Exploring Reduction Ring Theory in Theorema*

Alexander Maletzky

DK Computational Mathematics

Johannes Kepler University Linz, Austria

alexander.maletzky@dk-compmath.jku.at

July 2015

Abstract

This report presents the formal treatment of the theory of reduction rings in the Theorema system. We describe how the formalization is systematically structured into sub-theories and how we carried out the proofs with assistance of Theorema. For this, we devote a whole section to the special inference rules designed specifically for the verification of the theory. In addition, we also review the most important notions related to reduction rings and explain why one of them, the notion of *irrelativity* of reductors, had to be replaced by a slightly adjusted version.

Keywords: Computer-supported theory exploration, automated reasoning, reduction rings, Gröbner bases, Theorema

1 Introduction

The computer-supported exploration of a mathematical theory aims at first formalizing and then formally verifying, by means of automated or interactive proving, the theory in a computer system. In our case, the system of choice is Theorema [4, 5, 6], which was conceived by Buchberger in the mid-nineties as a system for supporting the "working mathematician" in all aspects of his everyday work. A few years ago it has undergone a major relaunch, finally leading

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

to version "Theorema 2.0" which, albeit incorporating all the main design principles of Theorema 1 and still being based on *Mathematica* [26], is very different compared to the old version of the system from the user-interface point of view. Hence, it must be noted that the theory exploration presented in this report was carried out entirely in Theorema 2.0.

The mathematical theory under consideration, *reduction ring theory*, is a natural candidate for being treated, in a completely formal way, by a mathematical assistant system. This is because large parts of the theory are extremely technical, with many complicated and lengthy definitions (the axioms in Section 2.1 serve as good examples in this respect) and tedious, but nonetheless mostly straight-forward, proofs. Therefore, delegating some of the more technical tasks to the system for being taken care of either in a fully automatic, or at least interactive manner, certainly is a great benefit. This claim is also supported by the fact that since their introduction in the early 1980s, and some extensions and generalizations by Stifter in the late 1980s / early 1990s, reduction rings have hardly seen any progress for more than 20 years now – and we do believe that progress *is* possible and has the potential to lead to interesting new insights and valuable results. For instance, a natural generalization of reduction rings are *non-commutative* reduction rings, which have not been considered so far. Sure, extending the theory to the non-commutative case is a highly non-trivial endeavour, but the availability of a solid and thorough formalization of the commutative case might ease some of the possible complications.

Although, up to our knowledge, the theory of reduction rings has never been subject to computer-supported theory exploration so far, in *any* system, there already exist formal treatments of *classical* Gröbner bases theory in ACL2 [13], Coq [23], Mizar [20] and Objective CAML [9]. Still, the *algorithmic* aspect of reduction rings (without any proofs) was implemented in Theorema in [3]. There, similar as in our approach, functors are used for constructing towers of domains in a generic way.

The report is structured as follows: Section 2 reviews the most important notions of reduction ring theory, mainly in order for the report to be self-contained, but also a couple of minor deviations of our formalization compared to existing literature are described there. Section 3 provides an overview of our formalization, i. e. how the theory is structured into sub-theories, how they are related to each other, and what exactly they contain. Section 4 outlines the Theorema-generated proof of the Main Theorem (Theorem 12) in more detail, serving as a concrete example of a non-trivial mathematical theorem and how it can be proved in Theorema. Section 4.1 contains some rather technical considerations related to the notions of *correlativity* and *irrelativity* (Definition 3) and may be skipped. Section 5 draws the focus from the object- to the meta level, by describing the special prover and special inference rules designed and used for the

formal verification of the theory in Theorema. Finally, Section 6 concludes the report with a short summary and outlook.

A preliminary and shorter version of this report has already been accepted for publication in the proceedings of the CASC'2015 conference [12]. The Theorema notebooks containing the formalization can be obtained from the author.

2 Reduction Rings and Gröbner Bases

Reduction rings were first introduced by Buchberger in 1984 [2] as a generalization of his *Gröbner bases* theory to much wider classes of domains than only polynomial rings over fields. Later, around 1988, Sabine Stifter further generalized Buchberger's approach [21, 22], such that, for instance, also $\mathbb{Z}_n$ (i. e. integer quotient rings modulo arbitrary n) could be turned into reduction rings.[1] Our formalization is mainly based on Stifter's approach, since it is the most general one. However, the differences between the two approaches by Buchberger and Stifter are only in the technical details anyway and will partly be pointed at explicitly in the remainder of this section. Furthermore, during the formalization of the theory it turned out that part of the definition of reduction rings due to [22] was slightly erroneous, in the sense that the Main Theorem (Theorem 12) did not hold in reduction rings in general. Still, this error could be fixed easily and will be explained in detail in Section 4.1.

The main purpose of reduction rings clearly is being able to solve ideal-theoretic problems, e. g. deciding ideal membership, algorithmically (or, at least, having a *procedure* for solving them if the domain in question is not an *algorithmic* reduction ring, cf. Section 2.1). Although the axioms defining reduction rings are quite complicated and technical, as will be seen below, they are satisfied by some well-known domains, if $\prec$, I_c and M_c^i are defined appropriately:

- fields,
- $\mathbb{Z}$, the ring of integers,
- $\mathbb{Z}_n := \mathbb{Z}/n\,\mathbb{Z}$ for arbitrary $n \in \mathbb{Z}^+$,
- the Gaussian integers $\{a + b\,\imath \mid a, b \in \mathbb{Z}\}$,
- $R \times R \ldots \times R$, i. e. the n-fold direct product of R, if R is a reduction ring,
- $R[X]$, i. e. the polynomial ring over R in the finitely many indeterminates X, if R itself is a reduction ring. This property does not come "by chance"

[1] This had wrongly been claimed already in [2]. See [21] for details.

but was a fundamental requirement when designing the reduction ring axioms in [2].

For the sake of completeness one must mention that there are many other generalizations of Gröbner bases besides reduction rings, both in the commutative and in the non-commutative setting: On the one hand, Zacharias [27], Kandri-Rody and Kapur [10], Pauer [17] and Sato *et al.* [19], among others, consider commutative polynomial rings over rings with certain properties. On the other hand, Kandri-Rody and Weispfenning [11], Mora [15], Reinert [18] and Mialebama Bouesso and Sow [14] deal with non-commutative algebras over fields or rings, only to name a few. As can be seen, there are hardly any variants of Gröbner bases in the spirit of reduction rings, where no polynomial structure of the underlying domain is required in the first place, but instead a "lifting" method for moving from an arbitrary reduction ring R to $R[X]$ (for a finite set of indeterminates X) is provided.

Although most of the contents of the remaining part of this section can also be found in the literature, in particular in [22], they are included here in order for the report to be self-contained. Still, there are also some slight deviations.

2.1 Definitions

Let in the sequel R always be a *commutative ring with unit*, endowed with a *partial Noetherian order relation* $\preceq$. Furthermore, every non-zero $c \in R$ must have a non-zero natural number I_c and sets $M_c^1, \ldots, M_c^{I_c} \subseteq R$ associated to it, where the latter are the sets of *multipliers* c may be multiplied with during a reduction process. Apparently, every c may have its own multipliers, in contrast to the classical setting of polynomials over fields, where the multipliers of every polynomial are precisely the monomials; this, in fact, is still the case in [2], where moreover I_c is restricted to 1. In [21] every c has its own multipliers, but I_c must be 2 for all c.

Define M_c (without superscript) as $M_c := M_c^1 \cup \ldots \cup M_c^{I_c}$. In the sequel, the typed variables a, b, c, z and m, possibly with subscripts, will always be elements of R.

Definition 1 (Reduction relation)**.** a reduces to b modulo c using m, written as $a \rightarrow_{m,c} b$, iff $b = a - m\,c \prec a$ and $m \in M_c$. If m is omitted, e. g. in $a \rightarrow_c b$, the meaning is that there exists $m \in M_c$ such that $a \rightarrow_{m,c} b$. If $C \subseteq R$, then $a \rightarrow_C b$ abbreviates $\exists_{c \in C}\, a \rightarrow_c b$.

For a fixed reductor r (i. e. an element of R or a set thereof), $\leftrightarrow_r$ denotes the symmetric, $\rightarrow_r^*$ the reflexive-transitive, and $\leftrightarrow_r^*$ the symmetric-reflexive-transitive closure of $\rightarrow_r$.

Note that $\rightarrow_r$ is Noetherian for any reductor r, since by definition $a \rightarrow_r b$ implies $b \prec a$ and $\preceq$ is Noetherian.

Definition 2 (Connectibility below)**.** Let r be an arbitrary reductor. Then a and b can be connected below z modulo r, written as $a \overset{\prec z}{\leftrightarrow}{}^*_r\ b$, iff there exist $e_1 = a, e_2, \ldots, e_{n-1}, e_n = b$ such that $e_i \leftrightarrow_r e_{i+1}$ for $1 \leq i \leq n-1$ and in addition $e_i \prec z$ for $1 \leq i \leq n$.

Now come the definitions of *irrelativity* and *correlativity*. The former can be found in the literature (in a slightly different variant, without referring to indices i and j), whereas the latter had to be introduced in order to fix the aforementioned problems in the definition of reduction rings.

Definition 3 (Irrelativity and correlativity)**.** Let $1 \leq i, j \leq I_{c_1}$. The pairs (m_1, c_1) and (m_2, c_2) are said to be *irrelative* w. r. t. i and j iff

- $c_1 \neq c_2$ or
- $i \neq j$, $m_1 \in M^i_{c_1}$ and $m_2 \in M^j_{c_1}$.

The pairs (m_1, c_1) and (m_2, c_2) are said to be *correlative* iff $c_1 = c_2$ and there exists $1 \leq k \leq I_{c_1}$ with $m_1 \in M^k_{c_1}$ and $m_2 \in M^k_{c_1}$.

Note that correlativity is *not* the negation of irrelativity: It may well be that there are $c_1, c_2, m_1, m_2 \in R$, $1 \leq i, j \leq I_{c_1}$ such that the pairs (m_1, c_1) and (m_2, c_2) are both irrelative w. r. t. i and j *and* at the same time also correlative.

Next we introduce the crucial notions of (minimal, non-trivial) common reducibles:

Definition 4 (Common reducible)**.** Let $1 \leq i, j \leq I_{c_1}$. a is called a *common reducible* of c_1 and c_2 w. r. t. i and j iff there exist m_1, m_2 such that $a \rightarrow_{m_1,c_1} a - m_1\, c_1$, $a \rightarrow_{m_2,c_2} a - m_2\, c_2$ and the pairs (m_1, c_1) and (m_2, c_2) are irrelative w. r. t. i and j.

Definition 5 (Non-trivial common reducible)**.** Let again $1 \leq i, j \leq I_{c_1}$. a is called a *non-trivial common reducible* of c_1 and c_2 w. r. t. i and j, written as $c_1 \ \triangle^a_{i,j}\ c_2$, iff it is a common reducible of c_1 and c_2 w. r. t. i and j (according to Definition 4) and there do *not* exist m_1, m_2 such that

- $a \rightarrow_{m_1,c_1} a - m_1\, c_1$,
- $a \rightarrow_{m_2,c_2} a - m_2\, c_2$,
- $a - m_1\, c_1$ is further reducible modulo c_2 using m_2 or vice versa, and

- the pairs (m_1, c_1) and (m_2, c_2) are irrelative w. r. t. i and j.

Definition 6 (Minimal non-trivial common reducible)**.** Let again $1 \leq i, j \leq I_{c_1}$. a is called a *minimal non-trivial common reducible* of c_1 and c_2 w. r. t. i and j, written as $c_1 \nabla^a_{i,j} c_2$, iff $c_1 \triangle^a_{i,j} c_2$ and there does not exist $a_0 \prec a$ with $c_1 \triangle^{a_0}_{i,j} c_2$.

Note that in the definitions of (minimal, non-trivial) common reducibles the two indices i and j are completely irrelevant if $c_1 \neq c_2$.

In addition to (minimal, non-trivial) common reducibles for *two* elements c_1, c_2 there is also the notion of (minimal) non-trivial common reducible for only one single element:

Definition 7 (Non-trivial common reducible for one element)**.** a is called a *non-trivial common reducible* for c, written as $c \triangle^a$, iff a is reducible modulo c and there do not exist m_1, m_2 such that $a \rightarrow_{m_k,c} a - m_k\, c$, for $k = 1, 2$, and $a - m_2\, c$ can be further reduced modulo c using m_1.

Definition 8 (Minimal non-trivial common reducible for one element)**.** a is called a *minimal non-trivial common reducible* for c, written as $c \nabla^a$, iff $c \triangle^a$ and there does not exist $a_0 \prec a$ with $c \triangle^{a_0}$.

It is important to note that $c \triangle^a$ ($c \nabla^a$) is *not* the same as $c \triangle^a_{i,j}$ ($c \nabla^a_{i,j} c$) for suitable i, j; see [21], page 6, for details.

There is only one more notion left to be introduced before we can state the definition of reduction rings:

Definition 9 (Critical-pair multipliers)**.** Assume $c_1 \nabla^a_{i,j} c_2$ for some $1 \leq i, j \leq I_{c_1}$. Then m_1, m_2 constitute *critical-pair multipliers* for c_1, c_2 and a w. r. t. i and j iff a can be reduced modulo c_k using m_k, for $k = 1, 2$, and the pairs (m_1, c_1) and (m_2, c_2) are irrelative w. r. t. i and j. In that case, the pair (b_1, b_2) with $b_k := a - m_k\, c_k$, for $k = 1, 2$, is called a *critical pair* of c_1, c_2 and a w. r. t. i and j.

Now we are ready to formally introduce reduction rings: R is a reduction ring iff it is a commutative ring with unit, endowed with a partial Noetherian ordering $\preceq$ and sets of multipliers M^i_c, as specified above, satisfying the following nine axioms (R0) – (R8):

(R0) If $c \neq 0$, then $1 \in M_c$.

(R1) If $c \neq 0$ and $m \in M_c$, then $-m \in M_c$.

(R2) If $c \neq 0$ and $m \in M_c$, then $m\, c \neq 0$. This axiom is only needed since we do not require R to be free of zero divisors.

(R3) For every $b, c \neq 0$: There exist $m_1, m_2, \ldots, m_n \in M_c$ such that $b = \sum_{i=1,\ldots,n} m_i$. This axiom is actually only needed to prove that $\equiv_B$, i. e. congruence modulo the ideal generated by the set B, coincides with $\leftrightarrow_B^*$.

(R4) If $a \neq 0$, then $0 \prec a$.

(R5) If $a \rightarrow_{m,c} b$, then there are $m_1, m_2, \ldots, m_x$ and $n_1, n_2, \ldots, n_y$ such that

- $a + d \rightarrow_{m_1,c} a + d - m_1\, c \rightarrow_{m_2,c} \ldots \rightarrow_{m_x,c} a + d - (m_1 + \ldots + m_x)\, c = b + d - (n_1 + \ldots + n_y)\, c \leftarrow_{n_y,c} \ldots \leftarrow_{n_2,c} b + d - n_1\, c \leftarrow_{n_1,c} b + d$ and
- $m_1 + \ldots + m_x = m + n_1 + \ldots + n_y$.

In short, $a + d$ and $b + d$ have a common successor.

(R6) If $a \rightarrow_{m_k,c} a - m_k\, c$, $k = 1, 2$, and (m_1, c) and (m_2, c) are correlative, then there are $n_1, n_2, \ldots, n_x$ such that

- $a - m_1\, c \leftrightarrow_{n_1,c} a - m_1\, c - n_1\, c \leftrightarrow_{n_2,c} \ldots \leftrightarrow_{n_y,c} a - (m_1 + n_1 + \ldots + n_y)\, c = a - m_2\, c$,
- $m_1 + n_1 + \ldots + n_y = m_2$, and
- $a - (m_1 + n_1 + \ldots + n_i)\, c \prec a$ for all $1 \leq i \leq y$.

In short, $a - m_1\, c$ and $a - m_2\, c$ can be connected below a. In [22] the requirement is "(m_1, c) and $(M - 2, c)$ not irrelative", which turned out not to be sufficient for proving Theorem 12.

(R7) If $c_1 \triangle_{i,j}^a c_2$ then there are $1 \leq k, l \leq I_{c_1}$, $\overline{a} \preceq a$ and m such that

- $c_1 \triangledown_{i,j}^{\overline{a}} c_2$,
- for every $e \neq 0$ and $m_0 \in M_e$, also $m\, m_0 \in M_e$,
- if $\overline{a} + e \prec \overline{a}$ then $a + m\, e \prec a$, for all e,
- if $b \rightarrow_e d$ then $m\, b \leftrightarrow_e m\, d$, for all b, d, e, and
- in case $c_1 = c_2$: If $\overline{a}$ is reducible modulo c_1 both using m_1 and m_2, and $m_1 \in M_{c_1}^k$ and $m_2 \in M_{c_1}^l$, then $m\, m_1 \in M_{c_1}^i$ and $m\, m_2 \in M_{c_1}^j$, or the other way round, for all m_1, m_2.

(R8) If $c \triangle^a$ then there are $\overline{a} \preceq a$ and m such that

- $c \triangledown^{\overline{a}}$,
- for every $e \neq 0$ and $m_0 \in M_e$, also $m\, m_0 \in M_e$,
- if $\overline{a} + e \prec \overline{a}$ then $a + m\, e \prec a$, for all e, and

– if $b \to_e d$ then $m\,b \leftrightarrow_e m\,d$, for all b, d, e.

This axiom is only needed to prove that the property of being a reduction ring is preserved in $R[X]$.

If R satisfies all of the nine axioms, it is a reduction ring. However, in order to effectively compute normal forms and Gröbner bases, and to effectively decide ideal membership, R in addition needs to fulfill certain *computability* and *finiteness* criteria. Deviating from the existing literature on reduction rings, where these criteria are simply part of the reduction ring axioms, we therefore introduce the notion of *algorithmic* reduction rings: R is an algorithmic reduction ring iff it is a reduction ring and moreover satisfies the five axioms (R9) – (R13):

(R9) For every $c_1, c_2 \neq 0$ and $1 \leq i, j \leq I_{c_1}$ the set $\{a | c_1 \nabla_{i,j}^a c_2\}$ is finite and can be computed algorithmically.

(R10) For every $c \neq 0$ the set $\{a | c \nabla^a\}$ is finite and can be computed algorithmically. Similar as (R8), this axiom is only needed in the proof that $R[X]$ is an algorithmic reduction ring if R is.

(R11) For every a, c reducibility of a modulo c can be effectively decided, and in case of reducibility a suitable multiplier can be computed algorithmically.

(R12) If $c_1 \nabla_{i,j}^a c_2$ at least *one* pair of critical-pair multipliers for c_1, c_2 and a w. r. t. i and j can be found algorithmically.

(R13) There does not exist an infinite sequence of sets $D_1, D_2 \ldots$ with $\mathrm{Red}(D_1) \subset \mathrm{Red}(D_2) \subset \ldots$, where $\mathrm{Red}(D) := \{a | a \text{ is reducible modulo } D\}$.

2.2 Gröbner Bases

It is a well-known fact that there are various equivalent characterizations of Gröbner bases in the classical setting of polynomials over a field. For instance, a set G is a Gröbner basis iff $\mathrm{lm}(\langle G \rangle) \subseteq \langle \mathrm{lm}(G) \rangle$ (where lm stands for "leading monomials" and $\langle \cdot \rangle$ denotes the ideal generated by its argument). Equivalently, G is a Gröbner basis iff $\to_G$ is Church-Rosser, or if $a \to_G^* 0$ for every $a \in \langle G \rangle$. In reduction rings, however, due to the lack of any polynomial structure, it is apparent that the first characterization by leading-monomial-ideals cannot be used. Therefore, it is the Church-Rosser property of the reduction relation induced by G that determines whether G is a Gröbner basis or not, leading to the following

Definition 10 (Gröbner basis)**.** A finite set $G \subseteq R$ is a *Gröbner basis* iff $\to_G$ is Church-Rosser, i. e. whenever $a \leftrightarrow_G^* b$ then there exists some d with $a \to_G^* d$ and $b \to_G^* d$.

The definition requires G to be finite, although in principle one could of course also consider *infinite* Gröbner bases; however, since the main purpose of Gröbner bases is to provide algorithmic means for solving ideal-theoretic problems, infinite sets would not be of great help.[2]

The characterization given in Definition 10 is *algebraic*, but not *algorithmic*: if R is infinite, there are infinitely many pairs a, b that have to be checked. Hence, what is needed is a *finite, algorithmic* criterion for deciding whether a given set is a Gröbner basis or not; this is the content of Theorem 12. Before we can state it, we need one more definition.

Definition 11 (Critical-pair connectibility)**.** Let $B \subseteq R$. Then B is said to have *connectible critical pairs*, denoted by $\textsc{cpConnectible}[B]$, iff for all $c_1, c_2 \in B$ (not necessarily distinct), all $1 \leq i, j \leq I_{c_1}$ and all a with $c_1 \nabla^a_{i,j} c_2$ there exists a critical pair (b_1, b_2) for c_1, c_2 and a w. r. t. i and j such that $b_1 \overset{\prec a}{\leftrightarrow}{}^*_B\, b_2$.

Theorem 12 (Main Theorem)**.** *Let R be a reduction ring and $G \subseteq R$ finite. Then G is a Gröbner basis iff G has connectible critical pairs.*

The proof of Theorem 12 is outlined in Section 4.[3] Now it should be clear that if R is not only a reduction ring, but even an *algorithmic* reduction ring, if can be effectively decided whether a given set G is a Gröbner basis or not. Moreover, if G is not, it can be *completed* in such a way that the new set still generates the same ideal but in addition is a Gröbner basis; this is achieved by Buchberger's critical-pair/completion algorithm in reduction rings:

Some remarks on Algorithm 1 are in place:

- Algorithm 1 is only an algorithm if R is an *algorithmic* reduction ring, otherwise it is a (possibly infinite) procedure. Termination follows essentially from axioms (R9) and (R13).
- In lines 2 and 11 also pairs of identical elements have to be taken into account. This is because singleton sets are not necessarily Gröbner bases.
- MNTCR is defined for sets $C \subseteq R^2$ as

$$\begin{aligned}\textsc{mntcr}(C) := \{(a,1,1,c_1,c_2) | (c_1,c_2) \in C, c_1 \neq c_2, c_1 \nabla^a_{1,1} c_2\} \cup \\ \cup \{(a,i,j,c,c) | (c,c) \in C, 1 \leq i < j \leq I_c, c \nabla^a_{i,j} c\}\end{aligned}$$

  Hence, *all* minimal non-trivial common reducibles for *all* combinations of indices i, j (with $i < j$) have to be taken into account. However, if $c_1 \neq c_2$, the indices are irrelevant and thus it is sufficient to only consider 1 and 1.

[2] In non-commutative polynomial rings there are finitely generated ideals without finite Gröbner basis, where one typically computes *truncated* Gröbner bases up to a certain degree.

[3] Actually, only the direction from right to left was proved. The other direction is obvious.

Algorithm 1 Buchberger's algorithm in reduction rings R

Input: $F = \{f_1, \ldots, f_n\} \subseteq R$
Output: $G \subseteq R$ s. t. $\langle G \rangle = \langle F \rangle$ and G is Gröbner basis

```
1:  function GB(F)
2:      P ← MNTCR({(f_i, f_j) | 1 ≤ i ≤ j ≤ n})
3:      G ← F
4:      while P ≠ ∅ do
5:          choose some p = (a, i, j, c_1, c_2) from P
6:          P ← P \ {p}
7:          (b_1, b_2) ← some critical pair for c_1, c_2 and a w. r. t. i and j
8:          Find h_1, h_2 with b_k →*_G h_k and h_k irreducible, for k = 1, 2
9:          h ← h_1 − h_2
10:         if h ≠ 0 then
11:             P ← P ∪ MNTCR({(h, h)} ∪ {(g, h) | g ∈ G})
12:             G ← G ∪ {h}
13:         end if
14:     end while
15:     return G
16: end function
```

- In contrast, only *one* critical pair needs to be considered in line 7.
- Instead of reducing the *difference* of c_1 and c_2, c_1 and c_2 must be reduced individually in line 8 to obtain h_1 and h_2. As long as G is no Gröbner basis, h_1 and h_2 are of course not unique.
- The algorithm could be made more efficient by adding the so-called *chain criterion* for avoiding unnecessary reductions. This criterion was introduced by Buchberger in [1] in the classical setting, but can easily be generalized to reduction rings.

If G is a Gröbner basis, membership of a in $\langle G \rangle$ can effectively be decided by reducing a to normal form w. r. t. G. Since $\to_G$ is Church-Rosser normal forms are unique, and furthermore the normal form of a is 0 iff $a \in \langle G \rangle$ (thanks to axiom (R3)).

2.3 Polynomial Reduction Rings

We briefly review the construction of polynomial reduction rings, as contained in [2, 21]. For this, let R as usual be a reduction ring and let $\leq$ be an admissible

term ordering on the monoid of power-products $[X]$. The typed variables s and t denote power-products, p and q denote polynomials, and the following notation will be used: $\mathrm{C}(p,t)$ denotes the coefficient of p at t, $\mathrm{H}(p,t)$ denotes the "higher part" of p w. r. t. t (i. e. all monomials with power-product $> t$), $\mathrm{lp}(p)$ denotes the leading power-product of non-zero p, and $\mathrm{lc}(p)$ is the leading coefficient of p (i. e. $\mathrm{lc}(p) = \mathrm{C}(p, \mathrm{lp}(p))$).

$R[X]$ can be made a reduction ring by defining the Noetherian order relation $\widetilde{\prec}$, the integers $\widetilde{I}_p$ and the sets of multipliers $\widetilde{M}_p^i$ as follows:

$$p \mathrel{\widetilde{\prec}} q \quad :\Leftrightarrow \quad \exists_t \, \mathrm{H}(p,t) = \mathrm{H}(q,t) \wedge \mathrm{C}(p,t) \prec \mathrm{C}(q,t) \tag{2.1}$$

$$\widetilde{I}_p \quad := \quad I_{\mathrm{lc}(p)} \tag{2.2}$$

$$\widetilde{M}_p^i \quad := \quad M_{\mathrm{lc}(p)}^i \cdot [X] \tag{2.3}$$

3 Overview of the Formalization in Theorema

Most of the theory presented in Section 2 has been formalized in the Theorema system, more precisely in Theorema 2.0 [24]. The formalization, as every formalization in Theorema, consists of formal (formulas and proofs thereof) and informal parts (explanatory text, drawings) making up the theory, distributed across several Theorema notebooks to achieve a structured hierarchical built-up of the theory in terms of sub-theories. The theory-structure of the formalization of reduction rings, as it is at the moment, is depicted in Figure 1, and Table 1 contains a short summary of the sizes of the individual components by means of formulas and proofs. As can be seen in Figure 1, any of the seven other theories depends, or will depend, directly on `ElementaryTheories`.

Theory	Formulas	Proofs	Proof Size (av./max.)
`ElementaryTheories`	433	228	21.7 / 94
`ReductionRings`	208	152	41.0 / 193
`Polynomials`	380	324	44.5 / 322
`Fields`	17	0	
`Integers`	20	0	
`IntegerQuotientRings`	19	0	
`PolyTuples`	66	0	
`GroebnerRings`	32	0	
	1175	**704**	**36.4 / 322**

Table 1: Number of formulas and proofs in the formalization. The proof size refers to the number of inference steps.

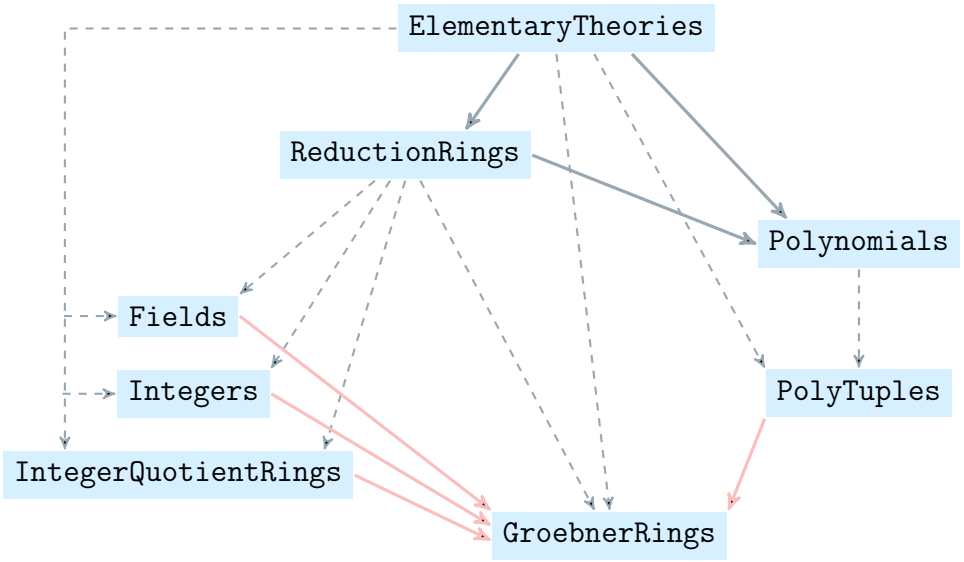

Figure 1: The structure of the formalization. An arrow from A to B denotes dependency of B on A, in the sense that formulas from A are used in B in proofs (blue) or computations (red). Dashed arrows denote future dependencies.

3.1 Theory `ElementaryTheories`

Theory `ElementaryTheories` is a collection of theories that are not directly related to reduction rings, but rather consist of definitions of and lemmas about basic notions and concepts, such as sets, integers and tuples. These notions are ubiquitous in almost every mathematical theory, hence it does not come as a surprise that all of the other seven components of the present formalization directly depend, or will directly depend, on `ElementaryTheories`.

Apart from the treatment of tuples and sequences, almost none of the lemmas and theorems stated in this notebook were proved formally using Theorema, simply because the focus of the work presented in this report was not on the systematic build-up of fundamental concepts, but rather on the formal treatment of reduction ring theory. These unproved lemmas and theorems are, of course, not concerned with deep mathematical insights, but state trivial facts like

$$(a \in \mathbb{Z} \wedge i \in \mathbb{Z}_{1,\ldots,a+1}) \Rightarrow (i \in \mathbb{Z}_{1,\ldots,a} \vee i = a + 1)$$

where, following Theorema notation, $\mathbb{Z}_{a,\ldots,b}$ denotes the set $\{x \in \mathbb{Z} | a \leq x \leq b\}$; a and b may be $\pm\infty$.

In the following subsections we describe the constituents of `ElementaryTheories` in more detail.

3.1.1 Sets

This part introduces basic notions related to sets, e. g. finite sets, the subset relation, unions of sets, and abstraction terms. Simple properties of these notions, mostly about finite sets and abstraction terms, like

$$\underset{A,B}{\forall} \left(\textsc{isFinite}[A] \land \textsc{isFinite}[B]\right) \Rightarrow \textsc{isFinite}[A \cup B]$$

are stated (without proof).[4]

In addition, functor DomainSets is introduced.[5] This functor maps a domain D to the domain of sets over D, i. e. the elements of $\textsc{DomainSets}[D]$ are all sets consisting of elements in D, which reads as

$$\underset{D,x}{\forall} \underset{\textsc{DomainSets}[D]}{\in} [x] \;:\Leftrightarrow\; \left(\textsc{isSet}[x] \land \underset{y \in x}{\forall} \underset{D}{\in} [y]\right)$$

in Theorema notation. Under-scripted non-binder symbols are *domain operations*, i. e. operations defined in the domain appearing as the underscript. The under-scripted "$\in$" is the *domain decision predicate*, i. e. the predicate that determines whether an object belongs to the respective domain or not.

3.1.2 Algebraic Structures

This part formally defines various algebraic categories, like groups, rings and fields, and notions related to them, like partial/total/Noetherian order relations and associative/commutative/distributive operations. One of the most important formulas contained in there, at least for developing the theory of reduction rings, is the *principle of Noetherian induction*:

$$\left(\underset{\preceq}{\exists} \left(\textsc{isNoetherian}[\preceq, D] \land \underset{\underset{D}{\in}[x]}{\forall} \left(\underset{\underset{D}{\in}[y]}{\forall} y \prec x \Rightarrow P[y]\right) \Rightarrow P[x]\right)\right) \;\Rightarrow\; \underset{\underset{D}{\in}[x]}{\forall} P[x]$$

3.1.3 Integers and Inductive Definitions

This part, on the one hand, is a mere collection of facts about natural numbers and integers, including

- the induction principle for natural numbers (in analogy to the Noetherian induction principle),
- linear arithmetic on $\mathbb{Z}$

[4] Following Theorema notation, function application is denoted by square brackets.

[5] For terminology in Theorema, especially "functor", "category" and "domain", we refer to [3].

- intervals of natural numbers and integers
- logical quantifiers ranging over intervals of integers.

As indicated above, the theory of natural numbers and integers was not developed systematically from scratch, but important and useful results were collected and stated without reference to any foundational definitions.

In addition to that, *finite sums* of the form $\sum_{i=a,\ldots,b} f[i]$ are introduced, by means of a recursive definition, and then a couple of their properties are stated, most of them being concerned with splitting sums into two parts.

3.1.4 Tuples

The theory of tuples was developed from scratch, starting from the very definition of tuples as functions whose domain is a finite interval of natural numbers of the form $\mathbb{N}_{1,\ldots,n}$. Elementary operations, such as concatenation ("JOIN"), append/prepend, rest/most, reverse, as well as tuple-abstraction, are introduced and some (more or less obvious) lemmas and theorems are stated (together with Theorema-generated proofs). The most important of these theorems are perhaps those that describe under which conditions on the constituents of a tuple-constructor (concatenation, append, ...) an arbitrary property P holds, either for all elements or for all pairs of consecutive elements, of the new tuple. For example, a binary relation P holds for all pairs of consecutive elements of JOIN$[S, T]$ iff it holds for all pairs of consecutive elements of S and T, and moreover it also holds for the pair consisting of the *last* element of S and the *first* element of T, given that both tuples are non-empty.

In analogy to the DOMAINSETS functor described in Section 3.1.1, a DOMAINTUPLES functor is introduced. This functor maps a domain D to the domain of tuples (of arbitrary length) over D; no further mathematical operations are defined in the functor.

3.1.5 Infinite Sequences

Infinite sequences are needed to handle axiom (R13) in the definition of reduction rings (see Section 2.1), in particular in conservation theorems. Similar as tuples, infinite sequences are defined as functions over $\mathbb{N}$, only that their domain is unbounded, of course. The most important concept formalized in the theory of infinite sequences is the notion of *sub-sequence* of a sequence S, and under which conditions on S certain sub-sequences (described by some property P that shall hold for all sequence-elements, or all pairs of consecutive elements, respectively) exist. As an example, the following (proved) formula is part of the

theory:

$$\underset{P,S}{\forall} \left(\underset{a\in\mathbb{N}}{\forall} \underset{i\geq a}{\exists} \underset{j>i}{\forall} P[S[i], S[j]] \right) \Rightarrow \underset{T}{\exists} \textsc{isSubsequence}[T, S] \wedge \underset{i<j}{\forall} P[T[i], T[j]]$$

Unlike for tuples, there is no nice syntax for accessing elements of sequences (e. g. with the index as subscript), but rather the i-th element of sequence S really has to be denoted by $S[i]$, i. e. the *function* S evaluated at i.

3.2 Theory `ReductionRings`

Theory `ReductionRings` is the core theory of the whole formalization of reduction rings, as expected. It contains the definitions of reduction rings and algorithmic reduction rings by means of the axioms listed in Section 2.1, as well as the definitions of many auxiliary notions (e. g. the reduction relation $\rightarrow$ and its symmetric and symmetric-reflexive-transitive closures).

Apart from lots of useful lemmas, the two main theorems stated and proved in this theory are the Main Theorem of reduction ring theory (Theorem 12) and the fact that in reduction rings ideal congruence coincides with the symmetric-reflexive-transitive closure of the reduction relation. The proof of the former serves as an example for proving in Theorema and is presented more thoroughly in Section 4. As can be seen in Table 1, the size of `ReductionRings` in terms of the number of formulas and proofs is quite moderate (at least compared to `ElementaryTheories` and `Polynomials`), which indicates that the proof of the Main Theorem is actually not that difficult[6].

The formal treatment of this particular theory is in principle finished by now; *every* result stated comes together with a formal, Theorema-generated proof, meaning that the only unproved formulas are definitions. However, it might be that in the future it turns out that additional results, holding in general in reduction rings, are needed to prove theorems in other theories; these theorems will have to be added to `ReductionRings` then.

3.3 Theory `Polynomials`

Theory `Polynomials` introduces the categories of *commutative power-products*, *reduction polynomials* and *algorithmic reduction polynomials*. The main contents of the theory in terms of theorems are three theorems that state that if $\mathcal{R}$ is a reduction ring and $\mathcal{T}$ is a domain of commutative power-products, and $\mathcal{P}$ is a domain belonging to the category of reduction polynomials over $\mathcal{R}$ and $\mathcal{T}$, then $\mathcal{P}$ is also a reduction ring. Moreover, if $\mathcal{R}$ is even an *algorithmic* reduction ring

[6] Still, 193 inference steps is quite something.

and $\mathcal{P}$ belongs to the category of *algorithmic* reduction polynomials, then $\mathcal{P}$ is an *algorithmic* reduction ring as well.

A domain $\mathcal{P}$ belongs to the category of reduction polynomial iff it is endowed with a coefficient function that maps any element p of the domain (i. e. any polynomial) and any power-product τ to the coefficient of p at τ; the axioms characterizing the category of reduction polynomials ensure that this coefficient function has all the properties it is supposed to have (i. e. how it behaves w. r. t. addition and multiplication, and that it may give non-zero coefficients only for finitely many power-products, for each polynomial). Besides the coefficient function, $\mathcal{P}$ also needs to provide a binary relation $\widetilde{\prec}$ and sets of multipliers $\widetilde{M}_p^i$, satisfying the properties stated in Section 2.3.

The category of algorithmic reduction polynomials is a subcategory of reduction polynomials, which additionally requires $\mathcal{P}$ to provide certain functions, e. g. a function that returns all minimal non-trivial common reducibles (mntcr) for p and q w. r. t. two indices i and j. This does not have to be stated using the very definition of mntcr in general reduction rings, but instead a more concise description of mntcr for polynomial domains, that captures precisely the essence of what it means to be a mntcr in a polynomial domain, is available. Something similar holds for multipliers suitable for reduction and for critical-pair multipliers. We do not go into details here, because this is already covered in [2] and [22].

As can be seen from Table 1, `Polynomials` is by far the largest theory in terms of the number of proofs. This stems from the fact that for showing that a domain belongs is an algorithmic reduction ring, in total 18 subgoals (including the 14 axioms (R0) – (R13)), of which most are highly non-trivial, have to be shown. Furthermore, although it is a well-known fact that a polynomial ring over a commutative ring with unit is itself a commutative ring with unit, proving this fact formally with Theorema turned out to be very tedious and lengthy. In general, most of the proofs in theory `Polynomials` are lengthy but involve comparatively little creativity, making them ideal candidates for being treated by a mathematical assistant system.

Actually, the Theorema-generated proofs of the 17 other subgoals (apart from $\mathcal{P}$ being a commutative ring with unit), follow more or less exactly the proofs contained in [2], with only slight modifications here and there due to the presence of different sets of multipliers and our new definitions of correlativity and irrelativity; the crucial (but obvious) property of irrelativity and correlativity needed for proving that (R7) is preserved by reduction polynomials is the following:

Lemma 13. *For all c, m_1, m_2, n_1, n_2 and all $1 \leq i, j \leq I_c$: If (m_1, c) and (m_2, c) are irrelative w. r. t. i and j, and (n_1, c) and (n_2, c) are correlative, then there exists*

$1 \leq k \leq I_c$ *s. t.*

- (n_l, c) *and* (m_1, c) *are irrelative w. r. t.* k *and* i*, for* $l = 1, 2$*, or*
- (n_l, c) *and* (m_2, c) *are irrelative w. r. t.* k *and* j*, for* $l = 1, 2$*.*

Moreover, for proving that (R13) is preserved by reduction polynomials one needs that the domain of power-products $\mathcal{T}$ is what we call a *Dickson domain*, i. e. a domain where Dickson's Lemma holds [7]. Since this does not follow from our rather general definition of commutative power-products as elements of cancellative commutative monoids, it must be required separately.[7]

As for `ReductionRings`, also the development of theory `Polynomials` is finished, at least unless it turns out that additional results are needed for proving that concrete domains belong to the categories of commutative power-products or (algorithmic) reduction polynomials.

3.4 Theories of Basic Domains and Functors

The collection of theories of basic domains and functors consists of `Fields`, `Integers`, `IntegerQuotientRings` and `PolyTuples`, each of them containing a functor which constructs a particular reduction ring.

3.4.1 Theory `Fields`

The functor in `Fields`, called ReductionField, takes as input a domain $\mathcal{K}$ and constructs a new domain $\mathcal{RK}$ where functions and relations, like the Noetherian ordering and the sets of multipliers, are defined in such a way that $\mathcal{RK}$ is an algorithmic reduction ring if $\mathcal{K}$ is a field. The formal proof of this fact has not been carried out yet but will most probably not be too difficult, at least compared to the proofs in theories `ReductionRings` and `Polynomials`; this is also the reason why `Fields`, at the moment, only contains 17 formulas (making up the definition of the functor). However, the functor can already be used in computations.

3.4.2 Theory `Integers`

The functor in `Integers`, called ReductionIntegers, does not take any input but merely constructs the domain $\mathcal{RZ}$ of reduction integers, i. e. the domain of integers endowed with all the functions and relations necessary for being an algorithmic reduction ring, according to [2]. As in `Fields`, the proof of this fact is still future work, but the functor itself can already be used for computations.

[7] We did not include it in the definition of commutative power-products, since it is not needed for proving preservation of any of the other axioms.

3.4.3 Theory `IntegerQuotientRings`

The functor in `IntegerQuotientRings`, called ReductionIQR, takes as input a value n (which is supposed to be a non-negative integer) and constructs the reduction ring $\mathcal{RZ}_{\backslash}$ of residue classes modulo n, according to [21]. Theorema-generated proofs are still missing also in this theory.

3.4.4 Theory `PolyTuples`

`PolyTuples` does not contain only one, but even five functors:

- PPTuples constructs the domain of (unordered) power-products represented as exponent tuples of length n of natural numbers. n is an input argument of the functor. The most important functions defined by this functor are addition, multiplication, division and LCM of two power-products, as well as the divisibility relation.
- Lex extends a domain $\mathcal{T}$ constructed by PPTuples by defining the binary relation $\leq$ on the new domain as the lexicographic order relation on $\mathcal{T}$.
- DegLex is similar to Lex, only that $\leq$ is defined as the degree-lexicographic order relation.
- DegRevLex is again similar to Lex, only that $\leq$ is defined as the degree-reverse-lexicographic order relation.
- PolyTuples takes two domains $\mathcal{R}$ and $\mathcal{T}$ and constructs the domain of polynomials over coefficient domain $\mathcal{R}$ and power-product domain $\mathcal{T}$, represented as tuples of pairs of coefficients and power-products. The functor does not rely on any particular representation of $\mathcal{T}$, meaning that it does not have to be a domain constructed by one of Lex, DegLex or DegRevLex.

Of course, PolyTuples is supposed to construct domains belonging to the category of algorithmic reduction polynomials, and hence also to the category of algorithmic reduction rings, if $\mathcal{R}$ is an algorithmic reduction ring itself and $\mathcal{T}$ is a commutative power-product domain (and also a Dickson domain). The proof of this fact, as well as the proofs that Lex, DegLex and DegRevLex construct commutative power-product- and Dickson domains, are still missing.

3.5 Theory `GroebnerRings`

Theory `GroebnerRings`, finally, puts together the computational aspects of reduction ring theory by introducing a functor, called GroebnerRing, that extends a given domain $\mathcal{R}$ by defining the function GB for actually computing Gröbner bases in $\mathcal{R}$. GB implements (a variant of) Algorithm 1 and is defined solely in terms of functions of $\mathcal{G}$, such that it is correct only if $\mathcal{R}$ is an algorithmic reduction ring (which, following a general principle in the design of functors in Theorema, is nowhere required in the *definition* of the functor). Furthermore, the so-called *chain criterion* for detecting useless steps in Algorithm 1 is also taken into account in function GB, making computations even more efficient. The chain criterion was introduced in [1] for the classical setting of polynomials over fields, but it is valid in general reduction rings, too.

Lots of sample computations of Gröbner bases using function GB have already been carried out, in all the reduction rings constructed by the functors listed in Section 3.4. As expected, the performances in terms of absolute timings are nowhere near of what one achieves with built-in *Mathematica* functions, hence we spare a detailed summary and comparison.

4 Proof of the Main Theorem

In this section we present the Theorema-generated proof of Theorem 12 in more detail. More precisely, we give an overview of the individual sub-proofs the proofs was split into and explain why the "old" definition of irrelativity due to [22] is not suitable.

The key result needed for proving Theorem 12 is the so-called *Generalized Newman Lemma*, which describes a very weak sufficient condition for a binary relation $\rightarrow$ to be Church-Rosser, without any appeal to reduction ring theory. It was first introduced and proved in [25] and is, of course, also used in [2]:

Lemma 14 (Generalized Newman Lemma)**.** *Let $\preceq$ be a partial Noetherian order relation and $\rightarrow$ a binary relation such that $a \rightarrow b$ implies $b \prec a$. If b_1 and b_2 can be connected below a (w. r. t. $\preceq$; cf. Definition 2) whenever $a \rightarrow b_1$ and $a \rightarrow b_2$, for all a, b_1, b_2, then $\rightarrow$ is Church-Rosser.*

Please note that although Lemma 14 holds for arbitrary Noetherian orderings $\preceq$ and binary relations $\rightarrow$, we only proved it for the particular family of reduction relations $\rightarrow_c$ according to Definition 1. Its proof is comparatively easy: In our formalization, where we follow precisely the proof in [25], it is the combination of six formulas which in turn can be proved without any trouble. One only needs to perform Noetherian induction and induction on $\mathbb{N}$ and use

reflexivity and transitivity of $\to^*$; this, among other basic properties of $\to^*$, $\leftrightarrow^*$ and $\overset{\prec z}{\leftrightarrow}{}^*$, is also needed for proving that congruence modulo the ideal generated by B coincides with $\leftrightarrow_B^*$.

Knowing Lemma 14 it is clear how to prove Theorem 12: It suffices to show that whenever $a \to_B b_1$ and $a \to_B b_2$ we also have $b_1 \overset{\prec a}{\leftrightarrow}{}_B^* b_2$, for any set B; this is what we call *local connectivity*:

Definition 15 (Local connectivity)**.** Let $B \subseteq R$. B is said to be *locally connective*, denoted by $\textsc{isLocConnective}[B]$, iff for all $c_1, c_2 \in B$ and all a, b_1, b_2 with $a \to_{c_1} b_1$ and $a \to_{c_2} b_2$ we have $b_1 \overset{\prec a}{\leftrightarrow}{}_B^* b_2$.

Summarizing, the direction from right to left in the Main Theorem is a consequence of the following

Theorem 16. *For all reduction rings R and all $B \subseteq R$:*

$$\textsc{cpConnectible}[B] \;\Rightarrow\; \textsc{isLocConnective}[B]$$

Theorem 16 is stated exactly as it is shown here in the formalization. Its proof goes along the same lines as the original pencil-and-paper proof in [2], only that correlativity and irrelativity, as well as the possibility of identical reductors (i. e. $c_1 = c_2$ in Definition 15), have to be taken into account. Also note that some important subgoals, most importantly the special case of identical reductors, are stated and proved separately. The reason for this is twofold: First, it reduces the size of the proof of Theorem 16, and second, the case of identical reductors appears more than once in the proof, making the availability of a reusable lemma desirable. Another subgoal that is stated and proved separately comprises the situation where $c_1 \ \triangle_{i,j}^a \ c_2$ (for the a in Definition 15 and indices $1 \leq i, j \leq I_{c_1}$), both if $c_1 \neq c_2$ and $c_1 = c_2$. These situations are the "interesting" ones, i. e. where one really needs axiom (R7) to prove local connectivity.

Figure 2 shows the *proof tree* corresponding to the Theorema-generated proof of Theorem 16. The node labeled with ① is where the two cases $c_1 \neq c_2$ (left subtree) and $c_1 = c_2$ (right subtree) are distinguished, and the node labeled with ② is where the two cases $\neg\, c_1 \ \triangle_{1,1}^a \ c_2$ (left subtree) and $c_1 \ \triangle_{1,1}^a \ c_2$ are distinguished. The right subtree of node ① is small because, as mentioned above, the case $c_1 = c_2$ is contained in a separate lemma. Figure 3 shows a little fragment of the automatically generated proof document, where the inference steps correspond to ① and its right subtree (in the proof document, c_1 and c_2 are named s and $\overline{s}$, respectively). The time spent for generating the proof in a dialog-based interaction with the system, using the fully-interactive proof strategy (see Section 5), was not quite 36 minutes.[8] Note that this proof is precisely the one

[8] After a couple of unsuccessful attempts, as usual.

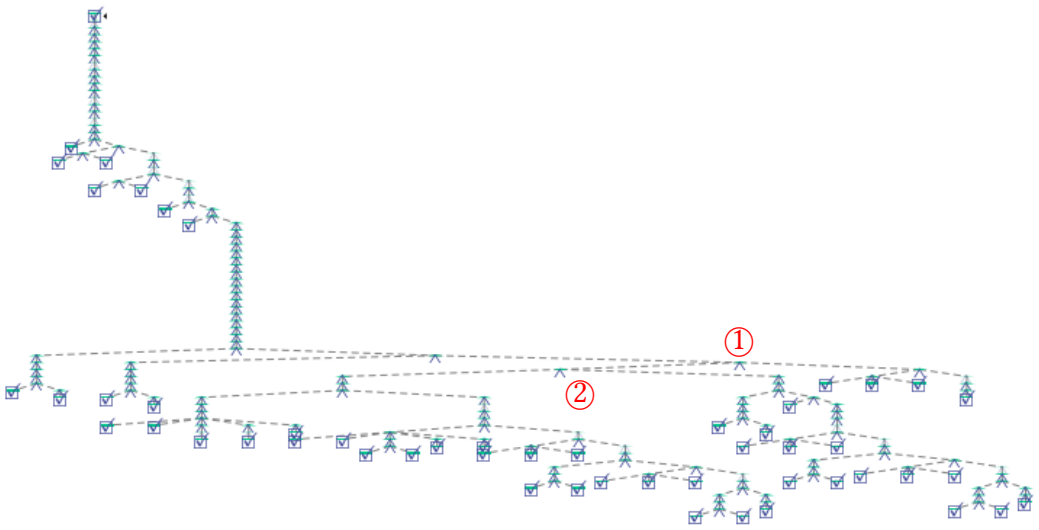

Figure 2: The proof tree corresponding to the proof of Theorem 16. Each node represents an inference step.

with the maximum of 193 inference steps in theory `ReductionRings`, as listed in Table 1.

4.1 Issues with Irrelativity

As mentioned in Section 2, the definition of irrelativity according to [22] is not suitable for proving the Main Theorem, making a re-definition necessary. In order to give evidence to this claim, we first present the "old" definition of irrelativity, which we call irrelative$_0$:

Definition 17 (Irrelative$_0$; [22], page 405)**.** Let R be a reduction ring and let $c_1, c_2, m_1, m_2 \in R$. The pairs (m_1, c_1) and (m_2, c_2) are said to be *irrelative*$_0$ iff

- $c_1 \neq c_2$ or
- there exists $1 \leq i \leq I_{c_1}$ such that $m_1 \in M_{c_1}^i$ and $m_2 \in M_{c_1} \backslash M_{c_1}^i$.

The first indication that irrelative$_0$ is not suitable is the fact that it is not symmetric in general, although it clearly should be: If $I_c = 2$, $M_c^1 \subset M_c^2$, $m_1 \in M_c^1$ and $m_2 \in M_c^2 \backslash M_c^1$, then apparently (m_1, c) and (m_2, c) are irrelative$_0$, but (m_2, c) and (m_1, c) are not.

A first attempt to make irrelative$_0$ symmetric could be to change it to irrelative$_S$, defined as follows:

Definition 18 (Irrelative$_S$)**.** Let R be a reduction ring and let $c_1, c_2, m_1, m_2 \in R$. The pairs (m_1, c_1) and (m_2, c_2) are said to be *irrelative*$_S$ iff

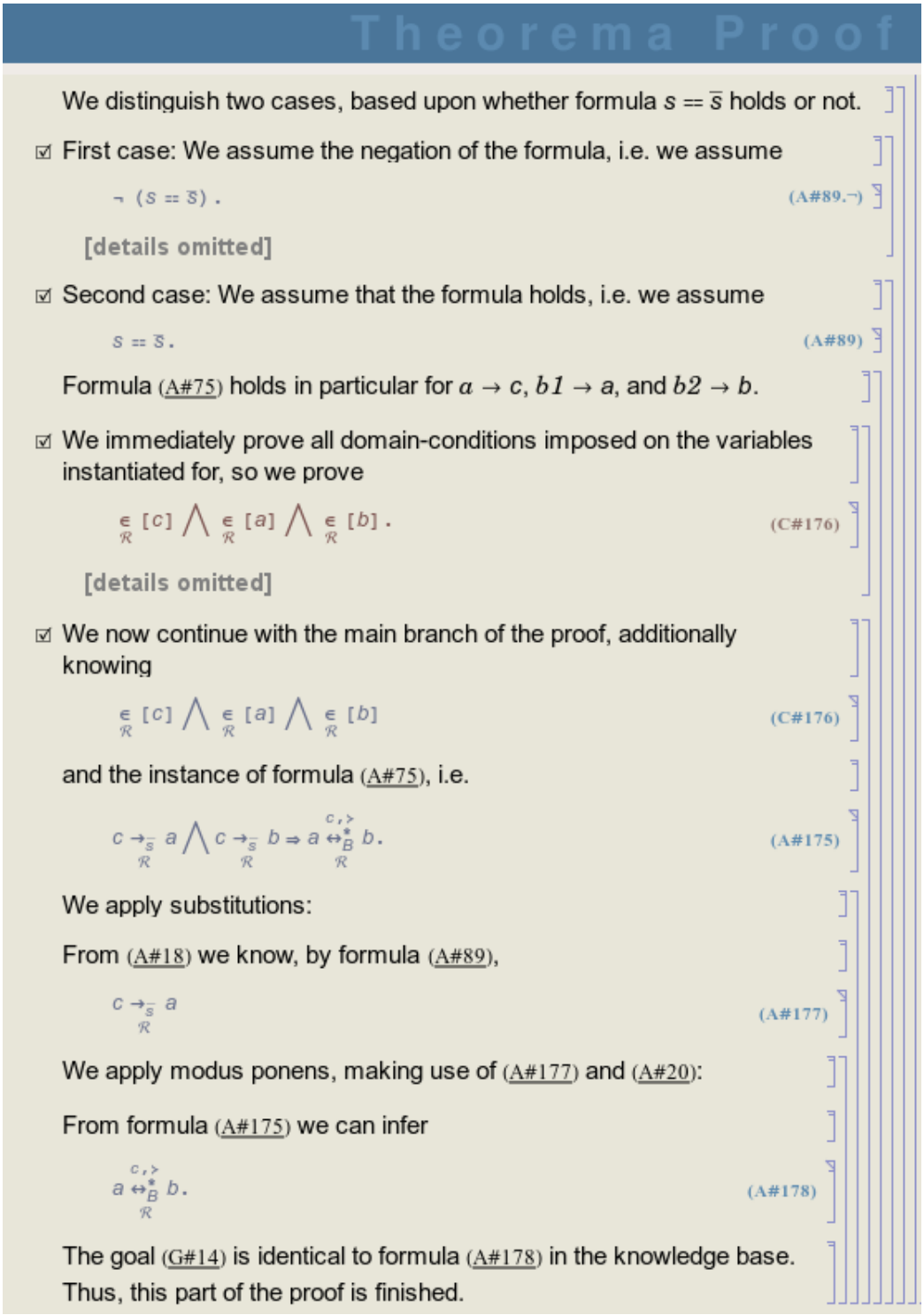

Figure 3: Part of the proof document of the proof of Theorem 16.

- $c_1 \neq c_2$ or
- there exist $1 \leq i \neq j \leq I_{c_1}$ such that $m_1 \in M^i_{c_1}$ and $m_2 \in M^j_{c_1}$.

Irrelative$_S$ already looks very similar to our definition of irrelativity, only that the indices i, j are not part of the notion itself, as free variables, but existentially quantified. In fact, irrelative$_S$ generalizes irrelativity due to [21], where it is only defined if $I_c = 2$ for all c. However, even irrelative$_S$ is still not suitable, for the following reason: The crucial property of irrelativity (with i, j existentially quantified) we would need for proving Theorem 12 is the following:

> For all $c, m_1, m_2, m_3 \in R$ with (m_1, c) and (m_2, c) irrelative$_S$ and (m_1, c) and (m_3, c) irrelative$_S$: (m_2, c) and (m_3, c) are not irrelative$_S$. (4.1)

Property (4.1) does not hold: If $I_c \geq 2$, $M^1_c \cap M^2_c \neq \emptyset$ and $m_1, m_2, m_3 \in M^1_c \cap M^2_c$, then all combinations of pairs (m_k, c) are irrelative$_S$, for $k = 1, 2, 3$.

Finally, after a thorough investigation of the proofs of Theorem 12 as well as the conservation theorem for polynomial domains, we arrived at our definitions of irrelativity, and also correlativity, as given in Definition 3. Instead of (4.1) there are now two crucial properties for proving the Main Theorem, due to the fact that the indices i and j are free variables in our definition of irrelativity. Either of them can easily be seen to be indeed satisfied:

> For all $c, m_1, m_2 \in R$ with (m_1, c) and (m_2, c) not correlative: There exist $1 \leq i, j \leq I_c$ such that (m_1, c) and (m_2, c) irrelative w. r. t. i and j. (4.2)

> For all $c, m_1, m_2, n_1, n_2 \in R$ and $1 \leq i, j \leq I_c$ with (m_1, c) and (m_2, c), (n_1, c) and (n_2, c) irrelative w. r. t. i and j: (m_k, c) and (n_k, c) are correlative, for $k = 1, 2$. (4.3)

5 Provers

In this section we describe the inference mechanisms (i. e. the *provers*) used for verifying the theory presented in the preceding sections. In Theorema, a prover consists of two (mostly independent) parts: a proof strategy (or, more precisely, a proof-search strategy), and a collection of inference rules. Inference rules transform proof situations, characterized by the current *proof goal* and the current *knowledge base*, to zero, one or more new, ideally simpler, proof situations. The inference rules used used in the present theory exploration naturally fall into

two categories: General-purpose predicate logic rules and special-purpose rules for the theories of integers, order relations, monoids and rings. The former will be described in Section 5.1, the latter in Section 5.2.

The proof strategy that was employed is a *fully interactive* strategy: the human user has full control over the inference rule to be applied to a proof situation, which branch in the proof tree to follow in case there are several possible ones, and what to do if a rule is applicable in several ways. All this happens in a dialog-oriented way, i. e. instead of writing a "proof script" that is later checked by the system, the user rather tries to find the proof by interacting with the system. The final result, still, is not only "proved" or "failed", but a nicely formatted, human readable document presenting the proof both by formal and informal means – just as always in Theorema.

Note that the fully-interactive proof strategy was implemented for exploring the theory of reduction rings, but it was designed sufficiently general and flexible to be used also together with other provers in other theory explorations.

5.1 General Inference Rules: RewriteInteractiveProver

The so-called *RewriteInteractiveProver* is a collection of general predicate-logic inference rules not attached to any particular theories and not depending on any knowledge about notions other than the usual logical quantifiers and connectives from predicate logic.[9] The inferences making up the prover are further distributed into five categories:

1. Propositional logic: Inference rules dealing with logical connectives in the obvious ways, e. g. distinguishing two cases based on a disjunction in the knowledge base.

2. Miscellaneous: Three more rules for logical connectives that are not as standard as the ones from the first category. In particular, one rule handles cases distinctions by first eliminating impossible cases and then distinguishing between the remaining ones, and the other rules simplify negated formulas by applying deMorgan rules for getting rid of negations that are not directly in front of predicate symbols.

3. Logical quantifiers: One rule for introducing "arbitrary but fixed" constants for variables bound by universal quantifiers in the proof goal, one rule for introducing Skolem constants for variables bound by existential quantifiers in the assumptions, and one rule for splitting a multi-range of

[9] The only exception being the *case distinction* construct, which is typically not considered part of core predicate logic.

one universal quantifier (in the goal) into several quantifiers with individual ranges. The latter rule is particularly useful in connection with certain higher-order backward rules (e. g. induction).

4. Rewriting: Several rules for all kinds of rewriting, for instance expanding explicit/implicit definitions, substituting equals by equals, or using (quantified) implications for reducing the goal.

5. Interactive rules: Inferences that cannot be applied completely automatically, but require a certain amount of user interaction. The most prominent examples of such rules are of course instantiation of quantifiers by suitable terms, or distinguishing two cases based upon whether an arbitrary formula holds or not. Another interactive rule gives the user full control over how to rewrite one or more formulas in the current proof situation by specific rewrite rules (unlike the inferences in the "Rewriting" category). And, last but not least, there is one rule that allows the user to add random formulas from the whole formalized theory to the proof, if they turn out to be needed.

The overall idea underlying the RewriteInteractiveProver is to apply standard propositional inferences as long as possible, and then rewriting the proof situation in a mostly goal-directed manner by performing substitutions and reducing to goal by backward reasoning. As the name suggests, the emphasis clearly lies on *interactive* proving where the human user decides which rule to apply in which way.

As for the rewriting, the default first-order rewriting mechanism currently in use in Theorema was enhanced by a higher-order one. At the moment, this mechanism is still part of the RewriteInteractiveProver and has not been integrated into the official release of Theorema, although this might change with future versions. Please also note that the higher-order rewriting mechanism is not complete in the sense that it can deal with all kinds of higher-order rules, but rather it treats a variable as higher-order only if it can evidently only be instantiated by one single term (that of course depends on the expression it is matched against), regardless of the expression it is matched against; otherwise it becomes a first-order variable, and application is only syntactic application instead of application modulo β-reduction. This, for instance, includes the case where the variable is applied only on *non-unifiable* arguments, each containing at least one *bound variable* outside the argument list of any other higher-order variable. Most[10] of the higher-order formulas one typically encounters in a theory exploration, like induction principles, are of such form.

[10]Though not all, as some examples in ElementaryTheories show.

The total number of inference rules in the RewriteInteractiveProver is 36, implemented by roughly 3200 lines of *Mathematica* code (this includes the functions for generating the interactive dialogs and the proof documents).

5.2 Special Inference Rules: ReductionRingProver

The *ReductionRingProver*, finally, extends the RewriteInteractiveProver by inference rules specifically designed to handle proof situations frequently arising in the theory of reduction rings. More precisely, the inferences only cover *basic* notions the theory is built upon, like tuples, order relations, and commutative rings, but no reduction-ring specific notions like reduction relations, irrelativity, etc.

The inferences are grouped into the following five classes, each being described in detail below: rules for intervals of integers, rules for tuple- and set membership, rules for order relations, rules for cancellative commutative monoids, and rules for commutative rings with unit. In total there are only eleven special inference rules, implemented by roughly 2400 lines of *Mathematica* code (including the functions for proof-document-generation).

5.2.1 Intervals of Integers

This group of rules contains only one single inference, which deals with proof situation where the proof goal is a (possibly universally quantified) integer-interval formula of the form $t \in \mathbb{Z}_{a,\ldots,b}$, or a conjunction thereof.

The rule basically employs closure properties of functions $+$, $-$ and $\cdot$, as well as some simple arithmetic truths (e. g. $b - (x - 1) \in \mathbb{Z}_{1,\ldots,b} \Leftrightarrow (b \in \mathbb{Z} \wedge x \in \mathbb{Z}_{1,\ldots,b})$) to successively simplify the proof goal. Knowledge about integer literals and other objects that belong to certain intervals of integers (e. g. the length of a tuple certainly belongs to $\mathbb{Z}_{0,\ldots,\infty}$) is used as well. Most of the knowledge implicitly used in this rule is also explicitly stated as an object-level formula in ElementaryTheories.

The reason for having a rule like this as part of the ReductionRingProver is simple: tuples play an important role in the theory, be it for defining transitive closures of reduction relations, or as summands in finite sums, and hence one very often has to deal with objects of the form T_i for a tuple T.[11] Now, properties of such objects can only be inferred if the index i can be shown to be between 1 and $|T|$, i. e. the length of T – and this is exactly expressed by the formula $i \in \mathbb{Z}_{1,\ldots,|T|}$.

[11] T_i denotes the i-th element of T, of course.

5.2.2 Tuples and Sets

This group contains two inference rules, one for tuples and one for sets. Since they behave completely analogously, we only describe the tuple-rule here.

The inference rule for tuples incorporates knowledge about the DomainTuples notion, a notion not built into standard Theorema but rather defined in ElementaryTheories, see Section 3.1.4. The inference rule itself is in fact a combination of the following basic inferences:

$$\frac{}{\underset{\text{DomainTuples}[D]}{\in} [T] \;\vdash\; \text{isTuple}[T]} \tag{5.1}$$

$$\frac{K \;\vdash\; i \in \mathbb{Z}_{1,\ldots,|T|}}{K, \underset{\text{DomainTuples}[D]}{\in} [T] \;\vdash\; \underset{D}{\in} [T_i]} \tag{5.2}$$

$$\frac{K \;\vdash\; \underset{\text{DomainTuples}[D]}{\in} [A] \qquad K \;\vdash\; \underset{\text{DomainTuples}[D]}{\in} [B]}{K \;\vdash\; \underset{\text{DomainTuples}[D]}{\in} [\text{join}[A, B]]} \tag{5.3}$$

$$\frac{K \;\vdash\; \underset{\text{DomainTuples}[D]}{\in} [A] \qquad K \;\vdash\; \underset{D}{\in} [a]}{K \;\vdash\; \underset{\text{DomainTuples}[D]}{\in} [(\text{append}|\text{prepend})[A, a]]} \tag{5.4}$$

$$\frac{K \;\vdash\; \underset{\text{DomainTuples}[D]}{\in} [A] \qquad K \;\vdash\; A \neq \langle\rangle}{K \;\vdash\; \underset{\text{DomainTuples}[D]}{\in} [(\text{rest}|\text{most})[A]]} \tag{5.5}$$

$$\frac{K \;\vdash\; \underset{\text{DomainTuples}[D]}{\in} [A]}{K \;\vdash\; \underset{\text{DomainTuples}[D]}{\in} [\text{reverse}[A]]} \tag{5.6}$$

$$\frac{K \;\vdash\; a \in \mathbb{Z} \qquad K \;\vdash\; b \in \mathbb{Z} \qquad K \;\vdash\; \forall_{i=a,\ldots,b}\, P[i] \Rightarrow \underset{D}{\in} [f[i]]}{K \;\vdash\; \underset{\text{DomainTuples}[D]}{\in} [\langle f[i] \,|_{i=a,\ldots,b}\, P[i]\rangle]} \tag{5.7}$$

The meaning of these basic inferences should be obvious. Only note that

$$\langle f(i) \,|_{i=a,\ldots,b}\, P(i)\rangle$$

in (5.7) is an *abstraction tuple*, in close analogy to abstraction terms of set theory. All of these inferences are justified by formulas explicitly stated in ElementaryTheories.

5.2.3 Order Relations

There are three inference rules contained in the group of ordering-rules: the first one ("orderingGoal") handles proof goals that are formulas of the form $a \preceq b$ or $a \npreceq b$ for orderings $\preceq$, the second one ("orderingKB") proof situations where formulas of that kind appear in the knowledge base, and the third one ("orderingEqualGoal") proof goals of the form $a = b$, if a and b are elements of an ordered domain. All of them are capable of dealing with three different kinds of orderings, namely

- partial irreflexive (i. e. asymmetric) orderings (here written as "PI" for the sake of brevity, in the formalization written as "isPartIrreflOrder"),
- partial reflexive (i. e. antisymmetric) orderings (here "PR", in the formalization "isPartReflOrder"), and
- total irreflexive orderings (here "TI", in the formalization "isTotalIrreflOrder").

All of PI, PR and TI are binary predicates, depending additionally on the domain D the respective relation is a partial/total ordering on. Furthermore,

$$\text{isOrderEmbedding}[\circ, \preceq, D]$$

states that $\circ$ is an order embedding w. r. t. $\preceq$ on D, i. e. that $a \circ c \preceq b \circ c$ whenever $a \preceq b$.

"orderingGoal" comprises the following basic inferences:

$$\frac{K \vdash \underset{D}{\in} [a]}{K, (\text{PI}|\text{TI})[\prec, D] \vdash a \nprec a} \tag{5.8}$$

$$\frac{K \vdash \underset{D}{\in} [a]}{K, \text{PR}[\preceq, D] \vdash a \preceq a} \tag{5.9}$$

TODO: Introduce n-ary version of domain membership predicate

$$\frac{K \vdash \underset{D}{\in} [a, b, c_1, \ldots, c_n]}{K, (\text{PI}|\text{TI})[\prec, D], b \prec c_1, c_1 \prec c_2, \ldots, c_n \prec a \vdash a \nprec b} \tag{5.10}$$

$$\frac{K \vdash a \neq b \qquad K \vdash \underset{D}{\in} [a, b, c_1, \ldots, c_n]}{K, \text{PR}[\preceq, D], b \preceq c_1, c_1 \preceq c_2, \ldots, c_n \preceq a \vdash a \npreceq b} \tag{5.11}$$

$$\frac{K \vdash \underset{D}{\in} [a, b, c_1, \ldots, c_n]}{K, (\text{PI}|\text{PR}|\text{TI})[\preceq, D], a \preceq c_1, c_1 \preceq c_2, \ldots, c_n \preceq b \vdash a \preceq b} \tag{5.12}$$

$$\frac{K \vdash \underset{D}{\in} [a,b,c] \qquad K \vdash a \preceq b}{K, (\textsc{pi}|\textsc{pr}|\textsc{ti})[\preceq, D], \textsc{isOrderEmbedding}[\circ, \preceq, D] \vdash a \circ c \preceq b \circ c} \tag{5.13}$$

"orderingKB" comprises the following basic inferences:

$$\frac{K \vdash \underset{D}{\in} [a]}{K, (\textsc{pi}|\textsc{ti})[\prec, D], a \prec a \vdash \Gamma} \tag{5.14}$$

$$\frac{K \vdash \underset{D}{\in} [a]}{K, \textsc{pr}[\preceq, D], a \not\preceq a \vdash \Gamma} \tag{5.15}$$

Finally, "orderingEqualGoal" comprises the following basic inferences:

$$\frac{K \vdash \underset{D}{\in} [b] \qquad K \vdash a \not\prec b \qquad K \vdash b \not\prec a}{K, \textsc{ti}[\prec, D], \underset{D}{\in} [a] \vdash a = b} \tag{5.16}$$

$$\frac{K \vdash \underset{D}{\in} [b] \qquad K \vdash a \preceq b \qquad K \vdash b \preceq a}{K, \textsc{pr}[\preceq, D], \underset{D}{\in} [a] \vdash a = b} \tag{5.17}$$

One remark on the inferences is still in place: Since Theorema's core logic is untyped, we cannot rely on the fact that, say, a, is an element of domain D if we are given a formula $a \preceq a$ and $\preceq$ is "defined" only on D. Hence, in all of the inferences above, membership of the involved terms in the respective domain always has to be checked explicitly.

5.2.4 Cancellative Commutative Monoids

Two special inference rules were designed to deal with *cancellative commutative monoids*, i. e. commutative monoids with the so-called *cancellation property*

$$x + z = y + z \;\Leftrightarrow\; x = y$$

Algebraic structures of that kind are needed when working with power-products of polynomials, as they have precisely the prescribed properties (w. r. t. multiplication, of course). The first inference rule simply reduces proof goals of the form $\underset{M}{\in} [1]$ and $\underset{M}{\in} [a \circ b]$, exploiting the facts that the neutral element 1 belongs to the monoid M and that the monoid operation $\circ$ is closed in$M$. Like the second inference rule described below, it is only applicable if $\textsc{isCCMonoid}[M]$ or $\textsc{isCommPPDomain}[M]$ appears in the current knowledge base.[12]

[12] $\textsc{isCommPPDomain}[M]$ expresses that M is a domain of commutative power-products, and thus by definition a cancellative commutative monoid.

The second rule is more involved: it simplifies goals of the form $a = b$, where a and b are elements of a cancellative commutative monoid, making use of the various properties of the monoid operation $\circ$ and the neutral element 1. Although this is straight-forward in principle, Theorema not being typed causes some trouble: Whenever we want to use, say, cancellativity to cancel a common subterm x of a and b, we have to make sure (by means of sub-proofs) that not only x, but also all other terms appearing in the argument list of any $\circ$ visited when traversing a and b to reach the respective occurrence of x – otherwise, cancellativity can simply not be used. Therefore, the rewrite-control underlying the inference rule does not blindly fully expand both sides of the equality and then cancels common subterms, but tries to be "smart" and perform only those rewrites that look promising, just to generate as few domain-membership side-conditions as possible. The far more complicated rule dealing with equality in commutative rings with unit, described below, behaves similarly.

5.2.5 Commutative Rings with Unit

As for cancellative commutative monoids, there are also special inference rules dealing with domain-membership and equality in commutative rings with unit, which is naturally a far more difficult task. One thing that complicates matters even more is the fact that besides the usual ring operations $+$, $-$ and $\cdot$ also finite sums of the form $\sum_{i=a,\ldots,b} f(i)$ have to be taken into account. Still, in principle the strategies for proving domain membership and equality are comparatively simple: Use closure properties of $+$, $-$ and $\cdot$ to successively reduce domain-membership goals, and in addition employ

$$\frac{K \vdash a \in \mathbb{Z} \qquad K \vdash b \in \mathbb{Z} \qquad K \vdash \forall_{i=a,\ldots,b} \underset{R}{\in} [f[i]]}{K, \textsc{isCommRing1}[R] \vdash \underset{R}{\in} [\sum_{i=a,\ldots,b} f[i]]} \tag{5.18}$$

to prove membership of sums.

Regarding equality, care has to be taken about domain-membership of terms when properties associativity/commutativity/distributivity/etc. of the various operations is to be used, exactly as in the rule for cancellative commutative monoids. To that end, the underlying rewrite-control at least tries to use these properties as few as possible. Please note that only the basic properties of $+$ are made use of when simplifying sum-expressions, e. g. associativity and commutativity are used to rewrite

$$\sum_{i=a,\ldots,b} (f[i] + g[i]) \longrightarrow \left(\sum_{i=a,\ldots,b} f[i]\right) + \left(\sum_{i=a,\ldots,b} g[i]\right)$$

given that all terms involved are in R. However, no "sum-splitting" and "sum-rearranging" theorems, e. g.

$$\sum_{i=a,\ldots,b} f[i] = \left(\sum_{i=a,\ldots,c} f[i]\right) + \left(\sum_{i=c+1,\ldots,b} f[i]\right) \quad \text{for } a \leq c \leq b$$

are used implicitly by the inference rule; instead, many of these are available explicitly as higher-order formulas in ElementaryTheories.

Besides the two rules for proving membership in R and for proving equality in R, also a third rule for simplifying known equalities is available. Internally, this rule relies on the same methods as the one for proving equality. Applicability of either of the three rules depends on the presence of isCommRing1$[R]$ or isReductionRing$[R]$ in the current knowledge base, since reduction rings are commutative rings with unit by definition.

5.3 General Remarks on Special Provers in Theorema

This subsection aims at clarifying the very purpose of having special provers with special inferences in the Theorema system. As already mentioned at the beginning of this section, proving in Theorema consists of two main components: the proof strategy and the collection of inference rules.

Usually, proof strategies are not very special in the sense they only can be used in particular theories, but in principle there are no limitations regarding their generality. However, at the moment all strategies implemented in Theorema 2.0 are of a rather general form that can effectively be used for basically all theories.

The component that really has the prospect of making provers special are the inference rules. One might argue that in principle only a few of these rules are sufficient to obtain a "reasonable" inference mechanism, e. g. natural deduction for higher-order predicate logic. Moreover, at the very core of every reasoner there in fact only needs to be a general (higher-order) conditional rewriting mechanism; Anything else (i. e. proofs and proof situations, inference rules, etc.) can then simply be formulated solely in the language of higher-order logic. Such an approach is pursued, for instance, in the well-known Isabelle system [16].

The philosophy in Theorema, however, is slightly different: Although we are of course aware of the fact that *in principle* only a few inference rules would suffice, we think that *in practice*, when developing a theory, one would also like to enhance the meta-level by special inference techniques to *efficiently* reason about new concepts and notions, without having to fall back to the very elementary rules of natural deduction and rewriting all the time.

In more concrete terms, there are at least two types of special inference rules one might envision:

1. Inference rules that are actually based on rewriting, but which have the rewrite rules "built-in" and maybe also implement a new control that applies these rules in a different way than the default, general-purpose rewriting mechanism usually employed in Theorema. As a simple example consider the theory of rings (cf. Section 5.2). When working in this theory one would certainly like to have a way for proving equality of two terms, making use of associativity, commutativity, etc. of the ring-operations $+$ and $\cdot$. Having a special inference rule available that incorporates all the knowledge about $+$ and $\cdot$ obviates the need for having the very definitions of rings, associativity, commutativity, etc. explicitly in the knowledge base, but instead it would be sufficient to only know, say, $\mathrm{isRing}[R]$ in order to prove $\underset{R}{0} \underset{R}{\cdot} \underset{R}{1} = \underset{R}{1} - \underset{R}{1}.\underset{R}{1}$. Furthermore, the various properties of $+$ and $*$ might be exploited in a more "clever"/efficient way than by simply rewriting both sides of the equality in all possible ways, until two identical expressions are found. In that sense, special inference rules can be understood as mere abbreviations of combinations of other, general ones.

2. Inference rules that enhance the logic of Theorema. Actually, Theorema is more of a logical "frame", and its logic is thus very elementary: It is *untyped* higher-order predicate logic with set theory, without a particular interpretation of, e. g., functions. Now, if someone wants to work in a strictly-typed logic, where functions are total mappings from one type to another that can only be defined by very special means (e. g. by primitive recursion), one can do so: Define new syntax for datatype- and function definitions and for type annotations (which should not pose any big issues thanks to the very flexible and extensible syntax of Theorema/*Mathematica*), and then implement a special inference rule that, when applied to a proof situation, type-checks all expressions and maybe also explicitly annotates all terms with their type; in short, the inference rule gives semantics to the newly introduced concept of types. Similarly, a recursively defined function could automatically be endowed with suitable induction rules for proving properties about it in an elegant way – this, for instance, is exactly what happens in Isabelle.

Summarizing, we do not claim that special inference rules cannot be simulated by other, elementary inference techniques as well. We do believe, however,

that efficient and "human-oriented" theory exploration profits from the possibility of designing and applying more and more elaborated and tailor-made inference techniques for all logics at all stages of the exploration. This, clearly, also means that at some point facilities for verifying new inference rules within the same logical system need to be available for ensuring the integrity of the system. Facilities of that kind are not yet integrated into Theorema,[13] meaning that new rules must be proved correct by pencil and paper.

6 Conclusion and Future Work

In the preceding sections we demonstrated how computer-supported mathematical theory exploration can be carried out in the Theorema system, following Theorema's paradigm of working in parallel on the object- and meta level: On the object level we formalized and formally verified large parts of the theory of reduction rings, and even managed to find and fix a subtle issue connected to the notion of irrelativity which, up to our knowledge, was not known before. On the meta level we designed special inference rules for efficiently dealing with concepts such as tuples and rings in proofs, and moreover even implemented a completely new, absolutely general, dialog-oriented interactive proof strategy.

Although we hope to have convinced the reader that lots of progress has already been made, there are still tasks left for the future. In the near future, we want to fill the remaining gaps in the formal verification; in particular, the "0"s in the "Proofs" column of Table 1 shall be replaced by positive numbers.

A more distant goal could be the investigation of *non-commutative* reduction rings. As of now, reduction rings were only considered in the commutative case, but making use of the existing formalization the extension to non-commutative domains might be practicable. A useful tool for undertakings like that might be some kind of a "proof synthesizer", i. e. a mechanism that takes a formula F to be proved and an existing proof P of a *similar* formula, and tries to construct, ideally *fully automatically*, a proof of F simply by re-doing the – suitably adjusted – inferences from P. All this is possible future work.

References

[1] Bruno Buchberger. A Criterion for Detecting Unnecessary Reductions in the Construction of Gröbner Bases. In E. W. Ng, editor, *Proceedings of the EUROSAM'79 Symposium on Symbolic and Algebraic Manipulation, Marseille, June 26-28, 1979*, volume 72 of *Lecture Notes in Computer Science*,

[13] There have already been first efforts in this direction, though; see [8]

pages 3–21. Copyright: Springer-Verlag, Berlin – Heidelberg – New York, 1979.

[2] Bruno Buchberger. A Critical-Pair/Completion Algorithm for Finitely Generated Ideals in Rings. In E. Boerger, G. Hasenjaeger, and D. Roedding, editors, *Logic and Machines: Decision Problems and Complexity (Proceedings of the Symposium "Rekursive Kombinatorik", Münster, May 23-28, 1983)*, volume 171 of *Lecture Notes in Computer Science*, pages 137–161. Copyright: Springer-Verlag, Berlin – Heidelberg – New York – Tokyo, 1984.

[3] Bruno Buchberger. Gröbner Rings in Theorema: A Case Study in Functors and Categories. Technical Report 2003-49, Johannes Kepler University Linz, Spezialforschungsbereich F013, November 2003.

[4] Bruno Buchberger, Adrian Craciun, Tudor Jebelean, Laura Kovacs, Temur Kutsia, Koji Nakagawa, Florina Piroi, Nikolaj Popov, Judit Robu, Markus Rosenkranz, and Wolfgang Windsteiger. Theorema: Towards Computer-Aided Mathematical Theory Exploration. *Journal of Applied Logic*, 4(4):470–504, 2006.

[5] Bruno Buchberger, Tudor Jebelean, Temur Kutsia, Alexander Maletzky, and Wolfgang Windsteiger. Theorema 2.0: Computer-Assisted Natural-Style Mathematics. *Journal of Formalized Reasoning*, 2015. to appear.

[6] Bruno Buchberger, Wolfgang Windsteiger, et al. *Theorema* – A System for Mathematical Theory Exploration. RISC, Johannes Kepler University Linz. `http://www.risc.jku.at/research/theorema/software/`.

[7] Leonard Eugene Dickson. Finiteness of the odd perfect and primitive abundant numbers with n distinct prime factors. *American Journal of Mathematics*, 35(4):413–422, 1913.

[8] Martin Giese and Bruno Buchberger. Towards Practical Reflection for Formal Mathematics. RISC Report Series 07-05, Research Institute for Symbolic Computation (RISC), University of Linz, Schloss Hagenberg, 4232 Hagenberg, Austria, 2007.

[9] J. Santiago Jorge, Victor M. Guilas, and Jose L. Freire. Certifying properties of an efficient functional program for computing Gröbner bases. *Journal of Symbolic Computation*, 44(5):571–582, May 2009.

[10] Äbdelilah Kandri-Rody and Deepak Kapur. Computing a Gröbner basis of a polynomial ideal over a Euclidean domain. *Journal of Symbolic Computation*, 6:37–57, 1988.

[11] Äbdelilah Kandri-Rody and Volker Weispfenning. Non-commutative Gröbner Bases in Algebras of Solvable Type. *Journal of Symbolic Computation*, 9:1–26, 1990.

[12] Alexander Maletzky. Automated Reasoning in Reduction Rings Using the Theorema System. In W. Koepf and E. V. Vorozhtsov, editors, *Proceedings of CASC'15 (Computer Algebra in Scientific Computing, Aachen, Germany, September 18–24)*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2015. to appear.

[13] Inmaculada Medina-Bulo, Francisco Palomo-Lozano, and Jose-Luis Ruiz-Reina. A verified Common Lisp implementation of Buchberger's algorithm in ACL2. *Journal of Symbolic Computation*, 45(1):96–123, January 2010.

[14] André Saint Eudes Mialebama Bouesso and Djiby Sow. Noncommutative Gröbner Bases over Rings. *Communications in Algebra*, 43(2):541–557, 2015.

[15] Teo Mora. An introduction to commutative and non-commutative Gröbner bases. *Theoretical Computer Science*, 134(1):131–173, 1994.

[16] Tobias Nipkow, Lawrence C. Paulson, Makarius Wenzel, et al. Isabelle. `https://isabelle.in.tum.de/`.

[17] Franz Pauer. Gröbner bases with coefficients in rings. *Journal of Symbolic Computation*, 42(11–12):1003–1011, 2007.

[18] Birgit Reinert. Gröbner bases in function rings – A guide for introducing reduction relations to algebraic structures. *Journal of Symbolic Computation*, 41(11):1264–1294, 2006.

[19] Yosuke Sato, Shaturo Inoue, Akira Suzuki, Katsusuke Nabeshima, and Ko Sakai. Boolean Gröbner bases. *Journal of Symbolic Computation*, 46(5):622–632, 2011.

[20] Christoph Schwarzweller. Gröbner Bases – Theory Refinement in the Mizar System. In M. Kohlhase, editor, *Mathematical Knowledge Management (4th International Conference, MKM 2005, Bremen, Germany, July 15-17)*, volume 3863 of *Lecture Notes in Artificial Intelligence*, pages 299–314. Springer Berlin Heidelberg, 2006.

[21] Sabine Stifter. A Generalization of Reduction Rings. *Journal of Symbolic Computation*, 4(3):351–364, 1988.

[22] Sabine Stifter. The Reduction Ring Property is Hereditary. *Journal of Algebra*, 140(89-18):399–414, 1991.

[23] Laurent Thery. A Machine-Checked Implementation of Buchberger's Algorithm. *Journal of Automated Reasoning*, 26:107–137, 2001.

[24] Wolfgang Windsteiger. Theorema 2.0: A System for Mathematical Theory Exploration. In Chee Yap and Hoon Hong, editors, *Proceedings International Congress on Mathematical Software (ICMS'2014), Seoul, Korea, August 5-8, 2014*, volume 8592 of *Lecture Notes in Computer Science (LNCS)*, pages 49–52, 2014.

[25] Franz Winkler and Bruno Buchberger. A Criterion for Eliminating Unnecessary Reductions in the Knuth-Bendix Algorithm. In *Colloqium on Algebra, Combinatorics and Logic in Computer Science*, pages 849–869, 1983.

[26] Wolfram Research, Inc. *Mathematica*. `http://www.wolfram.com/mathematica`.

[27] Gail Zacharias. Generalized Gröbner Bases in Commutative Polynomial Rings, 1978. Bachelor's thesis.