}$-top-reduced using** g during the $\mathcal{M}$-reduction of h to h'.

We say that h is **in $\mathcal{M}$-involutive normal form modulo** G or **$\mathcal{M}$-irreducible modulo** G if there is no $h' \in \mathbb{K}[X]$ with $h \to_{G,\mathcal{M}} h'$. The polynomial h' is an **$\mathcal{M}$-involutive normal form of** h **modulo** G if $h \to^*_{G,\mathcal{M}} h'$ and h' is in $\mathcal{M}$-involutive normal form modulo G. Let $\mathrm{NF}_{\mathcal{M}}$ be a function which assigns to each polynomial $h \in \mathbb{K}[X]$ an $\mathcal{M}$-involutive normal form of h modulo G.

We write NF, $\to_G$, and $\to^*_G$ as abbreviations for $\mathrm{NF}_{\mathcal{O}}$, $\to_{G,\mathcal{O}}$, and $\to^*_{G,\mathcal{O}}$, respectively, where $\mathcal{O} = (t[X])_{t \in T}$ denotes the ordinary division.

Let G and $\mathcal{M}$ be as in the previous definition. Note that there is no $h' \in \mathbb{K}[X]$ such that $0 \to_{G,\mathcal{M}} h'$. Thus 0 is in $\mathcal{M}$-involutive normal form and in particular it is an $\mathcal{M}$-involutive normal form of 0 and thus $\mathrm{NF}_{\mathcal{M}}(0, G) = 0$. Further, if h is in $\mathcal{M}$-involutive normal form modulo G then $\operatorname{supp} h \cap \langle \operatorname{lt} G \rangle_{\mathcal{M}} = \emptyset$.

The abbreviations NF, $\to_G$, and $\to^*_G$ are justified by the fact that they describe the usual concepts of normal form, one-step reduction modulo G, and reduction modulo G in Gröbner basis theory.

Definition 5.1.4. By $[g_1, g_2] := \operatorname{lcm}(\operatorname{lt} g_1, \operatorname{lt} g_2)$ we abbreviate the least common multiple of the leading terms of two polynomials $g_1, g_2 \in \mathbb{K}[X] \setminus \{0\}$.

Definition 5.1.5. Let $g_1, g_2 \in \mathbb{K}[X] \setminus \{0\}$. We define the **S-polynomial** of g_1 and g_2 by

$$\operatorname{spol}(g_1, g_2) := \frac{t_1 g_1}{\operatorname{lc}(t_1 g_1)} - \frac{t_2 g_2}{\operatorname{lc}(t_2 g_2)}$$

where $t_1, t_2 \in T$ are such that $\operatorname{lt}(t_1 g_1) = \operatorname{lt}(t_2 g_2) = [g_1, g_2]$.

5.2 Janet Basis Algorithm

Janet's idea of employing only non-multiplicative prolongations was used by Wu [Wu91] and Zharkov and Blinkov [ZB93] in a polynomial context.

In fact, Wu computes Gröbner bases with a characteristic set-like scheme employing the Thomas division. Basically, Wu's algorithm computes an autoreduced set F' from a given set F of polynomials, then adds multiples of polynomials in F' to F' to obtain F'' so that $\operatorname{lt}(F'')$ forms a complete set with respect to the Thomas division (cf. Definition 4.4.4). Then a set F''' is computed which consists of all products of polynomials from F'' with respect to their corresponding non-multiplicative variables. The polynomials from F''' are reduced with respect to F''. If all remainders are zero, the set F' is called a *well-behaved set* by Wu (which is the same as a reduced Gröbner basis). Otherwise the set of non-zero remainders joined with F' is taken as the new F and the process is iterated. Reduction and auto-reduction are the usual reductions known from Gröbner theory.

The translation of Janet's algorithm by Zharkov and Blinkov follows more closely the lines of Buchberger's algorithm where S-polynomials are replaced by non-multiplicative prolongations and the ordinary reduction is replaced by a restricted reduction where only multiplication by corresponding multiplicative variables is allowed. Although Zharkov and Blinkov speak of Janet-divisor in [ZB93], they use, in fact, a division which is nowadays called Pommaret division (cf. Definition 4.4.5). With *auto-reduce* and *reduce* applied in the restricted sense (cf. Definition 5.1.3), the algorithm works as follows. For a given set F compute an auto-reduced set F'. Then reduce for each element $f \in F'$ and each non-multiplicative variable x of $\operatorname{lt} f$ the x-prolongation xf with respect to F' and collect all non-zero remainders in R. If $R = \emptyset$ then stop with the output F'. Otherwise set $F := F' \cup R$ and iterate the process.

The examples reported in [ZB93] where this algorithm outperforms Buchberger's algorithm has drawn attention of other researchers to this alternative method of computing Gröbner bases, for example, [GB98a], [GBC98], [Ape98a], [CG00], [GBY01b], [Hem03].

Several variants of Janet's algorithm are known in a polynomial context. Let us start with a simple form where $\mathcal{J} = (t[Y_t])_{t\in T}$ always stands for the Janet division on the corresponding set $\operatorname{lt} G$ of leading terms. For this simple form we take advantage of the fact that the Janet division is disjoint according to Definition 4.6.1.

Algorithm 2 JanetBasis

Call: $G = \text{JanetBasis}(F)$

Input: $F \subset \mathbb{K}[X] \setminus \{0\}$ is a finite set of polynomials with pairwise distinct leading terms.

Output: G is a Janet basis of the ideal of F in $\mathbb{K}[X]$.

$G := F$
$C := \{(x, g) \mid g \in G \wedge x \in X \setminus Y_{\operatorname{lt} g}\}$
while $C \neq \emptyset$ do
  Choose (x, g) from C such that $\operatorname{lt}(xg)$ is $\prec$-minimal; $C := C \setminus \{(x, g)\}$
  $h := \operatorname{NF}_{\mathcal{J}}(xg, G)$
  if $h \neq 0$ then
    $G := G \cup \{h\}$
    $C := \{(x, g) \mid g \in G \wedge x \in X \setminus Y_{\operatorname{lt} g}\}$
return G

Obviously, Algorithm 2 is very similar to Buchberger's algorithm, which is given in a simple form as Algorithm 3.

Let us compare these two algorithms.

- Both algorithms are critical element/completion algorithms. Whereas in Algorithm 3 the critical elements are S-pairs (cf. line 2), Algorithm 2 forms non-multiplicative prolongations as the set C of critical elements.
- Both algorithms proceed by taking a critical element, reducing it and adjoining a non-zero remainder to the basis.
- The reduction in Algorithm 2 is done in a restricted way in the sense of Definition 5.1.3, whereas the ordinary reduction used in Algorithm 3.

Algorithm 3 GRÖBNERBASIS

Call: $G = \text{GRÖBNERBASIS}(F)$

Input: $F \subset \mathbb{K}[X] \setminus \{0\}$ is a finite set of polynomials.

Output: G is a Gröbner basis of the ideal of F in $\mathbb{K}[X]$.

1: $G := F$
2: $C := \{\{f, g\} \mid f, g \in G\}$
3: while $C \neq \emptyset$ do
4: Choose $\{f, g\}$ from C; $C := C \setminus \{\{f, g\}\}$
5: $h := \mathrm{NF}(\mathrm{spol}(f, g), G)$
6: if $h \neq 0$ then
7: $G := G \cup \{h\}$
8: $C := C \cup \{\{g, h\} \mid g \in G\}$
9: return G

- In fact, Algorithm 2 as well as Algorithm 3 compute S-polynomials. In Algorithm 3, this is obvious. In Algorithm 2, they appear when during a reduction in line 5 a top-reduction is performed.
- In line 8 of Algorithm 3, new critical elements are added to the set of critical elements whereas line 8 of Algorithm 2 all critical elements are recomputed due to the fact that the sets Y_t change if a new polynomial is added to the basis.
- As known from Gröbner theory, critical pairs can be processed in any order in Buchberger's algorithm. Janet's algorithm, however, requires a *normal selection* strategy in line 4.

We demonstrate a Janet basis computation on a non-trivial example which we continue in following sections.

Example 5.2.1. *Let $(x, y, z) = (x_1, x_2, x_3)$ and $X = \{x_1, x_2, x_3\}$. We compute a Janet basis of the following polynomials with respect to the degree reverse lexicographical term order. The prolongations are processed by a normal strategy, i. e., as the next prolongation we always choose one which has a minimal leading term with respect to the degree reverse lexicographical term order.*

$$f_1 = x^8y^5z^2 - x^4y^3z^3 + x^4y^3 - yz \tag{5.1}$$

$$f_2 = x^8y^4z^4 - x^4y^4z^3 + x^4y^2z^2 - y^2z \tag{5.2}$$

$$f_3 = x^6y^4z^9 - x^6y^2z^{11} + x^2y^2z^7 - x^2z^9 \tag{5.3}$$

We use a tabular form to denote the trace of the computation. Each line describes the polynomial h of an iteration of the while loop of Algorithm 2. The first column denotes the name of the polynomial which is added to the bases G. The second column contains a representation of h in terms of G. The third column shows the corresponding multiplicative variables to the whole polynomial which is written in column four. Each horizontal line represents the growth of the basis G and an adjustment of the multiplicative variables for each element of the basis. In this example the Janet division behaves "nicely" since the multiplicative variables only change for old polynomials after adding f_6. In fact, only *the multiplicative variables of f_3 change, that is why we have inserted a line for f_3 after f_6.*

Name	*Representation*	*Mult.*	*Polynomial*
f_1		x, y, z	$x^8y^5z^2 - x^4y^3z^3 + x^4y^3 - yz$
f_2		$x, \quad z$	$x^8y^4z^4 - x^4y^4z^3 + x^4y^2z^2 - y^2z$
f_3		y, z	$x^6y^4z^9 - x^6y^2z^{11} + x^2y^2z^7 - x^2z^9$
f_4	$yf_2 - z^2f_1$	y, z	$x^4y^5z^3 - x^4y^3z^5 + y^3z - yz^3$
f_5	xf_4	y, z	$x^5y^5z^3 - x^5y^3z^5 + xy^3z - xyz^3$
f_6	xf_5	y, z	$x^6y^5z^3 - x^6y^3z^5 + x^2y^3z - x^2yz^3$
f_3	*new multipliers*	z	$x^6y^4z^9 - x^6y^2z^{11} + x^2y^2z^7 - x^2z^9$
f_7	xf_6	y, z	$x^7y^5z^3 - x^7y^3z^5 + x^3y^3z - x^3yz^3$
f_8	$-xf_7 + zf_1$	$x, \quad z$	$x^8y^3z^5 - x^4y^3z^4 + x^4yz^3 - yz^2$
	$yf_8 - zf_2$		0
	$yf_3 - z^6f_6$		0
f_9	xf_3	z	$x^7y^4z^9 - x^7y^2z^{11} + x^3y^2z^7 - x^3z^9$
	$yf_9 - z^6f_7$		0
f_{10}	$-xf_9 + z^5f_2$	$x, \quad z$	$x^8y^2z^{11} - x^4y^4z^8 + x^4z^9 - y^2z^6$
	$yf_{10} - z^6f_8 + z^5f_4$		0

Figure 5.1 shows the cone decomposition of the monoid ideal generated by the leading terms during the run of the algorithm. We start with the decomposition of the leading term ideal of the given three polynomials. Each further picture describes the situation after a new element has been added to the basis. The pictures should be interpreted in the same way as in Example 4.4.8 using $x_1 = x$, $x_2 = y$, $x_3 = z$.

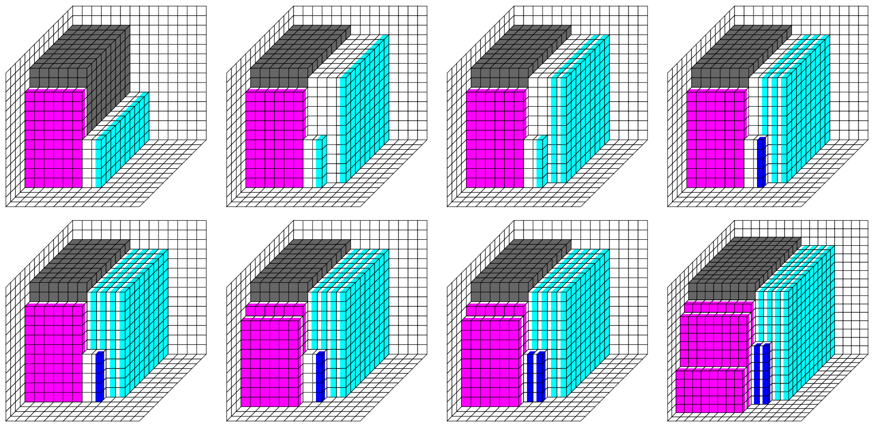


Figure 5.1: Cone decomposition corresponding to the Janet Basis Computation of Example 5.2.1

According to Algorithm 2, each time a new polynomial is added to the basis, we would have to re-reduce all prolongations that we have reduced so far. Such a re-reduction is not apparent in our table. In fact, for a Janet division Apel [Ape98a] as well as Gerdt and Blinkov [GB98a] show that each prolongation must only be considered once during the whole run of the Janet basis algorithm.

The "nice" behaviour of the Janet division in this example can also be seen in Figure 5.1. The colours do not change too drastically from one picture to the next. In fact, only the two-dimensional (cyan) cone of f_3 in pictures 1 to 3 becomes one-dimensional (blue) in picture 4 when the element f_6 has been added. The cones of other elements do not change after a basis growth.

The polynomial set $\{f_1, \ldots, f_{10}\}$ forms a Janet basis where already the subset $\{f_1, f_2, f_3, f_4, f_8, f_{10}\}$ forms a reduced Gröbner basis of $\mathrm{Id}_{\mathbb{K}[X]}(f_1, f_2, f_3)$.

Note that Algorithm 2 never computes the S-polynomial of f_2 and f_4. Only $\mathrm{spol}(f_1, f_2)$ and $\mathrm{spol}(f_1, f_4)$ are actually reduced. The first one appears in the top-reduction of the y-prolongation of f_2 and second one in the top-reduction of the x-prolongation of f_7.

We have $[f_1, f_4] = x^8y^5z^3 \lhd [f_2, f_4] = [f_1, f_2]$. By application of the Gebauer-Möller version of Buchberger's criteria, cf. [GM88, p. 279], the S-polynomials of f_2 and f_4 is discarded as being redundant in a Gröbner basis computation with application of criteria. Algorithm 2 automatically omits the S-pair $\{f_2, f_4\}$ due to the decomposition of the leading term ideal into disjoint cones and the way the algorithm proceeds.

The price one has to pay for such an automatic application of criteria is that one has generally more polynomials in a final Janet basis than in a reduced Gröbner basis. These additional polynomials can even give rise to S-polynomials that do not appear in a Gröbner basis computation. Consider, for example, $yf_9 - z^6f_7$. If $yf_3 - z^6f_6 = 0$, $f_9 = xf_3$, and $f_7 = xf_6$ is known, one can immediately conclude that $yf_9 - z^6f_7 = x(yf_3 - z^6f_6) = 0$ by lifting *the reduction to zero. In fact, Gerdt and Blinkov [GB98a] introduced a criterion which detects such situations and removes the superfluous reductions. In Chapter 7, we revisit the application of criteria for the involutive basis algorithm and give other criteria to avoid additional redundant reductions.*

5.3 Generalised Janet Basis Algorithm

The involutive basis algorithm is a generalisation of the Janet basis algorithm. Several variants of the involutive basis algorithm are known in a polynomial context, see, for example, [ZB93], [GB98b], [Ape98a], [CG00], [CHS01].

Our version is basically that of [Ape98a] and is given as Algorithm 4. It describes the basic form of the involutive basis algorithm, but in contrast to Algorithm 2, it incorporates the freedom of using also other partial divisions than just the Janet division.

Instead of requiring as input a family of partial divisions, we take here an algorithmic point of view and require a computable function PartialDivision which determines in each iteration of the while loop of Algorithm 4 the multiplicative and non-multiplicative variables for each leading term of the current basis. In order to make InvBasis correct and terminating, one must impose certain restrictions on the partial divisions that can be used as output of PartialDivision.

Apel [Ape98a] showed correctness and termination of Algorithm 4 if for each set U of terms, PartialDivision returns a U-admissible partial division which refines the Thomas division on U. Although Gerdt and Blinkov [GB98a] use a different involutive basis algorithm, their results on termination and correctness would carry over to Algorithm 4 if the collection of all outputs of PARTIALDIVISION forms a noetherian, constructive involutive division.

Since our concept of a suitable partial division is neither covered by Apel's theory nor by the one of Gerdt and Blinkov, we do not rely on their results, but prove termination and correctness of INVBASIS in Theorem 5.3.8 if for each set of terms U the algorithm PARTIALDIVISION returns a suitable partial division on U.

Note that by generalising the Janet division, we allow divisions which are not disjoint. This is reflected in Algorithm 4 by the introduction of the set Q of basis elements which are $\mathcal{M}$-top-reducible modulo $G \setminus Q$. In fact, we could weaken condition (4.19) of Definition 4.6.1 by just requiring

$$\mathcal{M}_u \cap \mathcal{M}_v \neq \emptyset \implies u \in \mathcal{M}_v \vee v \in \mathcal{M}_u.$$

This then corresponds to the fact that Algorithm 4 never accesses multiplicative variables of elements from Q. However, such a relaxation of condition (4.19) does not add anything new to the theory. In contrast, we would have to adjust the notion of $\mathcal{M}$-completeness in Definition 4.6.2 to join only the cones of those elements which are minimal in the sense that they do not lie in the cone of some other element.

Algorithm 4 INVBASIS

Call: $G = \text{INVBASIS}(F)$

Input: $F \subset \mathbb{K}[X] \setminus \{0\}$ is a finite set of polynomials with pairwise distinct leading terms.

Output: G is an involutive basis of the ideal of F in $\mathbb{K}[X]$.

$G := F$
$\mathcal{M} := (t[Y_t])_{t \in \operatorname{lt}(G)} := \text{PARTIALDIVISION}(\operatorname{lt} G)$
$Q := \{ g \in G \mid \exists f \in G \setminus \{g\} : \operatorname{lt} g \in \mathcal{M}_{\operatorname{lt}(f)} \}$
$C := \{ (x, g) \mid g \in G \setminus Q \wedge x \in X \setminus Y_{\operatorname{lt} g} \} \cup \{ (1, q) \mid q \in Q \}$
<u>while</u> $C \neq \emptyset$ <u>do</u>
  Choose (t, g) from C such that $\operatorname{lt}(tg)$ is $\prec$-minimal; $C := C \setminus \{(t, g)\}$
  $h := \mathrm{NF}_{\mathcal{M}}(tg, G \setminus Q)$
  <u>if</u> $h \neq 0$ <u>then</u>
    $G := G \cup \{h\}$
    $\mathcal{M} := (t[Y_t])_{t \in \operatorname{lt}(G)} := \text{PARTIALDIVISION}(\operatorname{lt} G)$
    $Q := \{ g \in G \mid \exists f \in G \setminus \{g\} : \operatorname{lt} g \in \mathcal{M}_{\operatorname{lt}(f)} \}$
    $C := \{ (x, g) \mid g \in G \setminus Q \wedge x \in X \setminus Y_{\operatorname{lt} g} \} \cup \{ (1, q) \mid q \in Q \}$
<u>return</u> G

Before we can state our termination and correctness proof of Algorithm 4, we need a few more definitions.

Definition 5.3.1. [Ape98b, p. 54] Let $G = \{g_1, \ldots, g_r\} \subseteq \mathbb{K}[X] \setminus \{0\}$ and $\prec$ be an irreflexive term order on T. For $t \in T$, let $\mathfrak{F}^G_{\preceq t}$ be the additive subgroup of $I := \operatorname{Id}(G)$ consisting of all polynomials $h \in I$ which can be represented in the form $h = \sum_{\varrho=1}^{r} h_\varrho g_\varrho$ where $h_\varrho \in \mathbb{K}[X]$ and either $h_\varrho = 0$ or $\operatorname{lt}(h_\varrho g_\varrho) \preceq t$ for all $\varrho \in \mathbb{N}_r$. The family $\left(\mathfrak{F}^G_{\preceq t}\right)_{t \in T}$ is a $\mathbb{K}[X]$-module filtration of I, the so-called **Gröbner filtration** of I. By $\mathfrak{F}^G_{\prec t}$ we denote the union $\bigcup_{s \prec t} \mathfrak{F}^G_{\preceq s}$.

Definition 5.3.2. [AH02] Let $G \subseteq \mathbb{K}[X] \setminus \{0\}$ and $f, g \in \mathbb{K}[X] \setminus \{0\}$. We define by $\mathcal{S}^G(f,g)$ the predicate $\operatorname{spol}(f,g) \in \mathfrak{F}^G_{\prec[f,g]}$. In places where G is clear from the context, we omit the upper index and simply write $\mathcal{S}(f,g)$.

The notation $\mathcal{S}(f,g)$ is just a short hand for saying that the S-polynomial has a "good" representation.

For the proof of correctness of Algorithm 4 it is helpful to have the following equivalence theorem at hand. It follows the lines of Theorem 5.1 in [Ape98a] but replaces admissibility by the weaker requirement of cycle-freeness.

Theorem 5.3.3. *Let $G \subseteq \mathbb{K}[X]$ be a finite set of non-zero monic polynomials. Let $\mathcal{M} = (t[Y_t])_{t\in T}$ be a partial division which is disjoint and cycle-free on $\operatorname{lt} G$. Then the following statements are equivalent.*

(i) G is an $\mathcal{M}$-involutive basis.

(ii) $\operatorname{lt} G$ is $\mathcal{M}$-complete and G is a Gröbner basis.

(iii) $\operatorname{lt} G$ is $\mathcal{M}$-complete and $\mathcal{S}(g,g')$ for all $g, g' \in G$.

(iv) For all $g \in G$ and $x \in X \setminus Y_{\operatorname{lt} g}$ there exist $f \in G$ and $t \in [Y_{\operatorname{lt} f}]$ such that $xg - tf \in \mathfrak{F}^G_{\prec \operatorname{lt}(xg)}$.

Proof. The implications (i) $\iff$ (ii) $\iff$ (iii) $\implies$ (iv) are trivial. We prove the implication (iv) $\implies$ (iii).

Assume (iv). Then $\operatorname{lt} G$ is not $\mathcal{M}$-extensible and hence $\mathcal{M}$-complete by Proposition 4.6.7.

Let $g, g' \in G$ with $\mathcal{S}(g,g')$. Then there exist $t, t' \in T$, such that $\operatorname{lt}(tg) = \operatorname{lt}(t'g') = [g,g']$. We construct sequences $(g_i)_{0\le i\le k}$, $(y_i)_{0\le i<k}$, and $(t_i)_{0\le i\le k}$ in the following way. Let $g_0 := g$ and $t_0 := t$. If $t \in \left[Y_{\operatorname{lt}(g_0)}\right]$ then we are finished. Otherwise, we find $y_0 \in X \setminus Y_{\operatorname{lt}(g_0)}$ which divides t_0. By (iv) there exist $g_1 \in G$ and $t_1 \in \left[Y_{\operatorname{lt}(g_1)}\right]$ with $y_0 g_0 - t_1 g_1 \in \mathfrak{F}^G_{\prec \operatorname{lt}(y_0 g_0)}$. Clearly, $\operatorname{lt}(y_0 g_0) = \operatorname{lt}(t_1 g_1)$. Now, if $[g,g'] = \operatorname{lt}\left(\frac{t_0}{y_0} \cdot t_1 g_1\right) \notin \mathcal{M}_{\operatorname{lt}(g_1)}$, we find $y_1 \in X \setminus Y_{\operatorname{lt}(g_1)}$ which divides $\frac{t_0 t_1}{y_0}$. Again by (iv) there exist $g_2 \in G$ and $t_2 \in \left[Y_{\operatorname{lt}(g_2)}\right]$ with $y_1 g_1 - t_2 g_2 \in \mathfrak{F}^G_{\prec \operatorname{lt}(y_1 g_1)}$. If $[g,g'] = \operatorname{lt}\left(\frac{t_0 t_1}{y_0 y_1} \cdot t_2 g_2\right) \notin \mathcal{M}_{\operatorname{lt}(g_2)}$ we can continue this process. The process must eventually stop, at index k, say, since $\mathcal{M}$ is cycle-free on $\operatorname{lt} G$.

Similarly, we construct sequences $(g'_i)_{0\le i\le k'}$ $(y'_i)_{0\le i<k'}$, and $(t'_i)_{0\le i\le k'}$ by starting with (t', g') instead of (t, g).

Clearly, by construction it holds $y_i g_i - t_{i+1} g_{i+1} \in \mathfrak{F}^G_{\prec \operatorname{lt}(y_i g_i)}$ for each i with $0 \le i < k$. Thus

$$\sigma := \sum_{i=0}^{k-1} \frac{[g,g']}{\operatorname{lt}(y_i g_i)} \cdot (y_i g_i - t_{i+1} g_{i+1}) \in \mathfrak{F}^G_{\prec[g,g']} \tag{5.4}$$

and similarly,

$$\sigma' := \sum_{i=0}^{k'-1} \frac{[g,g']}{\operatorname{lt}(y'_i g'_i)} \cdot (y'_i g'_i - t'_{i+1} g'_{i+1}) \in \mathfrak{F}^G_{\prec[g,g']}. \tag{5.5}$$

If we can show

$$\frac{[g,g']}{\mathrm{lt}(t_k g_k)} t_k g_k - \frac{[g,g']}{\mathrm{lt}(t'_{k'} g'_{k'})} t'_{k'} g'_{k'} \in \mathfrak{F}^G_{\prec [g,g']} \tag{5.6}$$

then $S(g,g')$ follows from

$$tg - t'g' = \sigma + \left(\frac{[g,g']}{\mathrm{lt}(t_k g_k)} t_k g_k - \frac{[g,g']}{\mathrm{lt}(t'_{k'} g'_{k'})} t'_{k'} g'_{k'} \right) - \sigma'. \tag{5.7}$$

By construction we have

$$[g,g'] \in \mathcal{M}_{\mathrm{lt}\, g_k} \cap \mathcal{M}_{\mathrm{lt}\, g'_k} \neq \emptyset. \tag{5.8}$$

So since $\mathcal{M}$ is disjoint on $\mathrm{lt}(G)$ we have $g_k = g_{k'}$ and (5.6) follows immediately. □

Lemma 5.3.4. *Let $U \subset T$ be a finite set of terms and let $\mathcal{M}$ be a partial division which is cycle-free on U. Furthermore assume that for two index sets $I, J \subseteq \mathbb{N}$ two $\mathcal{M}$-sequences $(u_i)_{i\in I}$ and $(v_j)_{j\in J}$ are given. Let $t_1, t_2 \in U$ with $t_1 \neq t_2$ and let there exist $i_1, i_2 \in I$, $j_1, j_2 \in J$ with $u_{i_1} = v_{j_1} = t_1$ and $u_{i_2} = v_{j_2} = t_2$. Then $i_1 < i_2$ implies $j_1 < j_2$.*

Proof. If $i_1 < i_2$ and $j_1 > j_2$ then we can construct an infinite $\mathcal{M}$-sequence on U, namely

$$u_{i_1}, u_{i_1+1}, \ldots, u_{i_2} = v_{j_2}, v_{j_2+1}, \ldots, v_{j_1} = u_{i_1}, \ldots$$

which clearly contradicts the assumption that $\mathcal{M}$ is cycle-free on U. □

Lemma 5.3.4 allows us to associate a partial order to a cycle-free partial division.

Definition 5.3.5. Let $U \subset T$ be a finite set of terms and let $\mathcal{M}$ be a partial division which is cycle-free on U. The relation $>_{\mathcal{M},U}$ is defined on U by $u >_{\mathcal{M},U} v$ if and only if there exists an $\mathcal{M}$-sequence $(u_i)_{0\leq i\leq k}$ on U such that $k > 0$, $u_0 = u$ and $u_k = v$. It is called the **$\mathcal{M}$-sequence semi-order on U**.

Remark 5.3.6. The relation $>_{\mathcal{M},U}$ is indeed a semi-order on U. It is obviously irreflexive. Transitivity follows by joining $\mathcal{M}$-sequences on U. Asymmetry follows from Lemma 5.3.4.

The following Lemma and Theorem generalise the corresponding statements in [Ape98a] by weakening the requirement on the partial division from admissible to cycle-free.

Lemma 5.3.7. *Let $G \subseteq \mathbb{K}[X]$ be a finite set of non-zero polynomials. Let $\mathcal{M} = (t[Y_t])_{t\in T}$ be a partial division which is disjoint and cycle-free on $\mathrm{lt}\, G$. Furthermore let $f \in \mathbb{K}[X] \setminus \{0\}$. Assume that for every $g \in G$ and every $x \in X \setminus Y_{\mathrm{lt}\, g}$ with $\mathrm{lt}(xg) \preceq \mathrm{lt}\, f$ there exists $g' \in G$ such that $\mathrm{lt}(xg) \in \mathcal{M}_{\mathrm{lt}(g')}$. Then $\mathrm{NF}_{\mathcal{M}}(f,G) = \mathrm{NF}(f,G)$.*

Proof. Let $h = \mathrm{NF}_{\mathcal{M}}(f, G)$, $t \in \operatorname{supp} h \cap \langle \mathrm{lt}\, G \rangle$, and $\Gamma := \{g \in G \,|\, \mathrm{lt}\, g \trianglelefteq t\}$. Among the elements of Γ choose g such that $\mathrm{lt}\, g$ is minimal with respect to the $\mathcal{M}$-sequence semi-order $>_{\mathcal{M},\mathrm{lt}(G)}$ on $\mathrm{lt}\, G$. Since h is $\mathcal{M}$-irreducible modulo G, it follows $t \notin \mathcal{M}_{\mathrm{lt}(g)}$. So let $x \in X \setminus Y_{\mathrm{lt}\, g}$ be a non-multiplicative variable of g such that $\mathrm{lt}(xg) \trianglelefteq t$. Since $\mathrm{lt}(xg) \preceq \mathrm{lt}\, f$, there exists $g' \in G$ such that $\mathrm{lt}(xg) \in \mathcal{M}_{\mathrm{lt}(g')}$. The relations $\mathrm{lt}\, g' \trianglelefteq t$ and $\mathrm{lt}\, g >_{\mathcal{M},\mathrm{lt}(G)} \mathrm{lt}\, g'$ follow but contradict the minimal choice of g. Therefore, h is Gröbner-irreducible and the claim of the lemma follows. □

Theorem 5.3.8. *Algorithm 4 terminates and fulfils its specification if for each set G which occurs during the run of Algorithm 4,* PartialDivision($\mathrm{lt}\, G$) *is a suitable partial division on* $\mathrm{lt}\, G$.

Proof. Let us denote by G^i, Q^i, and $\mathcal{M}^i$ the value of G, Q, and $\mathcal{M}$, respectively, before the i-th iteration of the <u>while</u> loop (lines 5–12).

Termination. By Dickson's Lemma, there exists $k \in \mathbb{N}$ such that

$$\left\langle \mathrm{lt}(G^i) \right\rangle = \left\langle \mathrm{lt}(G^k) \right\rangle \text{ for all } i \geq k. \tag{5.9}$$

Let (t, g) be the prolongation which is chosen in the k-th iteration in line 6 and $h = \mathrm{NF}_{\mathcal{M}}(tg, G^k \setminus Q^k)$. If there exists $g' \in G^k \setminus Q^k$ with $\mathrm{lt}(tg) \in \mathcal{M}^k_{\mathrm{lt}\,(g')}$ then $h = 0$ by (5.9) and Lemma 5.3.7. Note that Algorithm 4 applies a normal selection strategy. Otherwise, since $\mathcal{M}^k$ refines the Thomas division on $\mathrm{lt}\, G^k$, we have $\mathrm{lt}\, h = \mathrm{lt}(tg) \trianglelefteq \mathrm{lcm}(\mathrm{lt}\, G^k)$ and, therefore, $\mathrm{lcm}(\mathrm{lt}\, G^{k+1}) = \mathrm{lcm}(\mathrm{lt}\, G^k)$. Induction proves $\mathrm{lcm}(\mathrm{lt}\, G^i) = \mathrm{lcm}(\mathrm{lt}\, G^k)$ for all $i \geq k$. By Corollary 4.6.9 there exists $l \geq k$ such that $\mathrm{lt}(G^l)$ is $\mathcal{M}^l$-complete. Since the number of non-multiplicative prolongations of elements of G^l is finite, there are only finitely many iterations left.

Correctness. Let G be the output of InvBasis and let

$$\mathcal{M} := (t[Y_t])_{t \in \mathrm{lt}(G)} := \text{PartialDivision}(\mathrm{lt}\, G), \tag{5.10}$$

$$Q := \left\{ g \in G \,\middle|\, \exists f \in G \setminus \{g\} : \mathrm{lt}\, g \in \mathcal{M}_{\mathrm{lt}\,(f)} \right\}. \tag{5.11}$$

Trivially, $\mathcal{M}$ is also cycle-free on the subset $\mathrm{lt}(G \setminus Q)$. After adding the last element to the basis G, all non-multiplicative prolongations of elements of $G \setminus Q$ reduce to zero. From Proposition 4.6.7 we conclude that $\mathrm{lt}(G \setminus Q)$ is $\mathcal{M}$-complete, and hence also $\mathrm{lt}(G)$ is $\mathcal{M}$-complete.

After the last assignment of a value to C in line 4 or line 12 all reductions in line 7 lead to zero. By Theorem 5.3.3, (iv) we conclude that $G \setminus Q$ is an $\mathcal{M}$-involutive basis. Since all elements of Q can be represented as a $\mathbb{K}[X]$-linear combination of elements from $G \setminus Q$, the set $G \setminus Q$ is an involutive basis of $\mathrm{Id}\,(G)$. From $F \subseteq G$ it follows $\mathrm{Id}\,(F) \subseteq \mathrm{Id}\,(G)$. The opposite relation $\mathrm{Id}\,(F) \supseteq \mathrm{Id}\,(G)$ holds, because each element of G results from a $\mathbb{K}[X]$-linear combination of elements from F.

In conclusion, G is an involutive basis of $\mathrm{Id}\,(F)$. □

Chapter 6

Sliced Division

6.1 Motivation

We have seen in Example 5.2.1 that the Janet basis contains more elements than the corresponding reduced Gröbner basis. Indeed, by Theorem 5.3.3, it is obvious that this is in general the case.

A natural goal would be to apply a Janet-like algorithm and arrive at an involutive basis with fewer elements than in a Janet basis.

As we have seen in Remark 4.4.2 there are $n!$ different ways to determine a Janet division for a set of terms in n variables. In Example 5.2.1, we are in a better situation if the Janet division is computed with respect to the variable order $y > z > x$ instead of $x > y > z$. However, independent of the variable order, using a Janet division generally leads to many more elements in the final basis than in the reduced Gröbner basis. This can be seen from the following proposition.

Proposition 6.1.1. *Let $X = \{x_1, \dots, x_n\}$ be a set of variables, $d > n > 2$,*

$$m_0 := \prod_{i=1}^{n} x_i, \qquad m_j := \frac{m_0 x_j^d}{x_{j+1}} \quad \text{for } j = 1, \dots, n-1, \qquad m_n := \frac{m_0 x_n^d}{x_1}.$$

If $U := \{m_0, \dots, m_n\}$, $V \subset T$ such that V is $\mathcal{J}$-complete and $\langle V \rangle_{\mathcal{J}} = \langle U \rangle$ where $\mathcal{J}$ is a Janet division on V, then V contains an element of degree $2d + n - 2$ independent of the construction of V.

Proof. The set U is a set of minimal generators of $\langle U \rangle$. So a Janet-complete set V must contain all elements of U. According to Definition 4.4.1, the Janet division depends on an order of the variables. By symmetry, we can assume that the Janet division $\mathcal{J}$ on V is taken with respect to an order $>$ on X in which x_1 is the biggest. Let us denote by m the maximum of V with respect to the lexicographic extension of $>$ to T. According to Definition 4.4.1, m gets all variables as multiplicative. Obviously, $\deg_{x_1} m \geq d + 1 = \deg_{x_1} m_1$. Multiples of m_2 of the form $x_1^k m_2$ for $k \in \mathbb{N}$ are not divisible by any $u \in U \setminus \{m_2\}$. We conclude $x_1^k m_2 \in V$ for any $k < d$, since none of these elements has x_1 as Janet-multiplicative. Since $\deg(x_1^{d-1} m_2) = 2d + n - 2$, the proposition is proved. $\square$

Remark 6.1.2. The set U from Proposition 6.1.1 already forms an involutive basis which can be seen by giving m_0 the set X as multiplicative variables, m_i the set $X \setminus \{x_{i+1}\}$, $i = 1, \ldots, n-1$, and m_n the set $X \setminus \{x_1\}$, i. e., there is an involutive basis with elements of degree at most $d+n-1$. For $n = 3$ and $d = 6$ the situation is depicted for the Janet and Sliced Division in Example 4.4.8.

With respect to the number of elements, the Janet division is not optimal. By just focusing on the Janet division, Algorithm 4 can miss to realise that an intermediate basis is already an involutive basis, but only with respect to another partial division. So, one optimisation of Algorithm 4 would be to compute by PartialDivision a "good" partial division $\mathcal{M}$ in the sense that the leading terms of an intermediate basis form (at least "almost") an $\mathcal{M}$-complete set.

Another, slightly different, optimisation lies in a computation of a partial division such that the total number of prolongations that must be formed as the set C of critical elements in Algorithm 4 is minimal. Such a minimal cardinality is advantageous, because the smaller the cardinality of C the less iterations must be executed in Algorithm 4.

In the following section, we present an algorithm, the Sliced Division algorithm, which is "good" in the sense above.

6.2 Sliced Division Algorithm

We give here an algorithm which can be used (with minor adaptations of the output) as the sub-algorithm PartialDivision in Algorithm 4. In Theorem 5.3.8 we require PartialDivision to return a suitable partial division. Algorithm 5 returns such a division, but, in fact, even returns an admissible partial division. Our algorithm not only computes an admissible partial division for a given set U of terms but also a particular order on U so that the resulting division is admissible on U with respect to that order. This approach is different from Apel's submaximal refinement of a division with respect to some given order [Ape98a], because we do not require the order $\sqsubset$ in advance, but rather compute it dynamically during the run of the Sliced Division algorithm.

The basic objective for the Sliced Division algorithm was to find an admissible partial division on a set U of terms such that the cones attached to elements of U cover a relatively "big" part of the monoid ideal generated by U. This is achieved by assigning all variables as multiplicative to a term of minimal total degree and then applying such an assignment recursively (with fewer variables) to the remaining terms.

Definition 6.2.1. [Hem03] Let $U \subseteq T$ be a finite set of terms and

$$\text{LIST}(u_1[Y_{u_1}], \ldots, u_r[Y_{u_r}]) = \text{SlicedDivision}(X, U).$$

$(u[Y_u])_{u \in U}$ is called a **Sliced Division on** U.

The division mentioned in Remark 6.1.2 is an example of a Sliced Division. Another Sliced Division has already been given in Example 4.4.8.

Remark 6.2.2. Let us informally describe how the Sliced Division algorithm works. One should associate the set U of terms with its monoid ideal represented in some $\mathbb{N}^n$. As the first step, some element m of minimal total degree is taken from U and gets the full-dimensional cone assigned to it (a shifted copy of

Algorithm 5 SLICEDDIVISION

Call: $M := \textsc{SlicedDivision}(Y, U)$

Input: $Y \subseteq X$ is finite set of indeterminates.
$U = \{u_1, \ldots, u_r\} \subseteq w[Y]$ for some $w \in [X \setminus Y]$ where $r = \operatorname{card} U$.

Output: $M = \textsc{list}(v_1[Y_{v_1}], \ldots, v_r[Y_{v_r}])$ is a list of Y-cones such that
$U = \{v_1, \ldots, v_r\}$ and
$\forall i, j \left(1 \le i < j \le r \implies (v_i[Y] \cap v_j[Y_{v_j}] = \emptyset \vee v_i[Y_{v_i}] \subset v_j[Y_{v_j}])\right)$

1: $L := \textsc{list}()$ *# empty list*
2: **if** $U = \emptyset$ **then return** L
3: **if** $Y = \emptyset$ **then return** $\textsc{list}(w[Y])$ *# Note that here $U = w[Y] = \{w\}$ if $Y = \emptyset$.*
4: Choose $m \in U$ with $\deg m = \min \{\deg u \mid u \in U\}$
5: $C := (U \cap m[Y]) \setminus \{m\}$; $L_C := \textsc{SlicedDivision}(Y, C)$
6: $V := U \setminus m[Y]$ *# same as $V := U \setminus (C \cup \{m\})$*
7: **while** $V \neq \emptyset$ **do**
8: Choose $x \in Y$ such that $\deg_x(\gcd V) < \deg_x m$
9: $d := \deg_x(\gcd V)$
10: $S := \{v \in V \mid \deg_x v = d\}$
11: $V := V \setminus S$
12: $L_S := \textsc{SlicedDivision}(Y \setminus \{x\}, S)$
13: $L := \textsc{concat}(L_S, L)$
14: **return** $\textsc{concat}(L_C, \textsc{list}(m[Y]), L)$

$\mathbb{N}^n$). All elements which are divisible by m are treated recursively. Any of the other elements will get a cone which is at most of dimension $n-1$. By taking hyperplanes perpendicular to the direction of some variable x (chosen in each iteration step), slices are cut from the "outermost" part of the ideal until the cone of m is reached. The elements inside such a slice correspond to elements of V (cf. Algorithm 5) having minimal degree in x. Such slices are of dimension $n-1$ and thus can be treated recursively.

The condition $U \subseteq w[Y]$, $w \in [X \setminus Y]$ in the specification of Algorithm 5 just states that elements of U agree in the "non-multiplicative part".

The function CONCAT takes two lists $\textsc{list}(a_1, \ldots, a_r)$, $\textsc{list}(b_1, \ldots, b_s)$ and returns the concatenated list $\textsc{list}(a_1, \ldots, a_r, b_1, \ldots, b_s)$.

Example 6.2.3. *Let us compute a Sliced Division on the terms from Example 4.4.7. We start Algorithm 5 on $X = \{x, y\}$ and $U = \{x^{10}, x^9, x^3y^3, xy^9\}$. Obviously, x^3y^3 gets x and y as multiplicative, since it is the term of smallest total degree. Then we choose x in line 8 of Algorithm 5 and find that only xy^9 has a smaller degree in x than x^3y^3. In a recursive call, it gets y assigned as its multiplicative variable. Next, we choose y and have to execute* $\textsc{SlicedDivision}(\{x\}, \{x^{10}, x^9\})$. *In this recursive call, the element x^9 will be selected first and gets x as multiplicative whereas x^{10} falls into the set C. Another recursive call assigns x as multiplicative for x^{10}.*

In summary the multiplicative variables of the Sliced Division are given by the following table.

U	SD
x^{10}	x
x^9	x
x^3y^3	x, y
xy^9	y

A diagram similar to Figure 4.1 is shown as Figure 6.1.

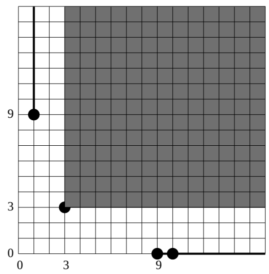


Figure 6.1: Sliced Division of Example 6.2.3

We proceed by showing termination and correctness of the Sliced Division algorithm.

Theorem 6.2.4. *[Hem03] Algorithm 5 terminates and meets its specification.*

Proof. One easily verifies that the parameters of recursive calls of the algorithm fulfil the input specification.

Termination. Because of lines 3, 6, and 7, the choice of x in line 8 is possible. There can be only finitely many while iterations (lines 7–13), since S contains at least one element and thus the cardinality of V decreases. In each recursive call the sum of the cardinalities of the input parameters is strictly smaller than $\operatorname{card} Y + \operatorname{card} U$. Thus the recursion depth is bounded, and hence SLICEDDIVISION terminates.

Correctness. Let Y and U be as specified in the input of Algorithm 5. The proof will be an induction on the number of elements in the input sets. Obviously, the algorithm is correct if it is started with empty input parameters. Let us assume that the algorithm is correct if the sum of the cardinalities of the input parameters is smaller than $\operatorname{card} Y + \operatorname{card} U$. Let $\text{LIST}(v_1[Y_{v_1}], \ldots, v_{r'}[Y_{v_{r'}}])$ be the output of the algorithm for Y and U. Consider the particular run where this output was produced. Let m, C, and V be as specified before the while loop, and let s be the number of actual iterations.

In the following let σ range over $1, \ldots, s$. The values of x, and S after the σ-th iteration will be denoted by y_σ and S_σ.

First, we must show that $r' = r$. By induction hypothesis, we have $\operatorname{card} C = \operatorname{card} L_C$ and $\operatorname{card} S_\sigma = \operatorname{card} L_{S_\sigma}$ for all σ. From lines 1, 13, and 14 and the fact that U is equal to the disjoint union $C \cup \{m\} \cup S_s \cup \cdots \cup S_1$, we can conclude $r = r'$.

Lines 5, 12, and 14 together with the induction hypothesis then imply $U = \{v_1, \ldots, v_r\}$.

Take i, j with $1 \le i < j \le r$. It remains to show that

$$v_i[Y] \cap v_j[Y_{v_j}] = \emptyset \text{ or} \tag{6.1}$$
$$v_i[Y_{v_i}] \subset v_j[Y_{v_j}] \tag{6.2}$$

holds. There are several cases to consider.

(i) $v_i, v_j \in C$: (6.1) or (6.2) follows by induction hypothesis.

(ii) $v_i \in C$, $v_j = m$: We have $Y_{v_j} = Y$ and (6.2) by construction of C.

(iii) $v_i \in C \cup \{m\}$, $v_j \in S_\sigma$ for some $\sigma \in \mathbb{N}_s$: By construction of m, C, and S_σ, we have $m[Y] \cap v_j[Y_{v_j}] = \emptyset$, because $\deg_{y_\sigma} v_j < \deg_{y_\sigma} m$ and $y_\sigma \in X \setminus Y_{v_j}$. Since $c[Y_c] \subset m[Y]$ for all $c \in C$, relation (6.1) holds.

(iv) $v_i \in S_\sigma$, $v_j \in S_{\sigma'}$ for some $\sigma, \sigma' \in \mathbb{N}_s$: If $\sigma = \sigma'$ then (6.1) or (6.2) follows from the induction hypothesis and Remark 4.3.2. Otherwise, $i < j$ implies $\sigma > \sigma'$ by line 14 and construction of L in line 13. The inequality $\deg_{y_{\sigma'}}(v_i t) > \deg_{y_{\sigma'}} v_j$ holds for $t = 1$ by construction of $S_{\sigma'}$ and remains true for any $t \in [Y]$. Since $y_{\sigma'} \notin Y_{v_j}$, we can conclude (6.1).

By line 14 there cannot be other cases. Thus, the correctness of Algorithm 5 is proved. □

Corollary 6.2.5. *[Hem03] Let $U \subseteq T$ be a finite set of terms, and*

$$\textsc{list}(u_1[Y_{u_1}], \ldots, u_r[Y_{u_r}]) = \textsc{SlicedDivision}(X, U).$$

Define the order $\sqsubset$ on U by $u_1 \sqsubset u_2 \sqsubset \cdots \sqsubset u_r$. Then the Sliced Division $\mathcal{M} = (u[Y_u])_{u \in U}$ is admissible on $(U, \sqsubset)$.

We want to emphasise the fact that instead of giving an order $\sqsubset$ in advance and then computing an admissible partial division with respect to that order, our algorithm only starts with a set of terms and returns a division and an order for which this division is admissible.

One easily verifies that SlicedDivision never calls itself with an empty set of variables if it is started with a non-empty set Y. Line 3 is only important if the algorithm is initially called with an empty set of variables or for a generalisation of the algorithm. In fact, algorithm SlicedDivision also meets its specification if an arbitrary element $m \in U$ in line 4 is chosen. For such a generalisation, however, it can happen that the algorithm calls itself with an empty set Y. One easily verifies that the proof of Theorem 6.2.4 does not depend on the particular choice of m. For an extensive treatment of the generalised algorithm see Section 6.4.

6.3 Sliced Division Refines Thomas Division

As Apel showed in [Ape98a], termination of the involutive basis algorithm (Algorithm 4) is guaranteed if the division used in its while loop refines the Thomas division, see also Lemma 4.5.1. In this section we show that a Sliced Division refines the Thomas division on the same set of terms and thus leads to a terminating involutive basis algorithm.

The following property of the Thomas division is easy to prove. In fact, it states that the Thomas division satisfies condition (4.4) of Definition 4.2.1.

Lemma 6.3.1. *Let $U, V \subseteq T$ be two sets with $U \supseteq V$ and $(t[Y_t])_{t\in T}$, $(t[Z_t])_{t\in T}$ be the Thomas divisions on U and V, respectively. Then $Y_t \subseteq Z_t$ for all $t \in T$.*

Proof. For all $t \in T \setminus U$ the equality $Y_t = Z_t = X$ holds. Trivially, $Y_t \subseteq Z_t = X$ is true for all $t \in U \setminus V$. For all $t \in V$ and all $x \in X \setminus Z_t$ there exists $t' \in V$ with $\deg_x t' > \deg_x t$ and, therefore $x \in X \setminus Y_t$. Hence, $Y_t \subseteq Z_t$. □

Theorem 6.3.2. *[Hem03] Let $Y \subseteq X$ be a set of variables, $U \subset T$ be a finite set of terms, and $\text{LIST}(u_1[Y_{u_1}], \ldots, u_r[Y_{u_r}]) = \text{SLICEDDIVISION}(Y, U)$. Then for the Thomas division $(t[Z_t])_{t\in T}$ on U the relation $Z_u \cap Y \subseteq Y_u$ holds for all $u \in U$.*

Proof. Since for $U = \emptyset$ there is nothing to prove, assume that $\operatorname{card} U = r > 0$ and that the theorem is correct if the number of terms is less than r. Consider the run of SLICEDDIVISION where $\text{LIST}(u_1[Y_{u_1}], \ldots, u_r[Y_{u_r}])$ was computed. Let s be the number of iterations of the <u>while</u> loop, and let C and m be as defined in the algorithm. After the σ-th iteration ($\sigma \in \mathbb{N}_s$) of the <u>while</u> loop the values of x and S are denoted by y_σ and S_σ, respectively. Then U is the disjoint union $C \cup \{m\} \cup S_s \cup \cdots \cup S_1$.

Trivially, $Z_m \cap Y \subseteq Y_m = Y$. From the induction hypothesis it follows that $Z_u \cap Y \subseteq Y_u$ for all $u \in C$. Now consider S_σ for some $\sigma \in \mathbb{N}_s$. Take any $u \in S_\sigma$. From the induction hypothesis we get $Z_u \cap (Y \setminus \{y_\sigma\}) \subseteq Y_u$ (see line 12 of Algorithm 5). By definition of the Thomas division, we have $y_\sigma \notin Z_u$ since $\deg_{y_\sigma} u < \deg_{y_\sigma} m$. Hence, $Z_u \cap Y \subseteq Y_u$ as claimed. □

The following corollary is Theorem 6.3.2 for the special case $X = Y$.

Corollary 6.3.3. *[Hem03] Let $U \subseteq T$ be a finite set of terms. Then a Sliced Division on U refines the Thomas division on U and, hence, is a suitable partial division on U.*

Proof. The statement follows by Definition 4.6.8 from Corollary 6.2.5, Proposition 4.6.15, and Theorem 6.3.2. □

6.4 Generalised Sliced Division

The algorithm SLICEDDIVISION was designed to construct an admissible partial division on a set U of terms such that the cones attached to elements of U cover a relatively "big" part of the monoid ideal generated by U. For theoretical considerations it is interesting to generalise the algorithm such that it can compute other known admissible divisions.

As we have already seen, the particular choice of m in line 4 of Algorithm 5 does not affect the correctness of Theorem 6.2.4. By a further generalised form of SLICEDDIVISION, one can compute not only the Janet division, but also Pommaret and Thomas division as we shall see in Section 6.4.2.

Let us first generalise the Sliced Division algorithm.

6.4.1 Admissibility

Let $U \subseteq T$ be a finite set of terms. $\text{SLICEDDIVISION}(X, U)$ always returns a partial division where at least one term of U has a full-dimensional cone. It cannot compute, for example, the Pommaret division for the terms $\{x_1x_2^2, x_1^2x_2\}$ since x_1 is not Pommaret-multiplicative for both terms. To produce admissible divisions on a set U where no element of U gets a cone corresponding to all possible variables, we introduce a distinct symbol $\infty \notin U$ and allow to choose in line 4 of SlicedDivision either an element of U or this symbol depending on whether or not the final division should contain a full-dimensional cone.

In order to simplify notation, for $m = \infty$, $Y \subseteq X$, and $U \subseteq T$, we require that the set $U \cap m[Y]$ is empty (no matter how $m[Y]$ is actually defined), and we declare $\deg_x u < \deg_x m$ for all $x \in X$, $u \in U$.

By using the same specification as that of SlicedDivision, the generalised version of Algorithm 5 is obtained by replacing lines 4, 8, and 14 by

4: $m := \textsc{ChooseFirst}(Y, U)$
8: $x := \textsc{ChooseNonMultiplier}(Y, V, m)$
14: <u>if</u> $m \in U$ <u>then</u> <u>return</u> $\textsc{concat}(L_C, \textsc{list}(m[Y]), L)$ <u>else</u> <u>return</u> L

with two sub-algorithms ChooseFirst and ChooseNonMultiplier. These algorithms should meet the following specifications.

S1: If $\emptyset \neq U \subseteq T$ is a finite set of terms, $Y \subseteq X$, and $m = \textsc{ChooseFirst}(Y, U)$ then $m \in U \cup \{\infty\}$.

S2: Let $\emptyset \neq V \subseteq T$ be a finite set of terms, $Y \subseteq X$, $m \in T \cup \{\infty\}$,

$$\forall v \in V\ \exists y \in Y : \deg_y v < \deg_y m$$

and $x = \textsc{ChooseNonMultiplier}(Y, V, m)$. Then

$$x \in Y \wedge \exists v \in V : \deg_x v < \deg_x m.$$

One easily verifies that the input specifications are always satisfied when these functions are called in the generalised Sliced Division algorithm. By lines 6 and 7 of Algorithm 5 it follows $\forall v \in V\ \exists y \in Y : \deg_y v < \deg_y m$.

Theorem 6.4.1. *[Hem03] The generalised version of the algorithm* SlicedDivision *terminates and meets its specification.*

Proof. The proof of termination and correctness of the generalised version is similar to the proof of Theorem 6.2.4. In fact, our proof of Theorem 6.2.4 has been formulated in such a way that it can even be taken literally. Note that the cases (i), (ii), and (iii) at the end of the proof cannot occur if $m = \infty$, since then we have $C = \emptyset$ and $v_i, v_j \in U$, but $\infty \notin U$. □

Remark 6.4.2. For given U and Y, one may think of ∞ as the term

$$m = \operatorname{lcm}(U) \prod_{y \in Y} y$$

as it fulfils all conditions that we posed on ∞. Since for $m = \infty$ the cone $m[Y]$ is never returned and $\deg_y v < \deg_y m$ will be true for all $y \in Y$ and $v \in V$, such an assignment would not cause any problem.

Definition 6.4.3. Let $U \subseteq T$ be a finite set of terms. A **generalised Sliced Division on** U is a partial division which is induced by the output of a generalised Sliced Division algorithm when started with input X and U.

Corollary 6.4.4. *Let $U \subseteq T$ be a finite set of terms, and let*

$$\text{LIST}(u_1[Y_{u_1}], \ldots, u_r[Y_{u_r}])$$

be the output of a generalised Sliced Division algorithm when started with input X and U. Define the order $\sqsubset$ on U by $u_1 \sqsubset u_2 \sqsubset \cdots \sqsubset u_r$. Then the generalised Sliced Division $\mathcal{M} = (u[Y_u])_{u \in U}$ is admissible on $(U, \sqsubset)$.

6.4.2 Classical Partial Divisions Revisited

We now show that the Janet, Thomas, and Pommaret divisions have a structure suitable to be computed by a generalised version of Algorithm 5.

For the whole section, let $W \subseteq T$ be a finite set of terms and let $\sqsubset = \prec_{\text{invlex}}$ be the inverse lexicographical order on W. Let $\mathcal{J}$, $\mathcal{P}$, and $\mathcal{T}$ be the Janet, Pommaret, and Thomas division on W, respectively. They are admissible on $(W, \sqsubset)$.

We present particular sub-algorithms such that the generalised form of Algorithm 5 computes one of the classical divisions if these sub-algorithms are used.

Definition 6.4.5. [Hem03] Let $\emptyset \neq Y \subseteq X$ and let $\emptyset \neq U, V \subseteq W$ be finite sets of terms. Let $m \in T \cup \{\infty\}$ and assume $\forall v \in V\ \exists y \in Y : \deg_y v < \deg_y m$. We define the following functions.

Janet case: Let $\text{CF}_{\mathcal{J}}(Y, U) := \max_{\prec_{\text{lex}}} U$. Let $\text{CNM}_{\mathcal{J}}(Y, V, m) := x_i$ where i is the smallest index such that $\deg_{x_i} v < \deg_{x_i} m$ for some $v \in V$.

Pommaret case: Let x_k be such that k is the smallest index of all variables of Y and let $U' := U \cap [(X \setminus Y) \cup \{x_k\}]$ then $\text{CF}_{\mathcal{P}}(Y, U)$ returns ∞ if $U' = \emptyset$, otherwise it returns the element of U' of smallest degree. Let $\text{CNM}_{\mathcal{P}} := \text{CNM}_{\mathcal{J}}$.

Thomas case: Let $\text{CF}_{\mathcal{T}}(Y, U)$ return $l := \operatorname{lcm}(U)$ if $l \in U$ and $\deg_y t \leq \deg_y l$ for all $y \in Y$ and $t \in W$, otherwise let it return ∞. Let $\text{CNM}_{\mathcal{T}}(Y, V, m) := x_i$ where i is the smallest index such that $\deg_{x_i} v < \deg_{x_i} m$ and $\deg_{x_i} v < \deg_{x_i}(\operatorname{lcm} W)$ for some $v \in V$.

Definition 6.4.6. [Hem03] Let $\mathcal{M}$ be a partial division and $\text{CF}_{\mathcal{M}}$ and $\text{CNM}_{\mathcal{M}}$ be functions satisfying the specifications **S1** and **S2** (cf. page 43) in place of ChooseFirst and ChooseNonMultiplier, respectively. The algorithm which is obtained from Algorithm 5 by replacing lines 4, 8, and 14 by

```
4: m := CF_M(Y,U)
8: x := CNM_M(Y,U,m)
14: if m ∈ U then return CONCAT(L_C, LIST(m[Y]), L) else return L
```

is called SlicedDivision$_{\mathcal{M}}$.

It is trivial to check that the functions $\text{CF}_{\mathcal{J}}$, $\text{CF}_{\mathcal{P}}$, and $\text{CF}_{\mathcal{T}}$ fulfil the property **S1**. $\text{CNM}_{\mathcal{J}}$ and $\text{CNM}_{\mathcal{P}}$ fulfil **S2**. For $\text{CNM}_{\mathcal{T}}$ we must show that there is some $y \in Y$ and some $v \in V$ in line 8 such that $\deg_y v < \deg_y m$ and $\deg_y v < \deg_y(\operatorname{lcm} W)$. If $m \in W$, this follows by the input specification given in Definition 6.4.6. If $m = \infty$, it follows from the definition of $\text{CF}_{\mathcal{T}}$.

Since $\textsc{SlicedDivision}_{\mathcal{J}}$, $\textsc{SlicedDivision}_{\mathcal{P}}$, and $\textsc{SlicedDivision}_{\mathcal{T}}$ are generalised Sliced Division algorithms, by Theorem 6.4.1, they return admissible partial divisions.

Theorem 6.4.7. *[Hem03] Let $Y \subseteq X$, $U = \{u_1, \dots, u_r\} \subseteq W$. Let $\mathcal{M} = (t[X_t])_{t \in T}$ where $\mathcal{M}$ stands for the Janet division $\mathcal{J}$, the Pommaret division $\mathcal{P}$, or the Thomas division $\mathcal{T}$ on W. Let $\forall u \in U : X_u \subseteq Y$ and*

$$\textsc{list}(v_1[Y_{v_1}], \dots, v_r[Y_{v_r}]) = \textsc{SlicedDivision}_{\mathcal{M}}(Y, U).$$

If $\mathcal{M} = \mathcal{P}$ is the Pommaret division then additionally assume that there exists $k \in \mathbb{N}_n$ such that $Y = \{x_k, \dots, x_n\}$. Then $X_u = Y_u$ for all $u \in U$.

Proof. The proof goes by induction on the cardinalities of the input parameters Y and U. All line numbers refer to $\textsc{SlicedDivision}_{\mathcal{M}}$. For empty input parameters there is nothing to show. Now consider $Y \subseteq X$ and $U \subseteq W$ and assume that the theorem is proved for all calls of $\textsc{SlicedDivision}_{\mathcal{M}}$ where the sum of the cardinalities of the input parameters is strictly less than $\operatorname{card} Y + \operatorname{card} U$. Let m, C, and V be as defined in lines 4–6. Let s be the number of iterations of the <u>while</u> loop, and let y_σ, d_σ, S_σ, and V_σ be the values of x, d, S, and V after the σ-th iteration ($\sigma \in \mathbb{N}_s$). By the input specification of $\textsc{SlicedDivision}_{\mathcal{M}}$, there exists an element $w \in [X \setminus Y]$ such that $U \subseteq W \cap w[Y]$. For line 12 we find that $S_\sigma \subseteq y_\sigma^{d_\sigma} w[Y \setminus \{y_\sigma\}]$, thus the input specification of $\textsc{SlicedDivision}_{\mathcal{M}}$ is satisfied. There are two cases to consider.

$m \in U$**:** By line 14, $Y_m = Y$ will be returned, it remains to show $X_m = Y$. Since $\operatorname{card} C, \operatorname{card} V < \operatorname{card} U$, the correct assignment of multiplicative variables for the elements of $U \setminus \{m\}$ follows from the induction hypothesis if we can show that the assumptions of the theorem are satisfied in lines 5 and 12. For line 5 this is trivial, since $C \subsetneq U$ and Y does not change. Note that $C = \emptyset$ if $\mathcal{M} = \mathcal{J}$.

For the remaining considerations, we treat the various cases separately.

$\mathcal{M} = \mathcal{J}$**:** From the assumption of the theorem, $X_m \subseteq Y$. By definition of $\mathcal{J}$ and $\textsc{cf}_{\mathcal{J}}$, we must have $X_m = Y$. By $m = \max_{\prec_{\text{lex}}} U$, line 6, the choice of y_σ through $\textsc{cnm}_{\mathcal{J}}$, and from line 10 we conclude $\deg_{y_\sigma} v < \deg_{y_\sigma} m$ for all $v \in S_\sigma$. By definition of the Janet division, y_σ is non-multiplicative for all elements of S_σ. Hence, the theorem is applicable for $(Y \setminus \{y_\sigma\}, S_\sigma)$ in line 12.

$\mathcal{M} = \mathcal{T}$**:** From the assumption of the theorem, $X_m \subseteq Y$. By definition of $\mathcal{T}$ and $\textsc{cf}_{\mathcal{T}}$, we must have $X_m = Y$. By $m = \operatorname{lcm}(U)$, line 6, the choice of y_σ through $\textsc{cnm}_{\mathcal{T}}$, and from line 10 we conclude $\deg_{y_\sigma} v < \deg_{y_\sigma} m$ for all $v \in S_\sigma$. By definition of the Thomas division, y_σ is non-multiplicative for all elements of S_σ. Hence, the theorem is applicable for $(Y \setminus \{y_\sigma\}, S_\sigma)$ in line 12.

$\mathcal{M} = \mathcal{P}$**:** Let $U' = U \cap [(X \setminus Y) \cup \{x_k\}]$. By definition of the Pommaret division, $X_u = Y$ for all $u \in U'$. By definition of $\textsc{cf}_{\mathcal{P}}$, $m = wx_k^a \in U'$ with the smallest possible a. Note that $U' \subseteq m[Y]$. By the choice of m and line 6, it follows that for all $v \in V$ $\deg_{x_k} v < a$ and $\exists j > k : \deg_{x_j} v > 0$. Therefore $y_\sigma = x_k$ and $x_k \notin X_v$. Hence, the theorem is applicable for $(Y \setminus \{x_k\}, S_\sigma) = (\{x_{k+1}, \dots, x_n\}, S_\sigma)$ in line 12.

$m = \infty$**:** This case does not occur if $\mathcal{M}$ is the Janet division. From lines 5 and 6 follows $C = \emptyset$ and $V = U$. Let $\sigma \in \mathbb{N}_s$. Correct assignment of the multipliers follows from the induction hypothesis applied to $(Y \setminus \{y_\sigma\}, S_\sigma)$ in line 12. We must show that the assumptions of the theorem are applicable.

If $\mathcal{M} = \mathcal{T}$ then $X_v \subseteq Y \setminus \{y_\sigma\}$ for all $v \in V$ follows from the definition of $\text{CNM}_\mathcal{T}$ and the definition of the Thomas division on W.

If $\mathcal{M}$ is the Pommaret division, we observe that $y_\sigma = x_k$ by $\text{CNM}_\mathcal{P}$ and that for all $v \in V$ there exists $j > k$ such that $\deg_{x_j} v > 0$, i. e., $X_v \subseteq \{x_{k+1}, \ldots, x_n\}$.

Hence, all assumptions of the theorem are satisfied in line 12 and we get $X_u = Y_u$ for all $u \in S_\sigma$ and consequently for all $u \in U$.

The theorem is proved. □

The following corollary states that Janet, Pommaret, and Thomas divisions are generalised Sliced Divisions.

Corollary 6.4.8. *[Hem03] Let $\mathcal{M}$ be the Janet, the Pommaret, or the Thomas division on W. Let $\text{LIST}(v_1[Y_{v_1}], \ldots, v_r[Y_{v_r}]) = \text{SLICEDDIVISION}_\mathcal{M}(X, W)$. $\mathcal{M}$ and the induced division of $(v[Y_v])_{v \in W}$ are W-equivalent.*

In general, it is not true that any admissible partial division can be obtained by a generalised version of Algorithm 5. This can be seen from the following example.

Example 6.4.9. *Take $X = \{x, y, z\}$, $U = \{xy, z^2, z\}$ with $xy \sqsubset z^2 \sqsubset z$ and corresponding sets of multiplicative variables*

$$Y_{xy} := \{z\}, \qquad Y_{z^2} := \{y\}, \qquad Y_z := \{x\}.$$

The induced partial division is admissible on U since

$$xy[X] \cap z^2[Y_{z^2}] = \emptyset, \qquad xy[X] \cap z[Y_z] = \emptyset, \qquad z^2[X] \cap z[Y_z] = \emptyset.$$

A generalised Sliced Division algorithm which should return the above division would first have to choose ∞ as m in line 4, i. e., $\text{CHOOSEFIRST}(X, U) = \infty$. So $V = U$ and any choice in line 8 of the generalised Sliced Division algorithm would not lead to the above division. Indeed, if x is chosen then we find that $S = \{xy\}$ and thus xy cannot get x as multiplicative as we see from line 12. Similarly, the choices of y or z do not lead to the desired division.

6.5 Sliced Division Tree

In this section we construct tree representations for a subclass of generalised Sliced Divisions and show how such trees allow a fast search for a divisor of a given term. This subclass is determined by an output of the generalised version of Algorithm 5 with the following restrictions on the sub-algorithms ChooseFirst and ChooseNonMultiplier.

T1: If $\emptyset \neq U \subseteq T$ is a finite set of terms, $Y \subseteq X$, and $m = \text{CHOOSEFIRST}(Y, U)$ then $m \in U$.

T2: Let $\emptyset \neq V \subseteq T$ be a finite set of terms, $Y \subseteq X$, $m \in T$,

$$\forall v \in V \, \exists y \in Y : \deg_y v < \deg_y m$$

and $x = \textsc{ChooseNonMultiplier}(Y, V, m)$. Then

$$x = \max_{>} \left\{ y \in Y \,\middle|\, \exists v \in V : \deg_y v < \deg_y m \right\}.$$

The order $>$ in **T2** for determining the maximum is an arbitrary order on the set X of variables and is fixed during the whole run of the generalised Sliced Division algorithm.

Definition 6.5.1. Let $U \subseteq T$ be a finite set of terms and let $>$ be an arbitrary order on X. The algorithm DirectedSlicedDivision is the generalised version of Algorithm 5 where the sub-algorithms ChooseFirst and ChooseNonMultiplier satisfy **T1** and **T2**, respectively.

A **$>$-directed Sliced Division on** U is a partial division which is induced by the output of DirectedSlicedDivision when started with input X and U.

Example 6.5.2. *From Definition 6.4.5 we see that* $\mathrm{CF}_{\mathcal{J}}$ *and* $\mathrm{CNM}_{\mathcal{J}}$ *satisfy the conditions **T1** and **T2**, respectively. Thus, the Janet division (with respect to* $>$*, cf. Definition 4.4.1) is a* $>$*-directed Sliced Division.*

In the sequel we fix the order $>$ to be defined by $x_1 > x_2 > \cdots > x_n$ and thus simply speak of a **directed Sliced Division** without explicitly mentioning the order on the variables.

The goal of Section 6.5 is to show how the structure of a directed Sliced Division $\mathcal{M}$ can be exploited to find an $\mathcal{M}$-divisor of a term t in a given set of terms U in a time independent of the cardinality of U. In fact, at the time when a Sliced Division is computed, we build a tree whose nodes correspond to the terms of U and then we search a divisor of t by taking advantage of such a tree structure. We show that one can find a divisor in $\mathcal{O}(n^2)$ steps where n is the number of variables. More precisely, we show that one can find a simplifier in $\mathcal{O}(nh)$ integer comparisons where $h = \min(n, r)$ and r is the number of terms in the tree. In practical computations the number of terms/polynomials usually dominates the number of variables, one can therefore say that the search time is independent of the degree and the number of terms in the actual basis.

A tree structure for the very special case of a Janet division has already been proposed in [GBY01a]. We improve their result from $\mathcal{O}(n + d)$ where d is the maximal total degree of the involved terms to $\mathcal{O}(n)$.

6.5.1 Construction of a Sliced Division Tree

In order to help to understand the construction of the tree structure (given below), we first give an informal sketch. As explained in Remark 6.2.2, the Sliced Division algorithm "cuts slices" from the monoid ideal generated by some given set U of terms. Now, instead of admitting to cut slices from an arbitrary side (which corresponds to an arbitrary choice of a variable in line 8 of the generalised version of Algorithm 5), we take first all possible slices corresponding to the biggest variable. This, in fact, turns the while loop into a for loop running from the biggest to the smallest variable. In each iteration all slices corresponding to the respective variable are cut starting from the smallest degree.

We turn the generalised Sliced Division algorithm into an algorithm which computes a directed Sliced Division in form of a tree rather than a list. Using the same choice functions satisfying **T1** and **T2**, respectively, the generalised Sliced Division algorithm and SLICEDDIVISIONTREE return just different representations of the same admissible partial division $\mathcal{M}$ and the same order $\sqsubset$ when started on the same input sets.

The algorithm to construct such a tree is presented as Algorithm 6. The call $\text{ARRAY}_n(\mathsf{nil})$ produces an array of size n where all entries are initialised by nil. If A is an array, we use the usual notation $A[i]$ for the i-th entry of the array A. In line 12 of Algorithm 6, the set V is partitioned into subsets L_d of elements of V with x_i-degree equal to d ($a \leq d < b$) and the remaining elements (which stay in V).

Algorithm 6 SLICEDDIVISIONTREE

Call: $M := \text{SLICEDDIVISIONTREE}(Y, U)$

Input: $Y \subseteq X$ is finite set of indeterminates.
$U = \{u_1, \ldots, u_r\} \subseteq w[Y]$ for some $w \in [X \setminus Y]$ where $r = \operatorname{card} U$.

Output: M is a Sliced Division tree of U and Y.

1: $A := \text{ARRAY}_n(\mathsf{nil})$
2: <u>if</u> $U = \emptyset$ <u>then</u> <u>return</u> nil
3: <u>if</u> $Y = \emptyset$ <u>then</u> <u>return</u> $\text{NODE}(\mathsf{nil}, w[Y], A)$
4: Choose $m \in U$
5: $C := (U \cap m[Y]) \setminus \{m\}$; $M_C := \text{SLICEDDIVISIONTREE}(Y, C)$
6: $V := U \setminus m[Y]$
7: <u>for</u> $i := 1$ <u>to</u> n <u>do</u>
8:     <u>if</u> $V = \emptyset$ <u>then</u> <u>return</u> $\text{NODE}(M_C, m[Y], A)$
9:     $a := \deg_{x_i}(\gcd V)$; $b := \deg_{x_i} m$; $s := b - a$
10:     <u>if</u> $s = 0$ <u>then</u> <u>next</u> i
11:     $S := \text{ARRAY}_s(\mathsf{nil})$
12:     $(L_a, \ldots, L_{b-1}, V) := \text{SLICE}(V, x_i, a, b)$
13:     <u>for</u> $d := a$ <u>to</u> $b - 1$ <u>do</u> $S[d] := \text{SLICEDDIVISIONTREE}(Y \setminus \{x_i\}, L[d])$
14:     $A[i] := S$
15: <u>return</u> $\text{NODE}(M_C, m[Y], A)$

Definition 6.5.3. Let $U \subseteq T$ be a finite set of terms and let

$$M = \text{SLICEDDIVISIONTREE}(X, U).$$

Then M is called a **Sliced Division tree of** U.

The order $\sqsubset$ on the input set U is, in analogy to Algorithm 5, induced by the process in which the slices are cut. It is determined by the following conditions on each node of the returned tree. Let $\text{NODE}(M_C, m[Y], A)$ be a node in a Sliced Division tree.

1. We define $c \sqsubset m \sqsubset a$ for all elements $c \in U$ in the sub-tree M_C and all elements $a \in U$ that are in one of the sub-trees stored in the array A.

2. Furthermore, if for $1 \le i < j \le n$ the element $a_i \in U$ is in a sub-tree stored in $A[i]$ and $a_j \in U$ is in a sub-tree stored in $A[j]$ then we define $a_j \sqsubset a_i$.

3. If for some $1 \le i \le n$ and $0 \le j < k < \deg_{x_i} m$ the elements $s_j, s_k \in U$ are stored in sub-tree $A[i][j]$ and $A[i][k]$, respectively, then we define $s_k \sqsubset s_j$.

Proposition 6.5.4. *Let $U \subseteq T$ be a finite set of terms and let*

$$M = \textsc{SlicedDivisionTree}(X, U).$$

Let furthermore the order $\sqsubset$ be determined as above. Then the nodes of M induce a partial division which is admissible on $(U, \sqsubset)$.

Proof. Similar to Algorithm 5, the algorithm SlicedDivisionTree constructs a tree node for an element m of U in each (recursive) call. So every element of the input set U corresponds bijectively to a node in the resulting tree. There are exactly $\operatorname{card} U$ nodes in the tree. The algorithm assigns multiplicative variables in exactly the same way as in the generalised version of Algorithm 5. Also the order $\sqsubset$ is defined in such a way that it agrees with the order that is induced by the generalised Sliced Division algorithm. So we can see a run of Algorithm 6 as a run of a generalised version of Algorithm 5 with additional commands to produce a tree. The tree structure is, however, irrelevant for the induced division. We conclude from Corollary 6.4.4 the claim of the proposition. □

We see from algorithm InvBasis (Algorithm 4 on page 32) that multiplicative variables of elements that lie in some cone (they correspond to elements in Q) are never accessed. So it is useless work if we really perform the recursive call in line 5 of Algorithm 6. By neglecting such recursive calls, the resulting tree has at most $r = \operatorname{card} U$ elements. Since by each recursive call the number of variables decreases by one and the tree does not contain more than r nodes, the Sliced Division tree is at most of depth $\min(n, r)$.

Since the construction of a Sliced Division tree is rather technical, let us consider an example.

We depict the tree in the following way, cf. Figure 6.2. Each box represents a node, i. e., a term m together with an array A of pointers to arrays of sub-trees. The entries of A are shown by edges that are labelled by the respective variable and that are missing if the entry is nil. For all terms below such a labelled edge, the respective variable is non-multiplicative. The black dots represent the arrays of the slices. The labels of the edges below a black dot correspond to the common degree (shown by a circle) of the terms in the sub-tree.

The multiplicative variables of a term can easily be read off from the tree. One follows the path from the root to the particular node and makes each variable that one encounters as a label on this path as non-multiplicative for the target term. All other variables are multiplicative for that term.

Example 6.5.5. *Let us follow the construction of a Sliced Division tree (with respect to $x > y > z$) of the set $U = \{x^6y^6, x^4y^3z, x^3y^3z^4, x^2y^3z^5, xy^7, xyz^5, y^4z\}$ where we always choose the degreewise smallest term in line 4 of Algorithm 6. The final tree together with a visual representation of the cone decomposition can be seen in Figure 6.2.*

In the following, we denote by A_t the array that is stored in the node corresponding to the term t.

The algorithm SLICEDDIVISIONTREE *starts by choosing a term of minimal total degree. In this example, $m = y^4z$ is chosen and it gets x, y, and z as multiplicative. This term becomes the root of the tree. Since no other term is divisible by y^4z, the set C is empty and $V = U \setminus \{y^4z\}$. Now, we divide V into two parts. Since $\deg_x(\gcd V) = \deg_x m$, there are no slices here and thus $A_{y^4z}[1]$ is set to* nil*. So we go on and divide V with respect to y. The first part contains elements of y-degree smaller than 4 and the second comprises the remaining elements. The first part is subdivided into sets which have equal degree in y. In our case we then have $S_{y1} = \{xyz^5\}$ and $S_{y3} = \{x^4y^3z, x^3y^3z^4, x^2y^3z^5\}$ which corresponds to two branches of the tree labelled with 1 and 3 below the edge that is labelled with y. For these elements the variable y is non-multiplicative.*

Now, we treat S_{y1}, S_{y3} and $V := V \setminus (S_{y1} \cup S_{y3})$. For S_{y1} we are already finished. The term xyz^5 gets x and z as multiplicative and the sub-tree corresponding to this term is stored in $A_{xyz^5}[2][1]$.

For S_{y3} the algorithm is applied recursively with respect to the variable set $\{x, z\}$. The term of smallest degree is x^4y^3z and x and z are its multiplicative variables. Both of the remaining terms can be separated with respect to x, because their degree is smaller than 4. The variable x becomes non-multiplicative and only z is left as multiplicative for $x^3y^3z^4$ and $x^2y^3z^5$. All elements of S_{y3} have their multipliers. $A_{x^4y^3z}[1][2]$ is set to the sub-tree with root $x^2y^3z^5$ and $A_{x^4y^3z}[1][3]$ to the sub-trees with root $x^3y^3z^4$. The sub-tree for S_{y3} has been computed. It is stored in $A_{y^4z}[2][3]$.

The remaining terms $\{xy^7, x^6y^6\}$ are treated in a similar fashion. The corresponding sub-tree is stored in $A_{y^4z}[3][0]$. All array entries that have not been mentioned explicitly are set to nil*.*

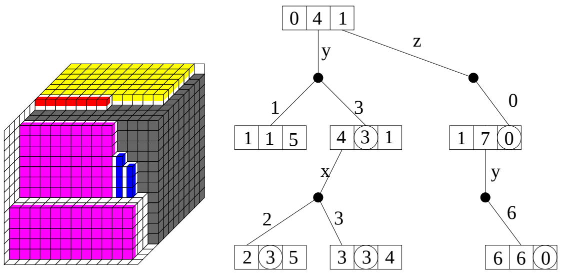


Figure 6.2: Cone decomposition and Sliced Division tree for the Sliced Division of Example 6.5.5

If we choose as m not the element of minimal total degree but the lexicographically biggest element, we will end up with a Janet division. The corresponding decomposition and tree is shown in Figure 6.3.

6.5.2 Search for a Divisor

Let us now describe how a Sliced Division tree can speed up the divisor search. The tree has been constructed in a recursive manner, so it is enough to describe

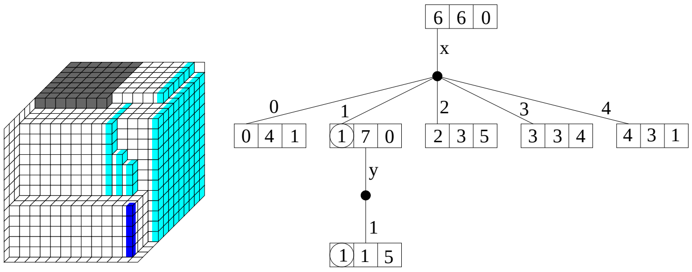


Figure 6.3: Cone decomposition and Sliced Division tree for the Janet division of Example 6.5.5

how the divisor search works at some node level. The formal algorithm is given as Algorithm 7.

Definition 6.5.6. Let $M = \textsc{node}(M_C, m[Y], A)$ be a Sliced Division tree of a finite non-empty set $U \subseteq T$. We denote by $\textsc{RootTerm}(M)$ the term m that is stored in the root node of the tree and by $\textsc{Sides}(M)$ the array A.

Let $\mathcal{M} = (u[Y_u])_{u\in U}$ be the admissible partial division corresponding to M. Given a term $t \in T$, we say that t **has an $\mathcal{M}$-divisor in** M if there is some $u \in U$ stored in some node of M such that $t \in u[Y_u]$.

Proposition 6.5.7. *Let $U \subseteq T$ be a finite non-empty set of terms, let*

$$M = \textsc{SlicedDivisionTree}(X, U),$$

and let $\mathcal{M} = (u[Y_u])_{u\in U}$ be the U-admissible partial division corresponding to M. Furthermore, let $m = \textsc{RootTerm}(M)$ and $A = \textsc{Sides}(M)$. A term $t \in T$ has an $\mathcal{M}$-divisor in M if and only if one of the following conditions holds

1. *$t \in \mathcal{M}_m$ or*
2. *t has an $\mathcal{M}$-divisor in $A[j][\deg_{x_j} t]$ for $j = \min_{i\in\mathbb{N}_n}(\deg_{x_i} m > \deg_{x_i} t)$.*

Proof. Note that we can safely ignore the branches of the tree that correspond to elements of U which have an $\mathcal{M}$-divisor in M. These branches stem from recursive calls in line 5 of Algorithm 6. Without loss of generality we can assume that no such branches exist. In this case, $\mathcal{M}$ is disjoint on U, and if there is an $\mathcal{M}$-divisor $u \in U$ of t then this u is unique.

Assume that neither the first nor the second condition holds, and let

$$j := \min_{i\in\mathbb{N}_n}(\deg_{x_i} m > \deg_{x_i} t).$$

For any $i < j$, sub-trees corresponding to $A[i]$ contain elements $u \in U$ with $\deg_{x_i} u < \deg_{x_i} m \leq \deg_{x_i} t$, but $x_i \notin Y_u$ by construction of M. So none of these elements is an $\mathcal{M}$-divisor of t.

Similarly, if $k > j$ then for each element $u \in U$ stored in a sub-tree corresponding to $A[k]$, it holds $\deg_{x_j} u \geq \deg_{x_j} m > \deg_{x_j} t$. Thus u is not even an ordinary divisor of t.

It is clear that t has no $\mathcal{M}$-divisor in $A[j][d]$ for any $d \neq \deg_{x_j} t$, since by construction of M each element $u \in U$ in the sub-tree $A[j][d]$ has x_j-degree equal to d and $x_j \notin Y_u$. □

Algorithm 7 FINDPARTIALDIVISOR

Call: $m := \textsc{FindPartialDivisor}(t, M)$

Input: $t \in T$
M is a Sliced Division tree of $U \subseteq T$.

Output: Let $\mathcal{M}$ be the admissible partial division on U corresponding to M. Then $m = u$ if there is $u \in U$ such that $t \in \mathcal{M}_u$ and $m =$ noDivisor otherwise.

1: $N := M; \quad j := 0$
2: <u>while</u> $j < n$ <u>do</u>
3:   <u>if</u> $N =$ nil <u>then</u> <u>return</u> noDivisor
4:   $j := j + 1$
5:   $m := \textsc{RootTerm}(M)$
6:   $d := \deg_{x_j} t$
7:   <u>if</u> $d < \deg_{x_j} m$ <u>then</u> # *t is not divisible by m*
8:     $A := \textsc{Sides}(M)$
9:     <u>if</u> $A[j] =$ nil <u>then</u> <u>return</u> noDivisor
10:     $N := A[j][d]; \quad j := 0$
11: <u>return</u> m

Proposition 6.5.8. *Algorithm 7 terminates and meets its specification.*

Proof. Let $U \subseteq T$ be a finite set of terms, $M = \textsc{SlicedDivisionTree}(X, U)$, and $t \in T$. As explained at the end of Section 6.5.1 the tree M has at most depth $h := \min(n, \operatorname{card} U)$. Hence, line 10 is executed at most h times during the whole computation. In each iteration of the <u>while</u> loop the variable j is incremented. It must eventually reach n (if the algorithm does not return beforehand) and the algorithm terminates.

Correctness follows from Proposition 6.5.7. □

Proposition 6.5.9. *Let $U \subseteq T$ be a finite set of terms, $t \in T$, and $M =$ $\textsc{SlicedDivisionTree}(X, U)$. The run of* $\textsc{FindPartialDivisor}(t, M)$ *takes at most $\mathcal{O}(n \cdot \min(n, \operatorname{card} U))$ integer comparisons to return its result.*

Proof. Line 10 of Algorithm 7 is executed at most $h := \min(n, \operatorname{card} U)$ times. Therefore, the <u>while</u> loop is iterated at most $n \cdot h$ times. Since the most costly operation of the loop appears in line 7, we conclude that at most $\mathcal{O}(nh)$ integer comparisons are needed. □

In Example 6.5.2 we have seen that the Janet division (with respect to $>$) is a $>$-directed Sliced Division (with respect to the same order $>$ on the variables X). So we can compute a Sliced Division tree corresponding to a Janet division by means of a particular variant of Algorithm 6. Proposition 6.5.9 tells us that we can find a Janet-divisor in such a tree in $\mathcal{O}(n^2)$ integer comparisons, since the number of given terms is usually much bigger than the number n of variables. In contrast to the complexity result of $\mathcal{O}(n + d)$ proposed in [GBY01a] where d is the maximal total degree of the involved terms, our complexity result is already independent of the number of terms in the tree and independent of their degrees. Additionally, we have presented a method to construct a tree structure for the whole class of directed Sliced Divisions and not only the Janet division.

In the sequel, we show that for the Janet division our tree structure allows a search time of $\mathcal{O}(n)$ in the worst case, since such a tree has an additional property which can be exploited.

Definition 6.5.10. We denote Algorithm 6 where in line 4 always the lexicographically biggest element is chosen by $\mathrm{SDT}_{\mathcal{J}}$.

Lemma 6.5.11. *Let $U \subseteq T$ be a finite set of terms and $M = \mathrm{SDT}_{\mathcal{J}}(X, U)$. Then the partial division induced by M is the Janet division.*

Proof. In Example 6.5.2 we have shown that the Janet division is a directed Sliced Division. Since we use the sub-algorithms $\mathrm{CF}_{\mathcal{J}}$ and $\mathrm{CNM}_{\mathcal{J}}$ in $\mathrm{SDT}_{\mathcal{J}}$ the claim follows from Theorem 6.4.7 and the fact that the algorithms DirectedSlicedDivision and SlicedDivisionTree assign the multiplicative variables in exactly the same way. ☐

A particularly "nice" property of a Sliced Division tree corresponding to a Janet division is that in each node one can predict equality of degrees of certain elements that occur at a "later" stage in Algorithm 7. This property, given through the following lemma, allows us to cut several branches in the search path of Algorithm 7 so that it improves its the worst case time complexity.

Lemma 6.5.12. *Let U and Y satisfy the input specifications of Algorithm 6 and let $M = \mathrm{SDT}_{\mathcal{J}}(Y, U)$. Then for all $j \in \mathbb{N}_n$*

$$\forall i, k \in \mathbb{N}_n : i \leq j < k \implies \forall v \in V_k : \deg_{x_i} v = \deg_{x_i} m \tag{6.3}$$

where V_j denotes the terms stored in the nodes of the sub-trees of $A[j]$.

Proof. Since $v \prec_{\mathrm{lex}} m$, we have $\deg_{x_1} v \leq \deg_{x_1} m$ for all $v \in U \setminus \{m\}$ and $V_1 = \{v \in U \setminus \{m\} \mid \deg_{x_1} v < \deg_{x_1} m\}$ by construction of M. So, for any $k > 1$ and any $v \in V_k$ the equation $\deg_{x_1} v = \deg_{x_1} m$ must hold. Let $j' \leq n$ and assume that (6.3) is correct for all $j < j'$. It suffices to prove

$$\forall k \in \mathbb{N}_n : j' < k \implies \forall v \in V_k : \deg_{x_{j'}} v = \deg_{x_{j'}} m \tag{6.4}$$

By induction hypothesis and since $v \prec_{\mathrm{lex}} m$, we have $\deg_{x_{j'}} v \leq \deg_{x_{j'}} m$ for all $v \in U_{j'} := U \setminus (V_1 \cup \cdots \cup V_{j'-1})$. From the construction of M follows

$$V_{j'} = \left\{ v \in U_{j'} \,\middle|\, \deg_{x_{j'}} v < \deg_{x_{j'}} m \right\}$$

and hence (6.4). ☐

Let $U \subseteq T$ be a finite set of terms, $M = \mathrm{SDT}_{\mathcal{J}}(X, U)$, and $t \in T$. Consider the run of FindPartialDivisor(t, M). If in line 10, the variable N gets the sub-tree $A[j][d]$ as its new value, we know from Lemma 6.5.12 that for all $i \leq j$ and all terms u and v in the sub-tree $A[j][d]$ the equality $\deg_{x_i} u = \deg_{x_i} v$ holds. These equalities make the assignment $j := 0$ in line 10 superfluous. Let us denote the algorithm which is obtained from algorithm FindPartialDivisor by removing the statement $j := 0$ from line 10 by $\mathrm{FPD}_{\mathcal{J}}$.

The following proposition is now obvious.

Proposition 6.5.13. *Let $U \subseteq T$ be a finite set of terms, $t \in T$, and $M = \mathrm{SDT}_{\mathcal{J}}(X,U)$. The run of $\mathrm{FPD}_{\mathcal{J}}(t,M)$ takes at most $\mathcal{O}(n)$ integer comparisons to return its result.*

Let us illustrate the divisor search on some examples.

Example 6.5.14. *We continue Example 6.5.5. Let us try to find a partial divisor for $t = x^5y^2$ in the tree represented by Figure 6.2. Let $m = y^4z$ be the root term. Since $\deg_x m < \deg_x t$ it is still possible that m could divide t. Then we find $\deg_y m > \deg_y t$. Thus we go along the y-branch of the tree and try to look for a divisor in the sub-tree that corresponds to $\deg_y t = 2$. Since there is no such sub-tree, there is no term in the tree that can be multiplied only by its multiplicative variables to reach t.*

As another example we take the term $t' = x^7yz^5$. Let $m = y^4z$ be the root term. Since $\deg_x m < \deg_x t'$ it is still possible that m could divide t. Then we find $\deg_y m > \deg_y t'$. Thus we go along the y-branch of the tree and try to look for a divisor in the sub-tree corresponding to $\deg_y t' = 1$. xyz^5 with multiplicative variables x and z is the only element in that sub-tree. Clearly, $x^7yz^5 \in xyz^5[\{x,z\}]$, so a divisor has been found.

Note that there is no divisor of t' in the tree corresponding to the Janet division in Figure 6.3. Since the x-degree of t' is bigger than 6, we must consider y. However, the y-degree of t' is smaller than 6 and there is no sub-tree corresponding to y below x^6y^6.

Example 6.5.15. *Recall Example 5.2.1. We take the leading terms of the final Janet basis $\{f_1, \dots, f_{10}\}$ as given in the table on page 30 and compute the corresponding tree for the Janet division (with respect to $x > y > z$). The cone decomposition and the corresponding Sliced Division tree are shown in Figure 6.4. The double circles around the 8 in Figure 6.4 correspond to Lemma 6.5.12 and the fact that in a Janet division elements of trees corresponding to some variable (y in our case) share the same degree in all bigger variables (x in our case).*

The construction of the tree goes as follows. The lexicographically biggest term is $x^8y^5z^2$. It becomes the root of the tree. All terms with a smaller x-degree are put under the branch which is labelled by x. Terms with x-degree equal to 8 are put under the branch labelled by y. There are exactly 3 sub-trees in the y-branch according to the y-degree of the terms, namely $x^8y^2z^{11}$, $x^8y^3z^5$, and $x^8y^4z^4$. In the x-branch we split the set of terms according to their x-degrees, which are 4, 5, 6, and 7, respectively. We have two terms of x-degree equal to 6, namely $x^6y^5z^3$ and $x^6y^4z^9$ and two terms of x-degree equal to 7, namely $x^7y^5z^3$ and $x^7y^4z^9$. In each case the first term is the lexicographically biggest and thus becomes the root of the sub-tree. The multiplicative variables corresponding to this tree (as shown in Figure 6.4) are exactly as given in the table on page 30. We just must follow the path from the root to each term and remove the variables encountered on labels of the path from the set of multiplicative variables for the respective term.

Let us search for a divisor of $t = x^8y^2z^4$ in this tree. We start by comparing the x-degree of t with the x-degree of the root term m. Since they are both equal to 8, we continue to test the y-degree. There we find that $\deg_y t < \deg_y m$. We follow the y-branch and take the sub-tree which corresponds to $\deg_y t$. Its root node is $m' := x^8y^2z^{11}$. By comparing the z-degrees of t and m', we realise that m' is not a Janet divisor of t. Since there is no node below the node of m', there is no Janet divisor of t at all in the tree.

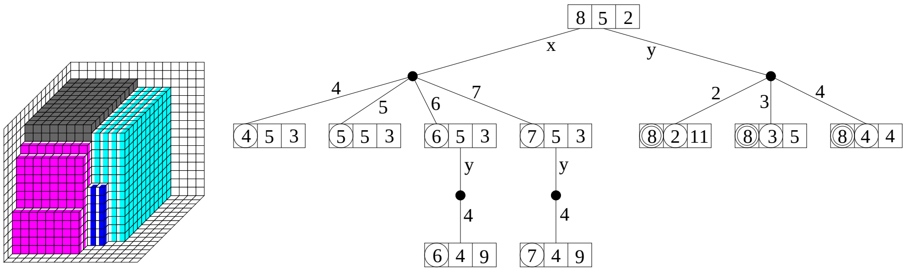


Figure 6.4: Cone decomposition and Sliced Division tree for the Janet division of Example 6.5.15

Example 6.5.16. *Let us consider the set $G = \{f_1, \ldots, f_7\}$ of Example 7.5.3 on page 74. G is an $\mathcal{M}$-involutive basis of the polynomials f_1, f_2, f_3 from Example 5.2.1 with respect to a Sliced Division $\mathcal{M}$. The cone decomposition of $\mathcal{M}$ and the corresponding Sliced Division tree are depicted in Figure 6.5. The tree contains fewer nodes than the tree in Figure 6.4, since G contains only seven elements.*

Let us search for a divisor of $t = x^8y^2z^4$ in this tree. We start by comparing the x-degree of t with the x-degree of the root term $m = x^4y^5z^3$. Since $\deg_x m < \deg_x t$, we cannot yet decide whether or not m divides t, since we must also compare the degrees of the other variables. We find $\deg_y m > \deg_y t$. Thus we go along the y-branch of the tree and try to look for a divisor in the sub-tree that corresponds to $\deg_y t = 2$. The only element in this subtree is $x^8y^2z^{11}$ and it is not a divisor of t. Thus we conclude that there is no $\mathcal{M}$-divisor of t in the tree.

Since G is an $\mathcal{M}$-involutive basis, the set $U := \operatorname{lt} G$ is $\mathcal{M}$-complete. Hence $\langle U \rangle = \langle U \rangle_{\mathcal{M}}$ and we conclude that there is no (ordinary) divisor of t in U. Note that we have thus verified that $t \notin \langle U \rangle$ by just touching two terms of U. A similar argument also holds for the (complete) Janet division of Example 6.5.15.

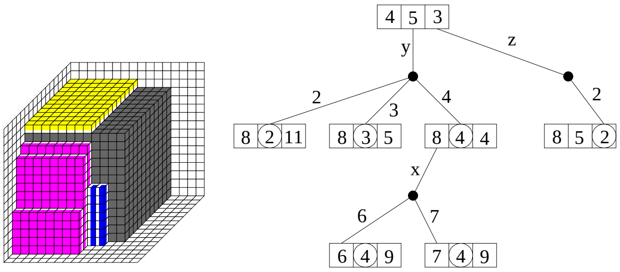


Figure 6.5: Cone decomposition and Sliced Division tree for the Sliced Division of Example 6.5.16

6.6 Sliced Dickson Lemma

The generalised form of the Janet basis algorithm is presented as Algorithm 4 with a normal selection strategy in line 6. Such a strategy is essential for the termination of the involutive basis algorithm as can be seen from Example 6.6.1. Also the assumption of a fair selection strategy, i. e., no prolongation stays forever in the set C of critical elements, is insufficient for ensuring termination. If, however, in Algorithm 4 the sub-algorithm PARTIALDIVISION is a generalised Sliced Division algorithm which satisfies certain properties then any prolongation selection strategy can be used.

Let us give a simple example of how it can happen that the involutive basis algorithm does not terminate if one deviates from the normal strategy.

Example 6.6.1. *[AH02] We use the lexicographical term order refining $u > v > w > x > y$. For the readers convenience we write the multiplicative variables in curly brackets behind each polynomial. The division refines the Thomas division in each iteration. Let the following 7 polynomials be given and consider Algorithm 4 with a selection of the next prolongation as given below.*

$$g_1 = x\ \{x\}, \quad g_2 = y\ \{y\}, \quad g_3 = u^2\ \{u\}, \quad g_4 = v^2\ \{v\}, \quad g_5 = w^2\ \{w\},$$
$$g_6 = ux - wy\ \{w, x\}, \quad g_7 = vy - wx\ \{w, y\}$$

First we reduce the prolongation ug_1 by g_6 and obtain $g_8 := wy\ \{w, y\}$. Then we reduce vg_8 by wg_7 and obtain $g_9 := w^2x\ \{w, y\}$. Now y is the only multiplicative variable for g_8. Then we add $g_{10} := ug_9 - w^2g_6 = w^3y\ \{w, y\}$. All variables are now non-multiplicative for g_5. We go on by adding $g_{11} := vg_{10} - w^3g_7 = w^4x\ \{w, y\}$, $g_{12} := ug_{11} - w^4g_6 = w^5y\ \{w, y\}$, etc. Each time a polynomial g_{i+1} $(i > 7)$ is added to the basis, the variable w becomes non-multiplicative for g_i. Obviously, this process is infinite by adding the polynomials $g_{2i+6} = w^{2i-1}y$, $g_{2i+7} = w^{2i}x$ $(i > 0)$.

The following theorem can be seen as Dickson's Lemma (cf. [Dic13]) for the case of Sliced Divisions. Let W be an infinite sequence of terms. The theorem basically says that under certain conditions on the partial division, which we describe as restrictions on the choice functions of the generalised Sliced Division algorithm, the set U of elements of a finite sub-sequence of W is sufficient so that any element of W has an $\mathcal{M}$-divisor in U where $\mathcal{M}$ is a generalised Sliced Division on U.

Theorem 6.6.2 (Sliced Dickson Lemma). *Let X be a set of n variables and $W = (t_i)_{i>0}$ be a family of terms from $T := [X]$. Let CF and CNM be two functions*

$$\mathrm{CF} : \mathbb{P}(X) \times \mathbb{P}_{\mathrm{fin}}(T) \to T$$
$$\mathrm{CNM} : \mathbb{P}(X) \times \mathbb{P}_{\mathrm{fin}}(T) \times T \to X$$

such that for all $Y \subseteq X$ and $\emptyset \neq U, U' \subseteq T$ with $\langle U \rangle = \langle U' \rangle$ holds

$$\mathrm{CF}(Y, U) = \mathrm{CF}(Y, U') \in U.$$

And if additionally $m \in T$, $x = \mathrm{CNM}(Y, U, m)$, and

$$\forall v \in U\ \exists y \in Y : \deg_y v < \deg_y m$$

then

$$x \in Y \wedge \exists v \in U : \deg_x v < \deg_x m,$$
$$\textsc{cnm}(Y, U, m) = \textsc{cnm}(Y, U', m).$$

Let $\mathcal{M}^k = \big(t[Y_t^k]\big)_{t \in T}$ *be the Sliced Division on* $\{t_1, \ldots, t_k\}$ *computed by the generalised version of Algorithm 5 with input* X *and* $\{t_1, \ldots, t_k\}$ *using for the sub-algorithms* ChooseFirst *and* ChooseNonMultiplier *the functions* cf *and* cnm, *respectively.*

Then there exists $N_0 \in \mathbb{N}$ *such that*

$$\forall N \geq N_0 \; \forall i > N \; \exists j \leq N : t_i \in \mathcal{M}_{t_j}^N. \tag{6.5}$$

The restrictions on the functions cf and cnm are basically such that choice depends on $\langle U \rangle$, i. e., on the minimal generators of $\langle U \rangle$, but not on the concrete elements of U.

Proof. The statement of the theorem is obviously correct if $n = 0$. Assume that the theorem is correct if the number of variables is less than n. In the sequel, we use for $k \in \mathbb{N}$, $x \in X$, $d \in \mathbb{N}$ the following notation:

$$W_k := \{t_1, \ldots, t_k\},$$
$$W^{x,d} := \{t \in W \mid \deg_x t = d\},$$
$$W_k^{x,d} := W^{x,d} \cap W_k.$$

By Dickson's Lemma we find an index N_1 such that W_{N_1} generates the same monoid ideal in T as W. We conclude that $m := \textsc{cf}(X, W_{N_1}) = \textsc{cf}(X, W_k)$ and $Y_m^k = X$ for all $k > N_1$. If $m[X] = \langle W \rangle$ we are done. Otherwise let

$$s := \deg m - \deg(\gcd(W_{N_1})), \qquad (s > 0),$$
$$x := \textsc{cnm}(X, W_{N_1} \setminus m[X], m).$$

By the assumptions on cnm, we have $x = \textsc{cnm}(X, W_k \setminus m[X], m)$ for all $k > N_1$. Let $d := \min\{\deg_x t \mid t \in W_{N_1}\}$. Since all elements of $W^{x,d}$ have the same degree in x, we find by the induction hypothesis an index $N_2 \geq N_1$ such that for all $i, k \geq N_2$ with $i > k$ and $t_i \in W^{x,d}$ there exists $t \in W_k^{x,d}$ such that $t_i \in \mathcal{M}_t^k$. Note that by construction of the Sliced Division for $W_k^{x,d}$, $k > N_2$, the variable x will not be in Y_t^k for any $t \in W^{x,d}$. Since $\deg m - \deg(\gcd(W_k \setminus W^{x,d})) < s$ for all $k \geq N_1$, we can assume that we have already found $N_3 \geq N_1$ such that for all $i, k \geq N_3$, $i > k$, $t_i \in W \setminus W^{x,d}$ there exists $t \in W \setminus W^{x,d}$ with $t_i \in \mathcal{M}_t^k$. Putting both cases together, we see that (6.5) holds for $N_0 = \max(N_2, N_3)$. ☐

The importance of Theorem 6.6.2 lies in the fact that under mild assumptions on the choice functions of the Sliced Division algorithm one can select the prolongations in the involutive basis algorithm in any order and still get a terminating algorithm.

Corollary 6.6.3. *If the sub-algorithm* PartialDivision *of Algorithm 4 is a generalised Sliced Division algorithm with choice functions that satisfy the conditions of Theorem 6.6.2 then any selection strategy can be used in line 6 of Algorithm 4 and the algorithm still terminates.*

Proof. Let us denote by G^i, $\mathcal{M}^i$, Q^i, and C^i the value of G, $\mathcal{M}$, Q, and C, respectively, before the i-th iteration of the while loop (lines 5–12 of Algorithm 4). The leading terms of elements h that are added in each iteration in line 9 together with the leading terms of the input set F form a sequence W. By the previous theorem, there exists $N \in \mathbb{N}$ such that $\langle \operatorname{lt} G^N \rangle = \langle \operatorname{lt} G^i \rangle$ for any $i \geq N$ and the set $\operatorname{lt}(G^N)$ is $\mathcal{M}^N$-complete. Then also $\operatorname{lt}(G^N \setminus Q^N)$ is $\mathcal{M}^N$-complete and thus $G^i = G^N$ for any $i \geq N$ since in any following iteration the reduction in line 7 leads to zero. Since the set C^N is finite the execution of the while loop must terminate. $\square$

Remark 6.6.4. The Janet division does not satisfy the assumptions of Theorem 6.6.2. However, if $\text{CF}(Y, U)$ always returns an element of U' where U' is minimal with the property $\langle U \rangle = \langle U' \rangle$ and $\text{CNM}(Y, V, m)$ always returns the biggest possible variable (see **T2** on page 47), then such a Sliced Division algorithm enables us to select the prolongations in line 6 of Algorithm 4 in any order without losing termination.

6.7 Examples

In this section we demonstrate the advantages of using Sliced Divisions in involutive bases computations. Further, we show that not every Sliced Division is maximal with respect to the refinement relation. Finally, we give an example where a given order $\sqsubset$ prevents a maximal division from being "maximal" in an intuitive sense, and where the Sliced Division algorithm produces a division and (another) order $\sqsubset'$ which is intuitively "better" and also maximal.

Let us compare involutive basis computations with respect to several suitable partial divisions. We have selected the well-known Cyclic4 and Cyclic6 examples as they appear in many collections of polynomial data[1]. They can also be found in [GB98a].

Example 6.7.1 (Cyclic4). *The Cyclic-4-root system is given by the following four polynomials.*

$$\begin{aligned}
f_1 &= x_4 + x_3 + x_2 + x_1 \\
f_2 &= x_3x_4 + x_2x_3 + x_1x_4 + x_1x_2 \\
f_3 &= x_2x_3x_4 + x_1x_3x_4 + x_1x_2x_4 + x_1x_2x_3 \\
f_4 &= x_1x_2x_3x_4 - 1
\end{aligned}$$

The table below represents involutive basis computations with respect to various suitable partial divisions and the degree inverse lexicographical term order as introduced on page 8. We use a modified normal selection strategy in Algorithm 4, i. e., prolongations with a minimal leading term are selected first but prolongations that have been selected in one iteration will not be selected in future iterations until each prolongation from the set C of critical elements has at least been selected once. The computations have been performed by our software package CALIX *which implements Algorithm 4 in* ALDOR, *cf. Chapter 8.*

[1] for example: `http://www.symbolicdata.org`

Division	N	P	R	Z
Thomas Division	*98*	*481*	*1144*	*387*
Janet Division	*37*	*147*	*456*	*114*
Sliced Division	*7*	*27*	*146*	*21*

The last four columns represent

(N) the number of elements in the final basis,

(P) the number of prolongations that are reduced in line 7 of Algorithm 4,

(R) the number of involutive reduction steps of the form $f \to_{G,\mathcal{M}} g$*, and*

(Z) the number of reductions to zero.

We see that the number of elements in the final involutive basis crucially depends on the partial division used during the computation. Usually, an involutive basis has more elements than a corresponding reduced Gröbner basis, but in this example the involutive basis with respect to the Sliced Division is a reduced Gröbner basis. The last three columns demonstrate that also the work needed to compute an involutive basis depends on the partial division. In all cases, the Sliced Division algorithm leads to a better performance of the involutive basis algorithm due to the fact that it produces a partial division which "better covers" the leading term ideal of the intermediate basis, i. e., fewer non-multiplicative prolongations have to be treated than in the case of the Thomas or Janet division.

Example 6.7.2 (Cyclic6). *The Cyclic-6-root system is given by the following six polynomials.*

$$\begin{aligned}
f_1 &= x_6 + x_5 + x_4 + x_3 + x_2 + x_1 \\
f_2 &= x_5x_6 + x_4x_5 + x_3x_4 + x_2x_3 + x_1x_6 + x_1x_2 \\
f_3 &= x_4x_5x_6 + x_3x_4x_5 + x_2x_3x_4 + x_1x_5x_6 + x_1x_2x_6 + x_1x_2x_3 \\
f_4 &= x_3x_4x_5x_6 + x_2x_3x_4x_5 + x_1x_4x_5x_6 + x_1x_2x_5x_6 \\
&\quad + x_1x_2x_3x_6 + x_1x_2x_3x_4 \\
f_5 &= x_2x_3x_4x_5x_6 + x_1x_3x_4x_5x_6 + x_1x_2x_4x_5x_6 \\
&\quad + x_1x_2x_3x_5x_6 + x_1x_2x_3x_4x_6 + x_1x_2x_3x_4x_5 \\
f_6 &= x_1x_2x_3x_4x_5x_6 - 1
\end{aligned}$$

The involutive basis computations are again done with respect to various suitable partial divisions, the degree inverse lexicographical term order, and a modified normal selection strategy as described in the previous example. The computation with respect to the Thomas division could not be completed within one hour, but it can easily be shown that such a basis has 7044 elements. The notation of the columns is as in Example 6.7.1.

Division	N	P	R	Z
Janet Division	*840*	*7120*	*348672*	*6286*
modified Janet Division	*385*	*4939*	*194378*	*4113*
Sliced Division	*46*	*1049*	*46144*	*869*

Due to the fact that the Janet division is disjoint on any intermediate basis, the first line shows with 840 far more elements in the final basis than a Janet basis has if one starts the computation with the 45 elements of the reduced Gröbner basis of this system. Such a Janet basis has 385 elements and can be computed with Algorithm 4 by using a modified Janet division starting with the polynomials $f_1, \ldots, f_6$. *The modifications are such that certain inclusions of cones are allowed. This modified Janet division* $\mathcal{J}'$ *has been computed by a generalised Sliced Division algorithm using the sub-algorithms* $\mathrm{CNM}_{\mathcal{J}'} := \mathrm{CNM}_{\mathcal{J}}$ *and* $\mathrm{CF}_{\mathcal{J}'}$ *with* $\mathrm{CF}_{\mathcal{J}'}(Y, U) := \max_{\prec_{\mathrm{lex}}} U'$ *where* $U' \subseteq U \subseteq T$ *is minimal with the property* $\langle U' \rangle = \langle U \rangle$, *cf. Definition 6.4.5 and Definition 6.4.6. A computation with such a modified Janet division approximately corresponds to a computation of a* minimal *Janet basis by an algorithm of Gerdt and Blinkov [GB98b], but without application of criteria.*

Again we see that the "big" cones of a Sliced Division lead to fewer reductions and also to fewer elements in the final involutive basis.

Remark 6.7.3. In the next chapter we treat the application of criteria in the context of involutive bases and run both examples again with a modified Algorithm 4 that applies criteria to avoid unnecessary reductions.

Although the Sliced Division algorithm often returns a relatively "good" admissible partial division, the following example shows that not every Sliced Division is maximal with respect to the refinement relation.

Example 6.7.4. *Let* $U := \{xy, yz^2, z^4\}$. *In the Sliced Division algorithm there is a choice to assign multiplicative variables to the elements of* U. *Clearly, since* xy *is of smallest degree, it will always get the full set of multiplicative variables. According to Algorithm 5,* $V = \{yz^2, z^4\}$ *and* $\gcd V = z^2$. *So one can choose either* x *or* y *in line 8. If we choose* x *then* yz^2 *gets* $\{y, z\}$ *as multiplicative and* z^4 *gets only* z *as multiplicative. If, however, we choose* y *first, then the set* S *in that iteration consists only of the element* z^4 *so that this element gets* $\{x, z\}$ *as multiplicative. In the next iteration, the only choice in line 8 is* x *and we obtain* $\{y, z\}$ *as multiplicative variables for* yz^2.

According to the choice of x *or* y *in line 8 one can obtain two different divisions* $\mathcal{M} = (u[Y_u])_{u \in U}$ *and* $\mathcal{N} = (u[Z_u])_{u \in U}$ *whose sets of multiplicative variables are shown in the table below.*

u	Y_u	Z_u
xy	$\{x, y, z\}$	$\{x, y, z\}$
yz^2	$\{y, z\}$	$\{y, z\}$
z^4	$\{z\}$	$\{x, z\}$

Interestingly, for both possible choices the induced order $\sqsubset$ *is* $xy \sqsubset yz^2 \sqsubset z^4$. *Since* $\mathcal{N}$ *properly refines* $\mathcal{M}$, *this example shows that the Sliced Division algorithm is not optimal in the sense that it always returns a maximal division.*

Although the Sliced Division is not always maximal, it has the advantage that the admissibility order $\sqsubset$ need not be given in advance. The concept of a maximal admissible division depends on an order $\sqsubset$. In Apel's approach this order has to be given before one can determine a maximal division.

Example 6.7.5. *Take the set* $U := \{x^2, y^2, xy\}$ *with the terms ordered by* $x^2 \sqsubset y^2 \sqsubset xy$. *The multiplicative variables in the maximal admissible division on* $(U, \sqsubset)$ *are* $\{x, y\}$, $\{y\}$, *and* $\emptyset$, *for* x^2, y^2, *and* xy, *respectively.*

Although the division in the above example is maximal, one would hardly call it a "good" division. The order $\sqsubset$ prevents the assignment of any multiplicative variable to the term xy. Regardless of the choices made during the run of Algorithm 5 started on U from the previous example, the returned division would always prove U to be a complete set. If xy gets y as multiplicative in the example above, this is a Sliced Division and it is maximal. This division is admissible but with respect to another order, namely $x^2 \sqsubset' xy \sqsubset' y^2$. In fact, the Sliced Division algorithm can also be seen as a method to find a "good" order with respect to which the division is admissible.

Chapter 7

Criteria for Involutive Bases

For the Gröbner basis algorithm many improvements are known, see, for example, [Buc79], [GM88], [GMN+91]. Since Buchberger's algorithm and the involutive basis algorithm are very similar, a natural question is whether one can adapt such improvements also to the involutive basis algorithm. We treat here the application of criteria in order to avoid the reduction of certain prolongations for which one can conclude in advance that their reduction is superfluous.

In [GB98a] already appears a criterion for the involutive case which Gerdt and Blinkov derive from Buchberger's chain criterion, cf. Lemma 7.1.1. In this chapter, we present a theorem which leads to certain criteria that extend the results of [GB98a].

Further, we show that the application of our criteria entail termination of the involutive basis algorithm independent of the prolongation selection strategy. This is an extension of the result obtained in Section 6.6, since now we prove termination without the additional assumptions on the (intermediate) suitable partial division as imposed in Theorem 6.6.2.

7.1 Preliminaries

Recall that we have fixed an irreflexive term order $\prec$ and that for two non-zero polynomials $f, g \in \mathbb{K}[X]$ we use the abbreviation $[f, g] := \mathrm{lcm}(\mathrm{lt}\, f, \mathrm{lt}\, g)$. Furthermore, we use in this chapter the notation $\mathcal{S}(f, g)$ as defined in Definition 5.3.2.

The following lemma is just Buchberger's chain criterion.

Lemma 7.1.1. *[Buc79] Let $G \subseteq \mathbb{K}[X]$ be a finite set of non-zero polynomials. For any $f, g, p \in G$ it holds*

$$\mathcal{S}(f,p) \wedge \mathcal{S}(p,g) \wedge \mathrm{lt}\, p \trianglelefteq [f,g] \implies \mathcal{S}(f,g).$$

The proof of our main theorem is based on the following characterisation of Gröbner bases.

Theorem 7.1.2. *[Ape98b, Theorem 5.4] Let $G = \{g_1, \ldots, g_r\} \subseteq \mathbb{K}[X]$. G is a Gröbner basis of $I := \mathrm{Id}\,(G)$ if and only if $\mathfrak{F}^G_{\preceq t} = \{h \in I \mid h = 0 \vee \mathrm{lt}\, h \preceq t\}$ for all $t \in T$.*

Definition 7.1.3. [AH02] Let $G = \{g_1, \dots, g_r\} \subset \mathbb{K}[X]$ be a set of non-zero polynomials. We define a quasi order $\trianglelefteq_G$ on G by $f \trianglelefteq_G g$ if and only if there exist $c \in \mathbb{K}$ and $t \in T$ such that $g - ctf \in \mathfrak{F}^G_{\prec \operatorname{lt}(g)}$. If $f \trianglelefteq_G g$, we say that f is an **ancestor** of g. The quasi order $\trianglelefteq_G$ is a partial order if all leading terms of G are pairwise distinct.

Note that $f \trianglelefteq_G g$ implies $\operatorname{lt} f \trianglelefteq \operatorname{lt} g$ and $\mathcal{S}(f, g)$.

7.2 Involutive Criteria Theorem

The following theorem adds another equivalent condition to those presented in Theorem 5.3.3. We give the theorem first and use in the proof some lemmata which we present afterwards. Let us emphasise the fact that the lemmata do not reference any partial division. In fact, in Theorem 7.2.1 the partial division is only used to form the special type of S-polynomials that are treated by the involutive basis algorithm. The rest is an adaptation of criteria from the theory of Gröbner bases to the special situation of involutive bases. Theorem 7.2.1 slightly generalises Theorem 3.1 of [AH02] by weakening the requirement on the partial division from admissibility to cycle-freeness. Nevertheless Remark 3.3 in [AH02] already covers the full generality of the following theorem.

Theorem 7.2.1. *[AH02] Let $G \subseteq \mathbb{K}[X]$ be a set of non-zero monic polynomials and let $\mathcal{M} = (t[Y_t])_{t \in T}$ be a cycle-free and disjoint partial division on $\operatorname{lt} G$. Moreover, let $\blacktriangleleft$ be an arbitrary irreflexive linear order on G.*

For all $g \in G$ and $x \in X \setminus Y_{\operatorname{lt} g}$ let there exist $f \in G$ such that $\operatorname{lt}(xg) \in \mathcal{M}_{\operatorname{lt}(f)}$ and one of the following conditions holds.

1. $xg - \frac{\operatorname{lt}(xg)}{\operatorname{lt} f} f \in \mathfrak{F}^G_{\prec \operatorname{lt}(xg)}$.
2. *There exist $g', f' \in G$ such that*
$$\operatorname{lt} g' \trianglelefteq \operatorname{lt} g, \quad \operatorname{lt} f' \trianglelefteq \operatorname{lt} f, \quad [g', f'] = \operatorname{lt}(g'f') \tag{7.1}$$
and either
   (a) $\operatorname{lt} f \triangleleft [g, f]$ *or*
   (b) $\operatorname{lt} f = [g, f] \wedge \exists f'' \in G : f'' \trianglelefteq_G f \wedge [f'', f'] \triangleleft [g, f]$.
3. *There exist $g', f', p \in G$ such that*
$$\operatorname{lt} g' \trianglelefteq \operatorname{lt} g, \quad \operatorname{lt} f' \trianglelefteq \operatorname{lt} f, \quad [p, g'] \triangleleft [g, f], \quad [p, f'] \triangleleft [g, f] \tag{7.2}$$
and either
   (a) $\operatorname{lt} f \triangleleft [g, f]$ *or*
   (b) $\operatorname{lt} f = [g, f] \wedge \exists f'' \in G : f'' \trianglelefteq_G f \wedge [f'', f'] \triangleleft [g, f]$.
4. *There exist $g', h, h' \in G$, $y \in X \setminus Y_{\operatorname{lt} h}$ such that*
$$\operatorname{lt} g' \trianglelefteq \operatorname{lt} g, \quad \operatorname{lt} h' \trianglelefteq \operatorname{lt} h, \quad [g', h'] \triangleleft \operatorname{lt}(xg) = \operatorname{lt}(yh), \quad h \blacktriangleleft g. \tag{7.3}$$

Then G is an $\mathcal{M}$-involutive basis of $I = \operatorname{Id}(G)$.

Proof. Since for all $g \in G$ and $x \in X \setminus Y_{\operatorname{lt} g}$ there exists $f \in G$ with $\operatorname{lt}(xg) \in \mathcal{M}_{\operatorname{lt}(f)}$, the set $\operatorname{lt} G$ is not $\mathcal{M}$-extensible, and, therefore, $\mathcal{M}$-complete by Proposition 4.6.7.

It remains to show that G is a Gröbner basis of I. According to Theorem 7.1.2 it is equivalent to show $p \in \mathfrak{F}^G_{\preceq \operatorname{lt}(p)}$ for all $p \in I \setminus \{0\}$.

Suppose there exists $p \in I \setminus \{0\}$ with $p \notin \mathfrak{F}^G_{\preceq \operatorname{lt}(p)}$. Let $t \in T$ be the minimal (with respect to $\prec$) term such that there exists $p \in I$ satisfying $p \in \mathfrak{F}^G_{\preceq t} \setminus \mathfrak{F}^G_{\prec t}$ and $\operatorname{lt} p \prec t$.

Let us first introduce a few auxiliary notions. The family $L = ((h_i, g_i))_{i \in \mathbb{N}_k}$ is called a ***t*-representation** if $h_i \in \mathbb{K}[X] \setminus \{0\}$, $g_i \in G$, $\operatorname{lt}(h_i g_i) \preceq t$, and $\operatorname{lt} g_i \neq \operatorname{lt} g_j$ for all $i, j \in \mathbb{N}_k$ with $i \neq j$. We define two operators on such t-representations by

$$\Sigma(L) := \sum_{i=1}^{k} h_i g_i$$

and

$$\Gamma(L) := \{g \in G \mid \exists i \in \mathbb{N}_k : g = g_i \wedge \operatorname{lt}(h_i g_i) = t \wedge t \notin \mathcal{M}_{\operatorname{lt} g}\}.$$

We say that L is a t-representation of p if $p = \Sigma(L)$.

If L is a t-representation, $\Sigma(L) \in \mathfrak{F}^G_{\preceq t} \setminus \mathfrak{F}^G_{\prec t}$, and $\operatorname{lt} \Sigma(L) \prec t$ then $\Gamma(L) \neq \emptyset$.

Among all t-representations choose $L = ((h_i, g_i))_{i \in \mathbb{N}_k}$ such that

1. $p := \Sigma(L) \in \mathfrak{F}^G_{\preceq t} \setminus \mathfrak{F}^G_{\prec t}$, $\quad \operatorname{lt} p \prec t$, and
2. $\max_{\blacktriangleleft} \Gamma(L)$ is minimal with respect to $\blacktriangleleft$.

Without loss of generality, let g_1 be the maximal element of $\Gamma(L)$ with respect to $\blacktriangleleft$. From the definition of $\Gamma(L)$ it follows that there exists $x \in X \setminus Y_{\operatorname{lt} g_1}$ such that x divides $\operatorname{lt} h_1$. By $\mathcal{M}$-completeness there exists $f \in G$ such that $\operatorname{lt}(x g_1) \in \mathcal{M}_{\operatorname{lt}(f)}$. The polynomial $g = g_1$, its non-multiplicative variable x, and the polynomial f must satisfy one of the conditions 1–4. We show that each of these conditions implies $\mathcal{S}(g_1, f)$, i. e.,

$$\operatorname{spol}(g_1, f) = x g_1 - \frac{\operatorname{lt}(x g_1)}{\operatorname{lt} f} f \in \mathfrak{F}^G_{\prec \operatorname{lt}(x g_1)}. \tag{7.4}$$

For the first condition this is obvious. For conditions 2 and 3 we apply Lemma 7.2.2 and 7.2.3, respectively, in order to show $\mathcal{S}(g_1, f)$. Finally, if condition 4 applies then we must have $\mathcal{S}(h, f)$ for the element h from condition 4 since by construction of L we conclude $\operatorname{spol}(h, f) = yh - \frac{\operatorname{lt}(yh)}{\operatorname{lt} f} f \in \mathfrak{F}^G_{\prec \operatorname{lt}(yh)}$. Application of Lemma 7.2.4 shows $\mathcal{S}(g_1, f)$. From (7.4) follows now

$$\operatorname{lt}(h_1) g_1 = \frac{\operatorname{lt}(h_1 g_1)}{\operatorname{lt} f} f + \hat{h} \tag{7.5}$$

for a suitable polynomial $\hat{h} \in \mathfrak{F}^G_{\prec t}$. Substituting the left hand side of (7.5) by the right hand side in the sum $\sum_{i=1}^{k} h_i g_i$ yields a new t-representation $L' = ((h'_i, g'_i))_{i \in \mathbb{N}_{k'}}$ of p. Moreover, $\Gamma(L') \subseteq (\Gamma(L) \setminus \{g_1\}) \cup \{f\}$. Neither $t \in \mathcal{M}_{\operatorname{lt} f}$ nor $f \blacktriangleleft g_1$ can hold, since then L' would contradict the minimality assumptions on L.

If $t \notin \mathcal{M}_{\operatorname{lt} f}$ and $g_1 \blacktriangleleft f$ we can assume, without loss of generality, $g_1' = f$ and repeat the above arguments. For $x' \in X \setminus Y_{\operatorname{lt} g_1'}$ such that $x' \trianglelefteq \operatorname{lt} h_1'$ there exist $f' \in G$ and $\hat{h}' \in \mathfrak{F}^G_{\prec t}$ satisfying $\operatorname{lt}(x' g_1') \in \mathcal{M}_{\operatorname{lt} f'}$ and $\operatorname{lt}(h_1') g_1' = \frac{\operatorname{lt}(h_1' g_1')}{\operatorname{lt} f'} f' + \hat{h}'$. Hence, again we can construct a representation L'' of p where now f' instead of g_1' appears in the set $\Gamma(L'')$.

Iteration of the above process yields an $\mathcal{M}$-sequence $\operatorname{lt} g_1, \operatorname{lt} f, \operatorname{lt} f', \ldots$ which must be finite since $\mathcal{M}$ is cycle-free. Thus, the process eventually terminates with a t-representation L^* of p which contradicts the minimality assumptions on L. In summary, the supposition of the existence of $p \in I$ satisfying $p \neq 0$ and $p \notin \mathfrak{F}^G_{\preceq \operatorname{lt}(p)}$ must have been wrong and the assertion of the theorem follows. □

In condition 4 of Theorem 7.2.1, the prolongations (x, g) and (y, h) are only distinguished by means of the order $\blacktriangleleft$. A tie breaking condition is necessary since otherwise it would allow us to conclude $\mathcal{S}(g, f)$ by relying on $\mathcal{S}(h, f)$ and to conclude $\mathcal{S}(h, f)$ by relying on $\mathcal{S}(g, f)$ although none of them is true.

One has, however, complete freedom to choose such a tie breaking order $\blacktriangleleft$. In fact, the theorem still remains valid if $\blacktriangleleft$ is replaced by a family $(\blacktriangleleft_t)_{t \in T}$ of irreflexive linear orders on G and the relation $h \blacktriangleleft_{\operatorname{lt}(xg)} g$ is used in condition 4.

We now present the three lemmata that have been used in the proof of Theorem 7.2.1.

Lemma 7.2.2. *[AH02] Let $G \subseteq \mathbb{K}[X]$ be a set of non-zero monic polynomials with distinct leading terms. Let $g, f \in G$. For each $u \in T$ with $u \prec [g, f]$ and each $h \in \operatorname{Id}(G) \setminus \{0\}$ assume $h \in \mathfrak{F}^G_{\preceq u} \implies h \in \mathfrak{F}^G_{\preceq \operatorname{lt}(h)}$. Furthermore, assume condition 2 of Theorem 7.2.1. Then $\mathcal{S}(g, f)$ holds.*

Proof. Let $g, f, g', f' \in G$ be such that (7.1) holds. From $[g, g'] \prec [g, f]$ it follows $\mathcal{S}(g, g')$. In addition, $\mathcal{S}(f', f)$ by $[f, f'] \prec [g, f]$ in subcase 2a, and $\mathcal{S}(f', f'')$ due to $[f', f''] \prec [g, f]$ and $\mathcal{S}(f'', f)$ because $f'' \trianglelefteq_G f$ in subcase 2b. Moreover, $\mathcal{S}(g', f')$ according to Buchberger's coprime criterion (Criterion2 in [Buc85]). By repeated application of Lemma 7.1.1 we conclude $\mathcal{S}(g, f)$. □

Lemma 7.2.3. *[AH02] Let $G \subseteq \mathbb{K}[X]$ be a set of non-zero monic polynomials with distinct leading terms. Let $g, f \in G$. For each $u \in T$ with $u \prec [g, f]$ and each $h \in \operatorname{Id}(G) \setminus \{0\}$ assume $h \in \mathfrak{F}^G_{\preceq u} \implies h \in \mathfrak{F}^G_{\preceq \operatorname{lt}(h)}$. Furthermore, assume condition 3 of Theorem 7.2.1. Then $\mathcal{S}(g, f)$ holds.*

Proof. Let $g, f, g', f', p \in G$ be such that (7.2) holds. From $[g, g'], [g', p], [p, f'] \prec [g, f]$ follows $\mathcal{S}(g, g')$, $\mathcal{S}(g', p)$, and $\mathcal{S}(p, f')$. While we deduce $\mathcal{S}(f', f)$ from $[f', f] \prec [g, f]$ in subcase 3a, we obtain $\mathcal{S}(f', f'')$ from $[f', f''] \prec [g, f]$ and $\mathcal{S}(f'', f)$ by the assumption $f'' \trianglelefteq_G f$ in subcase 3b. Finally, repeated application of Lemma 7.1.1 yields $\mathcal{S}(g, f)$. □

Lemma 7.2.4. *[AH02] Let $G \subseteq \mathbb{K}[X]$ be a set of non-zero monic polynomials with distinct leading terms. Let $s \in T$ be a term and for each $u \in T$ with $u \prec s$ and each $h \in \operatorname{Id}(G) \setminus \{0\}$ assume $h \in \mathfrak{F}^G_{\preceq u} \implies h \in \mathfrak{F}^G_{\preceq \operatorname{lt}(h)}$. Let $g, g', h, h' \in G$ be such that $\operatorname{lt} g' \trianglelefteq \operatorname{lt} g$, $\operatorname{lt} h' \trianglelefteq \operatorname{lt} h$, and $[g', h'] \lhd [g, f] = [h, f] = s$.*

If $\mathcal{S}(h, f)$ holds then $\mathcal{S}(g, f)$ holds, too.

Proof. From $[g, g'], [g', h'], [h', h] \prec s$ follows $\mathcal{S}(g, g')$, $\mathcal{S}(g', h')$, and $\mathcal{S}(h', h)$. By Lemma 7.1.1, we deduce $\mathcal{S}(g, h)$. Moreover, $\mathcal{S}(h, f)$ by assumption. We conclude $\mathcal{S}(g, f)$ by Lemma 7.1.1. □

Lemmata 7.2.3, and 7.2.4 and their proofs remain valid almost literally for left ideals generated by G of algebras of solvable type. Lemma 7.2.2 is based upon Buchberger's coprime criterion and, therefore, cannot be transfered.

Hence, after removing condition 2 also Theorem 7.2.1 including its proof holds in the more general situation of left ideals generated by G of algebras of solvable type. Only a few standard adaptations, cf. [AL88] or [KRW90], are necessary in the proof.

Corollary 7.2.5. *Theorem 7.2.1 holds in the case of disjoint continuous involutive divisions as well as for disjoint admissible partial divisions.*

Proof. Easy consequence of Propositions 4.6.10 and 4.6.15. □

7.3 Criteria for the Involutive Basis Check

Theorem 7.2.1 describes a situation in which the partial division $\mathcal{M}$ does not change. This corresponds to an algorithm which tests whether a given set G of polynomials is an $\mathcal{M}$-involutive basis. Such a check algorithm can be seen as a sub-algorithm of Algorithm 4. Each time the $\mathcal{M}$-involutive basis check fails, the failure causing polynomial is added to the basis and a new check is started. We give a check algorithm as Algorithm 8. Conditions 2–4 of the Theorem 7.2.1 are used in order to create criteria for omitting the reduction of certain non-multiplicative prolongations. The criteria are reflected in the notion of a redundant prolongation which we define below.

Algorithm 8 InvolutiveBasisCheck

Call: $h = \textsc{InvolutiveBasisCheck}(G, \mathcal{M})$

Input: $G \subset K[X] \setminus \{0\}$ is a finite set of polynomials with pairwise distinct leading terms.
$\mathcal{M} = (t[Y_t])_{t \in \mathrm{lt}(G)}$ is a cycle-free essentially disjoint partial division on $\mathrm{lt}\, G$.

Output: If G is an $\mathcal{M}$-involutive basis then $h = 0$, otherwise $h \in \mathrm{Id}\,(G)$ and $\mathrm{lt}\, h \notin \langle \mathrm{lt}\, G \rangle_{\mathcal{M}}$.

$Q := \{ g \in G \mid \exists f \in G \setminus \{g\} : \mathrm{lt}\, g \in \mathcal{M}_{\mathrm{lt}\,(f)} \}$
$C := \{ (x, g) \mid g \in G \setminus Q \wedge x \in X \setminus Y_{\mathrm{lt}\, g} \} \cup \{ (1, q) \mid q \in Q \}$
<u>while</u> $C \neq \emptyset$ <u>do</u>
  Choose (t, g) from C; $C := C \setminus \{(t, g)\}$
  <u>if</u> (t, g) is not redundant with respect to $(G, \mathcal{M})$ <u>then</u>
    $h := \mathrm{NF}_{\mathcal{M}}(tg, G \setminus Q)$
    <u>if</u> $h \neq 0$ <u>then</u> <u>return</u> h
<u>return</u> 0

In the sequel we use the following assumptions and notations. Let $G = \{g_1, \dots, g_r\} \subseteq \mathbb{K}[X]$ be a finite set of non-zero polynomials with $\mathrm{lt}\, g \neq \mathrm{lt}\, g'$ for all $g, g' \in G$ and let $\mathcal{M} = (t[Y_t])_{t \in T}$ be a cycle-free and essentially disjoint partial division on $\mathrm{lt}\, G$. Let idx be a function which assigns to each element $g_i \in G$ its index i. Let $H \subseteq G$ be minimal with the property $\langle \mathrm{lt}\, H \rangle = \langle \mathrm{lt}\, G \rangle$. We introduce two functions anc, wanc : $G \to G$ such that for all $g \in G$ the following relations hold:

$$\begin{aligned}\operatorname{anc} g &\trianglelefteq_G g,\\ \operatorname{lt}(\operatorname{wanc} g) \trianglelefteq \operatorname{lt}(\operatorname{anc} g) &\trianglelefteq \operatorname{lt} g,\\ \operatorname{wanc} g &\in H.\end{aligned}$$

Both functions are combined to define a family of functions $\operatorname{anc}_s : G \to G$ by

$$\operatorname{anc}_s f := \begin{cases} \operatorname{wanc} f & \text{if } \operatorname{lt}(\operatorname{anc} f) \lhd s \\ \operatorname{anc} f & \text{otherwise}\end{cases}$$

for all $s \in T$ and $f \in G$. By $\operatorname{anc} f$ and $\operatorname{wanc} f$ we abbreviate the notions **ancestor** and **weak ancestor** of f, respectively. Usually, s will be equal to $[g, f]$ when calling $\operatorname{anc}_s f$ or $\operatorname{anc}_s g$ during the investigation of a non-multiplicative prolongation xg which is involutively top-reducible by $f \in G$.

Let us justify the definition of anc_s. Under the made assumption that the leading term of the weak ancestor divides the leading term of the ancestor, the objects $\operatorname{anc}_s f$ and $\operatorname{anc}_s g$ can always serve as f' and g' in conditions 2–4 of Theorem 7.2.1.

Definition 7.3.1. Let $G \subseteq \mathbb{K}[X]$ be a set of non-zero polynomials with pairwise distinct leading terms. Let $\mathcal{M} = (t[Y_t])_{t \in T}$ be an essentially disjoint partial division on $\operatorname{lt}(G)$. Furthermore let

$$\begin{aligned}Q &:= \left\{ g \in G \,\middle|\, \exists f \in G \setminus \{g\} : \operatorname{lt} g \in \mathcal{M}_{\operatorname{lt}(f)} \right\},\\ C &:= \left\{ (x, g) \mid g \in G \setminus Q \wedge x \in X \setminus Y_{\operatorname{lt} g} \right\} \cup \left\{ (1, q) \mid q \in Q \right\},\end{aligned}$$

and let $H \subseteq G$ be minimal with the property $\langle \operatorname{lt} H \rangle = \langle \operatorname{lt} G \rangle$.

A prolongation (t, g) is called **redundant** with respect to $(G, \mathcal{M})$ (written: $\mathcal{C}(t, g, G, \mathcal{M})$) if for $s := \operatorname{lt}(tg)$ there exists $f \in G \setminus Q$ such that $s \in \mathcal{M}_{\operatorname{lt}(f)}$ and

$$\mathcal{C}_0(g, s) \vee \mathcal{C}_1(g, f, s) \vee \mathcal{C}_2(g, f, s) \vee \mathcal{C}_3(g, f, s, H) \vee \mathcal{C}_4(g, s, G \setminus Q, \mathcal{M}) \tag{7.6}$$

holds where the respective criteria are defined by

$$\begin{aligned}\mathcal{C}_0(g, s) :&\iff \operatorname{lt}(\operatorname{anc}_s g) \lhd \operatorname{lt} g = s,\\ \mathcal{C}_1(g, f, s) :&\iff [\operatorname{anc}_s g, \operatorname{anc}_s f] \lhd s,\\ \mathcal{C}_2(g, f, s) :&\iff \operatorname{lt}(\operatorname{anc}_s g) \cdot \operatorname{lt}(\operatorname{anc}_s f) = s,\\ \mathcal{C}_3(g, f, s, H) :&\iff \exists h \in H : [h, \operatorname{anc}_s f] \lhd s \wedge [h, \operatorname{anc}_s g] \lhd s,\\ \mathcal{C}_4(g, s, G, \mathcal{M}) :&\iff \exists (y, h) \in X \times G : [\operatorname{anc}_s g, \operatorname{anc}_s h] \lhd s = \operatorname{lt}(yh) \notin \mathcal{M}_{\operatorname{lt}(h)}\\ &\qquad \wedge \operatorname{idx}(h) < \operatorname{idx}(g).\end{aligned}$$

Remark 7.3.2. The criteria $\mathcal{C}_2$, $\mathcal{C}_3$, and $\mathcal{C}_4$ correspond to conditions 2–4 of Theorem 7.2.1. Note that $\mathcal{M}$ is disjoint on $G \setminus Q$ in the above definition.

In (7.6), the criterion $\mathcal{C}_0$ can only be true if $\operatorname{lt}(tg) = \operatorname{lt}(g)$, i. e., $g \in Q$. The condition $\operatorname{lt}(\operatorname{anc}_s g) \lhd \operatorname{lt} g$ is only true if a proper ancestor is known. A justification that we need not reduce elements $g \in Q$ for which $\mathcal{C}_0(g, \operatorname{lt} g)$ holds, is given in the proof of Proposition 7.3.3.

Criterion $\mathcal{C}_1$ describes a special case of $\mathcal{C}_3$, namely when $h = \operatorname{anc}_s g$ or $h = \operatorname{anc}_s f$. A separate criterion $\mathcal{C}_1$ seems to be reasonable since in contrast to $\mathcal{C}_3$ it can be tested without searching through the minimal basis H, but still applies in a lot of cases.

In $\mathcal{C}_4$ we have chosen $h \blacktriangleleft g$ from condition 4 of Theorem 7.2.1 to be defined by $\operatorname{idx}(h) < \operatorname{idx}(g)$. This is no restriction for the involutive basis check since the index of the polynomials in G can be assigned arbitrarily before starting Algorithm 8, but proves to be useful in the involutive basis completion algorithm, cf. Remark 7.4.3.

A difference to the conditions of Theorem 7.2.1 is that we use a deterministic function anc_s to select one out of possibly several polynomials in places where f', g', and h' appear in Theorem 7.2.1. In order to fully exploit Theorem 7.2.1, one could test all possible pairs $(g', f'), (g', h') \in G \times G$ matching the requirements of the conditions. However, it seems that the overhead caused by the tests is larger than the effect gained by the corresponding extended criterion. Nevertheless, this question remains open for further investigation.

Proposition 7.3.3. *Algorithm 8 terminates and meets its specification.*

Proof. Termination is obvious. To prove correctness, let G and $\mathcal{M}$ be as in the input specification of Algorithm 8. If the algorithm returns $h \neq 0$, we conclude $h \in \operatorname{Id}(G)$ and $\operatorname{lt} g \notin \langle \operatorname{lt} G \rangle_{\mathcal{M}}$ immediately. Otherwise, let Q be as in line 1. Clearly, $G \setminus Q$ satisfies the assumptions of Theorem 7.2.1 and thus is an $\mathcal{M}$-involutive basis of $I := \operatorname{Id}(G \setminus Q)$. We show that $q \in I$ for all $q \in Q$. Assume for a contradiction that $Q \setminus I \neq \emptyset$ and among all elements of $Q \setminus I$ let q be such that $\operatorname{lt} q$ is minimal with respect to the term order $\prec$. The relation $\operatorname{lt}(\operatorname{anc} q) = \operatorname{lt} q$ cannot hold since then the prolongation (t, q) is not redundant by Definition 7.3.1, but Algorithm 8 yields $\operatorname{NF}_{\mathcal{M}}(q, G \setminus Q) = 0$, a contradiction.

If $\operatorname{lt}(\operatorname{anc} q) \lhd \operatorname{lt} q$ then $\operatorname{anc} q \in I$ by assumption. From $\operatorname{anc} q \trianglelefteq_G q$ follows $\mathcal{S}^G(\operatorname{anc} q, q)$. By the minimality condition on q, we conclude $q \in I$ and thus have a justification for criterion $\mathcal{C}_0$.

It follows $Q \subseteq I$ and thus G is an $\mathcal{M}$-involutive basis of $\operatorname{Id}(G) = I$. ☐

Gerdt and Blinkov were the first to give a criterion for the involutive basis algorithm. For completeness let us quote here a nearly literal copy of the corresponding theorem from [GB98a].

Theorem 7.3.4. *[GB98a] Let $G \subseteq \mathbb{K}[X]$ be a finite set of non-zero monic polynomials with pairwise distinct leading terms. Let $\mathcal{M}$ be a partial division which is disjoint on $\operatorname{lt} G$. Let (x, g) be a non-multiplicative prolongation of $g \in G$. Then $\operatorname{NF}_{\mathcal{M}}(xg, G) = 0$ if the following holds*

$$\forall h \in G\, \forall u \in T : u \operatorname{lt} h \prec \operatorname{lt}(xg) \implies \operatorname{NF}_{\mathcal{M}}(uh, G) = 0, \tag{7.7}$$

$$\exists f, f', g' \in G \begin{bmatrix} \operatorname{lt} f' \trianglelefteq \operatorname{lt} f, & \operatorname{lt} g' \trianglelefteq \operatorname{lt} g, \\ \operatorname{lt}(xg) \in \mathcal{M}_{\operatorname{lt}(f)}, & [f', g'] \prec \operatorname{lt}(xg), \\ \multicolumn{2}{c}{\operatorname{NF}_{\mathcal{M}}(\frac{\operatorname{lt} f}{\operatorname{lt} f'} \cdot f', G) = \operatorname{NF}_{\mathcal{M}}(\frac{\operatorname{lt} g}{\operatorname{lt} g'} \cdot g', G) = 0} \end{bmatrix}. \tag{7.8}$$

Let us compare the above criterion with criterion $\mathcal{C}_1$. While Gerdt and Blinkov refer to elements $f', g' \in G$ such that $\operatorname{lt} f' \trianglelefteq \operatorname{lt} f$, $\operatorname{lt} g' \trianglelefteq \operatorname{lt} g$, and $\operatorname{NF}_{\mathcal{M}}(\frac{\operatorname{lt} f}{\operatorname{lt} f'} \cdot f') = \operatorname{NF}_{\mathcal{M}}(\frac{\operatorname{lt} g}{\operatorname{lt} g'} \cdot g') = 0$, we use $f' = \operatorname{anc}_s f$ and $g' = \operatorname{anc}_s g$. Therefore, f' and g' need to satisfy only the weaker assumptions $\operatorname{lt} f' \trianglelefteq \operatorname{lt} f$ and

$\operatorname{lt} g' \trianglelefteq \operatorname{lt} g$ in most cases. In the exceptional case $\operatorname{lt} f = \operatorname{lt}(xg)$, we make the additional assumption $f' \trianglelefteq_G f$. Even this additional condition is still weaker than $\mathrm{NF}_{\mathcal{M}}(\frac{\operatorname{lt} f}{\operatorname{lt} f'} \cdot f') = 0$ unless the non-multiplicative prolongations are processed by increasing leading term, i. e., if the normal strategy is used, where both conditions become equivalent. The normal strategy is encoded in Theorem 7.3.4 by condition (7.7). Our formulation of the criteria does not require a particular strategy in which the prolongations are processed.

7.4 Involutive Basis Completion Algorithm

Until now we discussed the check algorithm for the involutive basis property. The computation of an involutive basis from an arbitrary given finite generating set can be done by application of a completion algorithm. Each time the check algorithm fails for the current basis, the basis is enlarged by the failure causing element and the check algorithm is started again until it ends successfully. Due to the necessary changes of the partial division caused by the basis enlargements, such a process involves reductions of prolongations that have already been reduced in a previous run of the check algorithm. It is a natural wish to a avoid as many of such repeated reductions as possible. Total avoidance of repeated reductions is possible if an order $\sqsubset$ is fixed during the whole computation, i. e., each subset U of leading terms of elements of an intermediate basis is ordered by the restriction of a fixed order $\sqsubset$ of T to U, and after each enlargement of the basis, a partial division is chosen which is admissible on $(U, \sqsubset)$, cf. [Ape98a]. Studying why multiple reduction of the same prolongation can be avoided, one observes that Apel's proof relies on Buchberger's chain criterion.

Before we present the InvolutiveBasis algorithm, let us deal with the question of avoiding repeated reductions of a prolongation. From Theorem 5.3.3, condition (iv) we learn that the reduction of a prolongation (x, g), $g \in G$, $x \in X \setminus Y_{\operatorname{lt} g}$ during the involutive basis check serves for the verification of two conditions, namely, first the existence of $f \in G$ such that $\operatorname{lt}(xg) \in \mathcal{M}_{\operatorname{lt}(f)}$ and second the validity of $\mathcal{S}(g, f)$.

Consider the run of Algorithm 4 (page 32). Let us denote by G^i, $\mathcal{M}^i$, Q^i, and C^i the values of G, $\mathcal{M}$, Q, and C, respectively, before the i-th iteration of the while loop (lines 5–12). A prolongation $(t, g) \in C^k$, $k \in \mathbb{N}$, is said to be **explicitly reduced** during the i-th iteration ($i < k$) and $\mathcal{M}^i$-top-reduced using $h \in G^i \setminus Q^i$ (written: $\mathcal{R}(t, g, h)$) if $\operatorname{lt}(xg) \in \mathcal{M}^i_{\operatorname{lt}(h)}$ and the reduction step $\mathrm{NF}_{\mathcal{M}^i}(tg, G^i \setminus Q^i)$ has indeed been executed in the i-th iteration. Suppose in the k-th iteration a prolongation $(t, g) \in C^k$ is selected in line 6 which has already been explicitly reduced during the i-th iteration using h, i. e., $\mathcal{R}(t, g, h)$ holds. Furthermore, suppose that now we find $f \in G^k \setminus Q^k$ with $\operatorname{lt}(tg) \in \mathcal{M}^k_{\operatorname{lt}(f)}$. This is exactly the situation where tg will repeatedly be reduced if the reduction cannot be avoided by means of criteria. The $\mathcal{M}^k$-involutive basis check requires the verification of $\mathcal{S}^{G^k}(g, f)$. Provided $\mathcal{S}^{G^k}(h, f)$ holds we can deduce $\mathcal{S}^{G^k}(g, f)$ from $\mathcal{S}^{G^k}(g, h)$ and Buchberger's chain criterion. Since the property $\mathcal{S}^{G^i}(g, h)$ is independent of the partial division and preserved under enlargement of the basis, we can conclude $\mathcal{S}^{G^k}(g, h)$ from the explicit reduction of tg in the i-th iteration. If $h = f$ or $h = \operatorname{anc} f$, the relation $\mathcal{S}^{G^k}(h, f)$ follows immediately. Let us describe this case by the predicate

$$\mathcal{C}_r(t, g, G, \mathcal{M}) :\iff \exists f \in G : \mathrm{lt}(tg) \in \mathcal{M}_{\mathrm{lt}(f)} \wedge (\mathcal{R}(t, g, f) \vee \mathcal{R}(t, g, \mathrm{anc}\, f)).$$

It defines an effective version of condition 1 of Theorem 7.2.1. Let $s := [g, f] = \mathrm{lt}\,(tg)$. Also if $[\mathrm{anc}_s\, h, \mathrm{anc}_s\, f] \lhd s$ holds, we easily conclude $\mathcal{S}^{G^k}(g, f)$. In fact, this situation can effectively be tested by a slightly generalised version of criterion $\mathcal{C}_3$ which takes advantage of the knowledge of $\mathcal{R}(t, g, h)$. Even the case $[\mathrm{anc}_s\, h, \mathrm{anc}_s\, f] = s = \mathrm{lt}\,(\mathrm{anc}_s\, h) \cdot \mathrm{lt}\,(\mathrm{anc}_s\, f)$ can be handled by a generalised version of criterion $\mathcal{C}_3$ incorporating $\mathcal{C}_2$. In the remaining case, we have to ensure that a (perhaps multiple) non-multiplicative prolongation of h is either explicitly reduced starting with f or the reduction is omitted by another reason than application of a Buchberger like criterion involving g. If $t = 1$ and $g \in Q^k$ and the prolongation $(1, g)$ was explicitly reduced in the i-th iteration then the updating process of the ancestor as described in the following Remark 7.4.1 ensures that $\mathcal{C}_0$ is applicable in the k-th iteration ($i < k$) even if $h \neq f$, and both $\mathcal{R}(t, g, f)$ and $\mathcal{R}(t, g, \mathrm{anc}\, f)$ are false.

Our discussion makes clear that our criteria, in particular $\mathcal{C}_r$, $\mathcal{C}_0$, and (a generalised version of) $\mathcal{C}_3$, avoid multiple reductions of a prolongation in a huge number of cases by exploiting the knowledge of previously performed reductions. For an another efficient criterion to avoid repeated reductions, see Section 8.3.

Another aspect of a "good" dynamical property of the criteria is given in Remark 7.4.3. We propose to use the index function in $\mathcal{C}_4$ as stated in Definition 7.3.1 in order to let the history of reductions decide which of several potential prolongations need not be reduced.

So in contrast to the multiple reduction avoidance rule from [Ape98a], we still do not lose the freedom of changing $\sqsubset$ during the completion process. Note also that we slightly relax Apel's requirement of an admissible partial division.

Remark 7.4.1. [AH02] The functions anc and wanc reflect parts of the history. Let F be as in the input specification of Algorithm 9. We initialise the functions by setting for all $g \in F$

$$\mathrm{anc}\, g := g \quad \text{and} \quad \mathrm{wanc}\, g := h$$

where $h \in F$ is such that $\mathrm{lt}\, h$ is a $\trianglelefteq$-minimal divisor of $\mathrm{lt}\, g$.

Consider the run of the i-th iteration of the <u>while</u> loop (lines 5–13) of Algorithm 9 and let us denote by G, $\mathcal{M}$, Q, and C the values of the respective variables before starting this iteration. The values $\mathrm{anc}\, g$ and $\mathrm{wanc}\, g$ are updated during k-th iteration of the run of the algorithm in the following situations.

(i) Assume that a prolongation $(1, q) \in C$ ($q \in Q$) was chosen in line 6 and it was $\mathcal{M}$-top-reduced using some polynomial $f \in G \setminus Q$ in line 8. Then for all $g \in G$ with $\mathrm{anc}\, g = q$ set $\mathrm{anc}\, g := \mathrm{anc}\, f$ and $\mathrm{wanc}\, g := \mathrm{wanc}\, f$.

    Moreover, if $\mathrm{lt}\,(\mathrm{anc}\, f) \prec \mathrm{lt}\,(\mathrm{anc}\, q) \neq \mathrm{lt}\, q$, set $\mathrm{anc}\, q := \mathrm{anc}\, f$ and $\mathrm{wanc}\, q := \mathrm{wanc}\, f$.

(ii) Assume that a prolongation $(x, g) \in C$ ($x \in X$, $g \in G \setminus Q$) was chosen in line 6 and $\mathcal{M}$-top-reduced by some polynomial $f \in G \setminus Q$ with $\mathrm{lt}(xg) = \mathrm{lt}\, f$ in line 8. If $\mathrm{anc}\, f = f$ then for all $h \in G$ with $\mathrm{anc}\, h = f$ set $\mathrm{anc}\, h := \mathrm{anc}\, g$ and $\mathrm{wanc}\, h := \mathrm{wanc}\, g$. If $\mathrm{lt}\,(\mathrm{anc}\, g) \prec \mathrm{lt}\,(\mathrm{anc}\, f) \neq \mathrm{lt}\, f$, set $\mathrm{anc}\, f := \mathrm{anc}\, g$ and $\mathrm{wanc}\, f := \mathrm{wanc}\, g$.

Algorithm 9 InvolutiveBasis

Call: $G = \textsc{InvolutiveBasis}(F)$

Input: $F \subset \mathbb{K}[X] \setminus \{0\}$ is a finite set of polynomials with pairwise distinct leading terms.

Output: G is an involutive basis of the ideal of F in $\mathbb{K}[X]$.

1: $G := F$
2: $\mathcal{M} := (t[Y_t])_{t \in \operatorname{lt}(G)} := \textsc{PartialDivision}(\operatorname{lt} G)$
3: $Q := \{ g \in G \mid \exists f \in G \setminus \{g\} : \operatorname{lt} g \in \mathcal{M}_{\operatorname{lt}(f)} \}$
4: $C := \{(x, g) \mid g \in G \setminus Q \wedge x \in X \setminus Y_{\operatorname{lt} g}\} \cup \{(1, q) \mid q \in Q\}$
5: <u>while</u> $C \neq \emptyset$ <u>do</u>
6: Choose (t, g) from C such that $\operatorname{lt}(tg)$ is $\prec$-minimal; $C := C \setminus \{(t, g)\}$
7: <u>if</u> not $\mathcal{C}_r(t, g, G \setminus Q, \mathcal{M})$ and not $\mathcal{C}(t, g, G, \mathcal{M})$ <u>then</u>
8: $h := \operatorname{NF}_{\mathcal{M}}(tg, G \setminus Q)$
9: <u>if</u> $h \neq 0$ <u>then</u>
10: $G := G \cup \{h\}$
11: $\mathcal{M} := (t[Y_t])_{t \in \operatorname{lt}(G)} := \textsc{PartialDivision}(\operatorname{lt} G)$
12: $Q := \{ g \in G \mid \exists f \in G \setminus \{g\} : \operatorname{lt} g \in \mathcal{M}_{\operatorname{lt}(f)} \}$
13: $C := \{(x, g) \mid g \in G \setminus Q \wedge x \in X \setminus Y_{\operatorname{lt} g}\} \cup \{(1, q) \mid q \in Q\}$
14: <u>return</u> G

(iii) Assume that the prolongation $(t, g) \in C$ was $\mathcal{M}$-top-irreducible in line 8. Then its remainder h gets $\operatorname{anc} h := \operatorname{anc} g$ and $\operatorname{wanc} h := \operatorname{wanc} g$.

(iv) Assume that the prolongation $(t, g) \in C$ was $\mathcal{M}$-top-reducible in line 8. If the remainder h is non-zero then assign $\operatorname{anc} h := h$ and $\operatorname{wanc} h := f$ where $f \in G \cup \{h\}$ is such that $\operatorname{lt} f$ is a $\trianglelefteq$-minimal divisor of $\operatorname{lt} h$. Note, if the normal strategy is used in Algorithm 9, it always holds $f = h$.

The updates of anc and wanc described in (i) and (ii) analogously apply to Algorithm 8.

The idx function is updated so that each new element h in line 10 gets a bigger index than any of the elements in G in order to simulate the age of a polynomial.

Theorem 7.4.2. *Algorithm 9 terminates and fulfils its specification if for each set G which occurs during its run,* $\textsc{PartialDivision}(\operatorname{lt} G)$ *is a suitable partial division on* $\operatorname{lt} G$.

Proof. Termination follows as in the proof of Theorem 5.3.8. Correctness is a consequence of Proposition 7.3.3 and the remarks given before the introduction of the criterion $\mathcal{C}_r$ on page 70. □

Remark 7.4.3. As stated in Theorem 7.2.1, any order $\blacktriangleleft$ can be taken in criterion $\mathcal{C}_4$, for example, $g \blacktriangleleft g' :\Longleftrightarrow \operatorname{lt} g \prec \operatorname{lt} g'$ would be possible. However, having in mind not only the involutive basis check but also the involutive basis completion algorithm, it is preferable to use the "age" or "index relation" for $\blacktriangleleft$, i.e., $g_i \blacktriangleleft g_j :\Longleftrightarrow i < j$, where the elements of G are enumerated according to insertion time in G. The freedom to choose $\blacktriangleleft$ allows to circumvent the following situation. During a completion process, it may happen that a non-multiplicative prolongation (x, g) is explicitly reduced because its reduction preventing basis polynomial h is not yet part of the basis at this time. A good criterion could and should avoid this situation by preventing the reduction of the other prolongation (y, h) instead. This behaviour is ensured by using the age relation as $\blacktriangleleft$ in a criterion derived from condition 4 of Theorem 7.2.1.

Obviously, Theorem 7.2.1 still remains valid if the linear order $\blacktriangleleft$ is replaced by a family $(\blacktriangleleft_t)_{t\in T}$ of linear orders on G and the condition $h \blacktriangleleft_{\mathrm{lt}\,(xg)} g$ is used in condition 4. We do not exploit this fact so far. However, a possible improvement based on this observation would be to deviate from the age relation by making elements h small with respect to $\blacktriangleleft_{\mathrm{lt}\,(yh)}$ in case the reduction of the non-multiplicative prolongation (y, h) can be avoided according to one of the other criteria $\mathcal{C}_2$ or $\mathcal{C}_3$.

7.5 Examples

In this section we start by giving two detailed examples which demonstrate that our criteria $\mathcal{C}_3$ and $\mathcal{C}_4$ are not covered by the other criteria. Then we re-compute an example from Chapter 5 and show explicitly where our criteria apply.

It is clear that the application of criteria becomes more powerful the longer the polynomials are that would be involved in a reduction. Nevertheless, for the purpose of demonstration, we have chosen monomials in Examples 7.5.1 and 7.5.2. In both examples we use $X = \{x, y, z, t\}$ and a degree lexicographical term order refining $x \succ y \succ z \succ t$ as well as the Janet division with respect to this order on the variables.

In all examples we apply the normal strategy as specified in line 6 of Algorithm 9.

Example 7.5.1 (Application of $\mathcal{C}_3$). *[AH02] The polynomials $f_1 = x^2z$, $f_2 = xyzt$, and $f_3 = xy^2t$ form already a reduced Gröbner basis. We want to compute a Janet basis of the ideal generated by $\{f_1, f_2, f_3\}$. At the beginning, the above polynomials have $\{x, y, z, t\}$, $\{z, t\}$, and $\{y, z, t\}$ as their respective Janet-multiplicative variables. The next 2 prolongations, namely yf_2 and xf_2 reduce to zero by zf_3 and ytf_1, respectively. The remaining prolongation $f_4 := xf_3 = x^2y^2t$ is irreducible. After adding f_4 to the basis, y becomes non-multiplicative for f_1 and the other multiplicative variables remain unchanged. We have to check all prolongations again. According to the normal strategy, the next prolongation is $f_5 := yf_1$, which is irreducible. Again we have to consider all prolongations. Because of $\mathcal{C}_1(f_1, f_5, x^2yz)$ the prolongation yf_1 need not be reduced. Since $\mathcal{R}(y, f_2, f_3)$ and the multiplicative variables of f_3 have not changed, the (repeated) reduction of yf_2 is avoided. We can also avoid to reduce xf_2, because $\mathcal{C}_r(x, f_2, \{f_1, \ldots, f_5\}, \mathcal{J})$ follows from the fact that $\mathcal{R}(x, f_2, f_1)$ and $f_1 = \mathrm{anc}\, f_5$ are true, and $\mathrm{lt}\, f_5$ is a $\mathcal{J}$-divisor of $\mathrm{lt}\,(xf_2)$ where $\mathcal{J}$ is the Janet division on $\{f_1, \ldots f_5\}$. The reduction of the prolongation xf_3 is avoided, because $f_4 = \mathrm{anc}\, f_3$ and, thus, $\mathcal{C}_1(f_3, f_4, x^2y^2t)$ holds. There is only one prolongation left, namely $f_6 := yf_5 = x^2y^2z$ which is irreducible and thus added to the basis. All prolongations that have been considered before are again avoided by the same reasons as above. The prolongation yf_5 will not be reduced, because $f_5 = \mathrm{anc}\, f_6$ and, thus, $\mathcal{C}_1(f_5, f_6, x^2y^2z)$ holds. The prolongation zf_4 reduces to zero by tf_6. Its reduction cannot be avoided by the Gerdt/Blinkov version of Buchberger's chain criterion, cf. [GB98a]. Let $s := x^2y^2zt = \mathrm{lt}(zf_4)$. Also $\mathcal{C}_1(f_4, f_6, s)$ and*

$\mathcal{C}_2(f_4, f_6, s)$ are false. However, $\operatorname{anc} f_4 = f_3$ *and* $\operatorname{anc} f_6 = f_1$*, and there is a polynomial, namely f_2, with $[f_1, f_2] = x^2yzt \lhd s$, $[f_3, f_2] = xy^2zt \lhd s$. Therefore, $\mathcal{C}_3(f_4, f_6, s, \{f_1, \ldots, f_6\})$ is true and the reduction of zf_4 can be avoided.*

Example 7.5.2 (Application of $\mathcal{C}_4$). *[AH02] We check the Janet basis property for the polynomials $f_1 = xyz$, $f_2 = yt$, $f_3 = zt$, $f_4 = xf_2 = xyt$ and $f_5 = xf_3 = xzt$. The Janet-multipliers are then $\{x, y, z, t\}$, $\{y, z, t\}$, $\{z, t\}$, $\{x, y, t\}$, and $\{x, z, t\}$, respectively. The prolongations yf_3, xf_3, xf_2, all reduce to zero. Consider the prolongations zf_4 and yf_5. Both are involutively top-reducible by f_1. Now, for f_4 and f_5 the only proper divisibility relations are $f_2 \lhd f_4$ and $f_3 \lhd f_5$, but for any $i > 1$ we have $[f_1, f_i] = xyzt$. Therefore, neither of the criteria $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$ is applicable for zf_4 or for yf_5. However, by application of criterion $\mathcal{C}_4$ we need only reduce zf_4 but not yf_5.*

Example 7.5.3. *Let us use the same notation as in Example 5.2.1 and let us start an involutive basis computation of the polynomials $\{f_1, f_2, f_3\}$. We apply Algorithm 9 where in place of* PartialDivision$(\operatorname{lt} G)$ *in lines 2 and 11 we use* SlicedDivision$(X, \operatorname{lt} G)$*. The trace of the computation is given in the same way as in Example 5.2.1 by the following table. We also show the prolongations that are re-considered by Algorithm 9 and give in the last column a reason why their re-reduction can be avoided.*

Name	*Representation*	*Mult.*	*Polynomial*
f_1		x, y, z	$x^8y^5z^2 - x^4y^3z^3 + x^4y^3 - yz$
f_2		$x, \quad z$	$x^8y^4z^4 - x^4y^4z^3 + x^4y^2z^2 - y^2z$
f_3		y, z	$x^6y^4z^9 - x^6y^2z^{11} + x^2y^2z^7 - x^2z^9$
f_4	$yf_2 - z^2f_1$	x, y, z	$x^4y^5z^3 - x^4y^3z^5 + y^3z - yz^3$
f_1	*new multipliers*	x, y	$x^8y^5z^2 - x^4y^3z^3 + x^4y^3 - yz$
f_3	*new multipliers*	z	$x^6y^4z^9 - x^6y^2z^{11} + x^2y^2z^7 - x^2z^9$
f_5	$zf_1 - x^4f_4$	$x, \quad z$	$x^8y^3z^5 - x^4y^3z^4 + x^4yz^3 - yz^2$
	zf_1		*avoided by* $\mathcal{R}(z, f_1, f_4)$
	$yf_5 - zf_2$		0
	$yf_2 - x^4zf_4$		0 *(repeated)*
	$yf_3 - x^2z^6f_4$		0
f_6	xf_3	z	$x^7y^4z^9 - x^7y^2z^{11} + x^3y^2z^7 - x^3z^9$
	zf_1		*avoided by* $\mathcal{R}(z, f_1, f_4)$
	yf_5		*avoided by* $\mathcal{R}(y, f_5, f_2)$
	yf_2		*avoided by* $\mathcal{R}(y, f_2, f_4)$
	yf_3		*avoided by* $\mathcal{R}(y, f_3, f_4)$
	xf_3		*redundant by* $\mathcal{C}_1(f_3, f_6, x^7y^4z^9)$
	yf_6		*redundant by* $\mathcal{C}_1(f_6, f_4, x^7y^5z^9)$
f_7	$xf_6 - z^5f_2$	$x, \quad z$	$x^8y^2z^{11} - x^4y^4z^8 + x^4z^9 - y^2z^6$
	zf_1		*avoided by* $\mathcal{R}(z, f_1, f_4)$
	yf_5		*avoided by* $\mathcal{R}(y, f_5, f_2)$
	yf_2		*avoided by* $\mathcal{R}(y, f_2, f_4)$
	yf_3		*avoided by* $\mathcal{R}(y, f_3, f_4)$
	xf_3		*redundant by* $\mathcal{C}_1(f_3, f_6, x^7y^4z^9)$
	yf_6		*redundant by* $\mathcal{C}_1(f_6, f_4, x^7y^5z^9)$
	xf_6		*avoided by* $\mathcal{R}(x, f_6, f_2)$
	$yf_7 - z^6f_5$		0

The fact that a Sliced Division is used, leads to fewer elements in the final involutive basis. As in Example 4.4.8, we show in Figure 7.1 the cone decomposition of the leading term ideal each time the basis is enlarged by a new element.

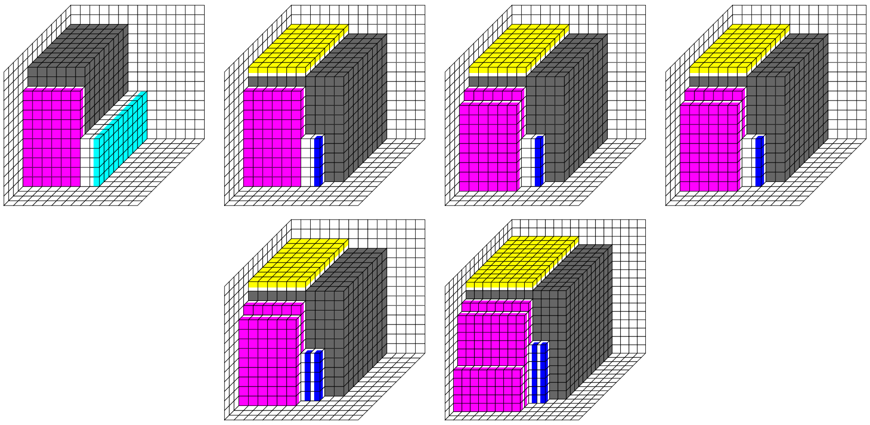


Figure 7.1: Cone decomposition corresponding to the Involutive Basis Computation of Example 7.5.3

Remark 7.5.4. In both Example 5.2.1 and Example 7.5.3 seven normal form computations are performed. However, during the latter computation the prolongation (y, f_2) is unnecessarily reduced twice, first by the polynomial f_1 and then by f_4. The criteria used in Algorithm 9 are not strong enough to avoid this repeated reduction of (y, f_2). As we have already mentioned on page 71, a generalised version of criterion $\mathcal{C}_3$ detects additional unnecessary repeated reductions. In the above example, at the time when (y, f_2) is selected for the second time, we find $h := f_1$ with $[h, f_4] = x^8y^5z^3 \lhd x^8y^5z^4 = \mathrm{lt}(yf_2) =: s$ as the only polynomial to satisfy the first part of criterion $\mathcal{C}_3$. Unfortunately, we have $[f_2, f_1] = s$ and thus criterion $\mathcal{C}_3$ is false. However, the knowlegde of $\mathcal{R}(y, f_2, f_1)$ is sufficient to avoid the reduction of (y, f_2) in this case.

7.6 Involutive Criteria vs. Buchberger's Criteria

Buchberger's algorithm checks whether a given set F of polynomials is a Gröbner basis by forming S-polynomials of all possible pairs of elements of F and for each such S-polynomial either avoids a reduction by means of certain criteria or performs a normal form computation to check zero-reduction explicitly, [Buc85].

In order to compare the involutive criteria with Buchberger's criteria applied in the Gröbner basis algorithm, we must explain how an S-polynomial of two given polynomials can occur and when it is *redundant* during the involutive basis algorithm. We start with an informal explanation of the correspondence between S-polynomials and prolongations and give a formal definition of the notion *redundant* afterwards.

S-polynomials are formed during the involutive basis algorithm (cf. Algorithm 9 on page 72) by multiplying basis elements by non-multiplicative variables until the product becomes reducible. At the time when an $\mathcal{M}$-top-reduction step of a prolongation is performed, which corresponds to an execution of line 6 of Algorithm 9, an S-polynomial is explicitly computed and eventually reduced. For the sake of simplicity, let us neglect for a while a possible tail reduction in Algorithm 9. Assume that there are two polynomials g' and f'. In order to form their S-polynomial, we usually have to compute a series of non-multiplicative prolongations $g_1 := y'g'$, $g_2 := y_1 g_1$, $g_3 := y_2 g_2$, etc. (and analogously for f') so that at some iteration of Algorithm 9 we arrive at two polynomials $g = g_k$ and $f = f_l$ with $\operatorname{lt}(xg) \in \mathcal{M}_{\operatorname{lt}(f)}$ where $\mathcal{M}$ is the partial division which is used in this iteration and x is a non-multiplicative variable for g with respect to $\mathcal{M}$. If the reduction of (x, g) cannot be avoided by means of criteria, this prolongation is explicitly reduced in line 6 of Algorithm 9. Note that the $\mathcal{M}$-top-reduction step creates $\operatorname{spol}(g, f)$ and equally $\operatorname{spol}(g', f')$. Additionally, we have $\operatorname{spol}(g', f') = \operatorname{spol}(g_i, f_j)$ for any g_i and f_j that originate from g' and f', respectively. In case we admit tail reductions, it still holds the weaker relation

$$\operatorname{spol}(g', f') - \operatorname{spol}(g_i, f_j) \in \mathfrak{F}^G_{\prec[g', f']}. \tag{7.9}$$

Since we apply criteria, not every possible S-polynomial is explicitly computed during the run of Algorithm 9. As the example below demonstrates, there are cases where an S-polynomial of two polynomials is never computed in an involutive basis algorithm even if no criteria are applied. Such an automatic removal of unnecessary reductions comes from the cone decomposition of the leading term ideal, i. e., the essentially disjoint partial division $\mathcal{M}$.

Example 7.6.1. *Take $X = \{x, y\}$, $f_1 = x^7 y$, $f_2 = xy^7$, and $f_3 = x^2 y^2$ where $\{x\}$, $\{y\}$, and $\{x, y\}$ are the corresponding sets of multiplicative variables. In order to check that $\{f_1, f_2, f_3\}$ is a Gröbner basis, we must test the S-polynomials of the S-pairs $\{f_1, f_2\}$, $\{f_1, f_3\}$, and $\{f_2, f_3\}$. The S-pair $\{f_1, f_2\}$ can be removed from the pair list in Buchberger's algorithm, since the chain criterion is applicable. In Algorithm 9 there are only the two prolongations (y, f_1) and (x, f_2) whose reduction leads to the computation of* $\operatorname{spol}(f_1, f_3)$ *and* $\operatorname{spol}(f_2, f_3)$*, respectively. An S-polynomial corresponding to $\{f_1, f_2\}$ does never occur in Algorithm 9.*

The following example shows that instead of none there might be several prolongations corresponding to one S-pair, or, more precisely, corresponding to several S-pairs which have "nearly" (in the sense of (7.9)) the same S-polynomial.

Example 7.6.2. *Let $X = \{x, y, z\}$ and $G = \{f_1, \ldots, f_5\}$. The partial division $\mathcal{M}$ is given in such a way that f_1 has all variables as multiplicative and for the other polynomials only z is multiplicative. Assume* $\operatorname{lt} f_1 = x^2 y^2$, $\operatorname{lt} f_2 = xz$, $\operatorname{lt} f_3 = xyz$, $\operatorname{lt} f_4 = x^2 yz$, *and* $\operatorname{lt} f_5 = xy^2 z$*. Let the relations* $f_2 = \operatorname{anc} f_3$ *and* $f_3 = \operatorname{anc} f_4 = \operatorname{anc} f_5$ *be given so that f_3 can be thought of as being obtained by prolongation from f_2, and f_4, f_5 by prolongation from f_3. Consider the S-pairs $\{f_2, f_1\}$ and $\{f_3, f_1\}$. Because of the above ancestor relations, there are two prolongations, namely (y, f_4) and (x, f_5), which lead to the computation of* $\operatorname{spol}(f_3, f_1)$ *at the $\mathcal{M}$-top-reduction step using f_1. Equally,* $\operatorname{spol}(f_2, f_1)$ *corresponds to the reduction of the above prolongations. Note that we have*

$$[f_2, f_1] = [f_3, f_1] = \operatorname{lt}(yf_4) = \operatorname{lt}(xf_5) = x^2 y^2 z =: s$$

and that

$$\operatorname{spol}(f_2, f_1) - \operatorname{spol}(f_i, f_1) \in \mathfrak{F}^G_{\prec s}$$

for $i = 3, 4, 5$.

The previous example shows that the relation between S-polynomials and and their respective prolongations is not functional. Note, however, that criterion $\mathcal{C}_4(f_5, s, G, \mathcal{M})$ holds. Application of $\mathcal{C}_4$ ensures that at most one prolongation is explicitly reduced in the above situation.

Let us derive a "good" definition for the correspondence between S-polynomials and respective prolongations. We consider a situation at the beginning of one iteration of the while loop of Algorithm 9, i. e., a finite set $G \subseteq \mathbb{K}[X] \setminus \{0\}$ of polynomials with distinct leading terms and a suitable partial division $\mathcal{M} = (t[Y_t])_{t \in \operatorname{lt} G}$ on $\operatorname{lt} G$ are given. We can assume that all elements of G are monic. Furthermore, let

$$
\begin{aligned}
Q &:= \left\{ g \in G \,\middle|\, \exists f \in G \setminus \{g\} : \operatorname{lt} g \in \mathcal{M}_{\operatorname{lt}(f)} \right\}, \\
C &:= \{ (x, g) \mid g \in G \setminus Q \wedge x \in X \setminus Y_{\operatorname{lt} g} \} \cup \{ (1, q) \mid q \in Q \},
\end{aligned}
$$

and let there be two functions anc and wanc with the properties given in Section 7.3. We additionally require

$$
\forall g \in G : \operatorname{anc}(\operatorname{anc} g) = \operatorname{anc} g. \tag{7.10}
$$

The latter condition basically says that the ancestor function always returns a "minimal" (known) ancestor. By using the update process of Remark 7.4.1, a careful implementation of Algorithm 9 can easily meet this specification, cf. Section 8.2.

Although in Example 7.6.2 it certainly holds $f_3 \trianglelefteq_G f_4$ and $f_2 \trianglelefteq_G f_4$ and one might guess that such a description seems natural for the above situation, the ancestor relation cannot be used for two reasons. First, it is not related to an actual execution of Algorithm 9 and second, it might include cases that we do not want in the correspondence. In order to see this, consider Example 7.6.2 and assume that we have another ancestor function which just tells us that $f_i = \operatorname{anc} f_i$ for $i \in \mathbb{N}_5$. Since we have not changed the polynomials, the relations $f_3 \trianglelefteq_G f_4$ and $f_2 \trianglelefteq_G f_4$ still hold, but in Algorithm 9 we have no way to recognise this except by actually computing and reducing the respective S-polynomials. The ancestor function is the only information about the prolongation history that we can access in Algorithm 9.

Assume there exists a corresponding prolongation $(t, g) \in C$ to the S-pair $\{g', f'\} \subseteq G$. We are looking for conditions to specify such a prolongation. First of all, (t, g) must be $\mathcal{M}$-top-reducible by some polynomial $f \in G \setminus Q$.

The requirement $g' = \operatorname{anc} g$ is too strong since this relation might not hold as $g' = f_2$, $g = f_4$, $f = f_1$ in Example 7.6.2 shows. Due to polynomials in an involutive basis which appear in addition to polynomials in a (reduced) Gröbner basis we must relax the above condition to $\operatorname{anc} g' = \operatorname{anc} g$. A polynomial g' for which $g' = \operatorname{anc} g'$ holds is either an element of the initial basis or appears in the involutive basis algorithm as a non-zero remainder of an explicit reduction of a prolongation that involves a top-reduction. For S-pairs $\{g', f'\}$ that actually appear also in Buchberger's algorithm, it would suffice to take $g' = \operatorname{anc} g$, because then we naturally have $g' = \operatorname{anc} g'$ and $f' = \operatorname{anc} f'$.

For the treatment below, it is convenient to introduce an auxiliary ancestor relation which depends on an ancestor function defined on G.

Definition 7.6.3. Let $g', g \in G$. The relation $\trianglelefteq_G^{\mathrm{anc}}$ is defined on G by

$$g' \trianglelefteq_G^{\mathrm{anc}} g :\Longleftrightarrow \operatorname{lt} g' \trianglelefteq \operatorname{lt} g \wedge \operatorname{anc} g' = \operatorname{anc} g. \tag{7.11}$$

The relation $g' \trianglelefteq_G^{\mathrm{anc}} g$ encodes the notion that g can be thought of as being obtained from g' by a series of (non-multiplicative) prolongations and tail reductions.

Lemma 7.6.4. *For any two polynomials $g, g' \in G$ holds*

$$g' \trianglelefteq_G^{\mathrm{anc}} g \implies g' \trianglelefteq_G g.$$

Proof. Let $g, g' \in G$ such that $\operatorname{lt} g' \trianglelefteq \operatorname{lt} g$ and $g^* := \operatorname{anc} g' = \operatorname{anc} g$. By the definitions of anc and $\trianglelefteq_G$ follows $g - \frac{\operatorname{lt} g}{\operatorname{lt} g^*} g^* \in \mathfrak{F}^G_{\prec \operatorname{lt}(g)}$ and $g' - \frac{\operatorname{lt} g'}{\operatorname{lt} g^*} g^* \in \mathfrak{F}^G_{\prec \operatorname{lt}(g')}$. Therefore,

$$g - \frac{\operatorname{lt} g}{\operatorname{lt} g'} g' = \left(g - \frac{\operatorname{lt} g}{\operatorname{lt} g^*} g^*\right) - \frac{\operatorname{lt} g}{\operatorname{lt} g'}\left(g' - \frac{\operatorname{lt} g'}{\operatorname{lt} g^*} g^*\right) \in \mathfrak{F}^G_{\prec \operatorname{lt}(g)}$$

and the claim of the lemma follows. □

Definition 7.6.5. Let $g', f' \in G$. Define the predicate $\mathcal{A}(g', f', t, g, f)$ by

$$(t, g) \in C \ \wedge \ f \in G \setminus Q \ \wedge \ g' \trianglelefteq_G^{\mathrm{anc}} g \ \wedge \ f' \trianglelefteq_G^{\mathrm{anc}} f \ \wedge \ \operatorname{lt}(tg) = [g', f'] \in \mathcal{M}_{\operatorname{lt}(f)}.$$

If t, g, f with $\mathcal{A}(g', f', t, g, f)$ or $\mathcal{A}(f', g', t, g, f)$ do not exist, we say that the S-pair $\{g', f'\}$ is **automatically omitted** with respect to $(G, \mathcal{M})$. The S-pair $\{g', f'\}$ is said to be **redundant** with respect to $(G, \mathcal{M})$ if for all t, g, f such that $\mathcal{A}(g', f', t, g, f)$ or $\mathcal{A}(f', g', t, g, f)$ holds, the prolongation (t, g) is redundant with respect to $(G, \mathcal{M})$.

We omit the phrase "with respect to $(G, \mathcal{M})$" if G and $\mathcal{M}$ are clear from the context.

Remark 7.6.6. Note that in Definition 7.6.5 f is uniquely determined if it exists, since $\mathcal{M}$ is disjoint on $\operatorname{lt}(G \setminus Q)$. Due to this disjointness, some S-polynomials are never computed in the involutive basis algorithm. They correspond to automatically omitted S-pairs. In Example 7.6.1, the S-pair $\{f_1, f_2\}$ is automatically omitted. Another instance of an automatically omitted S-pair has already been mentioned in the Janet basis computation of Example 5.2.1, namely the S-pair $\{f_2, f_4\}$. Obviously, each automatically omitted S-pair is redundant by the above definition.

In order to simplify the proofs of the propositions below, we provide the following lemmata and corollaries.

Lemma 7.6.7. *Let $(t, g) \in C$, $f \in G \setminus Q$ such that $\operatorname{lt}(tg) \in \mathcal{M}_{\operatorname{lt}(f)}$. If $\operatorname{anc} g = \operatorname{anc} f$ then the prolongation (t, g) is redundant.*

Proof. We have $\operatorname{lt} f \trianglelefteq \operatorname{lt}(tg) =: s$. If $t = 1$ then $g \in Q$. It follows

$$\operatorname{lt}(\operatorname{anc} g) = \operatorname{lt}(\operatorname{anc} f) \trianglelefteq \operatorname{lt} f \lhd s = \operatorname{lt} g,$$

since $f \notin Q$. Then $[\operatorname{anc} g, \operatorname{anc} f] \trianglelefteq \operatorname{lt} f \lhd s$. If $t \neq 1$ then

$$\operatorname{lt}(\operatorname{anc} g) = \operatorname{lt}(\operatorname{anc} f) \trianglelefteq \operatorname{lt} g \lhd s,$$

and consequently, $[\operatorname{anc} g, \operatorname{anc} f] \trianglelefteq \operatorname{lt} g \lhd s$. So in either case, the prolongation (t, g) is redundant because of criterion $\mathcal{C}_1$. □

Corollary 7.6.8. *Let $g', f' \in G$ and let t, g, f be such that $\mathcal{A}(g', f', t, g, f)$ and $\operatorname{anc} g' = \operatorname{anc} f$ holds. Then the prolongation (t, g) is redundant.*

Corollary 7.6.9. *Let $g', f' \in G$ with $\operatorname{anc} g' = \operatorname{anc} f'$. Then the S-pair $\{g', f'\}$ is redundant.*

Proof. Take any t, g, f such that $\mathcal{A}(g', f', t, g, f)$ or $\mathcal{A}(f', g', t, g, f)$ holds. The prolongation (t, g) is redundant by Corollary 7.6.8. Hence, $\{g', f'\}$ is redundant. □

As we know from Example 7.6.2, there might be several prolongations that correspond to one S-polynomial. We first show that at most one of them is not redundant since all the others can be avoided by means of criteria.

Proposition 7.6.10. *Let $g', f' \in G$ with $g' \neq f'$. Furthermore, let there be two triples (t, g, f) and (y, h, p) such that*

$$h \neq g \wedge \mathcal{A}(g', f', t, g, f) \wedge (\mathcal{A}(g', f', y, h, p) \vee \mathcal{A}(f', g', y, h, p))$$

holds. Then (t, g) or (y, h) is redundant.

Proof. Let $s := [g', f']$. Obviously, $f = p$ since $s \in \mathcal{M}_{\operatorname{lt}(f)}$, $s \in \mathcal{M}_{\operatorname{lt}(p)}$ and $\mathcal{M}$ is disjoint on $\operatorname{lt}(G \setminus Q)$. If $\mathcal{A}(f', g', y, h, f)$ holds then we have $\operatorname{anc} f' = \operatorname{anc} f = \operatorname{anc} h$ and (y, h) is redundant by Lemma 7.6.7. Now, consider the case that $\mathcal{A}(g', f', y, h, f)$ holds.

1. Assume $t \neq 1 \neq y$. We can conclude $\operatorname{lt}(\operatorname{anc} h) = \operatorname{lt}(\operatorname{anc} g) \trianglelefteq \operatorname{lt} g$. Therefore, $[\operatorname{anc} g, \operatorname{anc} h] \trianglelefteq \operatorname{lt} g \triangleleft s$ and by criterion $\mathcal{C}_4$ either (t, g) or (y, h) is redundant.

2. Assume $t = 1$ and $\operatorname{anc} g \neq g$. It follows $\operatorname{lt}(\operatorname{anc} g) \triangleleft \operatorname{lt} g = s$ and thus the prolongation (t, g) is redundant by application of criterion $\mathcal{C}_0$. An analogous statement holds for $y = 1$ and $\operatorname{anc} h \neq h$.

3. Assume $t = 1$ and $\operatorname{anc} g = g$. From $\operatorname{lt} g = \operatorname{lt}(\operatorname{anc} g) = \operatorname{lt}(\operatorname{anc} g') \trianglelefteq \operatorname{lt} g' \trianglelefteq \operatorname{lt} g$ follows $g = g'$. It implies $\operatorname{lt} g \triangleleft \operatorname{lt} h$ by $g' \trianglelefteq_G^{\operatorname{anc}} h$ and $h \neq g$. Since $\operatorname{lt}(tg) = \operatorname{lt}(yh)$, this case is impossible.

□

With the condition $\operatorname{lt}(tg) = [g', f']$, Definition 7.6.5 reflects the fact that proper multiples of S-polynomials are redundant. In Example 5.2.1, we explicitly reduce the prolongation (y, f_9) because Algorithm 2 does not apply criteria to avoid unnecessary reductions. If we apply Algorithm 9 to the three input polynomials $\{f_1, f_2, f_3\}$ of Example 5.2.1, the trace of the computation is the same (see the table on page 30), but the prolongation (y, f_9) is recognised to be redundant since the S-polynomial $\operatorname{spol}(f_9, f_7)$ is a proper multiple of $\operatorname{spol}(f_3, f_6)$.

Proposition 7.6.11. *Let $g', f', g_1', f_1' \in G$ with $g' \neq f'$, $[g', f'] \triangleleft [g_1', f_1'] =: s_1$, $g' \trianglelefteq_G^{\operatorname{anc}} g_1'$, and $f' \trianglelefteq_G^{\operatorname{anc}} f_1'$. Then $\{g_1', f_1'\}$ is redundant.*

Proof. Without loss of generality, we take any t_1, g_1, f_1 with $\mathcal{A}(g_1', f_1', t_1, g_1, f_1)$. It follows $\operatorname{anc} g' = \operatorname{anc} g_1' = \operatorname{anc} g_1 \trianglelefteq s$. Analogously, we derive $\operatorname{lt}(\operatorname{anc} f_1) \trianglelefteq s$ and thus $[\operatorname{anc} g_1, \operatorname{anc} f_1] \trianglelefteq s \triangleleft s_1$. Criterion $\mathcal{C}_1$ can be applied and avoids the reduction of (t_1, g_1). So $\{g_1', f_1'\}$ is redundant. □

Now we are ready to compare Buchberger's coprime and chain criterion with our criteria. Although only polynomials $p \in G$ for which $\operatorname{anc} p = p$ holds actually occur in Buchberger's algorithm, we do not restrict to only these situations, but consider all polynomials that appear in an involutive basis algorithm.

Let $g', f' \in G$ such that $g' \neq f'$, $[g', f'] = \operatorname{lt}(g'f')$ then, by application of the coprime criterion, the S-polynomial of g' and f' need not be reduced in Buchberger's algorithm. A similar statement holds for the involutive basis algorithm.

Proposition 7.6.12. *Let $g', f' \in G$ with $g' \neq f'$. Assume that $[g', f'] = \operatorname{lt}(g'f')$. Then either $\operatorname{Id}(G) = \mathbb{K}[X]$ or $\{g', f'\}$ is redundant with respect to $(G, \mathcal{M})$.*

Proof. The roles of g' and f' can be exchanged. So, without loss of generality, take t, g, f such that $\mathcal{A}(g', f', t, g, f)$ holds. If $t = 1$ and $\operatorname{lt} g' \lhd \operatorname{lt} g$ then (t, g) is redundant by criterion $\mathcal{C}_0$. If $t = 1$ and $g' = g$ then $[g', f'] = \operatorname{lt}(g'f')$ implies $f' = 1$, i. e., $\operatorname{Id}(G) = \mathbb{K}[X]$. Any algorithm which computes Gröbner bases could stop if $1 \in G$ is found. If $t \neq 1$ then (t, g) is redundant by criterion $\mathcal{C}_2$. □

Let $g', f', h' \in G$ be three pairwise different polynomials with $\operatorname{lt} h' \unlhd [g', f']$. In this case at most two of the three S-polynomials $\operatorname{spol}(g', f')$, $\operatorname{spol}(g', h')$, and $\operatorname{spol}(h', f')$ must be reduced in Buchberger's algorithm according to the chain criterion. The following proposition shows that in this situation also the involutive basis algorithm finds that at least one of the three S-pairs is redundant.

Proposition 7.6.13. *Let $g', f', h' \in G$ be three pairwise different polynomials with $\operatorname{lt} h' \unlhd [g', f']$. Then one of the S-pairs $\{g', f'\}$, $\{g', h'\}$, $\{h', f'\}$ is redundant.*

Proof. Assume for a contradiction that none of the above S-pairs is redundant.

The roles of g' and f' can be exchanged. So, without loss of generality, let the triple (t, g, f) be such that $\mathcal{A}(g', f', t, g, f)$ holds and (t, g) is not redundant. By Proposition 7.6.10 such a triple is unique. Furthermore let y, h, p be such that

$$\mathcal{A}(g', h', y, h, p) \vee \mathcal{A}(h', g', y, h, p) \tag{7.12}$$

holds and (y, h) is not redundant. Additionally, we can assume $\bar{y}$, $\bar{h}$, $\bar{p}$ such that

$$\mathcal{A}(f', h', \bar{y}, \bar{h}, \bar{p}) \vee \mathcal{A}(h', f', \bar{y}, \bar{h}, \bar{p}) \tag{7.13}$$

holds and $(\bar{y}, \bar{h})$ is not redundant. We distinguish several cases.

1. Assume $[g', h'] = [g', f']$. Since $\mathcal{M}$ is disjoint on $\operatorname{lt}(G \setminus Q)$, we conclude $p = f = \bar{p}$. If $\mathcal{A}(h', g', y, h, f)$ holds in (7.12) then $g' \unlhd_G^{\mathrm{anc}} f$. By using $\mathcal{A}(g', f', t, g, f)$, Corollary 7.6.8 implies that (t, g) is redundant, a contradiction. By Corollary 7.6.8, we can also conclude that $(\bar{y}, \bar{h})$ is redundant if $\mathcal{A}(f', h', \bar{y}, \bar{h}, f)$ holds in (7.13), because we know $f' \unlhd_G^{\mathrm{anc}} f$ by $\mathcal{A}(g', f', t, g, f)$; again a contradiction. In the remaining case, it holds $\mathcal{A}(g', h', y, h, f)$ and $\mathcal{A}(h', f', \bar{y}, \bar{h}, f)$. So we have $h' \unlhd_G^{\mathrm{anc}} f$. Contrary to the assumption, Corollary 7.6.8 implies that $(\bar{y}, \bar{h})$ is redundant.

2. Assume $[g', h'] \lhd [g', f']$ and $[f', h'] \lhd [g', f']$. From $[\operatorname{anc} g, h'] = [\operatorname{anc} g', h']$ follows $[\operatorname{anc} g, h'] \lhd [g', f']$ and, analogously, $[\operatorname{anc} f, h'] \lhd [g', f']$. Therefore, (t, g) is redundant by criterion $\mathcal{C}_3$. This, however, contradicts the assumption.

3. Assume $[g', h'] \lhd [g', f'] = [h', f']$. If $t = 1$ then $\operatorname{lt} g' \lhd \operatorname{lt} g = [g', f']$. The prolongation (t, g) is redundant because criterion $\mathcal{C}_0$ applies, a contradiction. Similar considerations lead to the contradiction that (y, h) is redundant if $y = 1$.

   Otherwise, we have $t \neq 1 \neq y$. Then (t, g) or (y, h) is redundant by criterion $\mathcal{C}_4$, again a contradiction.

So the assumption that none of the S-pairs $\{g', f'\}$, $\{g', h'\}$, $\{h', f'\}$ is redundant must have been false. □

7.7 Selection Strategy vs. Termination

We have already seen in Section 6.6 that the termination of Algorithm 4 is independent of the prolongation selection strategy as long as the partial division used during its execution is "sufficiently big". Algorithm 9 was obtained from Algorithm 4 by adding criteria to avoid unnecessary reductions. In this section, we show that Algorithm 9 terminates also without further restrictions on the partial divisions and without the necessity of a normal selection strategy in line 8.

Let us recall Example 6.6.1. If we apply Algorithm 9 instead of Algorithm 4, we see that the reduction step $vg_8 - wg_7$ would not be performed, because we have $s := [g_8, g_7] = vwy$, and because of

$$\begin{aligned}\operatorname{anc}_s g_8 &= \operatorname{wanc} g_8 = g_2 = y,\\ \operatorname{anc}_s g_7 &= \operatorname{wanc} g_7 = g_2 = y,\end{aligned}$$

and $y \lhd s$ criterion $\mathcal{C}_1(g_8, g_7, s)$ is true. This is the key-observation which leads to Theorem 7.7.1. The criteria tested in line 7 make Algorithm 9 terminating even independent of the selection strategy used in line 6.

Theorem 7.7.1. *[AH02] Algorithm 9 terminates for all inputs fulfilling its specification, independent of the selection strategy applied in line 6 if for each set G which occurs during its run,* PartialDivision($\operatorname{lt} G$) *is a suitable partial division on* $\operatorname{lt} G$.

Proof. Let us sketch the proof idea. We first show that there can be only finitely many top-reductions during any run of Algorithm 9, because many of the potential top-reductions are avoided due to criterion $\mathcal{C}_1$. Then, by the properties of a partial division which refines the Thomas division, there can be only finitely many top-irreducible prolongations.

By G^i and $\mathcal{M}^i$ we denote the value of G and $\mathcal{M}$, respectively, before the i-th iteration of the <u>while</u> loop starting in line 5 of Algorithm 9. Moreover,

$$G^\infty := \{ h \mid \exists i \in \mathbb{N} : h \in G^i \}$$

denotes the set of all polynomials which eventually belong to the basis G. We have the sequence of inclusions $G^1 \subseteq G^2 \subseteq \cdots$ and the resulting sequence $\langle \operatorname{lt} G^1 \rangle \subseteq \langle \operatorname{lt} G^2 \rangle \subseteq \cdots$ must become stationary, let us say $\langle \operatorname{lt} G^i \rangle = \langle \operatorname{lt} G^k \rangle$ for all $i \geq k$. Note that after each iteration the weak ancestors of all elements $h \in G^\infty$ satisfy $\operatorname{wanc} h \in G^k$. Until the k-th iteration this is obvious and later this follows easily since the leading term of any element of $G^\infty \setminus G^k$ has a proper divisor in $\operatorname{lt} G^k$. Hence, also after the k-th iteration no element of $G^\infty \setminus G^k$ may become a weak ancestor according to the specification of wanc.

Let

$$G^* := \left\{ h \in G^\infty \,\middle|\, \operatorname{lt} h \trianglelefteq \operatorname{lcm}(\operatorname{lt} G^k) \right\}.$$

Furthermore, let $X' := X \cup \{1\}$ and

$$R := \{ h \in G^\infty \mid \exists t \in X',\, g \in G^* : tg \to_{G^\infty}^* h \}.$$

Note that for the definition of R we use the ordinary reduction relation with respect to the fix set G^∞.

Suppose there exist $l \in \mathbb{N}$ and $h \in G^\infty$ such that

$$G^{l+1} \setminus G^l = \{h\},\ h \notin R,\ \text{and}\ \operatorname{anc} h = h$$

at creation time of h. Choose l to be minimal with the above property.

First of all, we must have $l \geq k$ since $G^k \subseteq R$ by construction of R. In order to satisfy $\operatorname{anc} h = h$ and $h \notin G^k$, the polynomial h must result from a top-reduction of some prolongation, i. e., there exist $g, f \in G^l$ and $t \in X'$ such that $\operatorname{lt}(tg) \in \mathcal{M}^l_{\operatorname{lt} f}$ and $\left(tg - \frac{\operatorname{lm}(tg)}{\operatorname{lm} f} f \right) \to_{G^\infty}^* h$.

Next, we deduce $g \notin G^*$ since otherwise it follows $h \in R$ because of $tg \to_{G^\infty}^* h$. Hence, $s := \operatorname{lt}(tg) \not\trianglelefteq \operatorname{lcm}(\operatorname{lt} G^k)$. If $\operatorname{anc} g = g$ then $g \in R$ by minimality of l. We must have $t \neq 1$ in this case since otherwise $g \to_{G^\infty}^* h$ leads to the contradiction $h \in R$. Hence, in any case we obtain $\operatorname{lt}(\operatorname{anc} g) \triangleleft s$ and, therefore, $\operatorname{anc}_s g = \operatorname{wanc} g \in G^k$.

The equality $\operatorname{lt}(\operatorname{anc} f) = \operatorname{lt} f = s$ is impossible since we would obtain $f \in R$ by minimality of l and, consequently, $h \in R$ because of $f \to_{G^\infty}^* h$ in this situation. Therefore, $\operatorname{lt}(\operatorname{anc} f) \triangleleft s$ and $\operatorname{anc}_s f = \operatorname{wanc} f \in G^k$.

In summary, we proved $\operatorname{anc}_s g, \operatorname{anc}_s f \in G^k$ and $s \not\trianglelefteq \operatorname{lcm}(\operatorname{lt} G^k)$. Consequently, $[\operatorname{anc}_s g, \operatorname{anc}_s f] \triangleleft s$ and $\mathcal{C}_1(g, f, s)$. However, this means that the prolongation (t, g) is redundant and Algorithm 9 would have omitted the reduction of tg during the l-th iteration, a contradiction.

In conclusion, the supposition of the existence of $h \in G^\infty$ such that $\operatorname{anc} h = h$ and $h \notin R$ must have been wrong.

Since the elements of G^∞ have pairwise distinct leading terms, the set G^* must be finite. Furthermore, it follows that the number of polynomials h which can be obtained as the result of a single Gröbner reduction step of a fixed polynomial f modulo G^∞ is finite. Noetherianity of the term order $\prec$ ensures that the number of polynomials h resulting from f by an arbitrary number of Gröbner reduction steps modulo G^∞ is still finite. Finally, finiteness of G^* and X implies finiteness of R.

Hence, there exists k' such that $\operatorname{anc} h \neq h$ for all $h \in G^\infty \setminus G^{k'}$, i. e., all elements added to G after the k'-th iteration result from top-irreducible prolongations. However, now, the properties of a partial division refining the Thomas division ensure $\operatorname{lcm}(\operatorname{lt} G^l) = \operatorname{lcm}(\operatorname{lt} G^{k'})$ for all $l \geq k'$. Hence, only a finite number of elements is added to $G^{k'}$. Therefore, there exists $k'' \geq k'$ such that $G^l = G^{k''}$ for all $l \geq k''$. Since $G^{k''}$ possesses only a finite number of prolongations, there are only finitely many iterations left. Hence, the algorithm terminates. □

Remark 7.7.2. The computation in Example 7.5.3 demonstrates that each time a new polynomial has been added to the basis, all prolongations are checked again. If the partial division does not change too drastically, most of the prolongations that were redundant in a previous iteration are then redundant by the same reason. Although checking criteria is usually cheaper than performing lengthy reductions, it does not seem to be satisfactory to test the same criteria again and again. Theorem 7.7.1 allows us to postpone the selection of prolongations that have been selected in a previous iteration as long as there are prolongations that have never been selected before. By such a strategy, the correctness of the algorithm is not lost because the prolongations are just selected *at the end*. If then one of such a previously considered prolongation cannot be proved to be redundant by some criterion, it is explicitly reduced and a new polynomial is added to the basis if the reduction does not lead to zero.

In particular, in the computation in Example 7.5.3, we can avoid any criteria test on prolongations which appear for the second time before f_7 is added to the basis. Thus, before f_7 appears in the trace of the computation on page 74, the seven prolongations that are marked *avoided* or *redundant* and the prolongation that is marked *repeated* can be removed. Note that the repeated reduction of (y, f_2) is avoided by a generalised version of criterion $\mathcal{C}_3$, cf. page 71.

One may modify any prolongation selection strategy by this process of postponing prolongations. Obviously, it destroys the selection strategy one had in mind in the first place. Therefore, extensive testing is necessary in order to find a good selection strategy for the involutive basis algorithm similar to what the sugar strategy [GMN+91] is for Buchberger's algorithm.

Example 7.7.3 (Cyclic4). *We continue Example 6.7.1 by computing the involutive bases with Algorithm 9, i. e., by performing involutive basis computations that include the application of our criteria. We apply a modified normal selection strategy as explained in Remark 7.7.2 and compute with respect to the degree inverse lexicographical term order. The computations have been performed by using our software package* CALIX *where Algorithm 9 is implemented, cf. Chapter 8.*

In the following table, we use the same notation as in the example on page 58.

Division	*N*	*P*	*R*	*Z*
Thomas Division	*98*	*101*	*151*	*7*
Janet Division	*37*	*38*	*53*	*5*
Sliced Division	*7*	*11*	*38*	*5*

Below we list the applicability of our criteria. For each division there are two lines. The first line counts how often each criterion holds independent of the others. The second line counts how often a criterion holds if all of the criteria in the previous columns do not hold.

Division	$\mathcal{C}_r$	$\mathcal{C}_0$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}_4$
Thomas Division	*11*	*0*	*359*	*103*	*359*	*244*
	11	*0*	*354*	*14*	*0*	*1*
Janet Division	*8*	*0*	*80*	*38*	*80*	*43*
	8	*0*	*77*	*19*	*0*	*5*
Sliced Division	*7*	*3*	*3*	*9*	*3*	*0*
	7	*0*	*0*	*9*	*0*	*0*

Since the Thomas and Janet divisions are disjoint on the respective sets, criterion $\mathcal{C}_0$ cannot apply.

Example 7.7.4 (Cyclic6). *By Algorithm 9, we compute an involutive basis of the polynomials from Example 6.7.2 with respect to the degree inverse lexicographical term order. We apply a modified normal selection strategy as explained in Remark 7.7.2. The columns of the tables below should be read as in the previous example.*

Division	*N*	*P*	*R*	*Z*
Janet Division	*840*	*1280*	*41038*	*446*
modified Janet Division	*385*	*1217*	*27862*	*386*
Sliced Division	*46*	*649*	*23275*	*469*

Better than in the previous example, we see that also our criteria $\mathcal{C}_0$, $\mathcal{C}_3$, and $\mathcal{C}_4$ apply in quite a number of cases.

Division	$\mathcal{C}_r$	$\mathcal{C}_0$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}_4$
Janet Division	*569*	*0*	*5349*	*1643*	*5408*	*3164*
	569	*0*	*4904*	*274*	*25*	*68*
modified Janet Division	*410*	*1168*	*4067*	*1489*	*4154*	*2405*
	410	*1144*	*3101*	*293*	*39*	*142*
Sliced Division	*129*	*138*	*134*	*176*	*138*	*0*
	129	*90*	*2*	*175*	*3*	*0*

Chapter 8

Implementation

The algorithms presented in this thesis have been implemented by the author in the software package Calix. All the necessary computations of our examples have been performed with this package. In fact, also the coloured representations of cone decompositions as presented, for example, in Figure 4.2 on page 16 have been produced with Calix. The package is written in the programming language Aldor[1].

Calix implements a parametric form of a critical element/completion procedure for polynomials. For an overview on critical pair/completion procedures see [Buc87]. In our case the parameters are specific *domains of computation*. A *domain* is a concept that is explained in detail in the Aldor User Guide [Ald02]. Domains in Calix are, for example, the coefficient ring, the terms, the polynomials, the simplifiers, the critical elements. By providing several domains and plugging them appropriately into the general critical element/completion framework, we obtain well-known instances such as Buchberger's algorithm and several variants of the involutive basis algorithm. Let us focus here on the involutive basis algorithm. We first demonstrate the features of Calix and how they can be accessed. In the second part of this chapter, we concentrate on selected implementation details and give a data structure to maintain the history of the involutive basis algorithm. In the third part, we take advantage of this data structure and give an additional efficient criterion to avoid repeated reductions.

8.1 Features of Calix

Calling the program without any parameter shows a help screen which explains the features of Calix. We give here only a brief description of the possible parameter combinations.

Parameters are given to the program by using a

```
KEYWORD=value
```

mechanism. Such parameters can either be given on the command line or can be extracted from a file. Up to special characters that must be escaped from the shell expansion by surrounding the value in single or double quotes, the parameter format on the command line and in a file are identical. Whereas on

[1] `http://www.aldor.org`

the command line parameters are separated by spaces, in a file each keyword must be at the beginning of a new line. Assume we have written the file database which contains the following lines.

```
BEGIN IB
VARS x,y,z
ORDER degrevlex
POLY=x^8*y^5*z^2 - x^4*y^3*z^3 + x^4*y^3 - y*z
POLY=x^8*y^4*z^4 - x^4*y^4*z^3 + x^4*y^2*z^2 - y^2*z
POLY=x^6*y^4*z^9 - x^6*y^2*z^11 + x^2*y^2*z^7 - x^2*z^9
METHOD=involutive
DIVISION=slicedCherry
STRATEGY=criteria
INFO=step
INFO=reduce(reduction)
POST=autoReduce
END IB
```

The `BEGIN` and `END` lines are wrappers in order to separate several data portions within one file. Each example in a file must have a unique name which is given both after the `BEGIN` and after the `END` keyword.

For example, the execution of

```
calix NAME=IB DB=database
```

on the command line produces the computation trace of Example 7.5.3 where the keyword `DB` specifies the file from which the example is read and the keyword `NAME` refers to the name of the specific data portion within the file.

The amount of information that is shown during the computation can be controlled by the keyword `INFO` as shown above. Depending on the value given to the `FORMAT` keyword a full computation trace (`FORMAT=long`) or only size information (`FORMAT=size`) of the intermediate data is shown.

If one calls the program by

```
calix FORMAT=cube OUT=trace.ps NAME=IB DB=database
```

a PostScript file trace.ps is created which contains the pictorial representation of the cone decompositions during the computation as shown on page 75.

In order to compute with respect to other term orders, one has to give another value to the `ORDER` keyword in the file above. All term orders described in Definition 3.0.5 are available by name, i. e., `lex`, `deglex`, `deginvlex`, and `degrevlex`. Additionally, on can specify term orders given by integer matrices. For example, a term order with respect to the matrix

$$\begin{pmatrix} 1 & 3 & 2 \\ -1 & 0 & 3 \end{pmatrix} \tag{8.1}$$

where ties are broken by the lexicographical order can be given in the above file database by writing `lex(1,3,2;-1,0,3)` instead of `degrevlex`.

CALIX is able to compute an involutive basis with respect to a number of partial divisions. At the moment we have implemented the

- classical divisions of Janet, Thomas, and Pommaret,

- Division I, Division II, and the Induced Division as defined in [Ger99],
- a submaximal refinement of the Janet division according to the submaximal refinement algorithm of [Ape98a],
- the modified Janet division that we have used in Examples 6.7.2 and 7.7.4,
- several variants of Sliced Division, in particular the Sliced Division tree algorithm.

By default computations are done over the field of rational numbers. In order to compute the above example in characteristic 32009 one simply adds RING=Z32009 on the command line or in the file.

Both Buchberger's algorithm as well as the involutive basis algorithm also work in the more general context of left ideals of algebras of solvable type, cf. [KRW90] and [Sei02]. If we add

```
BEGIN Weyl
VARS x,y,X,Y
ORDER=lex(0,0,1,1)
COMM=diff(X,x)
COMM=diff(Y,y)
POLY=X*Y+x
POLY=x*Y
END Weyl
```

to the file database and call

```
calix NAME=Weyl DB=database
```

then a left Gröbner basis of the left ideal generated by the above (skew) polynomials in the Weyl algebra over $\mathbb{Q}$ is computed.[2] Adding

```
METHOD=involutive
DIVISION=slicedCherry
STRATEGY=criteria
```

computes an involutive basis with respect to a Sliced Division.

In addition to the command line version, CALIX provides small MAPLE and MATHEMATICA functions so that the computations with CALIX can be started from within these computer algebra systems.

8.2 Selected Implementation Details

In this section we focus on several details of an actual implementation of the INVOLUTIVEBASIS algorithm. In order to concentrate on the main structure of Algorithm 9 in Section 7.4, we have left out how the history of the algorithm is managed. Now, we close the gap by providing implementation details.

In Section 7.3, we have introduced the ancestor and weak ancestor functions which are essential during the criteria test in the involutive basis algorithm. They describe the history of an actual run of the algorithm. The history in

[2] All variables commute except that $Xx = xX + 1$ and $Yy = yY + 1$.

Algorithm 9 is maintained by attaching to each polynomial g of the current basis a tuple

$$\left(a_g, w_g, i_g, Y_{\mathrm{lt}(g)}, s_g, r_g\right)$$

and by updating its values in each iteration.

The entries a_g, w_g, and i_g, represent the values of $\mathrm{anc}\,(g)$, $\mathrm{wanc}\,(g)$, and $\mathrm{idx}\,(g)$, respectively. With condition (7.10) in Section 7.6, we require

$$\forall g \in G : \mathrm{anc}\,(\mathrm{anc}\, g) = \mathrm{anc}\, g,$$

i. e., the ancestor function always yields a fixpoint of anc . It is, in fact, always a good idea to find a "small" ancestor. By selecting a "minimal" ancestor, one better takes advantage of the history of the involutive basis algorithm and it is more likely that one of our criteria applies. The above requirement can be met by iterating the access to the ancestor entry until a fixpoint is reached. In order to minimise the number of such iterations one should immediately re-set the ancestor so that it points to that fixpoint. Iterating the weak ancestor access is useless since w_g is by definition an element in the minimal basis and it is already updated appropriately by the rules given in Remark 7.4.1 on page 71.

The multiplicative variables $Y_{\mathrm{lt}\,(g)}$ are represented by a boolean array of length $n = \mathrm{card}\, X$, so that multiplicative and non-multiplicative variables can be extracted equally well. Note that during the computation of SLICEDDIVISION it is already decided which terms are divisible with respect to the so computed Sliced Division. Since during the involutive basis algorithm these terms are the leading terms of the current basis polynomials, one gets the computation of Q in lines 3 and 12 in Algorithm 9 for free if, instead of terms, one gives the basis polynomials (including their attached data) to the Sliced Division algorithm and collects the appropriate data. In case a Sliced Division tree is used during the involutive basis algorithm, it is not necessary to store $Y_{\mathrm{lt}(g)}$ explicitly for each polynomial, since the non-multiplicative (and hence also the multiplicative) variables of the simplifier polynomial (and only those are really needed) can easily be extracted during the divisor search as explained before Example 6.5.5. Similarly, the set C of critical elements in Algorithm 9 can be formed by traversing the Sliced Division tree.

The component s_g is a boolean array where for $t \in X \cup \{1\}$ the entry $s_g[t]$ is true if and only if the prolongation (t, g) has been selected in line 6 of Algorithm 9. The array s_g is used to implement the modified normal selection strategy which is explained in Remark 7.7.2. When the critical elements are created in lines 4 and 13 of Algorithm 9, we actually form two lists. A (relevant) prolongation (t, g) is put into the first or second list depending on whether $s_g[t]$ is false or true. In order to minimise memory usage, the creation of the second list can be postponed until all prolongations from the first list have been treated.

The component r_g is an array where for $t \in X \cup \{1\}$ the entry $r_g[t]$ is either nil or (a pointer to) the last simplifier f such that $\mathcal{R}(t, g, f)$ holds.

In the next section we demonstrate how we can benefit from the above data structure.

8.3 Implementation of Criteria

Recall Definition 7.3.1. We can distinguish two types of criteria. Criteria of the first kind, i.e., $\mathcal{C}_0$, $\mathcal{C}_1$, and $\mathcal{C}_2$, only involve the current prolongation (t, g), the corresponding simplifier f, and access to the (weak) ancestor. Note that $s = \operatorname{lt}(tg)$. With our data structure above, this type of criteria can be considered to be "cheap" in the sense that their test requires only $\mathcal{O}(n)$ operations and does not depend on the number of basis elements. Recall from Section 6.5.2 that even the top-simplifier f can be found by using a Sliced Division tree in $\mathcal{O}(n^2)$ and thus independent of the number of elements in the basis.

Criteria of the second kind, i.e., $\mathcal{C}_3$ and $\mathcal{C}_4$, require to find an intermediate polynomial h in the current basis. Whereas in criterion $\mathcal{C}_3$ only the minimal basis must be traversed, i.e., it is as expensive as the chain criterion in the Gröbner basis case, in criterion $\mathcal{C}_4$ the polynomial h must be searched in the whole basis.

In fact, also criterion $\mathcal{C}_r$ is of first kind. Let us reconsider the rules from Section 7.4 to avoid repeated reductions of prolongations. We have proposed, in addition to criterion $\mathcal{C}_r$, to extend criterion $\mathcal{C}_3$ by the knowledge of the history of the computation. By such an extension many unnecessary (repeated) prolongations are detected. In fact, it covers the case $\operatorname{anc} h = f$ which we have omitted in criterion $\mathcal{C}_r$ on page 70. However, due to the fact that in (7.6) criterion $\mathcal{C}_3$ is called with respect to the set H, i.e., a minimal subset of G for which $\langle \operatorname{lt} H\rangle = \langle \operatorname{lt} G\rangle$ holds, it may happen that some prolongations that need not be reduced a second time are not detected. Whereas criterion $\mathcal{C}_3$ requires to search an appropriate intermediate polynomial, we give here another criterion that can be implemented without traversing through the basis.

Let us use the notation before the introduction of criterion $\mathcal{C}_r$ on page 70 and repeat the considerations. We have a prolongation $(t, g) \in C^k$ in the k-th iteration which was explicitly reduced by h, i.e., $\mathcal{R}(t, g, h)$ holds, and now has a new $\mathcal{M}^k$-divisor $f \in G^k \setminus Q^k$. The relation $\mathcal{S}^{G^k}(g, f)$ can be concluded by Buchberger's chain criterion from $\mathcal{R}(t, g, h)$ and $\mathcal{S}^{G^k}(h, f)$. We have already covered the cases $h = f$ and $h = \operatorname{anc} f$ by criterion $\mathcal{C}_r$. Additionally, also $f = \operatorname{anc} h$ implies $\mathcal{S}^{G^k}(h, f)$. We describe this case by the condition

$$\exists f \in G \setminus Q \; \exists h \in G : \operatorname{lt}(tg) \in \mathcal{M}_{\operatorname{lt}(f)} \wedge \mathcal{R}(t, g, h) \wedge f = \operatorname{anc} h. \tag{8.2}$$

If this condition is true in line 7 of Algorithm 9, one can avoid the reduction of the prolongation (t, g) and immediately proceed to the next prolongation.

Condition (8.2) requires to traverse the whole basis in order to find an appropriate h. If we use a selection strategy which postpones the treatment of previously selected prolongations until no more untreated prolongations are available, then for each prolongation (t, g) there is at most one such h and it is stored in $r_g[t]$. Being more exact, there might be several such h in case a repeated reduction cannot be avoided and leads to a non-zero remainder. Despite this unfortunate possibility, we only store the last simplifier in $r_g[t]$. Focusing on the last simplifier and knowing f, condition (8.2) can be turned into the test

$$r_g[t] \neq \mathsf{nil} \wedge f = \operatorname{anc}(r_g[t]),$$

which does not require to traverse the basis. We call this criterion $\mathcal{C}_c$. The above test can be made very cheap by storing each polynomial in a unique place in memory and comparing only pointers. Criterion $\mathcal{C}_c$ is therefore a criterion of the first kind.

Though it seems to be natural to test criteria of the first kind before the other criteria, the order of applying the criteria is still an open question. We have put $\mathcal{C}_r$ and $\mathcal{C}_c$ first, since in our implementation we considered pointer comparison to be cheaper than checking divisibility or multiplication of terms. Nevertheless, we leave the investigation of the costs of our criteria open for future research.

Example 8.3.1 (Cyclic6). *After the introduction of criterion $\mathcal{C}_r$ on page 71, we proposed to generalise $\mathcal{C}_3$. Since we did not use such a generalised version in Example 7.7.4, the computation of an involutive basis with respect to the Sliced Division involved re-reductions of 21 prolongations. These prolongations are all reduced to zero. By a computation with the additional criterion $\mathcal{C}_c$, all repeated reduction are avoided.*

The entries for the Sliced Division in the tables of Example 7.7.4 then look as follows.

Division	*N*	*P*	*R*	*Z*
Sliced Division	*46*	*628*	*22498*	*448*

Division	$\mathcal{C}_r + \mathcal{C}_c$	$\mathcal{C}_0$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}_4$
Sliced Division	*226*	*138*	*134*	*176*	*138*	*0*
	129+97	*14*	*2*	*175*	*3*	*0*

Remark 8.3.2. Interestingly, application of the above criterion does not lead to any improvement for the involutive basis computations with the other two divisions of Example 7.7.4. Although in the involutive basis computation with respect to the modified Janet division, several prolongations appear more than once, they cannot be counted as repeated reductions. The reason is that always the first of such a *repeated* reduction happens to be of the form $tg - h$ with $\operatorname{lt}(tg) = \operatorname{lt} h$ so that according to our rule (ii) of Remark 7.4.1 the ancestor of h is either already different from h or updated to be $\operatorname{anc} g$. In any case, criterion $\mathcal{C}_0$ prevents any reduction of prolongations of the form $(1, h)$. In all of the above cases, the corresponding polynomial h becomes involutively top-reducible with respect to the final involutive basis. The reduction $tg - h$ can then be seen as a (not necessarily involutive) reduction of h by g and thus the *repeated* reduction of tg is, in fact, essential.

Chapter 9

Conclusion

In this thesis we have treated several topics in the context of polynomial involutive bases.

We have defined the concepts *essentially disjoint* and *cycle-free*, which describe parts of the conditions on a continuous involutive division due to Gerdt and Blinkov [GB98a], and we have taken *Thomas refinement* from Apel [Ape98a]. Our notion of a *suitable partial division* comprises these concepts and can count as a new foundation for terminating Janet-like algorithms. By removing restrictions, our approach can be seen as a generalisation and unification of the theories of Gerdt/Blinkov and Apel.

Furthermore, we have presented the Sliced Division algorithm. It is an efficient algorithm which produces a "good" partial division for an application in the involutive basis algorithm. The use of the Sliced Division algorithm as a sub-algorithm in the involutive basis algorithm positively affects the number of critical elements that have to be treated. We have demonstrated how a Sliced Division tree can be constructed for directed Sliced Divisions and how one can benefit from such a tree structure during the search phase of an involutive divisor of a term. The worst case complexity of a divisor search in such a tree is basically independent of the number of nodes in this tree and independent of the degrees of the terms in the tree. In addition to that, we could show that the use of certain types of Sliced Divisions allows to employ any prolongation selection strategy during the involutive basis algorithm without losing its termination.

Extending the results of Gerdt and Blinkov, we have introduced several new criteria in order to avoid unnecessary reductions in the involutive basis algorithm. Additionally, we have compared our criteria to Buchberger's coprime and chain criteria and have shown that they are equally strong. By showing that certain "bad" reductions are not performed in an involutive basis algorithm which applies our criteria, we could prove that in an involutive basis algorithm with criteria any prolongation selection strategy can be used without losing its termination.

Finally, we have demonstrated our implementation of the involutive basis algorithm in CALIX and have given some hints for an efficient implementation.

Since research tends to be infinite, but a thesis should be finite, unfortunately, we have to leave several problems open. We have mostly considered improvements to the involutive basis algorithm from a theoretical point of view and have demonstrated these improvements only by a few examples. Extensive computer tests with an efficient implementation are necessary in order to find a reasonable compromise of our theoretical results and practical applications. In particular, it remains an open question in which order our criteria should be applied, since this depends on their costs and their applicability in practical cases.

The flexibility of our generalised Sliced Division algorithm opens the field for experiments of running the involutive basis algorithm with different suitable partial divisions. It is a common belief that the involutive basis algorithm behaves better against Buchberger's algorithm with respect to intermediate coefficient growth. Since one of the differences of the two algorithms depends on the suitable partial division chosen during the run of the involutive basis algorithm, the connection between the chosen division and the effect on coefficient growth should be carefully investigated.

We have shown that the application of criteria enforces a termination of the involutive basis algorithm independent of the selection strategy that is used to choose the next prolongation. This opens a new field for further investigation on connections to Buchberger's algorithm. In particular, the *sugar strategy* [GMN+91] should be revisited in the context of involutive bases.

Although in this thesis we have not explicitly treated the applicability of involutive bases in the context of algebras of solvable type, most results can easily be carried over and remain valid. In particular, the criteria (except $\mathcal{C}_2$) and the strategy independence result as well as our work on suitable partial divisions can be used for involutive bases of left ideals in algebras of solvable type. Note that multiplication by different terms is not equally expensive in such a context. The running time of the involutive basis algorithm seems to be even more dependent on a "good" suitable partial division, since the division decides which elements are prolonged and kept in the basis. A careful choice of the suitable partial division depending on the commutation rules could avoid some unnecessary expensive multiplications during the reduction process.

Bibliography

[AH02] Joachim Apel and Ralf Hemmecke. Detecting unnecessary reductions in an involutive basis computation. RISC Linz Report Series 02-22, Research Institute for Symbolic Computation, Johannes Kepler Universität, 4040 Linz, Austria, Europe, October 2002.

[AL88] Joachim Apel and Wolfgang Laßner. An extension of Buchberger's algorithm and calculations in enveloping fields of Lie algebras. *Journal of Symbolic Computation*, 6(2–3):361–370, October–December 1988.

[AL94] William W. Adams and Philippe Loustaunau. *An Introduction to Gröbner Bases*. Number 3 in Graduate Studies in Math. American Mathematical Society, New York, 1994.

[Ald02] Aldor.org. *Aldor User Guide*, 2002. Available on the world wide web at `http://www.aldor.org`.

[Ape98a] Joachim Apel. The theory of involutive divisions and an application to Hilbert function computations. *Journal of Symbolic Computation*, 25(6):683–704, June 1998.

[Ape98b] Joachim Apel. *Zu Berechenbarkeitsfragen der Idealtheorie*. Habilitationsschrift, Universität Leipzig, Fakultät für Mathematik und Informatik, Augustusplatz 10–11, 04109 Leipzig, 1998.

[Buc65] Bruno Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. PhD thesis, Univ. Innsbruck, Dept. of Math., Innsbruck, Austria, 1965.

[Buc79] Bruno Buchberger. A criterion for detecting unnecessary reductions in the construction of Gröbner bases. In Edward W. Ng, editor, *Proc. EUROSAM '79*, volume 72 of *Lecture Notes in Computer Science*, pages 3–21. Springer Verlag, 1979.

[Buc85] 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. D. Reidel Publishing Company, Dordrecht, The Netherlands, 1985.

[Buc87] Bruno Buchberger. History and basic features of the critical-pair/completion procedure. *Journal of Symbolic Computation*, 3(1–2):3–38, February–April 1987. Rewriting techniques and applications (Dijon, 1985).

[BW93] Thomas Becker and Volker Weispfenning. *Gröbner Bases. A Computational Approach to Commutative Algebra*, volume 141 of *Graduate Texts in Mathematics*. Springer Verlag, New York, 1993.

[CG00] Yufu Chen and Xiao-Shan Gao. Involutive bases of partial differential polynomial systems. *Math.-Mechanization Research Preprints*, 19:11–32, 2000.

[CG01] Yufu Chen and Xiao-Shan Gao. Involutive directions and new involutive divisions. *Computers and Mathematics with Applications*, 41(7–8):945–956, 2001.

[CHS01] Jacques Calmet, Marcus Hausdorf, and Werner Markus Seiler. A constructive introduction to involution. Preprint, Universität Mannheim, Mannheim, Germany, 2001.

[Dic13] Leonard Eugene Dickson. Finiteness of the odd perfect and primitive abundant numbers with n distinct prime factors. *American Journal of Mathematics*, 35:413–422, 1913.

[Frö97] Ralf Fröberg. *An Introduction to Gröbner Bases*. John Wiley & Sons, 1997.

[GB98a] Vladimir P. Gerdt and Yuri A. Blinkov. Involutive bases of polynomial ideals. *Mathematics and Computers in Simulation*, 45:519–541, 1998.

[GB98b] Vladimir P. Gerdt and Yuri A. Blinkov. Minimal involutive bases. *Mathematics and Computers in Simulation*, 45:543–560, 1998.

[GBC98] Vladimir P. Gerdt, Matthias Berth, and Günter Czichowski. Involutive divisions in Mathematica: Implementation and some applications. In Jacques Calmet, editor, *Proceedings of the Sixth Rhine Workshop on Computer Algebra, March 31-April 3, 1998, Sankt-Augustin, Germany*, pages 74–91, 1998.

[GBY01a] Vladimir P. Gerdt, Yuri A. Blinkov, and Denis A. Yanovich. Construction of Janet bases I. Monomial bases. In Victor G. Ganzha, Ernst W. Mayr, and Evgenii V. Voronzhtsov, editors, *Computer Algebra in Scientific Computing, CASC 2001, Proceedings of the Fourth International Workshop on Computer Algebra in Scientific Computing, Konstanz, Germany, September 22–26, 2001*, pages 233–247. Springer Verlag, 2001.

[GBY01b] Vladimir P. Gerdt, Yuri A. Blinkov, and Denis A. Yanovich. Construction of Janet bases II. Polynomial bases. In Victor G. Ganzha, Ernst W. Mayr, and Evgenii V. Voronzhtsov, editors, *Computer Algebra in Scientific Computing, CASC 2001, Proceedings of the Fourth International Workshop on Computer Algebra in Scientific*

Computing, Konstanz, Germany, September 22–26, 2001, pages 249–263. Springer Verlag, 2001.

[Ger99] Vladimir P. Gerdt. Involutive division technique: Some generalizations and optimizations. *Zapiski Nauchnykh Seminarov POMI (St. Petersburg)*, 258:185–206, 1999. Submitted to CASC'98, International Conference "Computer Algebra in Scientific Computing", April 20-24, 1998, St. Petersburg, Russia. Available at `http://arXiv.org/abs/math.SC/9912030`.

[GM88] Rüdiger Gebauer and H. Michael Möller. On an installation of Buchberger's algorithm. *Journal of Symbolic Computation*, 6(2-3):275–286, October-December 1988.

[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, Bonn, Germany*, pages 49–54, New York, 1991. ACM Press.

[Hem02a] Ralf Hemmecke. Detecting unnecessary reductions in an involutive basis computation, June 2002. Talk presented at the Conference on Applications of Computer Algebra (ACA'02), Volos, Greece. Joint work with J. Apel.

[Hem02b] Ralf Hemmecke. An efficient method for finding a simplifier in an involutive basis computation (extended abstract). In Jacques Calmet, Marcus Hausdorf, and Werner M. Seiler, editors, *Proceedings of the Workshop on Under- and Overdetermined Systems of Algebraic or Differential Equations, March 18-19, 2002, Karlsruhe, Germany*, pages 87–95. Institut für Algorithmen und Kognitive Systeme, Universität Karlsruhe, March 2002. Local proceedings.

[Hem03] Ralf Hemmecke. Dynamical aspects of involutive bases computations. In Franz Winkler and Ulrich Langer, editors, *Symbolic and Numeric Scientific Computation—Proceedings of the Conference SNSC'01*, number 2630 in Lecture Notes in Computer Science. Springer Verlag, 2003. To appear.

[HSS02] Marcus Hausdorf, Werner Markus Seiler, and Rainer Steinwandt. Involutive bases in the Weyl algebra. *Journal of Symbolic Computation*, 34(3):181–198, September 2002.

[Jan20] Maurice Janet. Les systèmes d'équations aux dérivées partielles. *Journal de Mathématique. $8^{ème}$ série*, 3:65–151, 1920.

[Jan29] Maurice Janet. *Leçons sur le Systèmes d'Équations aux Derivées Partielles.* Gauthier-Villars, Paris, 1929.

[Kre93] Heinz Kredel. *Solvable Polynomial Rings.* Shaker, 1993. PhD thesis.

[KRW90] Abdelilah Kandri-Rody and Volker Weispfenning. Non-Commutative Gröbner Bases in Algebras of Solvable Type. *Journal of Symbolic Computation*, 9(1):1–26, January 1990.

[Pom78] Jean-François Pommaret. *Systems of Partial Differential Equations and Lie Pseudogroups*, volume 14 of *Mathematics and Its Applications*. Gordon and Breach Science Publishers, Inc., 1978.

[Riq10] Charles Riquier. *Les systèmes d'équations aux dérivées partielles*. Gauthier-Villars, Paris, 1910.

[Sei02] Werner Markus Seiler. A combinatorial approach to involution and δ-regularity I: Involutive bases in polynomial algebras of solvable type. Preprint, Universität Mannheim, Mannheim, Germany, 2002.

[SST99] Mutsumi Saito, Bernd Sturmfels, and Nobuki Takayama. *Gröbner Deformations of Hypergeometric Differential Equations*. Texts and Monographs in Symbolic Computation. Springer Verlag, 1999.

[Tho29] Joseph Miller Thomas. Riquier's existence theorems. *Annals of Mathematics*, 30(2):285–310, 1929.

[Tho34] Joseph Miller Thomas. Riquier's existence theorems. *Annals of Mathematics*, 35(2):306–311, 1934.

[Tho37] Joseph Miller Thomas. *Differential Systems*. American Mathematical Society, New York, 1937.

[Wu91] Wen-Tsün Wu. On the construction of Groebner basis of a polynomial ideal based on Riquier-Janet theory. *Systems Science and Mathematical Sciences*, 4(3):193–207, 1991.

[ZB93] Alexey Yu. Zharkov and Yuri A. Blinkov. Involution approach to solving systems of algebraic equations. In G. Jacob, N. E. Oussous, and S. Steinberg, editors, *Proceedings of the 1993 International IMACS Symposium on Symbolic Computation*, pages 11–16. IMACS, Laboratoire d'Informatique Fondamentale de Lille, France, 1993.

List of Algorithms

Index

Curriculum Vitae

Ralf Hemmecke

Born on July 22, 1967, Kölleda, Germany
German Nationality
Married since 2002
E-Mail: `ralf@hemmecke.de`

Education

1997–2003	Doctoral studies in Mathematics, Research Institute for Symbolic Computation (RISC), Johannes Kepler University Linz, Austria
1992–1993	Studies in Mathematics and Computer Science, University of Kent at Canterbury, England
1989–1996	Studies in Mathematics, University of Leipzig, Germany
1986	Abitur, Sömmerda, Germany

Work History

1997 to present	Research assistant, RISC, Johannes Kepler University Linz, Austria
1996–1997	Research assistant, Institut für Informatik, University of Leipzig, Germany
1994 to present	Beta-tester of the Aldor ($A^\sharp$, AxiomXL) compiler
1986–1989	Military service

Publications

Dynamical aspects of involutive bases computations. In F. Winkler and U. Langer, eds., *Symbolic and Numeric Scientific Computation—Proceedings of the Conference SNSC'01.* Springer, 2003. To appear.

CASA (with E. Hillgarter and F. Winkler). In J. Grabmeier, E. Kaltofen, and V. Weispfenning, eds., *Handbook of Computer Algebra: Foundation, Applications, Systems*, pp. 356–359. Springer, 2003.

Detecting unnecessary reductions in an involutive basis computation (with J. Apel). RISC Report 02-22, Johannes Kepler University, Linz, Austria, Oct 2002.

Continuously parameterized symmetries and Buchberger's algorithm. *Journal of Symbolic Computation*, 33(1):43–55, January 2002.

An evaluation of the state of the CASA system (with E. Hillgarter, W. Schreiner, and F. Winkler). RISC Report 98-16, Johannes Kepler University, Linz, Austria, Oct 1998.