Automated Reasoning in Reduction Rings using the *Theorema* System*

Alexander Maletzky

Doctoral College "Computational Mathematics" and RISC
Johannes Kepler University Linz, Austria
alexander.maletzky@dk-compmath.jku.at

Abstract. In this paper we present the computer-supported theory exploration, including both formalization and verification, of a theory in commutative algebra, namely the theory of reduction rings. Reduction rings, introduced by Bruno Buchberger in 1984, are commutative rings with unit which extend classical Gröbner bases theory from polynomial rings over fields to a far more general setting.
We review some of the most important notions and concepts in the theory and motivate why reduction rings are a natural candidate for being explored with the assistance of a software system, which, in our case, is the *Theorema* system. We also sketch the special prover designed and implemented for the purpose of semi-automated, interactive verification of the theory, and outline the structure of the formalization.

Keywords: Gröbner bases, reduction rings, mathematical theory exploration, automated reasoning, formalized mathematics, Theorema

1 Introduction

Automated reasoning, or, more precisely, computer-supported mathematical theory exploration, aims at the systematic creation of machine-checked, formal mathematics, either entirely by or at least with extensive support of software systems, where the meaning of "formal mathematics" may range from individual theorems over whole theories up to huge structured knowledge bases. In this paper, we demonstrate how a non-trivial theory in the realm of commutative algebra, namely the theory of reduction rings and Gröbner bases, can be formally developed in the *Theorema* software system, including semi-automated, interactive verification. Up to our knowledge, this is the first time this theory is the subject of computer-supported theory exploration in *any* software system, although it has to be mentioned that Gröbner bases in the *classical* setting (i. e. in polynomial rings over fields) have already undergone formal treatment in various flavors [15,11,5,6,7].

* This research was funded by the Austrian Science Fund (FWF): grant no. W1214-N15, project DK1

The theory of *reduction rings*, to be presented in more detail in Section 2, is a natural candidate for computer-supported exploration: a good deal of the notions and concepts involved have rather lengthy and technical definitions, which quite often leads to comparatively technical proofs with many case distinctions, subgoals, etc. Although the correctness of all results has already been established more than 20 years ago using "pencil and paper", the theory has hardly been extended and generalized since then (e. g. to non-commutative rings; see also Section 6), which is clearly a non-trivial task, but perhaps also due the aforementioned manifold of complicated (from the formal point of view) definitions. Further developing the theory, finally, is the main motivation and the long term goal of the project.

Before this goal can be achieved, however, the existing part has to be formalized and formally verified first, which is what will be presented in the rest of the paper. Still, one must also mention that the exploration of reduction ring theory in *Theorema* was started only recently and thus many things have not been completed yet: although the formal representation of the theory by means of a structured *Theorema* document is mostly finished, only a few results have been mechanically proved so far. Nevertheless, the completed proofs (one of them being sketched in Section 5) give a good impression of how semi-automated theorem proving in *Theorema* usually proceeds.

The structure of this paper is as follows: Section 2 very briefly reviews reduction rings and their relation to Gröbner bases. Section 3 contains an overview of the *Theorema* system, and Section 4 presents the *special prover* designed and implemented for the computer-supported verification of the formalization of reduction ring theory in *Theorema*. Afterward, Section 5 describes the outline of this formalization and demonstrates how computer-supported theorem proving in *Theorema* usually proceeds, by means of a concrete example. Finally, Section 6 summarizes the contents of this paper and provides an outlook over possible extensions of the work presented here.

2 Reduction Rings

The notion of *reduction ring* was introduced by Bruno Buchberger [1] in 1984 and later generalized and extended by Sabine Stifter [12,13]. In short, a reduction ring R is a commutative ring with unit, where in addition a Noetherian order relation $\prec$ and, for each ring element c, a subset $M_c \subseteq R$ are defined. The M_c are the sets of *multipliers*, i. e. they consist of those elements c may be multiplied with when reducing an arbitrary ring element modulo c. If $\prec$ and M_c have certain properties, and thus R is a reduction ring, then Gröbner bases of finitely generated ideals in R can be computed algorithmically from any given finite ideal basis. Moreover, if R is a reduction ring, then also $R[X]$ (the polynomial ring in indeterminates $X = \{x_1, \ldots, x_n\}$ over R) and R^n (the n-fold direct product of R) can be made reduction rings by defining $\prec$ and M_c appropriately. This already indicates that reduction rings do not necessarily have any polynomial structure. Furthermore, reduction rings not even have to be free of zero divisors;

for instance, $\mathbb{Z}_m$, the ring of residue classes modulo m, constitutes a reduction ring for *any* (not necessarily prime) m. Other examples of reduction rings are $\mathbb{Z}$, all fields, and thus also $\mathbb{Z}^n$, $K[X]$ (for fields K), etc.

It has to be remarked that many other generalizations of Gröbner bases theory, both in the commutative and non-commutative case, have been proposed by various authors (see, e. g., [1] for a review of commutative generalizations), and that the term "reduction ring" could thus also be understood in a broader sense, simply as a ring where a "reduction operation" is defined (in whatever way). However, the term "reduction ring" will always refer to reduction rings according to [13] in this paper.

2.1 Gröbner Bases in Reduction Rings

Many different characterizations of Gröbner bases exist in the literature, where most of them turn out to be equivalent if restricted to certain domains. For instance, in the classical setting of polynomials over a field, Gröbner bases can be characterized as sets G such that

(G1) every leading power-product of polynomials in $\langle G \rangle$ (i. e. the ideal generated by G) is a multiple of the leading power-product of at least one polynomial in G, or
(G2) the *reduction relation* modulo G is Church-Rosser, or
(G3) All S-polynomials of elements in G can be reduced to 0 modulo G.

It is clear that in reduction rings there is no analogue of (G1): elements of reduction rings, in general, cannot be decomposed into a "leading part" and into a "rest". Hence, the only candidates remaining are (G2) and (G3), and actually (G2) is taken as the definition of Gröbner bases in reduction rings. In the sequel, R is always assumed to be a reduction ring.

Definition 1 (Gröbner basis). *Let $G \subseteq R$ be finite. G is a* Gröbner basis *of the ideal it generates iff the reduction relation modulo G, i. e. $\rightarrow_G$, is Church-Rosser.*

Definition 1 provides an algebraic characterization of Gröbner bases, but not an algorithmic one: given a finite set $B \subseteq R$, it is in general not possible to find out *algorithmically* whether B is a Gröbner basis or not, and furthermore, to compute a Gröbner basis G of the ideal generated by B (if it exists, which is not yet clear either). Hence, what is needed is an analogue of (G3) in reduction rings, and indeed there is one.

Theorem 1 (Main Theorem). *Let $G \subseteq R$ be finite. Then G is a Gröbner basis of the ideal it generates iff for all $g_1, g_2 \in G$ (not necessarily distinct), all $i, j \in I_{g_1}$, and all non-trivial common reducibles a of g_1 and g_2 w. r. t. (i, j) there exists a critical pair (b_1, b_2) for g_1 and g_2 w. r. t. a and (i, j), such that $b_1 \leftrightarrow_G^{\prec a} b_2$.*

Unfortunately, due to the space limitations imposed on this paper, we cannot go further into the details of Theorem 1, nor provide the formal (and slightly technical) definitions of *non-trivial common reducibles*, *critical pair*, $\rightarrow_G$ and $\leftrightarrow_G^{\prec a}$; the interested reader is referred to [1,12,13] instead. Constructing a completely formal *Theorema* proof of Theorem 1 is one of the goals of the formal treatment of reduction ring theory described in this paper.

As in the classical setting of polynomials over a field, Theorem 1 provides both an algorithmic criterion for deciding whether a given set G is a Gröbner basis, and also for computing a Gröbner basis for the ideal generated by G in case it is not. The algorithm is almost exactly the same critical-pair/completion algorithm as Buchberger's algorithm in the classical setting, with three minor modifications:

- Not only pairs of *distinct* elements g_1, g_2 of G have to be considered, but also pairs where both constituents are identical.
- For a given pair (g_1, g_2), *all* minimal non-trivial common reducibles a (for *all* indices $i \neq j \in I_{g_1}$ if $g_1 = g_2$) have to be considered, not just one. Still, there are only finitely many, and considering only *one* critical pair (b_1, b_2) for g_1 and g_2 w. r. t. a and (i, j) is sufficient.
- Instead of reducing the difference $b_1 - b_2$ of b_1 and b_2 to normal form, both b_1 and b_2 have to be reduced separately to normal form. If the two normal forms are not identical, their difference has to be added to the basis.

Of course, the algorithm is only an algorithm relative to the computability of the basic ring operations, as well as the computability of normal forms and minimal non-trivial common reducibles. This can either be required by the axioms of reduction rings, as it is done in [1,12,13], or left as an additional degree of freedom. In the latter case, which is how it is done in the *Theorema*-formalization underlying this paper, one then has to distinguish between *algorithmic* and *non-algorithmic* reduction rings. All examples of reduction rings mentioned at the beginning of this section, however, are algorithmic, and furthermore being an algorithmic reduction ring carries over from R to $R[X]$ and to R^n.

3 The *Theorema* System – A Short Overview

Theorema [4,19] is a system for computer-supported mathematical theory exploration, conceived in the mid-90s by Bruno Buchberger and now developed under his guidance at RISC. One of the major goals of the project has ever since been the seamless integration of *proving*, *computing* and *solving* within one single software system, and thus supporting the working mathematician in his everyday life.

Recently, a completely new version of *Theorema* was released, called *Theorema* 2.0 [18]. Although the main design principles have not changed, and it is still based on *Mathematica* [20], the software was entirely redesigned and reimplemented from scratch, with substantial improvements compared to the previous version at all levels of its architecture. Since the research presented in this paper

was entirely carried out in *Theorema* 2.0, here and henceforth *Theorema* will always refer to *Theorema* 2.0 unless explicitly stated otherwise.

Besides the formal treatment of the theory of reduction rings presented in this paper, there are also other theories in the area of computer algebra developed with *Theorema*, which are worth being mentioned here in order to illustrate the capabilities of the system. All of them were developed in *Theorema* 1, but can easily be updated to function under the new version as well: the symbolic treatment of linear boundary problems by means of Green's functions and Green's operators, for instance, was developed by Markus Rosenkranz [10] with (at least partial) support from *Theorema*, and moreover, the resulting algorithm was implemented in the system in a generic, highly-structured way (following the *Theorema*-functor approach), see [14]. A second example of a non-trivial problem solved with *Theorema* is the problem of automatically synthesizing Buchberger's algorithm for computing Gröbner bases only from its specification, following the principle of *Lazy Thinking* [3,5].

An important concept in *Theorema* is the concept of *domains – functors – categories* [17,2]. In order for no confusion to arise, it must be pointed out already here that the terms *functor* and *category* have a slightly different meaning than in classical category theory, to be explained below. Also, the term *domain* does not necessarily refer to a ring without zero divisors (as usually in algebra), but rather to a general algebraic structure formed by a carrier together with operations defined on it.
In *Theorema*, a functor is essentially a function mapping domains (and possibly other entities) to domains, where in turn a domain is characterized by a carrier and operations. Hence, a functor typically takes as input a domain $\mathcal{A}$ and constructs a new domain $\mathcal{B}$, by defining $\mathcal{B}$'s carrier and operations in terms of $\mathcal{A}$'s carrier and operations. Moreover, in order to make things more compact, the carrier of a domain is usually not represented as a set, but as a further operation, namely a unary decision predicate that decides for any given object x whether x belongs to the respective domain or not. One of the simplest examples of a functor is the functor $\mathcal{DP}$ which takes as input two domains $\mathcal{D}_1$ and $\mathcal{D}_2$ and constructs the direct product $\mathcal{P}$ of $\mathcal{D}_1$ and $\mathcal{D}_2$, where all operations of $\mathcal{P}$ are defined component-wise in terms of the operations of the $\mathcal{D}_i$. Categories, finally, are classes of domains sharing common properties, e. g. the category of all commutative rings with unit, the category of all fields, and so on.

The formal development of reduction ring theory presented in this paper also follows the functor approach. Natural candidates for functors are, of course, the $\mathcal{POLY}$ functor that constructs the ring of multivariate polynomials over a given coefficient domain and power-product domain. The most important category then, obviously, is the category of reduction rings as described in Section 2.

4 The ReductionRingProver Special Prover

As many other systems for mathematical theory exploration, *Theorema* is not specialized to work only in one single mathematical domain, e. g. in algebra or

geometry, but its design allows for completely arbitrary mathematical content, formulated in the language of *higher-order predicate logic and set theory*, to be treated by the system. Still, this does not mean there is no specialization at all, but the specialization happens at a different level: an important aspect of the philosophy behind *Theorema*, and one that distinguishes it from other systems, is the idea that exploring a theory does not only happen at the object level, but also at the meta level. This means that when working in a certain theory T one should not have to fall back to the very elementary and general proving techniques of predicate logic all the time, but rather use more advanced techniques that eventually lead to more elegant and shorter proofs which ideally even resemble the way human mathematicians would proceed. These advanced techniques, however, might only be correct in T but not in general.

A concrete example for a theory T and special proving techniques is given by geometric theorem proving, i. e. where T can be thought of as the theory of real numbers with addition and multiplication. It is a well-known fact that Gröbner bases can be used for proving statements in T, simply by finding out whether a certain polynomial identity follows from a system of algebraic equations or not. Now, a mathematician working in this area most probably would like to automate this process, or more precisely, type in the statement he wants to prove into a computer system that internally automatically uses the method of Gröbner bases for proving or disproving it – and this is possible in *Theorema* by creating a *special prover* for geometric theorem proving. Such a prover will directly use Gröbner bases on the inference level and hence prove theorems in geometry automatically in a short and elegant way, precisely as desired by the human mathematician.

It has to be pointed out that a geometry-prover of the form sketched above does not yet exist in *Theorema* 2.0 (it was available in the old version of the system, though; see [8,9]). Instead, in the sequel another *Theorema* special prover, created for the treatment of the theory of reduction rings, is presented.

In *Theorema*, a prover consists of two parts which are mainly independent of each other: a collection of *inference rules* and a proving *strategy*. The inference rules describe how a certain proof situation, characterized by a set of assumptions and a proof goal, can be transformed into one or more simpler proof situations. The proving strategy guides the application of the inference rules, i. e. it specifies in which order the rules are tried, what to do if more than one rules are applicable, etc.

The proving strategy used for verifying the theory of reduction rings is a *fully interactive* strategy. This means that in each step the human operator has full control over the whole proof search: he decides which inference to perform (and in which way) and at which position in the proof to proceed. Nevertheless, if he feels that the current proof situation is simple enough for *Theorema* to automatically find a proof, it is still possible to trigger an automatic proof search as well.

Regarding inference rules, the ReductionRingProver actually does not consist of that many inference rules being special in the sense that they can only

be used in the theory of reduction rings, but not elsewhere; they are presented in the next two subsections.

One remark is still in place: Most functions and relations, e. g. $\prec$, are only defined for arguments of a particular domain $\mathcal{D}$. Since *Theorema* is not typed, however, a proof situation may well contain functions/relations applied on arguments *not* in that particular domain, leading to undefined expressions. Therefore, in order to reason correctly, domain-membership of all terms involved in an inference step must always be checked explicitly. In *Theorema*, membership of x in $\mathcal{D}$ is usually denoted by $\underset{\mathcal{D}}{\in}[x]$, and $\underset{\mathcal{D}}{\in}[x_1, \ldots, x_n]$ abbreviates $\bigwedge_{i=1,\ldots,n} \underset{\mathcal{D}}{\in}[x_i]$.

4.1 Order Relations

One of the ubiquitous objects in the theory of reduction rings are order relations of all kinds: every reduction ring is ordered by an arbitrary *partial* order relation, power products are ordered by *admissible total* orderings, and the divisibility relation on power products is a *monotonic* (w. r. t. multiplication) partial ordering. Moreover, some of the orderings have been defined as *irreflexive* and *asymmetric*, others as *reflexive* and *antisymmetric*. Although both kinds of order relations are more or less equivalent to each other (one can always make a reflexive ordering out of an irreflexive one, and vice versa), from the point of view of theorem proving it is desirable to be able to handle both kinds directly, without the need for any conversion taking place beforehand.

Let in the sequel $\prec$ always be defined on the domain $\mathcal{D}$. The following are the three ordering inference rules:

OrderingGoal If the proof goal is of the form $x \prec y$ or $\neg x \prec y$, various attempts for simplifying the goal are made, depending on whether the ordering $\prec$ is partial or total, and whether it is reflexive or irreflexive. For instance, if $\prec$ is any order relation, we have

$$\frac{\mathrm{K} \ \vdash \underset{\mathcal{D}}{\in}[x, y, z_1, \ldots, z_n]}{\mathrm{K}, x \prec z_1, z_1 \prec z_2, \ldots, z_{n-1} \prec z_n, z_n \prec y \ \vdash \ x \prec y}$$

by transitivity. Similarly, if $\prec$ is asymmetric, we have

$$\frac{\mathrm{K} \ \vdash \underset{\mathcal{D}}{\in}[x, y]}{\mathrm{K}, y \prec x \ \vdash \ \neg x \prec y}$$

All inferences of that kind are incorporated in the *OrderingGoal* inference rule in the ReductionRingProver.

OrderingKB If the set of assumptions of the current proof situation contains a formula of the form $x \prec x$ and $\prec$ is an irreflexive ordering, then the assumptions are apparently contradictory, so the proof is finished. This gives rise to the

following two inferences, both incorporated in the *OrderingKB* inference rule in the ReductionRingProver:

$$\frac{\mathrm{K} \ \vdash \underset{\mathcal{D}}{\in} [x]}{\mathrm{K}, x \prec x \ \vdash \ \Gamma}$$

if $\prec$ is irreflexive, and

$$\frac{\mathrm{K} \ \vdash \underset{\mathcal{D}}{\in} [x]}{\mathrm{K}, \neg x \prec x \ \vdash \ \Gamma}$$

if $\prec$ is reflexive.

OrderingEqualGoal If the proof goal is of the form $x = y$ and both x and y are elements of a domain that is totally ordered by $\prec$, then it suffices to prove both $\neg x \prec y$ and $\neg y \prec x$. Similarly, if $\prec$ is antisymmetric, it suffices to prove both of $x \prec y$ and $y \prec x$. This gives rise to the following two inferences, both incorporated in the *OrderingEqualGoal* inference rule in the ReductionRingProver:

$$\frac{\mathrm{K} \ \vdash \underset{\mathcal{D}}{\in} [x, y] \qquad \mathrm{K} \ \vdash \ \neg x \prec y \qquad \mathrm{K} \ \vdash \ \neg y \prec x}{\mathrm{K} \ \vdash \ x = y}$$

if $\prec$ is total, and

$$\frac{\mathrm{K} \ \vdash \underset{\mathcal{D}}{\in} [x, y] \qquad \mathrm{K} \ \vdash \ x \prec y \qquad \mathrm{K} \ \vdash \ y \prec x}{\mathrm{K} \ \vdash \ x = y}$$

if $\prec$ is antisymmetric.

4.2 Commutative Rings with Unit

Since every reduction ring by definition is a commutative ring with unit, it is desirable to have inference rules incorporating the logical axioms of the ring operations $+$ and $\cdot$ in a compact and easy-to-use way, such that it is not necessary to fall back to the very definitions of commutativity and distributivity to prove that, say, $x \cdot y + y \cdot z$ and $x \cdot (z + y)$ are equal. Rather, all this should happen fully automatically whenever the proof goal is an equality of two terms, where the outermost function symbol of at least one of the terms is $+$, $\cdot$, $-$ (the additive inverse) or $\sum$ (since also sums over several ring elements play an important role in reduction ring theory).

Fix now a commutative ring with unit $\mathcal{R}$ as the domain of discourse, i. e. $+$, $\cdot$, etc. are functions on $\mathcal{R}$. The following are the two special inference rules for commutative rings with unit (keep in mind that domain-membership in $\mathcal{R}$ of x and y in $x + y$ is not guaranteed and always has to be checked explicitly):

MembershipCommRings1 If one has to prove membership of a certain term in a commutative ring with unit, several properties of the functions $+$, $\cdot$, etc. are used in order to simplify the proof goal. Examples of inferences are

$$\frac{\mathrm{K} \vdash \underset{\mathcal{R}}{\in} [x] \qquad \mathrm{K} \vdash \underset{\mathcal{R}}{\in} [y]}{\mathrm{K} \vdash \underset{\mathcal{R}}{\in} [x+y]}$$

and

$$\frac{\mathrm{K} \vdash a \in \mathbb{Z} \qquad \mathrm{K} \vdash b \in \mathbb{Z} \qquad \mathrm{K} \vdash \underset{i=a,\ldots,b}{\forall} \underset{\mathcal{R}}{\in} [f(i)]}{\mathrm{K} \vdash \underset{\mathcal{R}}{\in} [\sum_{i=a,\ldots,b} f(i)]}$$

All inferences of that kind are incorporated in the *MembershipCommRings1* inference rule of the ReductionRingProver. In fact, since membership of certain terms in commutative rings with unit has to be proved very often, this is one of the most important special inference rules.

CommRing1Equal If the proof goal is of the form $a = b$, where the outermost function symbol of a or b is among $+$, $\cdot$, $-$ and $\sum$, both a and b are fully expanded using associativity and distributivity of the functions involved, and then the resulting terms are checked for being equal, further using commutativity of $+$ and $\cdot$. Of course, associativity/commutativity/distributivity of $+$ and $\cdot$ may only be exploited if the arguments of the respective functions belong to $\mathcal{R}$, which is checked analogously to the ordering-rules.

5 Formalized Reduction Rings

Theorema is a system for mathematical theorem exploration, opposed to isolated theorem proving. Thus, working in *Theorema* usually proceeds by developing a whole theory for what one wants to do, consisting of definitions, lemmas, theorems, computations, etc., all included in one or more *Theorema* notebooks and resembling the way how mathematical knowledge is presented in textbooks and articles. The theorems can be proved using definitions, lemmas, and special inference rules (see Section 4), and may then be used for carrying out sample computations or proving other theorems (although no restrictions are imposed on the order in which the theorems are proved: an unproved statement may well serve as knowledge for proving another statement).

5.1 Structure of the Formalization

We developed a formal *Theorema* theory for the theory of reduction rings. For this, we first introduced the category (in the sense of Section 3) **ReductionRing**. Having this category it is already possible to define the notion of Gröbner bases and state the Main Theorem of reduction ring theory, containing a finite criterion for checking whether a given set is a Gröbner basis or not.

We also introduced the categories **CommPPDomain** of (commutative) power-products and **ReductionPolynomialDomain** of polynomials over a coefficient domain $\mathcal{R}$ and a power-product domain, where objects like the Noetherian ordering[1] $\prec$ are defined in terms of the respective objects in $\mathcal{R}$ in such a way that if $\mathcal{R}$ is a reduction ring, then so is the polynomial ring. Moreover, the theorem containing this claim and the various lemmas needed for its proof are already part of the formalization as well.

Complementing the purely theoretical concepts of categories and theorems, the formalization consists of computational parts, too. In particular, it contains several *Theorema* functors for constructing concrete reduction rings, e. g. a functor that turns an arbitrary field into a reduction ring by endowing it with a suitable Noetherian order relation $\prec$ and other objects needed in reduction rings. Another example of a functor is the functor that turns a reduction ring $\mathcal{R}$ and a power-product domain $\mathcal{T}$ into the polynomial ring over $\mathcal{R}$ and $\mathcal{T}$, again endowed with suitable reduction-ring-objects (note that the resulting domain then belongs to category **ReductionPolynomialDomain**). Most importantly, however, there is a functor $\mathcal{GB}$ that takes as input an *algorithmic* reduction ring $\mathcal{R}$ and returns a new ring, where in addition also a function gb for computing Gröbner bases is defined. gb, of course, implements the critical-pair/completion algorithm sketched in Section 2.1 in terms of the operations of $\mathcal{R}$.

Regarding the formal verification of the formalization, we decided to begin with proving that if $\mathcal{P}$ is in category **ReductionPolynomialDomain**, the coefficient domain is in category **ReductionRing**, and the power-product domain is in category **CommPPDomain**, then also $\mathcal{P}$ is in category **ReductionRing**. Although this undertaking has been started only recently and, hence, it is not finished yet, first results, namely completely formal, machine-generated proofs of important lemmas, have already been achieved. One of them is presented in the following subsection.

Of course, the ultimate goal is the formal verification of *all* of reduction ring theory, not only of what is mentioned above. This is still future work.

5.2 Example: Noetherianity of $\prec$ in Polynomial Rings

As an example, we present here the formal statement of the theorem that $\prec$ in domains belonging to **ReductionPolynomialDomain** is Noetherian, and also sketch the main ideas behind its *Theorema*-generated proof.

Let in the sequel always $\mathcal{R}$ be a reduction ring, $\mathcal{T}$ a commutative power-product domain, and $\mathcal{P}$ the polynomial ring over $\mathcal{R}$ and $\mathcal{T}$. Since we have to refer to $\prec$ in $\mathcal{R}$, in $\mathcal{T}$ and in $\mathcal{P}$, we denote the relations by $\underset{\mathcal{R}}{\prec}$, $\underset{\mathcal{T}}{\prec}$ and $\underset{\mathcal{P}}{\prec}$, respectively, according to syntactic conventions regarding domain operations in *Theorema*. Before we can give the definition of $\underset{\mathcal{P}}{\prec}$, we need the auxiliary notions of H, lp and lc. The following is their definition in *Theorema* notation:

[1] Actually, the whole formalization is based upon the reverse relation $\succ$, but this should not lead to any confusion.

Definition 2 (H, lp, lc).

$$\underset{\substack{\in[p]\\ \mathcal{P}}}{\forall}$$

$$\underset{\substack{\in[\tau,\sigma]\\ \mathcal{T}}}{\forall} \mathrm{C}[\,\mathrm{H}[\,p,\tau],\sigma] \;=\; \begin{cases} \mathrm{C}[\,p,\sigma] \Leftarrow & \tau \underset{\mathcal{T}}{\prec} \sigma \\ 0 \quad \Leftarrow \text{otherwise} \end{cases} \tag{H}$$

$p \neq 0 \Rightarrow$

$$\mathrm{lp}[\,p] \;:=\; \underset{\substack{\in[\tau]\\ \mathcal{T}}}{\mathrm{the}}\ (\mathrm{C}[\,p,\tau] \neq 0 \wedge \underset{\substack{\in[\sigma]\\ \mathcal{T}}}{\forall}\ \tau \underset{\mathcal{T}}{\prec} \sigma \Rightarrow \mathrm{C}[\,p,\sigma] = 0) \tag{lp}$$

$$\mathrm{lc}[\,p] \;:=\; \mathrm{C}[\,p,\mathrm{lp}[\,p]] \tag{lc}$$

Some remarks on Definition 2 are in place:

- $\mathrm{C}[\,p,\tau]$ denotes the coefficient of polynomial p at power-product τ. It cannot be defined in general for arbitrary polynomial domains, but rather has to be defined in each concrete domain.
- $\mathrm{H}[\,p,\tau]$ denotes the *higher part* of p w. r. t. τ, i. e. the sum of those monomials of p whose power-product is strictly greater than τ. It is not defined explicitly, but only implicitly in terms of C.
- $\mathrm{lp}[\,p]$ denotes the *leading power-product* of p, given that p is non-zero. It is defined using *Theorema*'s "the" quantifier.
- $\mathrm{lc}[\,p]$, finally, denotes the *leading coefficient* of p.
- The first and the third line in the definition are so-called *global declarations*. They are simply prepended to all subsequent formulas, in order to ease reading and writing formulas in *Theorema*.
- Strictly speaking, not only the various versions of $\prec$ carry under-scripts in the formalization, but also all of C, H, etc. are under-scripted with $\mathcal{P}$, to explicitly state that they belong to domain $\mathcal{P}$. Here, these under-scripts are omitted for better readability.

Now we are able to define $\underset{\mathcal{P}}{\prec}$, again in *Theorema* notation:

Definition 3 ($\underset{\mathcal{P}}{\prec}$).

$$\underset{\substack{\in[p,q]\\ \mathcal{P}}}{\forall}$$

$$p \underset{\mathcal{P}}{\prec} q \;:\Leftrightarrow\; \underset{\substack{\in[\tau]\\ \mathcal{T}}}{\exists}\ (\mathrm{H}[\,p,\tau] = \mathrm{H}[\,q,\tau] \wedge \mathrm{C}[\,p,\tau] \underset{\mathcal{R}}{\prec} \mathrm{C}[\,q,\tau]) \tag{$\underset{\mathcal{P}}{\prec}$}$$

In the theorem claiming that $\underset{\mathcal{P}}{\prec}$ is Noetherian, one does not even need that $\mathcal{R}$ is a reduction ring, but only that $\underset{\mathcal{R}}{\prec}$ is a partial Noetherian ordering and that $\underset{\mathcal{T}}{\prec}$ is a total Noetherian ordering (which is implied by $\mathcal{R}$ being a reduction ring and $\mathcal{T}$ being a power-product domain, though).

Instead of presenting the theorem, which should be apparent, we present the key lemma for proving it, together with its proof:

Lemma 1 (Lemma for proving Noetherianity of $\underset{\mathcal{P}}{\prec}$).

$$\underset{\in[\tau]}{\underset{\mathcal{T}}{\forall}} \quad \underset{\in}{\underset{\text{DomainSets}[\mathcal{P}]}{\forall}}[A]$$

$$\left(A \neq \{\} \wedge \underset{p \in A}{\forall} p \neq 0 \Rightarrow \text{lp}[p] \underset{\mathcal{T}}{\prec} \tau\right) \Rightarrow \underset{p \in A}{\exists} \text{isMin}[\,p, A, \underset{\mathcal{P}}{\prec}] \tag{1}$$

DomainSets is a general *Theorema* functor returning the domain of all sets of elements belonging to the input domain $\mathcal{D}$, without any operations other than $\underset{\text{DomainSets}[\mathcal{D}]}{\in}$. $\text{isMin}[\,p, A, \underset{\mathcal{P}}{\prec}]$ states that no element in A is strictly less than p w. r. t. $\underset{\mathcal{P}}{\prec}$. Since its formal definition should be obvious, it is spared in this paper.

The *Theorema*-generated proof of Lemma 1 essentially proceeds by Noetherian induction on the power-product τ (recall that $\underset{\mathcal{T}}{\prec}$ is Noetherian), to be explained now step-by-step:

1. Perform Noetherian induction on τ, i. e. choose $\overline{\tau}$ and $\overline{A}$ arbitrary but fixed, assume

$$\overline{A} \neq \{\} \tag{A\#1}$$

$$\underset{p \in \overline{A}}{\forall} p \neq 0 \Rightarrow \text{lp}[p] \underset{\mathcal{T}}{\prec} \overline{\tau} \tag{A\#2}$$

$$\underset{\in[\tau]}{\underset{\mathcal{T}}{\forall}} \tau \underset{\mathcal{T}}{\prec} \overline{\tau} \Rightarrow \underset{\in}{\underset{\text{DomainSets}[\mathcal{P}]}{\forall}}[A] \quad \ldots \tag{IH}$$

   and prove

$$\underset{p \in A}{\exists} \text{isMin}[\,p, A, \underset{\mathcal{P}}{\prec}] \tag{G\#1}$$

   Formula (IH) of course is the induction hypothesis. Note that the principle of Noetherian induction does not have to be implemented as a special inference rule, but can be stated as a higher-order formula on the object level and then used like an inference rule on the meta level, thanks to the way how *Theorema* employs certain kinds of formulas (e. g. universally quantified implications) as rewrite-rules in proofs.
2. Distinguish two cases, based upon whether $0 \in \overline{A}$ or not. If $0 \in \overline{A}$ then 0 apparently witnesses the existential goal (G#1), so assume now $0 \notin \overline{A}$. From (A#2) we can readily infer

$$\underset{p \in \overline{A}}{\forall} p \neq 0 \wedge \text{lp}[p] \underset{\mathcal{T}}{\prec} \overline{\tau} \tag{A\#3}$$

3. Consider the set $P := \{\text{lp}[p] \underset{p \in \overline{A}}{|}\}$. P apparently is the non-empty (due to (A#2)) set consisting of all leading power-products of elements in $\overline{A}$, and therefore it contains a minimal element σ due to the Noetherianity of $\underset{\mathcal{T}}{\prec}$. We know

$$\text{isMin}[\,\sigma, P, \underset{\mathcal{T}}{\prec}] \tag{A\#4}$$

as well as

$$\sigma \underset{\mathcal{T}}{\prec} \overline{\tau} \tag{A\#5}$$

from (A#3).

4. Consider the set $C := \{\mathrm{lc}[p] \mid \underset{p \in \overline{A}}{} \mathrm{lp}[p] = \sigma\}$. As can easily be seen, C is a non-empty set consisting of elements of the coefficient domain $\mathcal{R}$, and since $\underset{\mathcal{R}}{\prec}$ is Noetherian, it contains a minimal element c. Thus we know

$$\mathrm{isMin}[\, c, C, \underset{\mathcal{R}}{\prec}] \tag{A\#6}$$

5. Consider the set $R := \{p - c \cdot \sigma \mid \underset{p \in \overline{A}}{} \mathrm{lp}[p] = \sigma \wedge \mathrm{lc}[p] = c\}$. R is again non-empty, consists of polynomials, and moreover satisfies

$$\underset{p \in R}{\forall}\ p \neq 0 \Rightarrow \mathrm{lp}[p] \underset{\mathcal{T}}{\prec} \sigma \tag{A\#7}$$

because all the leading monomials cancel by construction. This means that we can now use our induction hypothesis (IH), instantiated by $\tau \leftarrow \sigma$ and $A \leftarrow R$, and infer

$$\mathrm{isMin}[\, \overline{p}, R, \underset{\mathcal{P}}{\prec}] \tag{A\#8}$$

from (A#5) and (A#7), for some $\overline{p} \in R$.

6. Instantiate the existentially quantified goal (G#1) by $p \leftarrow \overline{p} + c \cdot \sigma$, and prove

$$\mathrm{isMin}[\, \overline{p} + c \cdot \sigma, \overline{A}, \underset{\mathcal{P}}{\prec}] \tag{G\#2}$$

which can be accomplished using Definition 3 together with (A#4), (A#6) and (A#8) (we spare the details).

The proof was generated interactively with the assistance of *Theorema*, following the six steps sketched above. Furthermore, ordering-rules, as described in Section 4.1, were used to abbreviate otherwise tedious inferences regarding, e. g., transitivity of $\underset{\mathcal{T}}{\prec}$.

Please note that "interactive proving" in *Theorema* does not mean to write down the proof in sufficient detail and afterward let the system check all inferences and fill small gaps, but rather to initiate a proof attempt where the system simply asks the human user for support whenever it does not know how to proceed, in a dialog-oriented manner. Nevertheless, as soon as the proof is finished (either with success or with failure), a nicely-structured, human readable proof document describing each single step is generated fully automatically, where formal contents are interspersed with informal explanatory text in English (or any other language). In other words, the proof outlined above is rather the *output* than the *input* of interactive proving in *Theorema*.

6 Conclusion and Future Work

The preceding sections illustrated how the formal treatment of reduction ring theory in *Theorema* can be conducted. On the object level, this consists of providing suitable definitions for all notions and concepts of the theory. On the inference level, it is desirable to have specialized inference rules at one's disposal for efficiently proving statements about all kinds of order relations and equality of terms in commutative rings with unit, such that the resulting proofs become short and elegant.

One of the long-term goals of the project is to extend the theory of reduction rings, but also of Gröbner bases in general, in several ways, making use of the existing formalization:

- Introduce non-commutative reduction rings and prove an analogue of Theorem 1.
- Find further basic reduction rings and functors that conserve the property of being a reduction ring.
- Investigate the relation of Gröbner bases (in polynomial rings over fields) and generalized Sylvester matrices, such that Gröbner bases can effectively be computed by triangularizing coefficient-matrices [16].

These possible extensions are among the main motivations for a formal treatment of reduction ring theory. Proofs in this theory tend to be lengthy, tedious (with many case distinctions etc.) and very technical, but still comparatively straightforward. Therefore, having a software system supporting the human mathematician in all aspects of theory exploration, being in *automatically* proving simple lemmas, keeping track of cases still to be considered in long proofs, or even carrying out test-computations with notions just introduced, certainly is a great benefit.

Acknowledgments I gratefully acknowledge the valuable discussions about formal mathematics and Gröbner bases with my PhD adviser Bruno Buchberger, and about *Theorema* with Wolfgang Windsteiger.
This research was funded by the Austrian Science Fund (FWF): grant no. W1214-N15, project DK1.

References

1. Buchberger, B.: A Critical-Pair/Completion Algorithm for Finitely Generated Ideals in Rings. In: Börger, E., Hasenjäger, G., Rödding, D. (eds.) Logic and Machines: Decision Problems and Complexity. LNCS, vol. 171, pp. 137–161, Springer-Verlag Berlin Heidelberg (1984)
2. Buchberger, B.: Gröbner Rings in Theorema: A Case Study in Functors and Categories. Tech. Rep. 2003-49, RISC Report Series, Johannes Kepler University Linz, Austria (2003)
3. Buchberger, B.: Towards the Automated Synthesis of a Gröbner Bases Algorithm. RACSAM (Revista de la Real Academia de Ciencias), Serie A: Mathematicas, vol. 98(1), 65–75 (2004)

4. Buchberger, B., Craciun, A., Jebelean, T., Kovacs, L., Kutsia, T., Nakagawa, K., Piroi, F., Popov, N., Robu, J., Rosenkranz, M., Windsteiger, W.: Theorema: Towards Computer-Aided Mathematical Theory Exploration. J. Applied Logic 4(4), 470–504 (2006)
5. Craciun, A.: Lazy Thinking Algorithm Synthesis in Gröbner Bases Theory. PhD thesis, RISC, Johannes Kepler University Linz, Austria (2008)
6. Jorge, J. S., Guilas, V. M., Freire, J. L.: Certifying properties of an efficient functional program for computing Gröbner bases. J. Symbolic Computation 44(5), 571–582 (2009)
7. Medina-Bulo, I., Palomo-Lozano, F., Ruiz-Reina, J.-L.: A verified Common Lisp implementation of Buchberger's algorithm in ACL2. J. Symbolic Computation 45(1), 96–123 (2010)
8. Robu, J., Ida, T., Tepeneu, D., Takahashi, H., Buchberger, B.: Computational Origami Construction of a Regular Heptagon with Automated Proof of its Correctness. In: Hong, H., Wang, D. (eds.) ADG 2004 (Automated Deduction in Geometry). LNCS, vol. 3763, pp. 19–33, Springer-Verlag Berlin Heidelberg (2006)
9. Robu, J.: Automated Proof of Geometry Theorems Involving Order Relation in the Frame of the Theorema Project. In: Pop, H. F. (ed.) KEPT 2007 (Knowledge Engineering, Principles and Techniques). Special issue of Studia Universitatis "Babes-Bolyai", Series Informatica, pp. 307–315 (2007)
10. Rosenkranz, M.: A New Symbolic Method for Solving Linear Two-point Boundary Value Problems on the Level of Operators. J. Symbolic Computation 39(2), 171–199 (2005)
11. Schwarzweller, C.: Gröbner Bases – Theory Refinement in the Mizar System. In: Kohlhase, M. (ed.) MKM 2005 (Mathematical Knowledge Management). LNAI, vol. 3863, pp. 299–314, Springer-Verlag Berlin Heidelberg (2006)
12. Stifter, S.: Computation of Gröbner Bases over the Integers and in General Reduction Rings. Diploma thesis, Institut für Mathematik, Johannes Kepler University Linz, Austria (1985)
13. Stifter, S.: The Reduction Ring Property is Hereditary. J. Algebra 140(89-18), 399–414 (1991)
14. Tec, L.: A Symbolic Framework for General Polynomial Domains in Theorema: Applications to Boundary Problems. PhD thesis, RISC, Johannes Kepler University Linz, Austria (2011)
15. Thery, L.: A Machine-Checked Implementation of Buchberger's Algorithm. J. Automated Reasoning 26, 107–137 (2001)
16. Wiesinger-Widi, M.: Gröbner Bases and Generalized Sylvester Matrices. PhD thesis, RISC, Johannes Kepler University Linz, Austria (to appear, 2015)
17. Windsteiger, W.: Building Up Hierarchical Mathematical Domains Using Functors in Theorema. In: Armando, A., Jebelean, T. (eds.) Calculemus 1999. ENTCS, vol. 23, pp. 401–419, Elsevier (1999)
18. Windsteiger, W.: Theorema 2.0: A System for Mathematical Theory Exploration. In: Yap, C., Hong, H. (eds.) ICMS 2014. LNCS, vol. 8592, pp. 49–52, Springer-Verlag Berlin Heidelberg (2014)
19. The *Theorema* System. `http://www.risc.jku.at/research/theorema/software/`
20. Wolfram *Mathematica*. `http://www.wolfram.com/mathematica/`