Fundamental study

Buchberger's algorithm:
The term rewriter's point of view

Reinhard Bündgen*

Wilhelm-Schickard-Institut, Universität Tübingen, D-72076 Tübingen, Germany

Received May 1993; revised October 1994
Communicated by M. Wirsing

Abstract

We analyse the relations between completion procedures for polynomials and terms and thereby show how Buchberger's algorithm for multivariate polynomials over finite fields and over the rationals can be simulated using term completion modulo AC. To specify the rational numbers an infinite term rewriting system is needed. However, for the simulation of each particular ideal completion a finite approximation of the infinite rule set is sufficient. This approximation can be constructed during the completion. The division operation needed in Buchberger's algorithm reduces to a narrowing procedure which becomes part of the critical pair computation process.

Contents

1. Introduction ........................................................................................................ 144
2. Preliminaries ................................................................................................. 147
  2.1. Abstract completion .............................................................................. 148
  2.2. Buchberger's algorithm ...................................................................... 149
  2.3. Knuth–Bendix completion .................................................................. 152
  2.4. Generalised symmetrisation ................................................................ 154
  2.5. Critical pair transformation .................................................................. 155
3. Polynomials as terms .................................................................................... 156
4. Simulating Buchberger's algorithm .............................................................. 160
  4.1. The simplification relation .................................................................... 160
  4.2. The termination of type-1 rules .............................................................. 162
  4.3. Superpositions ....................................................................................... 163
  4.4. The selection of the leading term ............................................................ 164
  4.5. The division problem ............................................................................ 166
5. Completion of polynomials over finite fields ................................................ 167
  5.1. Polynomials over prime fields ................................................................. 167
  5.2. The symmetrisation of G-rules ................................................................. 169
  5.3. Arbitrary finite fields ............................................................................. 171

* E-mail: buendgen@informatik.uni-tuebingen.de.

0304-3975/96/$15.00 © 1996 — Elsevier Science B.V. All rights reserved
SSDI 0304-3975(95)00061-5
1. Introduction

Completion procedures are probably the most powerful tools to reason with (quotient) structures presented by equations. Examples of such structures are finitely presented groups, ideals in rings and equational varieties in universal algebra. The purpose of completion procedures is to compute a canonical simplifier for a given structure such that equivalence of two objects can be decided by comparing their respective normal forms. Completion procedures are typically based on three operations that are non-deterministically applied to a set of equations and a set of rules: a simplification procedure which is normally described by a set of rules. The simplification procedure is applied to equations and rules. The set of rules is incrementally extended to a canonical simplifier by an orientation procedure which derives new rules from equations. The third operation is performed by a superposition procedure that derives new equations, called critical pairs, from two rules. In general completion procedures are semi-decision procedures in that a canonical simplifier will eventually be found if one exists but the termination of the procedure cannot be guaranteed.

The most famous completion procedures are Buchberger's algorithm [8] for polynomial ideals and the Knuth-Bendix completion procedure [38] for equational specifications over first order terms. Buchberger's algorithm computes a canonical simplifier for ideals in multivariate polynomial rings over fields. This simplifier is presented by a so-called Gröbner- or standard base of the ideal. Gröbner bases are effective tools to decide ideal membership, ideal inclusion, equivalence modulo an ideal and related problems. The Knuth-Bendix procedure computes on success a canonical term rewriting system for a given set of equations over first order terms. Given a canonical term rewriting system, the word problem over the input specification can be decided. Both procedures have been extended to be applicable to a wider range of domains. Thus there are extensions of Buchberger's algorithm to deal with polynomials over certain commutative rings [10,34,42,57]. The Knuth-Bendix procedure has been generalised to deal with finite equivalence classes of terms (e.g., terms including associative and commutative (AC) operators; [31,41,54]). Besides the aforementioned completion procedures...
there are further completion procedures for finitely presented algebraic structures like (Abelian) semi-groups, monoids, groups, rings, modules (see e.g., [5, 7, 43]).

The purpose of this paper is to elucidate the connection between Buchberger's algorithm and the Knuth–Bendix procedure. To be more precise, we want to describe Buchberger's algorithm by means of term rewriting systems and the Knuth–Bendix procedure. Thus showing that Buchberger's algorithm can be interpreted as a special instance of the Knuth Bendix procedure. Understanding in detail how the two procedures relate will hopefully help to translate and transfer findings from one area to the other.

The similarity between Buchberger's algorithm and the Knuth–Bendix completion procedure has long been observed. The first attempt to relate the two completion procedures was published by Loos [47]. He was followed by many other authors [2, 12, 13, 15, 31, 33, 35, 46, 51, 56, 59, 61]. The main finding of this research was the common classification of both procedures as completion procedures thereby interpreting s-polynomials as critical pairs. Most of the work was done in the spirit to show that the Knuth–Bendix procedure provides a more general formalism than Buchberger's algorithm. Therefore, the goal often was to simulate Buchberger's algorithm by term rewriting methods. Thus it was noticed by most authors that term completion modulo associativity and commutativity (AC) is needed to handle specifications of polynomial rings. It was already suggested in [46] to use the AC-canonical rewrite system for commutative rings for simulating Buchberger's algorithm. Kandr–Rody et al. [33] and Bündgen [15] succeeded to show how the extension of Buchberger's algorithm to deal with integral polynomials [42, 42] can be simulated by term completion modulo AC. However, Buchberger's algorithm for polynomials over fields could not be described by pure term rewriting systems. [35, 56] suggest common generalisations of the two procedures and [2] proposed to build-in the coefficient domain. On the theoretical side, some results on critical pair criteria found for Buchberger's algorithm [9] could already be carried over to term completion [39, 59, 60]. With regard to applications, it is notable that both procedures were used for theorem proving using formulae in exclusive-or normal form [26, 36] and for the completion of finitely presented commutative structures [33, 43].

Despite so much work on the connection between the two procedures, still some issues were open. There was no rigorous investigation on how to move from the variable free rule presentations in Buchberger's algorithm to term rewriting systems with variables. The major problem, however, was the translation of the orientation operation which seems much more complicated for Buchberger's algorithm than it is for the Knuth–Bendix procedure. Orienting polynomial equations among others involves exact divisions in a field. This seemed to be an insuperable problem because it is well known that fields cannot be specified as equational varieties.

\[\text{Indeterminates correspond to constants!}\]
\[\text{Le Chenadec [43] performed such studies for a number of algebraic structures, but he was not concerned about Buchberger's algorithm.}\]
In this paper, we show that Buchberger’s algorithm can be completely described in terms of term rewriting systems and their completion. In particular, we will present a simulation of Buchberger’s algorithm for polynomials over certain fields: arbitrary finite fields and the rational numbers. Simulating Buchberger’s algorithm means that the input polynomials are translated to an equational specification over first order terms. This specification will be completed by a Knuth-Bendix procedure modulo AC and the result can be transformed back to a Gröbner base. To be precise, the equational specification can be partitioned into a set of equations describing the polynomial domain and one ground equation for each input polynomial. The resulting canonical term rewriting system will then consist of a canonical specification $\mathcal{K}$ for the polynomial domain and a set of extension rules which corresponds to a Gröbner base (cf. Fig. 1).

Analysing this simulation, we show how the simplification, orientation and superposition operations of Buchberger’s algorithm correspond to certain parts of the term completion procedure. In particular the division problem can be solved: Exact division corresponds to certain unification problems that are solvable whenever there is a solution in the initial model (or equivalently in the ground term model). Thus it suffices that the equational specification, restricted to terms denoting coefficients, has an initial model that is isomorphic to the coefficient field. Note that it is possible to find equational specifications whose varieties (i.e., the classes of all models) do not represent fields but whose initial models are fields. We present such specifications for arbitrary finite fields. For the rational numbers, we can construct an arbitrary exact approximation of such a specification so that we can ensure that all divisions needed during the completion can be simulated. The orientation procedure as a whole is then simulated by a generalised symmetrisation process with regard to $\mathcal{K}$.
A second aspect of our simulation technique is that it describes a generic Buchberger’s algorithm for different coefficient domains because the equational specification for the polynomial domain can be partitioned into a prototype specification describing polynomials over a commutative ring $\mathcal{L}X$ and additional equations $\mathcal{H}$ describing the coefficient domain. Thus an AC-completion procedure where $\mathcal{L}X$ is fixed as some part of the input amounts to a generic Buchberger’s algorithm that can be parametrised by different coefficient domains. If we fix both $\mathcal{L}X$ and $\mathcal{H}$, we get a specific Buchberger’s algorithm.

The plan of the paper is as follows. In Section 2, we first settle the preliminaries for abstract completion, polynomial completion and term completion. In addition, we present our idea of a generalised symmetrisation procedure and a confluence criterion for critical pairs that will be needed in later proofs. In Section 3, we will present a canonical term rewriting system that specifies multivariate polynomials over commutative rings. This system reduces every ground term simulating a polynomial to its distributive normal form. It is our prototype specification for polynomials. Using this specification, we show in Section 4 how the operations of simplification, superposition and selection of leading terms as rule left-hand sides of Buchberger’s algorithm can be simulated. These simulations require that the rewrite rules have a particular format that corresponds to monic polynomials. Therefore, we must show that equations and rules of the required format will actually be computed and that all other equations and rules not belonging to that format can safely be neglected because they will eventually be deleted. This last proof is rather technical and depends on the coefficient domain chosen because it involves solving the division problem mentioned before. We show that for finite fields (Section 5) and for the rational numbers (Section 6), we always get rules of the required format. Therefore, we can simulate Buchberger’s algorithm for polynomials over these domains. In Section 7, we comment on some observations concerning our simulation. In Section 8, we summarise our results and conclude with some remarks on the relevance of our work.

2. Preliminaries

In this section, we first introduce the preliminaries and the notation needed in the rest of the paper. Since many notions are common to the domains of term rewriting and polynomial reductions, we will start with a description of abstract completion procedures (see also [22, 27]) and then proceed with the preliminaries of polynomial completion (see [4, 11, 25] for surveys) and term completion (surveys can be found in [21, 22, 29, 37, 55]). The reader familiar with these subjects may safely skip Sections 2.1–2.3 since we try to follow standard notation as far as possible.

Following these surveys, we explain our idea of a generalised symmetrisation process and we introduce a critical pair criterion which is an important tool used in the proofs of the remaining paper.
2.1. Abstract completion

Let $\to_{\mathcal{R}}$ be a binary relation over a domain $\mathcal{D}$. Then $\leftarrow_{\mathcal{R}}$ is the inverse relation of $\to_{\mathcal{R}}$, $\leftrightarrow_{\mathcal{R}}$, $\rightarrow_{\mathcal{R}}$, $\leftrightarrow^*_{\mathcal{R}}$ and $\rightarrow^*_{\mathcal{R}}$ are the symmetric-, transitive-, transitive and reflexive- and the symmetric, transitive and reflexive closures of $\to_{\mathcal{R}}$, respectively. The relation $\to_{\mathcal{R}}$ is terminating if there is no infinite chain $a_1 \to_{\mathcal{R}} a_2 \to_{\mathcal{R}} \cdots$.

A relation $\to_{\mathcal{R}}$ is confluent if for all $a, b, c \in \mathcal{D}$ such that $b \leftrightarrow_{\mathcal{R}} a \rightarrow_{\mathcal{R}} c$ there is a $d \in \mathcal{D}$ with $b \leftrightarrow_{\mathcal{R}} d \to_{\mathcal{R}} c$. We then write $b \downarrow_{\mathcal{R}} c$. To prove the confluence of a relation it often suffices to show a weaker condition. $\leftrightarrow_{\mathcal{R}}$ is locally confluent if for all $a, b, c \in \mathcal{D}$ with $b \leftrightarrow_{\mathcal{R}} a \rightarrow_{\mathcal{R}} c, b \downarrow_{\mathcal{R}} c$ follows. In [53], Newman showed that for all terminating relations, confluence is equivalent to local confluence.

An object $a \in \mathcal{D}$ is reducible by $\to_{\mathcal{R}}$ if there is another object $a' \neq a$ such that $a \rightarrow_{\mathcal{R}} a'$, otherwise $a$ is irreducible. If $\to_{\mathcal{R}}$ is confluent and terminating then $\rightarrow_{\mathcal{R}}$ is called canonical. Then for each object $a \in \mathcal{D}$ there is an irreducible object $a' \in \mathcal{D}$ with $a \rightarrow_{\mathcal{R}} a'$ and $a'$ is called the normal form of $a$ w.r.t. $\to_{\mathcal{R}}$. We write $a' = a_{\downarrow_{\mathcal{R}}}$.

A rule has the form $l \to r$ where the left-hand side (LHS) $l$ and the right-hand side (RHS) $r$ are patterns for a set of objects in $\mathcal{D}$. A rule may be applied to an object which 'contains' a part that 'fits' the pattern of $l$. This part will then be 'replaced' by a corresponding part which 'fits' the pattern of $r$. The 'part' to be replaced is called a redex. The exact meaning of 'containment', 'fitting' and 'replacement' depends on the domain $\mathcal{D}$. A set of rules $\mathcal{R}$ defines a reduction relation $\to_{\mathcal{R}}$ in the sense that $a \to_{\mathcal{R}} b$, if $b$ can be obtained from $a$ by a single application of a rule in $\mathcal{R}$. We therefore say $\mathcal{R}$ is terminating, confluent, canonical, etc. if $\to_{\mathcal{R}}$ is too. $\mathcal{D}_{\downarrow_{\mathcal{R}}}$ is the set of $\mathcal{R}$-normal forms of $\mathcal{D}$.

An equation of the form $a \leftrightarrow b$ is similar to a rule, with the difference that it may be applied in both directions. Thus a set of equations $\mathcal{E}$ defines a symmetric relation $\leftrightarrow_{\mathcal{E}}$.

Both rules and equations will be used to construct equality proofs among objects in $\mathcal{D}$. Therefore we write $=_{\mathcal{E}}$ for $\leftrightarrow_{\mathcal{E}}$ and $=_{\mathcal{E}}$ for $\rightarrow_{\mathcal{E}}$. Given an equivalence relation $=_{\mathcal{E}}$ defined by a set of equations $\mathcal{E}$, it would be rewarding to have a canonical rule set $\mathcal{R}$ with $=_{\mathcal{E}} = =_{\mathcal{R}}$. Then $a =_{\mathcal{E}} b$ holds iff $a_{\downarrow_{\mathcal{R}}} = b_{\downarrow_{\mathcal{R}}}$.

Completion procedures operate on a set of equations and a set of rules. Given a non-empty set of equations $\mathcal{E}$ and an empty set of rules the goal of a completion procedure is to derive the empty set of equations and a canonical set of rules $\mathcal{R}$ such that $=_{\mathcal{E}} = =_{\mathcal{R}}$. Completion procedures are based on three operations which are non-deterministically repeated until the desired result has been computed. First there is a simplification operation. The simplification operation must be terminating and it is described by the set of rules. Equations and (optionally) rules may be simplified. If an equation simplifies to a trivial equation (e.g., $a = a$) it will be deleted. The second operation is called orientation. It transforms an equation into a rule thereby restricting the applicability of the equation. The oriented rules must preserve the termination property of the simplification relation. The third operation deduces a new equation from a pair of rules. This step is called superposition. The criterion for the procedure
Delete: $$\frac{(\mathcal{E} \cup \{s \leftarrow t\}; \mathcal{R})}{(\mathcal{E}; \mathcal{R})}$$ if $s = t$

Compose: $$\frac{(\mathcal{E}; \mathcal{R} \cup \{s \rightarrow t\})}{(\mathcal{E}; \mathcal{R} \cup \{s \rightarrow u\})}$$ if $t \not\rightarrow \mathcal{R} u$

Simplify: $$\frac{(\mathcal{E} \cup \{s \rightarrow t\}; \mathcal{R})}{(\mathcal{E} \cup \{s \leftrightarrow u\}; \mathcal{R})}$$ if $t \rightarrow \mathcal{R} u$

Orient: $$\frac{(\mathcal{E} \cup \{s \leftrightarrow t\}; \mathcal{R})}{(\mathcal{E}; \mathcal{R} \cup \{l_i \rightarrow r_i\})}$$ if $l_i \succ r_i \text{ }^a$, $l_i = \{s \leftarrow t\} r_i$, $s \downarrow \{l_i \downarrow r_i\} t$

Collapse: $$\frac{(\mathcal{E}; \mathcal{R} \cup \{s \leftarrow t\})}{(\mathcal{E} \cup \{u \leftarrow t\}; \mathcal{R})}$$ if $\begin{cases} s \rightarrow \mathcal{R} u \text{ by} \\ l \rightarrow r \in \mathcal{R} \text{ where } s \nabla l \text{ }^b \end{cases}$

Superpose: $$\frac{(\mathcal{E}; \mathcal{R})}{(\mathcal{E} \cup \{s \leftrightarrow t\}; \mathcal{R})}$$ if $s \leftarrow \mathcal{R} u \rightarrow \mathcal{R} t$

$^a \succ \quad \not\subset \rightarrow \nabla$ is a terminating ordering on $\mathcal{D} \times \mathcal{D}$.

$^b \nabla \quad$ is a terminating ordering on $\mathcal{D} \times \mathcal{D}$.

![Fig. 2. Inference rules for abstract completion.](image)

to stop is that no new equations can be derived and no equations are left in the set of equations. In Fig. 2, we present an abstract completion procedure as a set of inference rules in a style suggested by Bachmair and Dershowitz [1, 21]. The composition and collapsing inference rules describe optional simplification steps. They are often included for efficiency reasons. With these inference rules a pair $(\mathcal{E}; \mathcal{R})$ of equations and rules can be transformed into a different pair $(\mathcal{E}'; \mathcal{R}')$ such that $(=_{\mathcal{E}} \cup =_{\mathcal{R}}) = (=_{\mathcal{E}'} \cup =_{\mathcal{R}'})$.

We use $\vdash$ for the inference relation. For example, we write $(\mathcal{E}; \mathcal{R}) \vdash^{*} (\mathcal{E}'; \mathcal{R}')$. If $\mathcal{E}' = \emptyset$ and no further new inference steps can be applied, then $\mathcal{R}'$ is canonical and we call $\mathcal{R}'$ the result of the completion. By abuse of notation, we will sometimes simply talk about completing $\mathcal{E} \cup \mathcal{R}$, or of completing $\mathcal{E}$ or $\mathcal{R}$ if $\mathcal{R}$ or $\mathcal{E}$ are empty, respectively.

2.2. Buchberger's algorithm

We consider multivariate polynomials in $\mathbb{k}[x_1, \ldots, x_r]$ where the coefficient domain $\mathbb{k}$ is a field and $\{x_1, \ldots, x_r\}$ is the set of indeterminates. A monomial $m$ is a polynomial of the form $m = c \alpha^\mathbf{x}$, where $0 \neq c \in \mathbb{k}$ and $\alpha^\mathbf{x}$ is a power product of indeterminates. For the purpose of this paper, we present all polynomials $p$ in distributive normal form. That is, $p$ is either the zero polynomial or a monomial or of the form $\sum_{i=1}^{n} m_i$, where the $m_i$ are monomials with pairwise distinct power products. We write $p[m]$ if the polynomial in distributive normal form contains the monomial $m$. Note that $p[m]$
does not follow from \( p = p' + m \). In the sequel, all definitions relate to polynomials represented in distributive normal form unless explicitly stated otherwise.

An ordering \( \geq \) on non-zero polynomials is called \textit{admissible} if the one-polynomial is minimal with regard to \( \geq \) and for all non-zero polynomials \( p, q, \) and \( m, m \cdot p \geq m \cdot q \) if \( p \geq q \). The \textit{leading term} \( LT(p) \) of a non-zero polynomial \( p \) is the monomial in the distributive normal form of \( p \) that is maximal with regard to \( \geq \). The \textit{leading coefficient} \( LC(p) \) of \( p \) is the coefficient of \( LT(p) \). The \textit{leading power product} \( LP(p) \) of \( p \) is the power product of \( LT(p) \). The \textit{reductum} of \( p \) is \( RED(p) = p - LT(p) \).

\textbf{Example 1.} Let \( \succeq_x \) be an ordering that orders power products lexicographically w.r.t. their exponent vectors. Since polynomial addition is associative and commutative the distributive normal form of a polynomial can be identified with the multiset of its monomials. The multiset extension of \( \succeq_x \) is an admissible ordering.

Let \( p = 3x_1^2x_2^2 + 8 + x_1x_3 + x_2^2x_3^2 \) then for the multiset extension of \( \succeq_x \)
\[
LT(p) = x_1^2x_3, \quad LC(p) = 1 \quad \text{and} \quad RED(p) = 3x_1^3x_2^3 + 8 + x_2^2x_3^2.
\]

Let \( P = \{p_1, \ldots, p_n\} \) be a set of polynomials in \( \mathbb{K}[x_1, \ldots, x_r] \). The \textit{ideal presented by} \( P \) is
\[
I = \{ p \mid p = \sum_{i=1}^{n} q_i p_i, \ p_i \in P, q_i \in \mathbb{K}[x_1, \ldots, x_r] \}
\]
and we write \( I = (p_1, \ldots, p_n) \). An ideal presentation \( (p_1, \ldots, p_n) \) describes an equivalence relation over polynomials modulo the set of equations \( \{ p_1 = 0, \ldots, p_n = 0 \} \).

\textit{Buchberger's algorithm} is a completion procedure that compiles an ideal presentation for \( I \) into another presentation \( G \) that can be interpreted as a rule set describing a canonical reduction relation. The resulting presentation \( G \) is called a \textit{Gröbner-} or \textit{standard base} for \( I \). In order to understand how Buchberger's algorithm works, let us describe the simplification, orientation and superposition operations for polynomials.

The polynomial reduction relation \( \rightarrow_p \) associated with a set \( P \) of polynomials and an admissible ordering is defined as follows: For \( p, q \in \mathbb{K}[x_1, \ldots, x_r] \), \( p \) reduces to \( q \) modulo \( P \) if there is a polynomial \( r \in P \), such that \( p[m \cdot LT(r)] \) and \( q = p - mr \) for some monomial \( m \). That is, a multiple of \( LT(r) \) in \( p \) is replaced by the respective negative multiple of \( RED(r) \). Thus a polynomial \( p \) may be interpreted as rule a \( LT(p) \rightarrow \neg RED(p) \). It is convenient to present rules as monic polynomials (i.e., each polynomial is divided by its leading coefficient). Note that normalising a polynomial \( p \) with regard to a rule \( LT(q) \rightarrow \neg RED(q) \) computes some remainder of \( p \) divided by \( q \). It has been shown that \( \rightarrow_p \) is terminating for every set of monic polynomials \( P \) and every admissible ordering.

Orienting a polynomial equation \( p = q \) into a rule means to select the leading term of \( p - q \) divided by its coefficient as the left-hand side expression and to take the

---

3 For an algebraic characterisation of Gröbner bases see elsewhere in the literature. For our purposes the above characterisation suffices.
\[ G \leftarrow \text{BA}_{\mathcal{K}}(P, \succ) \]

[Buchberger's algorithm. \( P \) is a set of polynomials presenting an ideal \( \mathcal{I} \) in \( \mathcal{K}[x_1, \ldots, x_r] \) where \( \mathcal{K} \) is a field. \( \succ \) is an admissible ordering on polynomials. Then \( G \) is the Gröbner base of \( \mathcal{I} \).]

1. [Initialise.] \( E := \{ p \mapsto 0 \mid p \in P \} \); \( R := \emptyset \).

2. [Simplify.] while the simplify-inference rule applies do
   \((E; R) := \text{Simplify}((E; R))\).

3. [Delete.] while the delete-inference rule applies do \((E; R) := \text{Delete}((E; R))\).

4. [Stop?] if \( E = \emptyset \) then return \( G = \{ l \mapsto r \mid l \rightarrow r \in R \} \) and stop.

5. [Orient.] Let \( a \leftarrow b \in E \); \( E := E \setminus \{ a \leftarrow b \} \); \( l := \text{LT}(a - b)/\text{LC}(a - b) \);
   \( r := -\text{RED}(a - b)/\text{LC}(a - b) \); \( R := R \cup \{ l \mapsto r \} \).

6. [Collapse.] while the collapse-inference rule applies do \((E; R) := \text{Collapse}((E; R))\).

7. [Compose.] while the compose-inference rule applies do \((E; R) := \text{Compose}((E; R))\).

8. [Superpose.] Compute all those critical pairs \( C \) of rules in \( R \) where the rule \( l \mapsto r \) participates. \( E := E \cup C \); \( R := R \cup \{ l \mapsto r \} \).

   continue with step 2.

---

* The leading term (LT) and the reductum (RED) are computed w.r.t. \( \succ \).
* Here we identify pairs and equations. Equivalently we could compute the set \( S \) of all s-polynomials of \( \{ r \mapsto l \mid l \rightarrow r \in R \} \) and let \( C = \{ s = 0 \mid s \in S \} \).

---

negated reductum of \( p - q \) divided by the leading coefficient as the right-hand side expression. Thus \( p = q \) becomes the rule

\[
\frac{\text{LT}(p - q)}{\text{LC}(p - q)} \rightarrow -\frac{\text{RED}(p - q)}{\text{LC}(p - q)}.
\]

Remember that polynomials in an ideal presentation must be interpreted as set to zero.

Let \( p_1, p_2 \) be two polynomials. \( p_1 \) and \( p_2 \) superpose if their leading terms have a non-trivial greatest common divisor, i.e., \( \gcd(\text{LT}(p_1), \text{LT}(p_2)) = q \neq 1 \). Let \( LT(p_1) = q \cdot p_1' \) and \( LT(p_2) = q \cdot p_2' \) such that \( \text{lcm}(p_1, p_2) = LT(p_1) \cdot p_2' = LT(p_2) \cdot p_1' \) is the least common multiple of \( LT(p_1) \) and \( LT(p_2) \) then

\[
\text{spol}(p_1, p_2) = p_1 \cdot p_2' - p_2 \cdot p_1'.
\]

is the s-polynomial of \( p_1 \) and \( p_2 \). The pair \((\text{RED}(p_1) \cdot p_2', \text{RED}(p_2) \cdot p_1')\) is called the critical pair of \( p_1 \) and \( p_2 \). The s-polynomials of any two members of an ideal \( \mathcal{I} \) are also in that ideal. Therefore \( \text{spol}(p_1, p_2) = 0 \) can be considered as an equation derived from \( p_1 \) and \( p_2 \). Buchberger showed that a set of polynomials \( G \) presents a canonical reduction relation if and only if all s-polynomials of any two members of \( G \) reduce to zero. Fig. 3 presents an implementation of Buchberger's algorithm.
2.3. Knuth–Bendix completion

Throughout this paper we denote the finite set of sorted, ranked function symbols and constants by $\mathcal{F}$ and the set of variables by $\mathcal{V}$. Each function symbol is implicitly accompanied by a sort description $f : s_1 \times \cdots \times s_n \rightarrow s$ for $f \in \mathcal{F}$ and the $s, s_1$ are elements of a fixed set of sorts. In the same way each variable is assigned a fixed sort. Terms are defined recursively as follows: Each constant or variable which is assigned the sort $s$ is a term of sort $s$. If $t_1, \ldots, t_n$ are terms of sorts $s_1, \ldots, s_n$, respectively, and $f : s_1 \times \cdots \times s_n \rightarrow s$ then $f(t_1, \ldots, t_n)$ is a term of sort $s$. Nothing else is a (well formed) term. $T(\mathcal{F}, \mathcal{V})$ denotes the set of all well formed terms. A term $t$ without variables is called a ground term. $T(\mathcal{F})$ is the set of ground terms. Let $p$ be a position in a term $t$. Then $t\lfloor_p$ denotes the subterm of $t$ at position $p$, and the result of replacing the subterm $t_1\lfloor_p$ of a term $t_1$ by a term $t_2$ is denoted by $t_1[t_2]_p$.

A substitution $\sigma : \mathcal{V} \rightarrow T(\mathcal{F}, \mathcal{V})$ is a mapping from the set of variables into the set of terms. If we apply a substitution $\sigma$ to a term $t$ we write $t\sigma$. In $t\sigma$ all variables $x$ occurring in $t$ are replaced simultaneously by $\sigma(x)$. $t\sigma$ is called an instance of $t$. If $s\sigma = t\sigma$ for some substitution $\sigma$, then $\sigma$ is called a unifier of $s$ and $t$. The most general unifier of $s$ and $t$ $\mu = mgu(s, t)$ is a unifier of $s$ and $t$, such that all unifiers of $s$ and $t$ are instances of $\mu$. $t\mu$ is then called the most general common instance.

A set of term equations $\mathcal{E}$ induces a symmetrical relation $\rightarrow_\mathcal{E}$ such that $s \rightarrow_\mathcal{E} t$ if for some position $p$ and some substitution $\sigma$, $s\lfloor_p = u\sigma$, $t = s[\sigma]_p$ and either $u \leftrightarrow v \in \mathcal{E}$ or $v \leftrightarrow u \in \mathcal{E}$.

A rewrite rule is a pair of terms $l \rightarrow r$ such that all variables in $r$ occur in $l$. A term rewriting system is a set of rewrite rules. A term rewriting system $\mathcal{R}$ induces a rewrite relation $\rightarrow_\mathcal{R}$ such that $s \rightarrow_\mathcal{R} t$ if for some position $p$ and some substitution $\sigma$, $s\lfloor_p = l\sigma$, $t = s[\sigma]_p$ and $l \rightarrow r \in \mathcal{R}$. We say a term rewriting system $\mathcal{R}$ is confluent, terminating, canonical, etc. if $\rightarrow_\mathcal{R}$ is.

The termination property of a term rewriting system $\mathcal{R}$ is undecidable in general but there exist powerful criteria to guarantee the termination of term rewriting systems. These criteria involve showing that $\rightarrow_\mathcal{R}$ is included in a terminating ordering on terms. See [20] for a survey. Orienting an equation $u \leftrightarrow v$ either yields the rule $u \rightarrow v$ or the rule $v \rightarrow u$. Which of the two orientations is to be chosen is normally controlled by the term ordering used for the termination proof.

The superposition operation for term rewriting systems consists in computing critical pairs. Let $l_1 \rightarrow r_1, l_2 \rightarrow r_2 \in \mathcal{R}$, $p$ is a non-variable position in $l_2$ and $\mu$ be such that $l_1\mu = l_2\lfloor_p\mu$ is the most general common instance (meci) of $l_1$ and $l_2\lfloor_p$ (i.e., $\mu = mgu(l_1, l_2\lfloor_p)$). Then $(l_2[r_1]_p\mu, r_2\mu)$ is called a critical pair of $l_1 \rightarrow r_1$ and $l_2 \rightarrow r_2$, $l_2\mu$ is a superposition term, and

$l_2[r_1]_p\mu / l_2\mu \leftarrow r_2\mu$

is a critical peak of the two rules. A critical pair $(t_1, t_2)$ is confluent if $t_1 \downarrow_\mathcal{R} = t_2 \downarrow_\mathcal{R}$.

Knuth and Bendix showed that a terminating term rewriting system is confluent iff all its critical pairs are confluent. They proposed a completion procedure to compute a
Knuth–Bendix completion procedure.

\( \mathcal{R} \leftarrow \text{KB}(\mathcal{E}, \succ) \)

\( \mathcal{E} \) is a set of term equations and \( \succ \) is a terminating ordering over \( T(\mathcal{F}, \text{Var}) \). Then \( \mathcal{R} \) is a canonical term rewriting system with \( =_{\mathcal{R}} = =_{\mathcal{E}} \).

1. [Initialise.] \( \mathcal{R} := \emptyset \).
2. [Simplify.] while the simplify-inference rule applies do
   \( (\mathcal{E}; \mathcal{R}) := \text{Simplify}(\mathcal{E}; \mathcal{R}) \).
3. [Delete.] while the delete-inference rule applies do \( (\mathcal{E}; \mathcal{R}) := \text{Delete}(\mathcal{E}; \mathcal{R}) \).
4. [Stop?] if \( \mathcal{E} = \emptyset \) then return \( \mathcal{R} \) and stop.
5. [Orient.] Let \( a \leftrightarrow b \in \mathcal{E} \) then \( \mathcal{E} := \mathcal{E} \setminus \{a \leftrightarrow b\} \);
   if \( a \succ b \) then \( \{l := a; r := b; \mathcal{R} := \mathcal{R} \cup \{l \rightarrow r\}\} \);
   else if \( b \succ a \) then \( \{l := b; r := a; \mathcal{R} := \mathcal{R} \cup \{l \rightarrow r\}\} \);
   else stop with failure.
6. [Collapse.] while the collapse-inference rule applies do
   \( (\mathcal{E}; \mathcal{R}) := \text{Collapse}(\mathcal{E}; \mathcal{R}) \).
7. [Compose.] while the compose-inference rule applies do
   \( (\mathcal{E}; \mathcal{R}) := \text{Compose}(\mathcal{E}; \mathcal{R}) \).
8. [Superpose.] Compute all those critical pairs \( \mathcal{C} \) of rules in \( \mathcal{R} \) where the rule \( l \rightarrow r \) participates. \( \mathcal{E} := \mathcal{E} \cup \mathcal{C}^a \);
   continue with step 2

Fig. 4. Knuth–Bendix Procedure.

canonical term rewriting system for an equational specification \( \mathcal{E} \) [38]. An implementation of the Knuth–Bendix completion procedure is presented in Fig. 4. In contrast to Buchberger’s algorithm, the Knuth–Bendix procedure does not always terminate. It may fail due to a non-orientable equations or it may run for ever.

As mentioned in the introduction, some operations on polynomials like addition and multiplication are both associative and commutative (AC). Associative and commutative operators like + are called AC-operators. Unfortunately, equations describing these properties like

\( x + y \leftrightarrow y + x \) and \( (x + y) + z \leftrightarrow x + (y + z) \)

cannot be oriented without destroying the termination property of any term rewriting system. Therefore associativity and commutativity are treated as built-in properties of AC-operators. In the presence of AC-operators, all notions of equality, matching, unification, reduction, termination, confluence, etc. must be treated modulo AC. That is, they must be extended to AC-equivalence classes of term. With these extensions it is possible to design a term completion procedure for terms with AC-operators [41, 54, 31]. In this article, we use the procedure proposed by Peterson and Stickel, but we are confident that the other procedures can be used as well.
Two major differences between the plain Knuth-Bendix procedure and the Peterson-Stickel procedure should be noted. First, the most general unifier of two terms modulo AC consists in general of a set of substitutions $M$ such that any AC-unifier is an instance of an element of $M$. Hence there may be more than one critical pair between two rules superposing at a position. Second, AC-extension rules must be added to some rules. A set of rules containing all necessary AC-extension rules is called AC-compatible (see [54] for details.). During completion, the generation of extension rules can be considered as a part of the orientation procedure or as derived from a 'critical pair' between a rule and the associativity law.

The term rewriting systems we need in this article are always term rewriting systems modulo associativity and commutativity. Therefore we will often omit saying 'modulo AC'. In particular, we will use $\rightarrow\text{AC}$ to denote rewriting relations modulo AC.

An equational specification is a pair $(\mathcal{F}, \mathcal{E})$. If the signature $\mathcal{F}$ is understood we often speak of the equational specification $\mathcal{E}$. The class of all models of $(\mathcal{F}, \mathcal{E})$ is called the variety of $(\mathcal{F}, \mathcal{E})$. If we speak of a model or variety of a term rewriting system $\mathcal{R}$ we mean the model or variety of $\{l \leftarrow r \mid l \rightarrow r \in \mathcal{R}\}$. According to a theorem of Birkhoff [6] an equation $s \leftarrow t$ is valid in the variety of $(\mathcal{F}, \mathcal{E})$ iff $s =_\mathcal{E} t$. Hence equality in the variety of a canonical term rewriting system is decidable. Very often one is only interested in a standard model of $(\mathcal{F}, \mathcal{E})$, the so called initial model which is isomorphic to the $\mathcal{E}$-equivalence classes of ground terms. It is therefore also called the ground term model. The initial model of a canonical term rewriting system $\mathcal{R}$ is isomorphic to the ground terms in $\mathcal{R}$-normal form. Term completion procedures are complete in the sense that every ground equation valid in the initial model can eventually be proven unless the orientation step fails [28].

2.4. Generalised symmetrisation

Symmetrisation as a part of a completion process has first been discovered in the context of group completion [14] and has then been generalised by Le Chenadec [43] to the completion of other finitely presented algebraic structures like (Abelian) semigroups, monoids, groups, (commutative) rings and modules.

Given a canonical term rewriting system $\mathcal{R}_S$ describing an algebraic structure $S$, the symmetrisation of a ground equation $s \leftarrow t$ (or rule $s \rightarrow t$, resp.) consists of a restricted completion of $\{s \leftarrow t\}; \mathcal{R}_S$. During symmetrisation only those critical pairs are computed in which a rule of $\mathcal{R}_S$ participates. The result of symmetrising a ground equation (or rule) is often predictable in that it associates to each ground equation (or rule) a set of extension rules that fit certain patterns. Typically, symmetrisation cannot only be applied to ground equations and ground rules but also to all equations and rules having the format of the extension rules. That is, the set of rules computed by symmetrisation is closed under the symmetrisation operation. In this case, we call all rules and equations symmetrisable that have the format of extension rules.

Now consider the case that all critical pairs of each two extension rules yield symmetrisable equations. Then the completion of ground equations (or symmetrisable
equations) modulo $\mathcal{R}_S$ can be interpreted as a more abstract completion procedure with symmetrisation as the orientation procedure and computation of critical pairs between extension rules as the superposition procedure. In this abstract completion procedure all terms are immediately reduced to $\mathcal{R}_S$-normal form. Note that term completion with symmetrisation is still an implementation of a pure Knuth-Bendix completion (modulo AC) unless advantage is taken of the knowledge that extension rules always have a predictable format. Completion with symmetrisation w.r.t. $\mathcal{R}_S$ can be interpreted as completion of ground equations modulo the equational theory presented by $\mathcal{R}_S$.

It will turn out that our simulation of Buchberger's algorithm fits into that scheme of an abstract completion modulo a canonical term rewriting system describing polynomials. Thus Buchberger's algorithm can be seen as a term completion procedure modulo the theory of polynomials where the knowledge of the format of the extension rules and critical pairs is exploited. That is, each polynomial corresponds to a set of extension rules or equations.

2.5. Critical pair transformations

In this section, we describe a technique for transforming critical pairs into simpler ones. This technique is essential both for keeping many proofs in this paper of a tractable size and to cut down the complexity of the simulation. During the completion a lot of redundant critical pairs will be computed. Many of these critical pairs can be safely eliminated using a critical pair criterion based on the following theorem [16]:

**Theorem 1.** Let $\mathcal{R} \cup \{l \to r\}$ be an AC-compatible and terminating term rewriting system and let $\mathcal{E}$ be a set of equations such that $(\mathcal{E}; \mathcal{R} \cup \{l \to r\}) \vdash^* (\mathcal{E}'; \mathcal{R} \cup \{l' \to r'\})$. Further let $\mathcal{R} \cup \{l' \to r'\}$ be AC-compatible and terminating. If there is a position $q$ in $l$ and a substitution $\tau$ such that $l|_q =_{AC} l'\tau$, then all critical pairs of $\mathcal{R} \cup \{l \to r\}$ are confluent if both $(l[l'\tau]_q, r)$ and all critical pairs of $\mathcal{R} \cup \{l' \to r'\}$ are confluent w.r.t. $\mathcal{R} \cup \{l' \to r'\}$.

With this theorem, it is possible to deduce any consequence of the equations and rules in $\mathcal{E}$ and $\mathcal{R}$ by computing new critical pairs between orientable equations and rules or between two orientable equations. This can be repeated until a 'convenient' rule is deduced. Whenever the theorem applies we can forget all critical pairs derived in intermediate deductions. Specialising Theorem 1 to deductions containing exactly one superposition yields a new inference rule for critical pair transformations:

$$
\frac{(\mathcal{E} \cup \{l \leftrightarrow r\}; \mathcal{R})}{(\mathcal{E} \cup \{l' \leftrightarrow r'\}; \mathcal{R})} \quad \text{if} \quad \begin{cases} l \Rightarrow r, \quad l' \Rightarrow r', \quad l \triangleright l',^4 \quad (l', r') \text{ is a } \mathcal{R}\text{-normalised AC-critical pair of } l \to r \text{ and } \mathcal{R}, \\ (l, r) \text{ is confluent by } \mathcal{R} \cup \{l' \to r'\}. \end{cases}
$$

---

$^4\triangleright$ is a terminating ordering on $\mathcal{E} \times \mathcal{E}$
The applicability of this inference rule can effectively be tested and it can be implemented. It turns out to be very helpful during symmetrisation; in particular in presence of AC-operators (see [16] for an investigation of this phenomenon).

3. Polynomials as terms

We want to specify the abstract data type *polynomial* by a many-sorted canonical term rewriting system. As a matter of fact we will present here a canonical term rewriting system $\mathcal{K}$ describing $\mathbb{Z}[x_1, \ldots, x_n]$. $\mathcal{K}$ will serve as some kind of prototype for all other rewrite specifications of polynomial domains in that they can be derived from $\mathcal{K}$ by adding new constants and rules. As we will see in Section 4 many aspects of the simulation of Buchberger’s algorithm can already be demonstrated using $\mathcal{K}$.

We specify polynomials using a many-sorted term algebra in order to stress that implementations of polynomials are of a type that is constructed from a type for the coefficient domain and a type for the Abelian monoid of indeterminates. We need a coefficient sort $\text{Coeff}$, a sort for power products $\text{Ind}$ (short for indeterminate) and a sort $\text{Poly}$ for polynomials. The set of operators $\mathcal{F}$ together with their signatures and intended meanings is listed in Table 1. The addition and multiplication operators are considered to be both associative and commutative. Using this signature expressions of integral polynomials can easily be translated to ground terms. For example, the polynomial expression $3x^2y + (-y + 2)(x - 1)$ translates to

$$M(1 + 1 + 1, X, X, Y) \oplus ((M(-1, Y) \oplus M(1 + 1, I)) \odot (M(1, X) \oplus M(-1, I))).$$

The laws known for the coefficients, indeterminates and polynomials can be specified by equations which in turn can be compiled into a canonical term rewriting system. From now on we consider $a, b, c, d$ to be variables of sort $\text{Coeff}$; $x, y, z$ to be variables of sort $\text{Ind}$; and $f, g, h$ to be variables of sort $\text{Poly}$ if they occur in terms of our polynomial.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Domain</th>
<th>Range</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0:</td>
<td>-&gt; Coef</td>
<td></td>
<td>Coefficient zero</td>
</tr>
<tr>
<td>1:</td>
<td>-&gt; Coef</td>
<td></td>
<td>Coefficient one element</td>
</tr>
<tr>
<td>-:</td>
<td>Coef</td>
<td>Coef</td>
<td>Additive inversion for coefficients</td>
</tr>
<tr>
<td>+:</td>
<td>Coef x Coef</td>
<td>Coef</td>
<td>Coefficient addition (AC)</td>
</tr>
<tr>
<td>::</td>
<td>Coef x Coef</td>
<td>Coef</td>
<td>Coefficient multiplication (AC)</td>
</tr>
<tr>
<td>I:</td>
<td>-&gt; Ind</td>
<td></td>
<td>$x^0$, one-element of Ind</td>
</tr>
<tr>
<td>X, Y, X_i</td>
<td>-&gt; Ind</td>
<td></td>
<td>Indeterminates</td>
</tr>
<tr>
<td>.:</td>
<td>Ind x Ind</td>
<td>Ind</td>
<td>Indeterminate multiplication (AC)</td>
</tr>
<tr>
<td>M:</td>
<td>Coef x Ind</td>
<td>Poly</td>
<td>Monomial constructor</td>
</tr>
<tr>
<td>Ω:</td>
<td>-&gt; Poly</td>
<td></td>
<td>Zero polynomial</td>
</tr>
<tr>
<td>⊕:</td>
<td>Poly x Poly</td>
<td>Poly</td>
<td>Polynomial addition (AC)</td>
</tr>
<tr>
<td>⊙:</td>
<td>Poly x Poly</td>
<td>Poly</td>
<td>Polynomial multiplication (AC)</td>
</tr>
</tbody>
</table>
algebra. The integers and all coefficient domains in consideration are commutative rings with ones. The variety of commutative rings with ones can be described by the following set of equations

\[ \mathcal{E}_x = \{ a + 0 \leftrightarrow a, \quad a + (-a) \leftrightarrow 0, \]
\[ a \cdot 1 \leftrightarrow a, \quad a \cdot (b + c) \leftrightarrow (a \cdot b) + (a \cdot c) \}. \]

Completion of \( \mathcal{E}_x \) modulo AC yields the well-known canonical term rewriting system for commutative rings with unit elements [54] that is given in Specification 1:

**Specification 1.**

\[ \mathcal{X} = \{ 1 : a + 0 \rightarrow a, \quad 6 : a + (-a) \rightarrow 0, \]
\[ 2 : a \cdot 0 \rightarrow 0, \quad 6x : b + (a + (-a)) \rightarrow b, \]
\[ 3 : a \cdot 1 \rightarrow a, \quad 7 : -(a + b) \rightarrow -a + (-b), \]
\[ 4 : -0 \rightarrow 0, \quad 8 : a \cdot -b \rightarrow -(a \cdot b), \]
\[ 5 : -(a) \rightarrow a, \quad 9 : a \cdot (b + c) \rightarrow (a \cdot b) + (a \cdot c) \}. \]

In Specification 1 and all following specifications, rules are labelled. This allows us to refer to the rule \( i : l \rightarrow r \in \mathcal{R} \) by \( \mathcal{R}.i \). Rules with labels \( ix \) (for \( i \in \mathbb{N} \)) are AC-extension rules [54] of the rules labelled by \( i \). These extension rules are needed for technical reasons if AC-reductions are to be based on AC-matches at a fixed position.

The ground term model of \( \mathcal{X} \) is isomorphic to the integers because only terms of the form

\[ 0, \ -1, \ 1, \ -1 + 1, \ 1 + 1, \ -1 + -1 + -1, \ 1 + 1 + 1, \ldots \]

are irreducible. The canonical rewrite rule specification of polynomials over commutative rings with unit elements is given in Specification 2.

**Specification 2.** \( \mathcal{XX} = \mathcal{X} \cup \mathcal{X} \) where

\[ \mathcal{X} = \{ 1 : x.I \rightarrow x, \]
\[ 2 : M(0,x) \rightarrow \Omega, \]
\[ 3 : M(a,x) \oplus M(b,x) \rightarrow M(a + b,x), \]
\[ 3x : f \oplus M(a,x) \oplus M(b,x) \rightarrow f \oplus M(a + b,x). \]
\[ 4 : M(a,x) \odot M(b,y) \rightarrow M(a \cdot b,x,y), \]
\[ 4x : f \odot M(a,x) \odot M(b,y) \rightarrow f \odot M(a \cdot b,x,y), \]
\[ 5 : f \odot \Omega \rightarrow f, \]
\[ 6 : f \odot \Omega \rightarrow \Omega, \]
\[ 7 : f \odot M(1,I) \rightarrow f. \]
\[ 8 : f \odot (g \oplus h) \rightarrow (f \odot g) \oplus (f \odot h), \]
\[ 9 : (f \odot M(a,x)) \oplus (f \odot M(b,x)) \rightarrow f \odot M(a + b,x), \]
\[ 9x : g \oplus (f \odot M(a,x)) \oplus (f \odot M(b,x)) \rightarrow g \oplus (f \odot M(a + b,x)), \]
\[ 10 : f \odot M(a + 1,I) \rightarrow f \odot (f \odot M(a,I)), \]
Rule 1 defines an Abelian monoid, thus the ground normal forms of the sort \( \text{Int} \) are isomorphic to the power products over the indeterminates. Rules 5–8 and 11 specify some ring axioms for polynomials (existence of a zero, a one and of additive inverses, and the distributivity law). The remaining rules are needed to define operations on monomials. In particular, monomials with common power products must be combined. Note that we did not define explicit operators for the additive inverse of polynomials and scalar multiplication. Such operators could easily be defined by reducing them to a multiplication of a polynomial with an appropriate monomial term. The ground normal forms w.r.t. \( \mathcal{F} \) of sort \( \text{Poly} \) are of the forms

\[
\Omega, M(k,X), M(k_1,\vec{X}_1) \oplus \ldots \oplus M(k_n,\vec{X}_n),
\]

where the \( k, k_i \neq 0, \vec{X}, \vec{X}_i \) are ground terms in normal form and the \( \vec{X}_i \) are disjoint within a sum of monomials for all \( 1 \leq i \leq n \). Thus the canonical form for polynomials produced by \( \mathcal{F} \) is the distributive normal form which is convenient for Gröbner basis computations. We will denote the isomorphism from polynomials to terms in \( \mathcal{F} \)-normal form by \( \psi \). If it is clear from the context, \( \psi \) may also be extended to map coefficients to terms of sort \( \text{Coeff} \).

**Example 2.**

<table>
<thead>
<tr>
<th>Polynomial</th>
<th>Term ( \psi(p) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x^2y - 3 )</td>
<td>( M(1, X \cdot X \cdot Y) \oplus M(-1 + -1 + -1, I) )</td>
</tr>
<tr>
<td>( -x^3 + 2y^2 + 1 )</td>
<td>( M(-1, X \cdot X \cdot X) \oplus M(1 + 1, Y \cdot Y) \oplus M(1, I) )</td>
</tr>
<tr>
<td>0</td>
<td>( \Omega )</td>
</tr>
<tr>
<td>8</td>
<td>( M(1 + 1 + 1 + 1 + 1 + 1 + 1, I) )</td>
</tr>
<tr>
<td>( 2(x + y) )</td>
<td>( M(1 + 1, X) \oplus M(1 + 1, Y) )</td>
</tr>
<tr>
<td>( x^2 + y - 2y )</td>
<td>( M(1, X \cdot X) \oplus M(-1, Y) )</td>
</tr>
</tbody>
</table>

Note, that we do not allow for overloading of zeroes \( (0 \neq \Omega) \), ones \( (1 \neq I \neq M(1, I)) \), addition \( (1 + M(1 + 1, X) \) is not a well-formed term \) or multiplication \( ((1 + 1) \cdot M(1, Y) \) and \( 1 \cdot X \) are not well-formed).

The confluence of \( \mathcal{F} \) has been shown using the AC-completion procedures of the ReduX- and REVE-rewriting laboratories [17, 44]. To prove the termination of \( \mathcal{F} \), we show that every left-hand side in \( \mathcal{F} \) is greater than its corresponding right-hand side w.r.t. some simplification ordering [19]. Our simplification ordering is based on a polynomial interpretation of terms using the following interpretation function:

**Definition 1.** Let \( \Phi : T(\mathcal{F}, \text{Var})/\text{AC} \rightarrow \mathbb{R}[\text{Var}] \) be a class of functions from the set of terms modulo AC to the set of multivariate polynomials over the reals. \( \Phi \) is
inductively defined by the following constraints:

\[
\begin{align*}
\Phi(x) &= x & \text{for every variable } x \in \mathcal{V}ar, \\
\Phi(0) &> 6, \Phi(1) \geq 2, \Phi(I) \geq 6, \Phi(\Omega) \geq 6, \\
\Phi(X_i) &\geq 6 & \text{for every ind-constant } X_i, \\
\Phi(a + b) &= \Phi(a) + \Phi(b) + 5, \\
\Phi(a \cdot b) &= \Phi(a)\Phi(b), \\
\Phi(x, y) &= \Phi(x)\Phi(y), \\
\Phi(-a) &= K_1\Phi(a) + 2, & \text{where } K_1 \geq 7/5, \\
\Phi(M(a, x)) &= (\Phi(a) + K_2)\Phi(x), & \text{where } K_2 \geq 5, \\
\Phi(f \oplus g) &= \Phi(f) + \Phi(g) + 5, \\
\Phi(f \odot g) &= \Phi(f)\Phi(g).
\end{align*}
\]

We define an ordering \( \succcurlyeq_p \) on polynomials such that for two polynomials \( f \) and \( g \), \( f \succcurlyeq_p g \) if \( f \) is eventually greater than or equal to \( g \) (i.e., \( \exists \overline{y} : \forall \overline{x} : \overline{x} > \overline{y} \Rightarrow f(\overline{x}) \geq g(\overline{x}) \)). This ordering induces an ordering \( \succeq_\Phi \) on terms with \( t_1 \succeq_\Phi t_2 \) if \( \Phi(t_1) \succeq_p \Phi(t_2) \). The orderings \( \succeq_\Phi \) are terminating because \( \Phi(t) \geq 2 \) for all ground terms \( t \) and \( \Phi(x) \) is strictly ascending for \( x \geq 2 \). Now we can conclude that \( \mathcal{X} \mathcal{X} \) is terminating because \( l \succeq_\Phi r \) for all \( l \rightarrow r \in \mathcal{X} \mathcal{X} \). For a survey on termination proofs for term rewriting systems see [20].

**Theorem 2.** \( \mathcal{X} \mathcal{X} \) is canonical modulo AC.

**Notation.** For the ease of presentation, we will introduce a more compact notation for terms representing polynomials. \( \overline{X}, \overline{Y}, \overline{Z} \) or any indexed version of these vectors represents a power product in ground normal form. We will use \( k, n \) or any indexed version of these letters as repetition factors of coefficients. Thus \( ka \) is a short hand for

\[
\underbrace{a + \cdots + a}_{k\text{-times}}
\]

if \( k \) is positive otherwise it means

\[
\underbrace{-a + \cdots + -a}_{-k\text{-times}}
\]

\( k \) is an abbreviation for \( k1 \) if \( k \) is positive or \(-k(-1)\) otherwise. We also write \( a^k \) for

\[
\underbrace{a \cdot \cdots \cdot a}_{k\text{-times}}
\]

if \( k \geq 0 \) and \( \bigoplus_{i=1}^n M(k_i, \overline{X}_i) \) for \( M(k_1, \overline{X}_1) \oplus \cdots \oplus M(k_n, \overline{X}_n) \), where sums of zero terms denote the constant \( \Omega \). Optional parts in a term will be put in brackets. Thus \( a[+b] \) means \( a \) or \( a + b \). In a rule, optional parts on both sides of the arrow belong together. In our examples, the reader can easily detect which of several optional terms in a rule or an equation are related.
4. Simulating Buchberger’s algorithm

In order to simulate Buchberger’s algorithm for a set of given input polynomials \( \mathbf{P} \), we first augment \( \mathcal{K} \) by a set of equations \( \mathcal{H} \) specifying the coefficient domain. Then we translate each \( P \in \mathbf{P} \) to a ground equation \( \psi(P) \rightarrow \Omega \). Now completing

\[ \mathcal{K} \cup \mathcal{H} \cup \{ \psi(P) \rightarrow \Omega \mid P \in \mathbf{P} \} \]

modulo AC simulates Buchberger’s algorithm applied to \( \mathbf{P} \). It may of course be advantageous to compute first the canonical term rewriting system \( \mathcal{K} \) for \( \mathcal{K} \cup \mathcal{H} \) before ‘entering’ the polynomials.

**Example 3.** Suppose we want to compute the Gröbner base of

\[ \mathbf{P} = \{ x^2 y - x^2 + 2xy, y^2 - y + 1 \} \]

in \( \mathbb{Z}/(2\mathbb{Z})[x, y] \).

First we compute the AC-canonical term rewriting system \( \mathcal{K} \) from \( \mathcal{K} \cup \{ 1+1 \rightarrow 0 \} \). \( \mathbf{P} \) can be translated to the set of equations

\[ \mathcal{P} = \{ M(1, X.Y) \oplus M(-1, X.X) \oplus M(1 + 1, X.Y) \rightarrow \Omega, \]
\[ M(1, Y.Y) \oplus M(-1, Y) \oplus M(1, I) \rightarrow \Omega \}. \]

Given an appropriate term ordering, AC-completion of \( \mathcal{K} \cup \mathcal{P} \) yields the canonical term rewriting system

\[ \mathcal{K} \cup \{ M(a, Y.Y) \rightarrow M(a, Y) \oplus M(-a, I), \]
\[ M(a, Y.Y.x) \rightarrow M(a, Y.x) \oplus M(-a, x), \]
\[ M(a, X.X) \rightarrow M(a + a, X.Y) \oplus M(-a + -a, X), \]
\[ M(a, X.X.x) \rightarrow M(a + a, X.Y.x) \oplus M(-a + -a, X.x) \}

from which we can extract the Gröbner base \( \mathbf{G} = \{ y^2 - y + 1, x^2 - 2xy + 2x \} \).

We will show that complete systems like that of Example 3 will actually be computed and how a Gröbner base can be extracted from the resulting system. Therefore, we will investigate in this section, which parts of the term completion procedure correspond to the simplification, the superposition procedure and the selection of leading terms within the orientation procedure. These operations can be analysed independently from the coefficient domain (i.e., \( \mathcal{K} \)) by looking at \( \mathcal{K} \) only. The analysis of operations depending on the coefficient domain will be deferred to the next two sections.

4.1. The simplification relation

For the rest of this section, we will assume that rewrite rules denoting polynomials always have a particular format.

Let \( P \) be a monic polynomial. Then \( P \) can be interpreted as the rule \( LT(P) \rightarrow -RED(P) \). When we translate this rule to the term domain we obtain the ground rule
Definition 2. Given the monic polynomial \( P = \bar{x} + \sum_{i} k_i \bar{x}_i \) where \( \bar{x} > \bar{x}_i \) for all \( i \) and \( \psi(\bar{x}) = x, \psi(\bar{x}_i) = \bar{x}_i \), then
\[
M(1, \bar{x}) \to \bigoplus_i M(k_i, \bar{x}_i)
\]
is the ground rule (g-rule) associated with \( P \) and
\[
M(a, \bar{x}) \to \bigoplus_i M(k_i a, \bar{x}_i), \quad M(a, \bar{x} x) \to \bigoplus_i M(k_i a, \bar{x}_i x)
\]
(where \( k_i a \) is the \( \mathcal{N} \mathcal{X} \)-normal form of \( \psi(k) \cdot a \)) are the extension rules (e-rules) associated with \( P \). The first e-rule is of type \( e_1 \) and the second of type \( e_2 \).

Ground rules can also be associated with non-monic polynomials. Their left-hand side is then of the form \( M(k, \bar{x}) \) for \( k \neq 1 \). We will say ground and extension rules associated with monic polynomials are of type \( I \). Note that given a g-rule or an e-rule, we can uniquely identify its associated polynomial. E-rules associated with \( P \) apply to any monomial term (with top operator \( \mathcal{R} \mathcal{D} \)) that contains \( X \). That is, they apply to any monomial term that corresponds to a multiple of \( LT(P) \) and they replace their redex by a corresponding multiple of \( \psi(-\text{RED}(P)) \). Therefore we see that term reductions w.r.t. e-rules correspond to polynomial reductions w.r.t. the associated polynomials. Yet the result of an e-rule reduction does not correspond to a polynomial in distributive normal form. The reduced term translates to a polynomial expression that can be interpreted as an intermediate result of computing a polynomial reduction. A subsequent normalisation by \( \mathcal{N} \mathcal{X} \) may be needed to complete the simulation of a polynomial reduction.

Example 4. Consider the polynomial \( p = x^2 + 2xy^2 + 2y \) and the polynomial \( q = xy - y^2 + 1 \). With \( x > y \), \( q \) can be interpreted as the polynomial rule \( xy \rightarrow y^2 - 1 \) and
\[
x^2 + 2xy^2 + 2y \rightarrow (q) x^2 + 2y^3.
\]
Translating \( p \) to a ground term yields
\[
\psi(p) = M(1, XX) \oplus M(1 + 1, YXY) \oplus M(1 + 1, Y).
\]
The e-rules associated with \( q \) are
\[
E(q) = \{ M(a, XY) \rightarrow M(a, YY) \oplus M(-a, I), \quad M(a, XYx) \rightarrow M(a, YYx) \oplus M(-a, x) \}.
\]
Then reducing \( \psi(p) \) by the second rule of \( E(q) \) and normalising by \( \mathcal{N} \mathcal{X} \), we obtain a simulation of the polynomial reduction by \( q \):
\[
\begin{align*}
\psi(p) \rightarrow_{E(q)} M(1, XX) \oplus M(1 + 1, Y) \oplus M(1 + 1, YYY) \oplus M(-1 + -1, Y) \\
\rightarrow_{\mathcal{N} \mathcal{X}}^* M(1, XX) \oplus M(1 + 1, YYY).
\end{align*}
\]
Theorem 3. Let $P$ be a monic polynomial and $P'$, $P''$ be polynomials such that $P' \rightarrow_{(P)} P''$. Let $E$ be the extension rules associated with $P$. Then there is a term $t$ such that $\psi(P') \rightarrow_{E} t$ and the $\mathcal{KX}$-normal form of $t$ is $\psi(P'')$.

As we will see below, simulating Buchberger's algorithm, in the way explained above, always results in a canonical term rewriting system consisting of $\mathcal{KX}$ and the set of extension rules associated with the elements of the Gröbner base. Therefore the Gröbner base can be derived from the result (cf. Example 3).

4.2. The termination of type-I rules

In the last subsection, we have shown that the e-rules associated with a polynomial $P$ together with $\mathcal{KX}$ simulate polynomial reductions by $P$. We must now show that term rewriting systems consisting only of $\mathcal{KX}$ and e-rules are terminating.

Termination of term rewriting systems is typically proved by designing a term ordering that includes the rewrite relation. The difficulty is now in finding a term ordering that is compatible with $\mathcal{KX}$ and that on ground terms in $\mathcal{KX}$-normal form simulates an admissible polynomial ordering for Buchberger's algorithm. In addition we require that e-rules can be proven to be terminating by such a term ordering.

Here we look for a polynomial interpretation ordering that refines $\Phi$ from Definition 2 and that orients g- and e-rules according to the lexicographic ordering used for Buchberger's algorithm. The lexicographic ordering for monomials is a multiset extension [23] of a total ordering on the indeterminates (all occurrences of indeterminates in a product are counted). The lexicographic ordering for polynomials $\succ$ is a (multi)set extension of the lexicographic ordering for monomials. Here all monomials in the distributive normal form of a polynomial are counted.

We will not be able to present such a polynomial interpretation ordering a priori, but we will show that instances of the polynomial interpretation function $\Phi$ exist which simulate the lexicographical ordering on polynomials and orient all e-rules from left to right. In particular, the following two inequalities must hold:

**type-e1 rules:** $(\Phi(a) + K_2)\Phi(\bar{X}) \succ_p \sum_k((\Phi(k)a) + K_2)\Phi(\bar{X}_i) + 5) - 5$

**type-e2 rules:** $(\Phi(a) + K_2)\Phi(\bar{X})\Phi(x) \succ_p \sum_k((\Phi(k)a) + K_2)\Phi(\bar{X}_i)\Phi(x) + 5) - 5$

For $K_2 \geq 1$ the first inequality implies the second and the inequality for the corresponding g-rule. For all ground terms $t$, $\Phi(t) \geq 2$. To determine the interpretations of the $\text{Ind}$-constants, let $k$ be the maximal coefficient, $n$ be the maximal number of monomials and $r$ be the maximal degree of a power product occurring in a left- or right-hand side of a rule or an equation. These numbers are not known a priori, but at any given moment during the completion process such numbers exist and can be determined. Now for every constant $X \in \text{Ind} \setminus \{I\}$ let

$$\Phi(X) > n(2k\Phi(\bar{X}')) + 1),$$
where \( \vec{X} \) is the greatest power product with degree less than or equal to \( r \) such that \( \psi^{-1}(X) \succ \psi^{-1}(\vec{X}) \). In addition, the interpretation of the smallest indeterminate must be greater than \( n \). Then for given \( n, r \) and \( K_2 \) we have found an instance \( \phi \) of \( \Phi \) that induces a terminating term ordering compatible with \( \mathcal{A} \mathcal{X} \) and all e-rules and hence simulates \( \succ \). Let us call this term ordering \( \succ_{\phi} \). Thus we get

**Theorem 4.** The term rewriting system \( \mathcal{A} \mathcal{X} \cup E \) where \( E \) consists of e-rules is terminating modulo \( \mathcal{A} \mathcal{C} \).

For different coefficient domains \( \mathcal{K} \), the polynomial interpretation function \( \phi \) must be further refined. As we will see in the following sections this will pose no problems. The existence of an appropriate \( \phi \) can be formulated as a quantifier elimination problem [18, 58] and is therefore decidable. In general, the problem is too complex to be solved mechanically. But, since the existence of appropriate orderings is assured, we can orient \( g \)- and e-rules according to the lexicographic ordering of the associated polynomials.

### 4.3. Superpositions

In order to simulate the computation of the s-polynomial of two polynomials \( P_1 \) and \( P_2 \), we must compute the critical pairs of their associated e-rules. Two e-rules can only superpose at the top positions of their left-hand sides. Thus both the \( \text{Coeff} \)-terms and the \( \text{Ind} \)-terms of the left-hand side monomials must unify modulo \( \mathcal{A} \mathcal{C} \). The coefficient terms unify trivially because they consist of distinct variables. Unifying two ground terms of type \( \text{Ind} \) with variable extensions results in computing the least common multiple of both terms. If one of the power product terms is ground this term must be the most general common instance if the terms unify at all. It is actually sufficient to compute only critical pairs between two e2-rules because these pairs subsume those involving e1-rules. Let us consider the two e2-rules

\[
\alpha : M(a, \vec{X}.x) \rightarrow \bigoplus_i M(k_i a, \vec{X}_i.x) \quad \text{and} \quad \beta : M(b, \vec{Y}.y) \rightarrow \bigoplus_i M(k_i b, \vec{Y}_i.y),
\]

where \( \vec{X} = \vec{X'}, \vec{Y} = \vec{Y'}, \vec{Z} = \vec{X'}, \vec{Y'}, \vec{Z'} \) for disjoint \( \vec{X}' \) and \( \vec{Y'} \). Thus the most general common instances of \( \vec{X}x \) and \( \vec{Y}x \) modulo \( \mathcal{A} \mathcal{C} \) are \( \{\vec{Z}, \vec{Z}x\} \). In the cases where \( \vec{Y} = \vec{Z'} \) or \( \vec{X} = \vec{Z'} \) we have \( \text{mgci}_{\mathcal{A} \mathcal{C}}(\vec{X}, \vec{Y}, x) = \{\vec{Z}\} \) or \( \text{mgci}_{\mathcal{A} \mathcal{C}}(\vec{X}, \vec{Y}) = \{\vec{Z}\} \). If the \( \text{Ind} \)-terms in the left-hand sides of the rules unify we get the critical peaks:

\[
\bigoplus_i M(k_i a, \vec{X}_i[z]) \quad \bigoplus_i M(k_i b, \vec{Y}_i[z]).
\]

These critical pairs correspond exactly to the critical pairs of the polynomial rules associated with \( \alpha \) and \( \beta \). The AC-mgci of the \( \text{Ind} \)-terms corresponds to the least common multiples of the power products in the leading terms. The fact that AC-unification in finitely presented commutative monoids reduces to least common multiple computations (for ground- and extension rules) has already been pointed out in [3].
Note that all critical pairs of two e-rules have a particular format. Both terms of the pairs contain a single coefficient variable and possibly a single indeterminate variable but no further variables. The terms of the critical pairs thus correspond to two polynomials that are 'multiplied by' the variable(s). More precisely substituting a 1 for the coefficient variable and an I for the indeterminate variable yields ground terms that correspond to the polynomials whose difference makes up the s-polynomial.

Example 5. Consider the two polynomials $xy^2 + 2y^2 + z$ and $xyz + 2yz - y - 3$. Given an admissible ordering induced by $x \succ y \succ z$, we find the superposition

$$-2y^2z - z^2 \leftarrow xy^2z \rightarrow -2y^2z + y^2 + 3y$$

and thus the s-polynomial $-y^2 - 3y - z^2$. The type-2 extension rules associated with the two polynomials are

$$M(a,XY.Y.x) \rightarrow M(-a + -a, Y.Y.x) \oplus M(-a, Z.x),$$

$$M(a,XY.Z.x) \rightarrow M(-a + -a, Y.Z.x) \oplus M(a, Y.x) \oplus M(a + a + a, x).$$

Their critical peaks are

$$M(-a + -a, Y.Y.Z[.x]) \leftarrow M(a,XY.YZ[.x]) \rightarrow M(-a + -a, YYZ[.x])$$

$$\oplus M(-a, Z.Z[.x]) \oplus M(a, Y.[x])$$

$$\oplus M(a + a + a, Y.[x]).$$

This corresponds exactly to the rewrite ambiguity for polynomials.

In case $\vec{x}$ and $\vec{y}$ do not overlap at all, it is easy to verify that all corresponding critical pairs are confluent. This corresponds to the fact that the s-polynomial of $P_1$ and $P_2$ reduces to 0 if gcd(LT($P_1$), LT($P_2$)) = 1.

4.4. The selection of the leading term

The orientation of an equation $s \leftrightarrow t$ is a symmetrical process for term rewriting systems: Either $s$ becomes the left-hand side of the rule and $t$ its right-hand side or vice versa. For Buchberger's algorithm, orientation of polynomials (or equivalently of polynomial equations) is a more complex process. Consider the equation $P = Q$. First the maximal monomial (w.r.t. an admissible ordering) of the distributive normal form of $P - Q$ must be determined. This leads to the equation $LT(P - Q) = -RED(P - Q)$ which can be oriented from left to right according to the ordering. Finally, the polynomial rule must be divided by its leading coefficient unless it is 1.

In this subsection, we disregard the leading coefficient and tackle only the problem of, given a (simulated) polynomial equation, how do we obtain a rewrite rule corresponding to $LT(P - Q) \rightarrow -RED(P - Q)$. In the domain of polynomials the transformation
of the equation is possible by adding appropriate polynomials to both sides of the equation because polynomials are an Abelian group w. r. t. addition. The selection of the leading term as the left-hand side of a rule can be simulated in the term rewriting environment as a symmetrisation process in the Abelian group of polynomial addition with monomials as the generators of the group (cf. [47]). To obtain a single generator as a left-hand side of a symmetrised rule, the term ordering must compare (sums of) generators according to a multiset ordering induced by a total ordering on the generators. This condition is surely met for \( \succ_{\phi} \) proposed in Section 4.2. Critical pair transformations play a central role to providing short proofs for and small costs of this process.

Let \((\bigoplus_i M_i, \bigoplus_i N_i)\) be a critical pair reduced to \(\mathcal{K}\)-normal form. W. l. o. g., let \(M_1\) be the maximal monomial in this pair. Then a superposition with \(\mathcal{K}.11x\) (cf. Specification 2) results in

\[
M_1 \leftarrow M_1 \oplus \bigoplus_{i > 1} M_i \oplus ((\bigoplus_{i > 1} M_i) \odot M(-1,I)) \oplus \bigoplus_i N_i \oplus ((\bigoplus_{i > 1} M_i) \odot M(-1,I))
\]

which can be oriented from left to right. This rule makes the original critical pair confluent. Using Theorem 1, the old critical pair can be 'transformed' into a new one.

If both sides of the original critical pair contain the maximal power product, then the monomials containing these power products must first be 'put on one side' by an additional transformation involving \(\mathcal{K}.11x\). Let \(P, Q\) and \(R\) be any terms of sort \(\text{Poly}\) and \((P \oplus Q, P \oplus R)\) a critical pair such that \(Q \succ R\). Then superposing \(P \oplus Q \rightarrow P \oplus R\) and \(\mathcal{K}.11x\) results in the following critical peak:

\[
Q \leftarrow P \oplus (P \odot M(-1,I)) \oplus Q \leftarrow P \oplus (P \odot M(-1,I)) \oplus R
\]

which reduces to the critical pair \((Q,R)\). \((P \oplus Q, P \oplus R)\) is confluent using \(Q \rightarrow R\) (and its AC-extension rule). Therefore it can be replaced by \((Q,R)\).

**Example 6.** Let us select the leading term of the critical pair from Example 5:

\[
(M(-a + -a, Y.Y.Z[x]) \oplus M(-a, Z.Z[x]),
M(-a + -a, Y.Y.Z[x]) \oplus M(a, Y.Y[x]) \oplus M(a + a + a, Y[x]).
\]

Orienting this pair from the right to the left and superposing the so obtained rule with \(\mathcal{K}.11x\) yields the critical mountain

\[
M(-a + -a, Y.Y.Z[x]) \oplus (M(-1,I) \odot M(-a + -a, Y.Y.Z[x])) \oplus M(a, Y.Y[x]) \oplus M(a + a + a, Y[x]).
\]
The corresponding critical pair can again be oriented from the right to the left and the resulting rule can also be superposed with $\mathcal{L}$.11x:

$$
M(a, Y.Y[x]) \\
\oplus M(a + a + a, Y[x]) \\
\oplus (M(-1, I) \odot \\
M(a, Y.Y[x]) < M(a + a + a, Y[x])) \\
\oplus (M(-1, I) \odot \\
M(a + a + a, Y[x])) \\
\oplus M(-a, Z.Z[x])
$$

A further orientation step and normalisation by $\mathcal{L} \mathcal{X}$ yield the rule

$$
M(a, Y.Y[x]) \rightarrow M(-a + -a + -a, Y[x]) \oplus M(-a, Z.Z[x]).
$$

Using this rule and $\mathcal{L} \mathcal{X}$ the original critical pair reduces to the common normal form $M(-a + -a, Y.Y[Z[x]]) \oplus M(-a, Z.Z[x])$ and thus it may be transformed into the new rule.

4.5. The division problem

Let us now reconsider our original goal: the simulation of Buchberger's algorithm by AC-completion of a set of ground equations (of sort $Pdy$) and $\mathcal{N}X$. So far we have shown how polynomial reductions can be simulated by e-rules and $\mathcal{N}X$ and that critical pairing of e-rules corresponds to computing s-polynomials. Then we have shown how simulated polynomial equations can be transformed to equations (or rules) where the leading term is isolated on one side. The last operation can be used to transform $\mathcal{N}X$ normalised ground equations (of sort $Pdy$) into g-rules. To show that the simulation is correct we must still show that

1. e-rules can be derived from corresponding ground equations;
2. when completing ground equations (of sort $Pdy$) together with $\mathcal{N}X$ only e-rules are persistent i.e., all other equations and all rules neither belonging to $\mathcal{N}X$ nor being an e-rule will eventually be deleted and
3. the completion of $\mathcal{N}X$ and a single ground equation of sort $Pdy$ terminates given a fair completion strategy.

The solution of the above three points largely depends on the solution of the division problem: How can we simulate the division of a polynomial (rule/equation) by its leading coefficient?

Exact division means that for some quotient $c_1/c_2$, the equation $c_2 \cdot x = c_1$ must have a solution. Therefore if $c_1$ and $c_2$ are represented by the ground terms $t_1$ and $t_2$, then a ground term $t'$ must exist such that $t_2 \cdot t' \leftrightarrow_{\mathcal{R}}^* t_1$ for a term rewriting system $\mathcal{R}$ specifying a coefficient domain in which division by $c_2$ is possible, i.e., $t_2 \cdot x$ and $t_1$ must be unifiable modulo $\mathcal{R}$. More precisely, since $t_1$ is ground, $t_2 \cdot x$ must match $t_1$ modulo $\mathcal{R}$. If $\mathcal{R}$ is canonical then unifiers modulo $\mathcal{R}$ can be computed by a narrowing procedure [24, 30, 40]. The key inference step of narrowing is an operation very similar to critical pair computation. It can in fact be simulated by completion [21, 48].
Now our idea is to specify the coefficient domain by a canonical term rewriting system in such a way that the reciprocals of all ground terms (of sort \texttt{Coef}) exist. We can even weaken this requirement to demand only the existence of the reciprocals of those ground terms that occur as leading coefficients. Then the narrowing procedure computing the divisions is a part of the term completion procedure simulating Buchberger's algorithm, i.e., the critical pairs between \texttt{g}- or \texttt{e}-rules and \texttt{H} simulate the division operation whereas other critical pairs contribute to simulating the raw structure of Buchberger's algorithm.

The solution of the division problem clearly depends on the coefficient domain and therefore on its rewrite specification. In the next two sections, we will describe these solutions for the case where the coefficient domain is a finite field (Section 5) or the rational numbers (Section 6).

5. Completion of polynomials over finite fields

In this section, we describe the aspects of the simulation of Buchberger's algorithm that are specific to polynomial completion over finite fields. We will first present a refinement \texttt{ZpX} of \texttt{X} that specifies multivariate polynomials over prime fields. Then we show that the completion of a ground equation of sort \texttt{Poly} and \texttt{ZpX} always yields a canonical term rewriting system consisting of \texttt{ZpX} and the \texttt{e}-rules associated with the ground equation. In Subsection 5.3, we extend our results to arbitrary finite fields.

The initial algebras (ground term models) of all specifications \texttt{H} presented in this section are isomorphic to the polynomials over the specified finite fields. More precisely, the ground term model of \texttt{H} restricted to the coefficient sort is isomorphic to a finite field \texttt{GF(q)}. Therefore \( t_2 \cdot x \) and \( t_1 \) are unifiable modulo \texttt{H} for all non-zero ground terms \( t_1 \) and \( t_2 \). Hence division is effective in the term specification of \texttt{GF(q)} and the division problem is solved.

5.1. Polynomials over prime fields

Let us now look for a rewrite specification for polynomials over integers modulo some prime \( p \). We begin with specifications for residue rings of the integers modulo \( p \) (\( \mathbb{Z}/(p\mathbb{Z}) \)) where \( p \) is prime. Therefore we complete the equational description

\[
\mathcal{E}_X = \mathcal{E}_X \cup \{1 + \cdots + 1 \leftarrow 0\}_{p \text{ times}}
\]

of commutative rings with ones of characteristic \( p \) and we get the following canonical term rewriting system.
Specification 3. $\mathcal{Y}_p =$

\begin{align*}
1 : a + 0 & \rightarrow a, & 5 : -a & \rightarrow (p - 1)a, \\
2 : a \cdot 0 & \rightarrow 0, & 6 : pa & \rightarrow 0, \\
3 : a \cdot 1 & \rightarrow a, & 6x : b + pa & \rightarrow b, \\
4 : a \cdot (b + c) & \rightarrow (a \cdot b) + (a \cdot c). & & \end{align*}

The initial model of $\mathcal{Y}_p$ is isomorphic to the ground term model of $\mathcal{Y}_p$ represented by the set of $\mathcal{Y}_p$-ground normal forms \{0, 1, 1 + 1, \ldots, (p - 1)1\} and thus it is isomorphic to the field $\mathbb{Z}/(p\mathbb{Z})$. Although there is no equation defining a multiplicative inverse for every term, like

\((*) : a \cdot (a)^{-1} \rightarrow 1,\)

we can prove the following theorem which can be considered as an inductive consequence of $\mathcal{Y}_p$:

**Theorem 5.** Let $p$ be a prime number. Then for every ground term $t$ that does not reduce to 0 by $\mathcal{Y}_p$ there is a ground term $t'$ such that $t \cdot t' \rightarrow_{\mathcal{Y}_p} 1$.

**Proof.** It is sufficient to prove the above claim for terms in ground normal form. Let $\psi$ be a bijective mapping from $\mathbb{Z}/(p\mathbb{Z})$ to the set of $\mathcal{Y}_p$-ground normal forms. Then $t \cdot \psi(\psi^{-1}(t)^{-1}) =_{\mathcal{Y}_p} 1$. For all terms $t$ in ground normal form other than 0, $t \succ 1$ (take e.g., the term ordering $\succ_\phi$ proposed later in this section). By compatibility of $\succ$, this holds also for a product of two ground normal forms. Thus we have $t \cdot \psi(\psi^{-1}(t)^{-1}) \rightarrow_{\mathcal{Y}_p} 1$. \(\Box\)

**Example 7.** Let $p = 5$ and $t = ((1 + 1) \cdot (1 + 1 + 1)) + 1 + 1$. Now $t \rightarrow_{\mathcal{Y}_p} 1 + 1 + 1$ and $\psi^{-1}(t) = 3$. Therefore $t' = \psi(2) = 1 + 1$ and $t \cdot t' \rightarrow_{\mathcal{Y}_p} 1$.

We can extend $\mathcal{Y}_p$ to a canonical specification of multivariate polynomials over $\mathbb{Z}/(p\mathbb{Z})$ by completing $\mathcal{Y}_p \cup \mathcal{X}$:

**Specification 4.** $\mathcal{Y}_p \mathcal{X} = \mathcal{Y}_p \cup \mathcal{X}$ where

\begin{align*}
1 : x.I & \rightarrow x, \\
2 : M(0,x) & \rightarrow \Omega, \\
3 : M(a,x) \oplus M(b,x) & \rightarrow M(a + b,x), \\
3x : f \oplus M(a,x) \oplus M(b,x) & \rightarrow f \oplus M(a + b,x), \\
4 : M(a,x) \odot M(b,y) & \rightarrow M(a \cdot b,x,y), \\
4x : f \odot M(a,x) \odot M(b,y) & \rightarrow f \odot M(a \cdot b,x,y), \\
5 : f \oplus \Omega & \rightarrow f, \\
6 : f \odot \Omega & \rightarrow \Omega, \\
7 : f \odot M(1,I) & \rightarrow f, \\
8 : f \odot (g \oplus h) & \rightarrow (f \odot g) \oplus (f \odot h), \\
9 : (f \odot M(a,x)) \oplus (f \odot M(b,x)) & \rightarrow f \odot M(a + b,x), & \end{align*}
The confluence of XPX was again proved using the ReDuX system. To prove the termination of XPX, we take the polynomial interpretation function $\Phi$ of Section 3 with one small change:

$$\Phi(-a) = K_1(\Phi(a) + 5), \text{ where } K_1 \geq p - 1.$$ 

**Theorem 6.** For every prime number $p$, $\mathcal{X}_p \mathcal{X}$ is canonical modulo AC.

**Remark 1.** The new polynomial interpretation function $\Phi$ can also be specialised to a function which orients g- and e-rules according to the lexicographic ordering.

**Remark 2.** Note that the rules $\mathcal{X}.11$ and $\mathcal{X}.11x$ have been deleted from $\mathcal{X}_p$. The additive inverse of a polynomial is now described by a $(p - 1)$-fold addition of that polynomial as opposed to a multiplication with $M(-1, I)$. Therefore, the rules $\mathcal{X}.p.11$ and $\mathcal{X}.p.11x$ take the role of $\mathcal{X}.11$ and $\mathcal{X}.11x$ for the selection of the leading term (cf. Section 4.4).

**5.2. The symmetrisation of g-rules**

The symmetrisation of a g-rule $\alpha$ w.r.t. $\mathcal{X}_p \mathcal{X}$ coincides with the completion of $\mathcal{X}_p \mathcal{X} \cup \{\alpha\}$ because all critical pairs between the type-I g- and e-rules associated with a single polynomial are confluent (only g- and e1-rules superpose). Our goal is to show that the result of this completion is a canonical term rewriting system $\mathcal{X}_p \mathcal{X} \cup E$ where $E$ is the set of e-rules associated with $\alpha$. For the moment we assume that the left-hand side of $\alpha$ contains a non-trivial power product. The following lemma states the confluence of the resulting system.

**Lemma 7.** Let $E$ be the set of type-e1 rules associated with a g-rule $l \rightarrow r$ where $l = M(1, \bar{X})$, and $\bar{X} \neq I$. Then all critical pairs among rules in $\mathcal{X}_p \mathcal{X} \setminus E$ are confluent modulo AC.

**Proof.** Only rules $\{\mathcal{X}.p.1, \mathcal{X}.p.3, \mathcal{X}.p.4, \mathcal{X}.p.9, \mathcal{X}.p.9x\}$ superpose with e-rules. Then the proof is by straightforward analysis of the corresponding critical pairs. Q.E.D.

Next we must explain that the e-rules will actually be deduced during the completion of $\mathcal{X}_p \mathcal{X}$ and $\alpha$. Let

$$\alpha : M(k, \bar{X}) \rightarrow \bigoplus_i M(k_i, \bar{X}_i)$$
be in \( \mathcal{X}_p \mathcal{X} \)-normal form. Superposing \( \alpha \) with the rule \( \mathcal{X}_p.4 \), we get the critical peak

\[
M(k \cdot a, \overline{x}) \leftarrow M(k, \overline{x}) \circ M(a, x) \leftarrow (\bigoplus_i M(k_i, \overline{x}_i)) \circ M(a, x).
\]

This critical pair can be normalised and oriented from left to right. If \( k = 1 \), the resulting rule

\[
\alpha_2 : M(a, \overline{x}) \rightarrow \bigoplus_i M(k_i a, \overline{x}_i)
\]

is of type e2. Superposing \( \alpha_2 \) with \( \mathcal{X}_p.1 \) then results in a rule \( \alpha_1 \) of type e1. If \( k > 1 \) we get a rule

\[
\alpha_3 : M(k a, \overline{x}) \rightarrow \bigoplus_i M(k_i a, \overline{x}_i).
\]

Since \( \alpha \) is irreducible, \( k < p \) and \( k a \) is irreducible. Therefore \( k a \) AC-unifies with the left-hand side of \( \mathcal{X}_p.6x \). Since \( p \) is prime, for every \( k < p \) there is a \( k' \) such that \( k k' = 1 \mod p \). Then there must exist a \( p' \) such that \( k k' = pp' + 1 \). Now the following critical peak can be created from rule \( \mathcal{X}_p.6x \) and rule \( \alpha_3 \):

\[
M(a, \overline{x}) \leftarrow M(kk' a, \overline{x}) \circ \bigoplus_i M(k_i k' a, \overline{x}_i).
\]

We orient the critical pair from left to right,

\[
\alpha_2 : M(a, \overline{x}) \rightarrow \bigoplus_i M(k_i k' a, \overline{x}_i),
\]

and we show that \( \alpha_3 \) is confluent w. r. t. \( \mathcal{X}_p \mathcal{X} \cup \{ \alpha_2 \} \):

\[
M(k a, \overline{x}) \rightarrow_{\{ \alpha_1 \}} \bigoplus_i M(k_i k' a, \overline{x}_i) \rightarrow^{* \{ \mathcal{X}_p \mathcal{X} \cup 6x \} } \bigoplus_i M(k_i a, \overline{x}_i).
\]

Note that this critical pair transformation simulates a division of an equation by \( k \) and thus enables us to make the rules 'monic'. \( \alpha_2 \) is again of type e2. The rest carries over from the case \( k = 1 \). Note that in both cases \( \alpha_1 \) reduces \( \alpha \) making it confluent in \( \mathcal{X}_p \mathcal{X} \cup \{ \alpha_1 \} \) such that by Theorem 1 no other critical pairs of \( \alpha \) need to be considered. Hence we have

**Lemma 8.** Let \( l \rightarrow r \) be a g-rule with \( l = M(k, \overline{x}) \), and \( \overline{x} \neq I \). Then \( (\emptyset, \mathcal{X}_p \mathcal{X} \cup \{ l \rightarrow r \}) \vdash^* (\emptyset, \mathcal{X}_p \mathcal{X} \cup E) \), where \( E \) consists of all e-rules associated with \( l \rightarrow r \) and \( \mathcal{X}_p \mathcal{X} \cup E \) is confluent.

If the left-hand side of a g-rule is \( M(k, l) \), then according to the term ordering described in Section 4.2 its right-hand side must be \( \Omega \). Thus we get a trivial term
rewriting system where all polynomials reduce to $\Omega$ because superposing $M(a,x) \to \Omega$ with rule $\mathcal{X}_p.7$ results in the rule $f \to \Omega$.

**Theorem 9.** Let $l \to r$ be a g-rule. Then
- $(\emptyset; \mathcal{X}_p \cup \{l \to r\}) \vdash^* (\emptyset; \mathcal{X}_p \cup E);
- $E$ consists either of all e-rules associated with $l \to r$ or of the rule $f \to \Omega$;
- $\mathcal{X}_p \cup E$ is confluent and terminating modulo AC.

**Proof.** By Theorem 4, Lemma 8 and the preceding argument. □

5.3. Arbitrary finite fields

So far we have considered polynomials whose coefficient domains are prime fields. Let us now extend our results to polynomials over arbitrary finite fields. The class of all finite fields has been completely classified, the main results being:

- Finite fields of the same order are identical up to isomorphisms.
- If $\mathbb{K}$ is a finite field of order $q$ then $q = p^r$ for some prime number $p$. $\mathbb{K}$ is then called the Galois field of order $q$ denoted by $\text{GF}(q)$.
- $\text{GF}(q)$ is isomorphic to some field extension $\mathbb{Z}/(p\mathbb{Z})(x)$ of $\mathbb{Z}/(p\mathbb{Z})$ where $x$ is a root of an irreducible minimal polynomial $m_\alpha(x)$ of degree $r$. Thus $\text{GF}(q) \cong \mathbb{Z}/(p\mathbb{Z})[x]/(m_\alpha(x))$ holds.

Using the last relation, we can give an equational specification whose ground term model is isomorphic to $\text{GF}(q)$. From now on let $q = p^r$. For $r > 1$, we must add a new constant $A$ of sort $\text{Coef}$ to the signature $\mathcal{F}$. Let $\psi$ be a mapping from $\mathbb{Z}/(p\mathbb{Z})[x]$ to $T(\mathcal{F})$ which maps $x$ to $A$. Then the completion of $\mathcal{E}_{\mathcal{F}_q} = \mathcal{X}_p \cup \{\psi(LT(m_\alpha(x))) \cdot \psi(-\text{RED}(m_\alpha(x)))\}$ results in

$$\mathcal{E}_{\mathcal{F}_q} = \mathcal{X}_p \cup \{A^r \to \psi(-\text{RED}(m_\alpha(x))), \ a \cdot A^r \to a \cdot \psi(-\text{RED}(m_\alpha(x)))/\mathcal{X}_p\}.$$ 

To prove the termination of $\mathcal{E}_{\mathcal{F}_q}$, it is sufficient to choose $\Phi(A)$ large enough. Further, it is easy to see that the last two rules do not superpose with any rule in $\mathcal{X}_p$. Thus $\mathcal{E}_{\mathcal{F}_q}$ is confluent. For $q = p^1$ we set $\mathcal{E}_{\mathcal{F}_q} = \mathcal{X}_p$.

**Theorem 10.** The set $T(\mathcal{F}) \upharpoonright \mathcal{F}_q$ of ground normal forms of $\mathcal{F}_{\mathcal{F}_q}$ is isomorphic to $\text{GF}(q)$.

**Proof.** $T(\mathcal{F}) \upharpoonright \mathcal{F}_q$ contains terms of the form 0, $t'$ and $\sum_i t_i$, where the $t'$ and $t_i$ are either 1 or products of $A$ and no term in the sum may occur more than $p - 1$ times. Thus $|T(\mathcal{F}) \upharpoonright \mathcal{F}_q| = p^r = q$. For each ground term $t \neq 0$ in $\mathcal{F}_{\mathcal{F}_q}$-normal form, $\psi^{-1}(t)(y)^{-2} = \psi(t)^{-1}$. Then $t^* = t^{(q-2)} \upharpoonright \mathcal{F}_{\mathcal{F}_q}$ is the multiplicative inverse of $t$ and $t \cdot t^* \to^* 1$. Thus $T(\mathcal{F}) \upharpoonright \mathcal{F}_q$ is isomorphic to $\text{GF}(q)$. □

---

5 For an introduction to the theory of finite fields needed in this section see e.g., [45, Chap. 6].
Example 8. Let \( q = 2^4 = 16 \). Then \( m_q(x) = x^4 + x + 1 \) is an irreducible minimal polynomial of degree 4 and \( GF(16) \cong \mathbb{Z}/(2\mathbb{Z})[x]/(m_q(x)) \cong \mathbb{Z}/(2\mathbb{Z})/(x) \).

\[
\begin{align*}
\mathcal{F}_{16} &= \{ a + 0 \rightarrow a, \quad -a \rightarrow a, \\
&\quad a \cdot 0 \rightarrow 0, \quad a + a \rightarrow 0, \\
&\quad a \cdot 1 \rightarrow a, \quad b + a + a \rightarrow b, \\
&\quad a \cdot (b + c) \rightarrow (a \cdot b) + (a \cdot c), \quad A \cdot A \cdot A \cdot A \rightarrow A + 1,
&\quad a \cdot A \cdot A \cdot A \rightarrow a \cdot A + a \}.
\end{align*}
\]

To find the inverse of \( A \cdot A \) we must solve the equation \( A \cdot A \cdot a = \_1 \). From the theory of finite fields it is known that \( (x^2)^{-1} = x^{13} \). Translating this to terms we get

\[
A^{13} \rightarrow_{\mathcal{F}_{16}}^{*} A^3 + A^2 + 1
\]

and

\[
A \cdot A \cdot (A^3 + A^2 + 1) \rightarrow_{\mathcal{F}_{16}}^{*} 1.
\]

Thus we have found a \( \mathcal{F}_{16} \)-ground normal form denoting the multiplicative inverse of \( A \cdot A \) solving the above unification problem.

If \( q = p^1 \) let \( \mathcal{F}_q \mathcal{X} = \mathcal{X}_p \mathcal{X} \). Otherwise the completion of \( \mathcal{F}_q \cup \mathcal{X}_p \) results in the canonical term rewriting system

\[
\mathcal{F}_q \mathcal{X} = \mathcal{X}_p \mathcal{X} \cup \{ A' \rightarrow \psi(-RED(m_s(\alpha))), \quad a \cdot A' \rightarrow (a \cdot \psi(-RED(m_s(\alpha)))) \downarrow_{\mathcal{X}_p} \}.
\]

The termination proof for \( \mathcal{F}_q \mathcal{X} \) carries over from \( \mathcal{F}_q \) and \( \mathcal{X}_p \mathcal{X} \). Since no new critical pairs can be created \( \mathcal{F}_q \mathcal{X} \) is confluent too. Computing Gröbner bascs in \( GF(q)[x_1, \ldots, x_n] \) with \( \mathcal{F}_q \mathcal{X} \) differs from their computation as described in the previous sections only in that g-rules of the form

\[
M(A', \vec{X}) \rightarrow \bigoplus_i M(A_i, \vec{X}_i)
\]

can occur where the \( A' \) and the \( A_i \) are terms of sort \( \text{Coef} \) in ground normal form which may contain the new constant \( A \). By completeness of the completion procedure, the equation

\[
M(1, \vec{X}) \leftrightarrow (\bigoplus_i M(A_i, \psi^{-1}(A')^{-1}, \vec{X}_i)) \downarrow_{\mathcal{F}_q \mathcal{X}}
\]

must be derivable. Then e-rules can be derived as shown in Section 5.2. These e-rules reduce the original g-rules making them confluent such that the critical pair transformation criterion applies. By the same reasoning new types of e-rules can be derived from g-rules as intermediate results. These new e-rules are g-rules ‘multiplied’ by a coefficient variable \( a \). All other aspects of the completion with \( \mathcal{F}_q \mathcal{X} \) carry over from completion using \( \mathcal{X}_p \mathcal{X} \).
5.4. Simulating Buchberger's algorithm for finite fields

In the preceding sections, we have shown that the completion of polynomials in the polynomial ring \( \mathbb{F}(q)[x_1, \ldots, x_r] \) can be simulated using term completion modulo associativity and commutativity. As input to the AC-completion algorithm we need

1. \( \mathbb{F}_q \mathcal{X} \),
2. a set \( P \) of polynomials encoded as set \( \mathcal{P} \) of ground equations and
3. the term ordering \( \succsim \) described in Section 4.2.

As a result of the simulation we get

\[
\mathbb{F}_q \mathcal{X} \cup \mathcal{I}
\]

where \( \mathcal{I} \) is the set of extension rules associated with the polynomials in the Gröbner basis of \( P \).

The termination of the term completion procedure is ensured by the same arguments that apply for the termination of Buchberger’s algorithm. It is well known that for quite a few commutative algebraic structures (like Abelian semi-groups, -monoids, -groups, -rings, -modules, -algebras over rings) there exists an AC-canonical term rewriting system for each finitely generated and finitely presented such structure [3,43]. Similar results have been reported in [49] and [52] for equational theories presented only by a set of ground equations including associative and commutative operators and by [50] for idempotent Abelian semi-groups and monoids.

We can now translate all features of Buchberger’s algorithm for \( \mathbb{F}(q)[x_1, \ldots, x_r] \) to the language of term rewriting:

- Multivariate polynomials in disjunctive normal form are isomorphic to the ground terms of sort \( \text{Poly} \) in \( \mathbb{F}_q \mathcal{X} \)-normal form.
- Lexicographical orderings over polynomials can be simulated by a polynomial interpretation ordering for terms.
- The reduction relation associated with a polynomial maps to the rewrite relation described by its associated e-rules.
- Critical pairs between polynomials correspond to critical pairs between rules of type e2.
- The orientation procedure of Buchberger’s algorithm is simulated by a symmetrisation process which computes all extension rules associated with an equation.

The algorithm \textit{FFPCOMPLETE} in Fig. 5 presents a term completion procedure using a strategy that models the algorithm \textit{BAK} of Fig. 3. The function \textit{Complete} is an arbitrary implementation of a term completion procedure modulo AC. Hence we get our final result.

**Theorem 11.** For every finite field \( \mathbb{F}(q) \), the computation of Gröbner bases in the polynomial ring \( \mathbb{F}(q)[x_1, \ldots, x_n] \) by Buchberger’s algorithm using the lexicographic ordering can be simulated using term completion modulo associativity and commutativity.
\( R^* \leftarrow \text{FFPCOMPLETE}(\mathcal{F}_q \mathcal{X}, \mathcal{P}, \succ) \)

[Finite field polynomial AC-term completion with ‘Buchberger strategy’.
\( \mathcal{F}_q \mathcal{X} \) is a canonical term rewriting system whose initial model is \( \text{GF}(q)[x_1, \ldots, x_r] \).
\( \mathcal{P} \) is a set of ground equations of sort \( \text{Poly} \) and \( \succ \) is a terminating term ordering that includes the lexicographical ordering on multivariate polynomials described by ground terms. Then \( R^* \) is the canonical term rewriting system derived from \( \mathcal{F}_q \mathcal{X} \cup \mathcal{P} \).]

(1) [Initialise.] \( R := \mathcal{F}_q \mathcal{X} \); \( \mathcal{E} := \mathcal{P} \).
(2) [Simplify.] while the simplify-inference rule applies do
\( (\mathcal{E}; R) := \text{Simplify}((\mathcal{E}; R)) \).
(3) [Delete.] while the delete-inference rule applies do \( (\mathcal{E}; R) := \text{Delete}((\mathcal{E}; R)) \).
(4) [Stop?] if \( \mathcal{E} = \emptyset \) then return \( R^* = R \) and stop.
(5) [Orient.] Let \( a \leftrightarrow b \in \mathcal{E} \); \( \mathcal{E} := \mathcal{E} \setminus \{a \leftrightarrow b\} \).
    (5.1) [Symmetrise.] \( R' := \text{Complete}((\{a \leftrightarrow b\}; \mathcal{F}_q \mathcal{X}, \succ)); \)
    \( E := R' \setminus \mathcal{F}_q \mathcal{X} \).
    (5.2) [Trivial ideal.] if \( \exists t \in T(\mathbb{F}, \forall \text{var}) : t \rightarrow \Omega \in E \) then
    return \( R^* = R' \) and stop.
(6) [Collapse.] \( R := R \cup E \); while the collapse-inference rule applies do
\( (\mathcal{E}; R) := \text{Collapse}((\mathcal{E}; R)) \).
(7) [Compose.] while the compose-inference rule applies do
\( (\mathcal{E}; R) := \text{Compose}((\mathcal{E}; R)) \).
(8) [Superpose.] Compute the set \( \mathcal{E} \) of all critical pairs of \( e2 \)-rules in \( R \) where a \( e2 \)-rule of \( E \) participates. \( \mathcal{E} := \mathcal{E} \cup \mathcal{E} \); continue with step 2.

Fig. 5. Algorithm FFPCOMPLETE.

6. Completion of polynomials over the rationals

In this section, we present a solution to the division problem for the case of the coefficient field being the set of rational numbers. Again our goal is to find an equational specification whose initial model is isomorphic to the coefficient domain we compute in.

Completion of the \( g \)-rule associated with the polynomial \( 2xy - y + 1 \) will, among others, produce a rule of the form

\[ M(a + a, X.Y) \rightarrow M(a, Y) \oplus M(-a, I) \]

(cf. Section 5.2). In order to make this rule ‘monic’ we need a term representation of the reciprocal of \( 2 \) in the rationals. Obviously, this reciprocal cannot be specified by just adding equations to \( \mathcal{X} \). We must extend our signature \( \mathbb{F} \). The easiest way to specify \( 1/2 \) is to add a new constant \( \frac{1}{2} \) to \( \mathbb{F} \) and the equation \( \frac{1}{2} + \frac{1}{2} \leftrightarrow 1 \) to \( \mathcal{X} \). The
critical pair of our original rule and the new equation (oriented from left to right) is

\[ (M(1,X,Y), \ M(\frac{1}{2},Y) \oplus M(-\frac{1}{2},I)) \]

which can be oriented to a type-I ground rule. Thus we must introduce new constants and equations for each reciprocal in \( \mathbb{Q} \). To construct all reciprocals in \( \mathbb{Q} \), given operators for addition, multiplication and negation, it suffices to define the reciprocals of all primes. Even with this restriction, we are forced to extend \( \mathcal{F} \) and \( \mathcal{F} \) by infinitely many constants and equations. Luckily we can relax the requirement that all divisions in \( \mathbb{Q} \) be effective. Namely, it suffices that only the divisions by leading coefficients generated during the completion are effective. In order to specify the rational numbers by the ground term model of an equational specification, we need an infinite set of equations that may be thought of as the limit of a series of equation sets \( \mathcal{E}_\mathbb{N} \).

The initial model of each \( \mathcal{E}_k \) is isomorphic to an extension \( \mathbb{Z}(k) = \mathbb{Z}(\frac{1}{p_1}, \ldots, \frac{1}{p_k}) \) of \( \mathbb{Z} \) for distinct prime numbers \( p_1, \ldots, p_k \). We will show that for each \( \mathcal{E}_k \) there exists a canonical term rewriting system \( \mathcal{L}(p_1, \ldots, p_k) \) which is extensible to a canonical term rewriting specification \( \mathcal{S}(p_1, \ldots, p_k) \) of polynomials over \( \mathbb{Z}(\frac{1}{p_1}, \ldots, \frac{1}{p_k}) \). In addition, for each ideal \( \mathfrak{I} \) in \( \mathbb{Q}[x_1, \ldots, x_r] \) there is an extension \( \mathbb{Z}(k) \) of \( \mathbb{Z} \) such that the Gröbner base of \( \mathfrak{I} \) in \( \mathbb{Q}[x_1, \ldots, x_r] \) is equal to the Gröbner base of \( \mathfrak{I} \) in \( \mathbb{Z}(k)[x_1, \ldots, x_r] \). We show how, starting with a specification of \( \mathcal{L} \), the rewrite specification of \( \mathcal{Z}(k) \) can be computed incrementally during the simulation of Buchberger’s algorithm and thereby present a method to work with an infinite term rewriting system.

6.1. A canonical term rewriting system for \( \mathbb{Z}(1/n_1, \ldots, 1/n_m) \)

In this section, we present a canonical term rewriting system for extension rings of \( \mathbb{Z} \). These extensions can be specified as finitely presented commutative rings. General results on the completion of finitely presented rings can be found in [43].

We begin with presenting a canonical term rewriting system for \( \mathbb{Z}(1/n) \), which is the commutative ring generated by 0, 1 and \( \frac{1}{n} \) where \( n(1/n) = 1 \). In this extension of \( \mathbb{Z} \), every power of a prime factor of \( n \) has an inverse. In order to specify \( \mathbb{Z}(1/n) \) for \( n \geq 2 \in \mathbb{N} \) by a term rewriting system, we add a new constant \( \frac{1}{n} \) of sort \( \text{Coeff} \) to \( \mathcal{F} \) and complete

\[ \mathcal{E}_\mathcal{S}(n) = \{ \frac{1}{n} + \cdots + \frac{1}{n} \leftrightarrow 1 \} \cup \mathcal{E}_\mathcal{S}. \]

The resulting canonical term rewriting system is

**Specification 5.**

\[ \mathcal{S}(n) = \mathcal{S} \cup \{ \alpha : \frac{1}{n} + \cdots + \frac{1}{n} \rightarrow 1, \]
Lemma 12. For every $n \geq 2 \in \mathbb{N}$, the term rewriting system $\mathcal{X}(n)$ is terminating modulo AC.

Proof. We show the existence of a polynomial interpretation $\phi$ which is an instance of $\Phi$ of Section 3, such that $\succ_\phi$ orients all rules in $\mathcal{X}(n)$. The termination of rules $\mathcal{X}(n).\alpha$ through $\mathcal{X}(n).\beta x$ poses no problem if $\Phi(\frac{1}{n}) > 1$.

$$\gamma : \Phi(-\frac{1}{n}) = K_1 \Phi(\frac{1}{n}) + 2 > \Phi(-1 + \frac{1}{n} + \cdots + \frac{1}{n}) = (n-1)\Phi(\frac{1}{n})$$

$$(n-1)\text{-times}$$

$$+ K_1 \Phi(1) + 5(n-1) + 2$$

if $K_1 > n$ and $\Phi(\frac{1}{n}) > K_1 \Phi(1) + 5n$.

$$\delta : \Phi(-(a \cdot \frac{1}{n})) = K_1 \Phi(\frac{1}{n}) \Phi(a) + 2 > \Phi(-a + (a \cdot \frac{1}{n}) + \cdots + (a \cdot \frac{1}{n}))$$

$$(n-1)\text{-times}$$

$$= (n-1)\Phi(\frac{1}{n}) \Phi(a) + K_1 \Phi(a) + 5(n-1) + 2$$

if $K_1 > n$ and $\Phi(\frac{1}{n}) > K_1 + 5n$.

The constraints found for $\Phi$ do not conflict with any other constraint on $\Phi$. Thus an instance $\phi$ of $\Phi$ exists such that $\succ_\phi$ is a simplification ordering that orients all rules in $\mathcal{X}(n)$ from left to right. \qed

Theorem 13. For every $n \geq 2 \in \mathbb{N}$, the term rewriting system $\mathcal{X}(n)$ is AC-canonical.

Proof. The rules $\mathcal{X}(n).\alpha$ through $\mathcal{X}(n).\delta$ do not superpose. It is easy to verify that all critical pairs between rules in $\{\mathcal{X}(n).\alpha, \ldots, \mathcal{X}(n).\delta\}$ and $\{\mathcal{X}(n).1, \ldots, \mathcal{X}(n).9\}$ are confluent. Thus the theorem follows from Lemma 12. \qed
Example 9. The canonical term rewriting system describing \( \mathbb{Z}(1/6) \) is

\[
\mathcal{Z}(6) = \mathcal{Z} \cup \{ \alpha : \frac{1}{6} + \frac{1}{6} + \frac{1}{6} + \frac{1}{6} + \frac{1}{6} \rightarrow 1, \\
\alpha x : a + \frac{1}{6} + \frac{1}{6} + \frac{1}{6} + \frac{1}{6} + \frac{1}{6} \rightarrow a + 1, \\
\beta : (a \cdot \frac{1}{6}) + (a \cdot \frac{1}{6}) + (a \cdot \frac{1}{6}) + (a \cdot \frac{1}{6}) \\
\hspace{1cm} + (a \cdot \frac{1}{6}) + (a \cdot \frac{1}{6}) + (a \cdot \frac{1}{6}) \rightarrow a, \\
\beta x : b + (a \cdot \frac{1}{6}) + (a \cdot \frac{1}{6}) + (a \cdot \frac{1}{6}) \\
\hspace{1cm} + (a \cdot \frac{1}{6}) + (a \cdot \frac{1}{6}) + (a \cdot \frac{1}{6}) \rightarrow b + a, \\
\gamma : -\frac{1}{6} \rightarrow -1 + \frac{1}{6} + \frac{1}{6} + \frac{1}{6} + \frac{1}{6} + \frac{1}{6}, \\
\delta : -(a \cdot \frac{1}{6}) \rightarrow -a + (a \cdot \frac{1}{6}) + (a \cdot \frac{1}{6}) \\
\hspace{1cm} + (a \cdot \frac{1}{6}) + (a \cdot \frac{1}{6}) + (a \cdot \frac{1}{6}), \}
\]

where \( \frac{1}{6} \) is a new coefficient constant. The rational numbers \( \frac{1}{2}, \frac{1}{3} \) and \( \frac{1}{6} \) can be translated to the terms \( \frac{1}{6} + \frac{1}{6}, \frac{1}{6} + \frac{1}{3} \) and \( \frac{1}{6} \), respectively, and all divisions by 2 and 3 are possible.

Given a finite set \( S \subseteq \mathbb{Q} \), we can construct a canonical term rewriting system for an extension \( \mathbb{Z}' \) of \( \mathbb{Z} \) such that each element of \( S \) has an inverse in \( \mathbb{Z}' \). The idea is to compute \( \mathcal{Z}(n) \) for a sufficiently large \( n \). However, this is not satisfactory for the following reasons:

- Assume we have constructed a term rewriting system \( \mathcal{R}_S \) with respect to which every element in \( S \) has an inverse. If we now want to obtain such a term rewriting system for a new set \( S \cup \{s\}, s \in \mathbb{Q} \), we must start all over again.
- AC-matching and AC-unification are extremely expensive for large, non-linear terms. Such non-linear terms will occur in the left-hand sides of the rules \( \mathcal{Z}(n) \cdot \beta \) and \( \mathcal{Z}(n) \cdot \beta x \).

Therefore, we want to present canonical term rewriting systems for extensions of \( \mathbb{Z} \) which can be extended incrementally. Clearly, to find the inverse of any number \( n \), only the inverses of its prime factors must be known. Following this idea, we want to construct canonical term rewriting systems for multiple extensions \( \mathbb{Z}(1/n_1, \ldots, 1/n_m) \) of \( \mathbb{Z} \), where the \( n_i \) are distinct prime numbers. We must add new constants \( \frac{1}{n_1}, \ldots, \frac{1}{n_m} \) to the signature \( \mathcal{F} \) and to get the canonical specification of such an extension we must complete

\[
\mathcal{X}(n_1, \ldots, n_m) = \{ n_1 \frac{1}{n_1} \leftrightarrow 1, \ldots, n_m \frac{1}{n_m} \leftrightarrow 1 \} \cup \mathcal{X}.
\]

This results in

Specification 6. \( \mathcal{X}(n_1, \ldots, n_m) = \mathcal{X}(n_1) \cup \cdots \cup \mathcal{X}(n_m) \cup \)

\[
\{ e_{ij} : \frac{1}{n_i} \cdot \frac{1}{n_j} \rightarrow -1 + \underbrace{\frac{1}{n_i} + \cdots + \frac{1}{n_i}}_{u_{ij}\text{-times}} + \underbrace{\frac{1}{n_j} + \cdots + \frac{1}{n_j}}_{v_{ij}\text{-times}} \mid 1 \leq i < j \leq m, \\
v_{ij}n_i + u_{ij}n_j = n_in_j + 1, \ 0 < u_{ij} < n_i, \ 0 < v_{ij} < n_j \ \text{minimal} \} \]
Lemma 14. For every set of distinct prime numbers \( \{n_1, \ldots, n_m\} \), the term rewriting system \( \mathcal{Z}(n_1, \ldots, n_m) \) is terminating modulo AC.

Proof. We only have to show that the \( \varepsilon \)-rules terminate. The rest follows from Lemma 12.

\[
\varepsilon : \Phi(h_i \cdot h_j) = \Phi(h_i) \Phi(h_j) > \Phi(-1 + u_{ij} h_i + v_{ij} h_j)
\]

\[
= u_{ij} \Phi(h_i) + v_{ij} \Phi(h_j) + 5(u_{ij} + v_{ij} - 1) + K_i \Phi(1) + 2
\]

\[\text{if } \Phi(h_i) > v_{ij} \Phi(h_j)/(\Phi(h_j) - u_{ij}) + (K_i \Phi(1) + 5(u_{ij} + v_{ij} - 1) + 2)/(\Phi(h_j) - u_{ij}).\]

From Lemma 12 we inherit the constraints \( K_1 > \max\{n_i \mid 1 \leq i \leq m\} \) and \( \Phi(1/n_i) > \max_j\{u_{ij}, v_{ij} \mid 1 \leq j \leq m\} \). Since all constraints are resolvable, there is an instance \( \phi \) of \( \Phi \) that orients all rules in \( \mathcal{Z}(n_1, \ldots, n_m) \) if \( K_1 > \max\{n_i \mid 1 \leq i \leq m\} \) and \( \Phi(1/n_i) > \max_j\{u_{ij}, v_{ij} \mid 1 \leq j \leq m\} \).

Lemma 15. For every set of distinct prime numbers \( \{n_1, \ldots, n_m\} \), all critical pairs in the term rewriting system \( \mathcal{Z}(n_1, \ldots, n_m) \) are confluent modulo AC.

Proof. All critical pairs between \( \alpha-, \beta-, \gamma-, \delta- \) rules on the one hand and rules of \( \{\mathcal{Z}(n)\cdot 1, \ldots, \mathcal{Z}(n)\cdot 9\} \) on the other hand are confluent by Theorem 13. Proving the confluence of critical pairs between \( \varepsilon \)-rules and \( \{\mathcal{Z}(n)\cdot 1, \ldots, \mathcal{Z}(n)\cdot 9\} \) is straightforward. The remaining proof that all critical pairs between \( \alpha-, \beta-, \gamma-, \delta- \) and \( \varepsilon \)-rules are confluent is very technical. It uses some number theoretical arguments. For details see [16].

Theorem 16. For every set of distinct prime numbers \( \{n_1, \ldots, n_m\} \), the term rewriting system \( \mathcal{Z}(n_1, \ldots, n_m) \) is canonical modulo AC.

Proof. Follows from Lemmas 14 and 15.

Example 10. The canonical term rewriting system describing \( \mathcal{Z}(1/2, 1/3) \) is

\[
\mathcal{Z}(2, 3) = \mathcal{Z} \cup \{ \alpha_2 : \begin{array}{c} \frac{1}{2} + \frac{1}{2} \rightarrow 1, \\ \alpha_2 x : a + \frac{1}{2} + \frac{1}{2} \rightarrow a + 1, \\ \beta_2 : (a \cdot \frac{1}{2}) + (a \cdot \frac{1}{2}) \rightarrow a, \end{array} \}
\]
\[ \beta_2 x : b + (a \cdot \frac{1}{2}) + (a \cdot \frac{1}{2}) \rightarrow b + a, \]

\[ \gamma_2 : -\frac{1}{2} \rightarrow -1 + \frac{1}{2}, \]

\[ \delta_2 : -(a \cdot \frac{1}{2}) \rightarrow -a + (a \cdot \frac{1}{2}), \]

\[ \alpha_3 : \frac{1}{3} + \frac{1}{3} + \frac{1}{3} \rightarrow 1, \]

\[ \alpha_3 x : a + \frac{1}{3} + \frac{1}{3} + \frac{1}{3} \rightarrow a + 1, \]

\[ \beta_3 : (a \cdot \frac{1}{2}) + (a \cdot \frac{1}{2}) + (a \cdot \frac{1}{2}) \rightarrow a, \]

\[ \beta_3 x : b + (a \cdot \frac{1}{2}) + (a \cdot \frac{1}{2}) + (a \cdot \frac{1}{2}) \rightarrow b + a, \]

\[ \gamma_3 : -\frac{1}{3} \rightarrow -1 + \frac{1}{3} + \frac{1}{3}, \]

\[ \delta_3 : -(a \cdot \frac{1}{3}) \rightarrow -a + (a \cdot \frac{1}{3}) + (a \cdot \frac{1}{3}), \]

\[ \epsilon_{2,3} : \frac{1}{2} \cdot \frac{1}{3} \rightarrow -1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{3}, \]

\[ \epsilon_{2,3} x : a \cdot \frac{1}{2} \cdot \frac{1}{3} \rightarrow -a + (a \cdot \frac{1}{2}) + (a \cdot \frac{1}{3}) + (a \cdot \frac{1}{3}), \]

where \( \frac{1}{2} \) and \( \frac{1}{3} \) are new coefficient constants. All rational numbers that can be described by \( \mathcal{F}(6) \) can also be described by \( \mathcal{F}(2,3) \). In particular 1/2, 1/3 and 1/6 can be represented by the terms \( \frac{1}{2}, \frac{1}{3} \) and \( -1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{3}. \) \( \square \)

6.2. Symmetrisation in \( \mathbb{Z}(1/n_1, ..., 1/n_m)[x_1, ..., x_r] \)

The term rewriting system \( \mathcal{F}(n_1, ..., n_m)\mathcal{X} = \mathcal{F}(n_1, ..., n_m) \cup \mathcal{X} \) specifies the polynomial ring over \( \mathbb{Z}(1/n_1, ..., 1/n_m). \)

**Theorem 17.** For all sets \( \{n_1, ..., n_m\} \) of distinct prime numbers, the term rewriting system \( \mathcal{F}(n_1, ..., n_m)\mathcal{X} \) is canonical modulo AC.

**Proof.** The interpretation \( \Phi \) of Section 3 is compatible with the extensions of \( \Phi \) described in Lemma 14. Thus \( \mathcal{F}(n_1, ..., n_m)\mathcal{X} \) is terminating. It is easy to see that the only critical pairs between rules in \( \mathcal{F}(n_1, ..., n_m) \) and \( \mathcal{X} \) result from superposing \( \alpha x \)- and \( \beta x \)-rules with rules in \( \{\mathcal{X}.10, ..., \mathcal{X}.12x\} \). The associated critical pairs are all confluent. Thus \( \mathcal{F}(n_1, ..., n_m)\mathcal{X} \) is canonical. \( \square \)

In the context of completion with \( \mathcal{F}(n_1, ..., n_m)\mathcal{X} \), we have to consider a new rule type. Before defining this type, we introduce some notation. We use \( \bar{A}, \bar{B} \) or any indexed version of these to denote a term of sort Coef in ground normal form. \( \bar{A}a \) (where \( a \) is a coefficient variable) denotes the \( \mathcal{F}(n_1, ..., n_m) \)-normal form of \( (\bar{A} \cdot a) \). Now the following rule types can be defined:

**Specification 7 (rule types).**

\[ \text{II.0 : } M(k, \bar{X}) \rightarrow \bigoplus_i M(\bar{A}_i, \bar{X}_i), \quad \text{II.1 : } M(k, \bar{X}x) \rightarrow \bigoplus_i M(\bar{A}_i, \bar{X}_i.x), \]

\[ \text{II.2 : } M(ka, \bar{X}) \rightarrow \bigoplus_i M(\bar{A}_ia, \bar{X}_i), \quad \text{II.3 : } M(ka, \bar{X}x) \rightarrow \bigoplus_i M(\bar{A}_ia, \bar{X}_i.x), \]
Rules of types II.0 and III.0 are also called g-rules. Type-III rules subsume the type-II rules. Let \( \tilde{A} \) be a ground term such that there exists another ground term \( \tilde{A}' \) and
\[
\tilde{A} \cdot \tilde{A}' \rightarrow_{\mathfrak{X}(n_1, \ldots, n_m)} 1.
\]
Then by completeness of the Knuth–Bendix completion, the following type-I rules can be derived for all the type-III rules above:
\[
M(\tilde{a}, \tilde{X}[.x]) \rightarrow \bigoplus_i M(((\tilde{A}_i \cdot \tilde{A}') a), \tilde{X}_i[.x]).
\]

The original type-III rules are confluent w.r.t. the derived type-I rule. Therefore every type-III rule with an invertible coefficient term in its left-hand side can be transformed into a type-I rule by Theorem 1. Otherwise, if \( \tilde{A} \) and \( \mathfrak{X}(n_1, \ldots, n_m)\mathfrak{X} \) are such that there is no multiplicative inverse of \( \tilde{A} \) in the set of \( \mathfrak{X}(n_1, \ldots, n_m)\mathfrak{X} \)-ground normal forms then for every type-III rule a type-II rule
\[
M(k_1 a, \tilde{X}[.x]) \rightarrow \bigoplus_i M(\tilde{B}_i a, \tilde{X}_i[.x])
\]
can be derived from the completion of the type-III rule and \( \mathfrak{X}(n_1, \ldots, n_m)\mathfrak{X} \), where
\[
\psi^{-1}(\tilde{A}) = k_1/k_2 \text{ for } k_1, k_2 \text{ relative prime, and } \tilde{B}_i = \psi(k_2 \psi^{-1}(\tilde{A}_i)).
\]

**Example 11.** Let us consider how the following rule \( \rho_1 \) of type III.2
\[
\rho_1 : M(a + (a \cdot \frac{1}{3}), X.X) \rightarrow M(a, Y)
\]
can be transformed into a type-II.1 rule by symmetrisation with \( \mathfrak{X}(2,3)\mathfrak{X} \). The rule \( \rho_1 \) superposes with rule
\[
\mathfrak{X}.9 : a \cdot (b + c) \rightarrow (a \cdot b) + (a \cdot c)
\]
resulting in the critical peak
\[
M(a + b + (a \cdot \frac{1}{3}) + (b \cdot \frac{1}{3}), X.X) \leftarrow M((a + b) + ((a + b) \cdot \frac{1}{3}), X.X) \rightarrow M(a + b, Y).
\]
The resulting critical pair can be oriented from the left to the right and one more superposition with the rule \( \mathfrak{X}.9 \) results in the rule
\[
\rho_2 : M(a + b + c + (a \cdot \frac{1}{3}) + (b \cdot \frac{1}{3}) + (c \cdot \frac{1}{3}), X.X) \rightarrow M(a + b + c, Y).
\]

\( \rho_2 \) can now be superposed with
\[
\mathfrak{X}(2,3)\beta_3x : b + (a \cdot \frac{1}{3}) + (a \cdot \frac{1}{3}) + (a \cdot \frac{1}{3}) \rightarrow b + a
\]
and yields the critical peak
\[ M(a + a + a + (a \cdot \frac{1}{2}) + (a \cdot \frac{1}{2}), X \cdot X) \]
\[ + (a \cdot \frac{1}{2}) + (a \cdot \frac{1}{2}), X \cdot X) \]
\[ M(a + a + a, Y). \]

Again we can orient the critical pair from the left to the right and get the type-II rule
\[ \rho_3 : M(a + a + a + a, X \cdot X) \rightarrow M(a + a + a, Y) \]

we looked for. Note that this type-II rule could also be derived in \( \mathcal{Z}(3) \mathcal{X} \). Finally, two further superpositions with
\[ \mathcal{Z}(2, 3) \rho_2 : b + (a \cdot \frac{1}{2}) + (a \cdot \frac{1}{2}) \rightarrow b + a \]
result in the "monic" rule

\[ \rho_4 : M(a, X \cdot X) \rightarrow M((a \cdot \frac{1}{2} \cdot \frac{1}{2}) + (a \cdot \frac{1}{2} \cdot \frac{1}{2}) + (a \cdot \frac{1}{2} \cdot \frac{1}{2}), Y) \]

which makes the rules \( \rho_1, \rho_2 \) and \( \rho_3 \) confluent allowing for the application of Theorem 1. Starting with \( \rho_1 \) the derivation of rule \( \rho_4 \) simulates the divisions of \( \rho_1 \) by the leading coefficient of its associated polynomial \( \frac{1}{2}x^2 - y \).

6.3. Computing Gröbner bases in \( \mathbb{Q}[x_1, \ldots, x_r] \)

Given a specification \( \mathcal{Z}(n_1, \ldots, n_m) \mathcal{X} \), we can compute a Gröbner base for an ideal associated with a set of g-rules if every irreducible type-III rule created during the completion can be made 'monic'. That is, the leading coefficients of all these type-III g-rules have their inverse elements in \( T(\{0, 1, \frac{1}{n_1}, \ldots, \frac{1}{n_m}, -, +, *\}) \). In this case the completion procedure is the same as in the case of finite fields.

To compute Gröbner bases for arbitrary ideals using term completion, we need an infinite rewrite system specifying \( \mathcal{Z}(1/n_1, \ldots, 1/n_k, \ldots) \{x_1, \ldots, x_r\} \) where \( \{1/n_1, \ldots, 1/n_k, \ldots\} \) denotes a sequence of all inverted primes. Of course this is not practicable; but to complete any given ideal, only divisions by a finite number of primes must be defined. It is hardly possible to predict which set \( \{n_1, \ldots, n_m\} \) of prime numbers must be chosen such that the specification \( \mathcal{Z}(n_1, \ldots, n_m) \mathcal{X} \) can be used to compute the Gröbner base for a given set of polynomials. We propose two scenarios to overcome these difficulties:

**Lucky guess:** This is not a very intelligent solution, but it is of course possible to guess the right set of primes for which the completion succeeds. If there is a heuristic to determine the leading coefficients computed during the completion, lucky guess would be a very good solution. Also if an upper bound for the coefficients is known, 'guessing' all primes less than or equal to this bound yields an effective algorithm.

**Incremental approximation of \( \mathbb{Q}[x_1, \ldots, x_r] \):** In this case we start the completion with \( \mathcal{R} = \mathcal{Z} \mathcal{X} \) as the specification of the polynomial domain and a set of g-rules describing the ideal. Whenever we encounter an irreducible type-II e-rule

\[ M(ka, \bar{X}_i[x]) \rightarrow \bigoplus_i M(\bar{A}_i a, \bar{X}_i[x]) \]
which cannot be made monic, we set $\mathcal{F} := \mathcal{F} \cup \{\frac{1}{n_i}\}$ and

$$\mathcal{R} := \text{Complete} \left( \left\{ \frac{1}{n_i} + \cdots + \frac{1}{n_i} \leftrightarrow 1 \right\}; \mathcal{R} \right),$$

where $n_i$ is the smallest prime number that divides $k$ and has not been introduced to $\mathcal{F}$.

Let $\text{EXTEND}$ be a procedure which performs this update of $\mathcal{F}$ and $\mathcal{R}$ (see Fig. 7 in the Appendix). Note that the factorisation needed in $\text{EXTEND}$ can also be simulated by term rewriting methods as described in the Appendix. Now we can formulate the completion algorithm $\text{RPCOMPLETE}$ in Fig. 6 for ideal completion in $\mathbb{Q}[x_1, \ldots, x_r]$.

$\text{RPCOMPLETE}$ simulates completion with an infinite set of rules. More precisely, we complete an infinite canonical set of rules together with a finite set of ground equations/rules. The completion is correct because the resulting canonical term rewriting system is $\mathcal{R}^* = \mathcal{F}(n_1, \ldots, n_m)\mathcal{X} \cup \mathcal{G}$, where $\{n_1, \ldots, n_m\}$ is a set of prime numbers and $\mathcal{G}$ is a set of type-I e-rules. Type-I e-rules do not superpose with any rule of sort $\text{Coeff}$. Thus no critical pairs between a rule in $\mathcal{G}$ and a rule in $\mathcal{F}(n_1, \ldots, n_k, \ldots)\mathcal{X} \setminus \mathcal{F}(n_1, \ldots, n_m)\mathcal{X}$ exist.

### 7. Miscellaneous remarks

#### 7.1. On strategies for completion

Completion procedures are non-deterministic procedures. That is, we are free to use any ordering in that the inference rules of Fig. 2 are to be applied. A fixed scheme that determines the order in which the inference rules must be applied is called a strategy.

It is fair if all equations are eventually considered in an inference step (delete, simplify or orient) and if all critical pairs of all rules are created (unless their deletion is ensured by a critical pair criterion). Thus for (semi-)decision procedures only fair strategies are of interest. In the same way, it does not matter to which rule or equation (and at which position) an inference step is applied. This non-determinism may be fixed by a strategy too.

The complexity of the completion process depends strongly on the strategy used. In particular, there are strategies preferred for Buchberger's algorithm and others that are advantageous for term completion. The strategies used in $\text{BA}_k$, $\text{KB}$, $\text{FFPCOMPLETE}$ and $\text{RPCOMPLETE}$ are not necessarily optimal but fair completion strategies. They have been chosen to highlight the analogies between these procedures. Note that any other implementation using a different strategy of Buchberger's algorithm can be simulated too. Besides, changing the strategy of these procedures does not change their functionality.

The strategies of $\text{FFPCOMPLETE}$ and $\text{RPCOMPLETE}$ (cf. p. 174 and 183) give priority to symmetrisation. This not only helps us to see the analogy between the orientation step in Buchberger's algorithm and symmetrisation, but is also essential to keep the complexity of the simulation at a moderate level.
R* ← RPCOMPLETE(℘, ⊳)

[Rational polynomial AC-term completion with ‘Buchberger strategy’.

℘ is a set of ground equations of sort Poly and ⊳ is a terminating term ordering which includes the lexicographical ordering on multivariate polynomials described by ground terms. Then R* = ℳ(n₁, ..., nₙ)ℳ ∪ ℘ is a canonical term rewriting system where {n₁, ..., nₙ} is a set of prime numbers and ℘ is a set of type-I r-rules. The polynomials associated with ℘ describe a Gröbner base in the domain of rational polynomials of the ideal described by ℘.]

(1) [Initialise.] ℳ := {0, 1, −, +, •, −, •, M, ⊗, ⊘}; ℳ := ℳ; ℳ := ℳ; ℘ := ℘; ℘ := ℘; ℘ := ℘.

(2) [Simplify.] while the simplify-inference rule applies do (◇; ℘) := Simplify((◇; ℘)).

(3) [Delete.] while the delete-inference rule applies do (◇; ℘) := Delete((◇; ℘)).

(4) [Stop?] if ℘ = ∅ then return R* = ℘ and stop.

(5) [Orient.] Let a ⇒ b ∈ ℘; ℘ := ℘ \ {a ⇒ b}; ℘ := ∅;

(5.1) [Symmetrise.] R' := Complete(({a ⇒ b}; ℳ, ⊳);

E := E ∪ (R' \ ℳ);

(5.2) [Trivial ideal.] if ∃t ∈ T(ℳ, Var) : t ⇒ Ω ∈ E then return R* = ℘ and stop;

(5.3) [Extend Specification.] if ∃ t ∈ T(ℳ, Var) : t ⇒ Ω ∈ E then

{ EXTEND(k, ℳ, ℳ, ⊳ ; ℳ, ℳ) * ; continue with step 5.2. }

(6) [Collapse.] ℘ := ℘ ∪ ℳ ∪ E; while the collapse-inference rule applies do (◇; ℘) := Collapse((◇; ℘)).

(7) [Compose.] while the compose-inference rule applies do (◇; ℘) := Compose((◇; ℘)).

(8) [Superpose.] Compute the set G if all critical pairs of type-I r-rules in ℘ where a rule in E participates. ℘ := ℘ ∪ G; continue with step 2.

* The parameters in front of the ‘;’-sign are input parameters and those following the ‘;’ are output parameters.

Fig. 6. Algorithm RPCOMPLETE.

7.2. A generic Buchberger algorithm

We have seen that Knuth–Bendix completion modulo AC can simulate Buchberger’s algorithm for polynomials over finite fields or the rational numbers. More precisely, given a canonical rewrite specification ℳ for polynomials over a field Ω and a set of ground equations ℘ describing an ideal presentation ℘, Complete((℘, ℳ), ⊳) simulates the computation of the Gröbner base of ℘. Here Complete is any fair implementation of the Knuth–Bendix procedure modulo AC. Now ℳ is the result of completing an equational specification ℘ refining the canonical rewrite systems ℳ.
Instead of first computing $\mathcal{K}$ and then completing $\mathcal{P}$ modulo $\mathcal{K}$ we can compute the Gröbner base of $\mathcal{P}$ by completing $(\mathcal{E}, \mathcal{P}, \mathcal{K})$. Thus by fixing the input $\mathcal{K}$ and keeping $\mathcal{E}$ variable, we obtain a generic completion procedure for polynomials over any commutative coefficient rings that are equationally specifiable. This is not only restricted to fields. Such a simulation of (an extension of) Buchberger’s algorithm for integral polynomials (i.e., $\mathcal{E} = 0$) is shown in [15]. [16] handles the case of arbitrary modular commutative rings (i.e., $\mathcal{E} = \{a + \cdots + a \rightarrow 0\}$) and conjectures that (an extension of) Buchberger’s algorithm for polynomials over any finitely presented ring can be simulated that way.

It is well known that for univariate polynomials Buchberger’s algorithm reduces to Euclid’s algorithm for polynomial gcd computations, and for linear input polynomials Buchberger’s algorithm reduces to Gauss’ algorithm. Thus term completion also provides generic procedures for those two algorithms.

7.3. Extension rule complexity

As indicated above, the way we specify polynomials by term rewriting systems allows us to vary the polynomial rings we want to work with by simply adding new equations (and constants or operators) to the specification of the coefficient ring. In this way, it is also possible to specify multivariate polynomials recursively (i.e., $\mathcal{K}[[x_1, x_2]] = ((\mathcal{K}[[x_2]])[x_1])$). However, it turns out that this leads to very complex specifications because in a strictly sorted algebra we may not overload operations in $\mathcal{K}[[x_2]]$ with the corresponding operations in $(\mathcal{K}[[x_2]])[x_1]$ (e.g., the canonical term rewriting system describing $(\mathcal{K}[[x_2]])[x_1]$ contains 37 rules plus AC-extensions). In addition, the simulation of polynomial completion becomes very expensive because the number of extension rules associated with an arbitrary polynomial increases drastically. So for example, for $(\mathcal{K}[[y]])[x]$ with $y \rightarrow x$ there are 8 extension rules associated with $x^4 + 2x^2 + 4x$, 16 with $x^2y + 2x$ and 64 with $2xy - x^2 - 2x$. The number of extension rules associated with an arbitrary polynomial may even serve as a measure of ‘complexity’ for polynomial completion procedures: completion of integral polynomials (8 extension rules [15]) is more complicated than completion of polynomials over fields (2 extension rules); and using the distributed normal form as data structure for polynomials in Buchberger’s algorithm is much more efficient than using a recursive structure as proposed above.

7.4. Semi-compatibility vs. compatibility

A reduction relation $\rightarrow_{\mathcal{E}}$ is compatible w.r.t. some operation $o$ if for two objects $a$ and $b$, $o(a) \rightarrow_{\mathcal{E}} o(b)$ follows from $a \rightarrow_{\mathcal{E}} b$. A reduction relation $\rightarrow_{\mathcal{E}}$ is called

---

6 As we have seen it suffices that the coefficient domain is the initial model of the equational specification.
semi-compatible if from $a \rightarrow_{\mathcal{R}} b$ follows that $o(a)$ and $o(b)$ have a common $\mathcal{R}$-normal form.

It can be proved that a rewrite relation is compatible w. r. t. application of substitution\(^7\) and w. r. t. replacement of subterms in a term:

\[
\begin{align*}
S & \rightarrow_{\mathcal{R}} T \Rightarrow \begin{cases} 
S\sigma & \rightarrow_{\mathcal{R}} T\sigma \\
us[p] & \rightarrow_{\mathcal{R}} u[t][p]
\end{cases}
\end{align*}
\]

The polynomial reduction relation is only semi-compatible w. r. t. addition and multiplication of polynomials\(^8\):

\[
\begin{align*}
p \rightarrow_{\mathcal{R}} pq & \Rightarrow \begin{cases} 
(p\sigma)_{\downarrow_p} = (q\sigma)_{\downarrow_p} \\
(p+r)_{\downarrow_p} = (q+r)_{\downarrow_p}
\end{cases}
\end{align*}
\]

The observation of this difference between term rewriting and polynomial reductions gave rise to the assumption that there might be a profound difference between the respective completion procedures \cite{13}. In particular the proofs of the respective critical pair theorems could not be unified. This question has been resolved in \cite{16, Chap. 3} using proofs based on subconnectedness rather than confluence for the critical pair theorem of Knuth and Bendix.

The loss of the compatibility property is a typical phenomenon for normalised reductions. That is, objects are always presented in some kind of normal form (e.g., the distributed normal form for polynomials or reduced words for finitely presented groups). Semi-compatibility can also be observed for our rewrite simulation if we consider only terms in $X\mathcal{X}$-normal form.

To see that $(S \oplus P)_{\downarrow_{X\mathcal{X}}(i \rightarrow_r)} (T \oplus P)_{\downarrow_{X\mathcal{X}}}$ does not follow from $S \rightarrow_{\{i \rightarrow_r\}} T$, consider $P = S \circ M(-1,I)$.

**Corollary 18.** Let $S$, $T$ and $P$ be terms of sort $\mathcal{R}dy$ and $\mathcal{R} = X\mathcal{X} \cup E$ where $E$ is the set of $e$-rules associated with a set of monic polynomials. If $i \rightarrow_r \in \mathcal{R}$ such that $S \rightarrow_{\{i \rightarrow_r\}} T$ then $(S \oplus P)_{\downarrow_{X\mathcal{X}}}, (T \oplus P)_{\downarrow_{X\mathcal{X}}}$ and $(S \circ P)_{\downarrow_{X\mathcal{X}}}, (T \circ P)_{\downarrow_{X\mathcal{X}}}$ are confluent w. r. t. $\mathcal{R}$ modulo $AC$.

Corollary 18 relates our approach to the work of Kandri-Rody et al. \cite{35} who propose to compare Buchberger's algorithm with the completion of a term rewriting system modulo a simplification relation where the simplification procedure must satisfy a so-called orthogonality property w. r. t. the reduction relation. In the procedure proposed by Kandri-Rody et al. only fully simplified terms should be reduced and only the critical pairs of the reduction relation must be considered. The distinction between a rewrite relation and a simplification relation is not necessary in our approach because we can prove that the term rewriting systems we construct are confluent.

Normalised rewriting has recently been investigated in \cite{50} with the goal of optimising term completion procedures.

---

\(^7\)This property is sometimes called stability.

\(^8\)It is compatible w. r. t. the multiplication of non-zero polynomials.
8. Conclusion

We showed that for fixed coefficient domains Buchberger's algorithm can be simulated by term completion modulo associativity and commutativity. Our simulation can be interpreted as a completion of a finitely presented algebra in the spirit of [43]. More precisely, Buchberger's algorithm can be interpreted as completion of ground equations modulo a canonical term rewriting describing the theory of polynomials. On an even more abstract level, Buchberger's algorithm is a Knuth-Bendix procedure modulo an equational theory describing polynomials. Thus we can claim that Buchberger's algorithm is a specialised form of term completion modulo AC. A key outcome of our work is that division of polynomials by leading coefficients reduces to a unification problem in the term rewriting simulation. The solution of this unification is a part of the term completion process. The solvability of unification problems presupposes their solvability in the ground term model. Therefore, the initial algebra semantics of the term rewriting system representing the polynomial ring plays a central rôle.

We do not propose to replace Buchberger's algorithm by a term completion procedure. On the contrary, Buchberger's algorithm is in general much more efficient than its simulations by term completion. However, we argue that term completion is a good means to study algebraic completion procedures. The structure of term rewriting systems and term completion procedures is easy and well-understood. Many complicated features of algebraic completion (e.g. symmetrisation or complicated rewrite relations for the generalised Buchberger's algorithm for polynomials over rings, see also [16]) can be easily explained in the setting of term completion. Therefore term completion provides an appropriate framework to present, explain and illustrate various algebraic completion procedures. Buchberger's algorithm is highly optimised w.r.t. the algebraic domain it is supposed to work for. Translating it into a term completion procedure elucidates which parts of the original procedure are essential for completion (the parts that are invariant w.r.t. different term completion strategies, -orderings and coefficient representations) and which features are included for efficiency reasons only (e.g., the left-hand sides of the rules contain only monomials, or computing with polynomials in distributive normal form only). Therefore it should be much easier to modify or generalise an existing algebraic completion procedure given a term completion presentation. Once a sound modified completion procedure has been found, it can still be optimised and adapted to its input data.

Acknowledgements

I am grateful to Prof. Loos for initiating this work and for many discussions. I wish to thank Lars Langemyr for inspiring conversations and Prof. Küchlin for helpful comments. I also want to thank Sibylle Schupp, Beatrice Amrhein and Andreas Weber for critically reading this article.
Appendix A. Signature extension and factorisation

We show that AC-matching is a very powerful mechanism. It is even possible to simulate the factorisation operation needed to extend $F$ and $\cal{X}$ in the completion procedure $\text{RPCOMPLETE}$. Fig. 7 presents the procedure $\text{EXTEND}$ and the function $\text{FACTORIZE}$ is shown in Fig. 8. In the algorithms, $\{x \mapsto t\}$ denotes a substitution mapping $x$ to $t$ and $t(\lambda)$ is the root symbol of the term $t$.

\begin{verbatim}
\text{EXTEND}\left(t,F,R,\succ; F^* R^*\right)

[Extend term rewriting system.]
$t = ka$ is a term of sort $\text{Coeff} \ (a \in \cal{V}ar)$. $F$ is a signature. $R$ is a canonical term rewriting system. $\succ$ is a terminating ordering such that the term 1 is minimal. Let $k'$ be the smallest factor of $k$ such that $k' \cdot k'' =_R 1$ cannot be derived for a $k'' \in T(F)$. Then $F^* = F \cup \{\frac{1}{k'}\}$ for a new constant $\frac{1}{k'}$ and $R^*$ is a canonical term rewriting system for $\langle\{k'/t \mapsto 1\}; R\rangle$.]

(1) [Factorize $k$.] $P := \text{FACTORIZE}(+, t)$.
(2) [Select smallest factor.] Let $p \in P$ be the smallest term such that no left-hand side of a rule in $R$ matches $p$; let $\{a\} = \cal{V}ar(t)$ and $b \in \cal{V}ar$.
(3) [Extend signature.] Let $\pi \notin F$ be a constant symbol of sort $\text{Coeff} ; F^* := F \cup \{\pi\}$.
(4) [Extend term rewriting system.] $R^* := \text{COMPLETE}(\{p\{a \mapsto \pi \mapsto 1\}; R, \succ\} \Box$
\end{verbatim}

---

\begin{verbatim}
P \leftarrow \text{FACTORIZE}(o, t)

[Factorize. $o$ is an AC-operator and $t = \overbrace{t' \circ \ldots \circ t'}^{n\text{-times}}$ is a term of sort $S$ such that $t'(\lambda) \neq o$. Then $P = \{p_1, \ldots, p_n\}$ is the set of terms such that $p_i = x \overbrace{\circ \ldots \circ x}^{n_i\text{-times}}$ $x \in \cal{V}ar$ of sort $S$ and $n_i$ is a prime factor of $n$.]

(1) [Initial.] Let $x, y \in \cal{V}ar; t' := t; q := y; q_s := x \circ y; P := \emptyset$.
(2) [Divide.] while $q \sigma \neq t'$ do
\{ $q := q \circ y; q_s := q_s \circ y;$ if no $p \in P$ matches $q$ then
  if $q \sigma = t'$ for some substitution $\sigma$ then $P := P \cup \{q\}; t' := \sigma(y)$ \}
\end{verbatim}

---

Fig. 7. Algorithm $\text{EXTEND}$.  

Fig. 8. Algorithm $\text{FACTORIZE}$.  

\begin{verbatim}
\text{FACTORIZE}(o, t)

[Factorize. $o$ is an AC-operator and $t = \overbrace{t' \circ \ldots \circ t'}^{n\text{-times}}$ is a term of sort $S$ such that $t'(\lambda) \neq o$. Then $P = \{p_1, \ldots, p_n\}$ is the set of terms such that $p_i = x \overbrace{\circ \ldots \circ x}^{n_i\text{-times}}$ $x \in \cal{V}ar$ of sort $S$ and $n_i$ is a prime factor of $n$.]

(1) [Initial.] Let $x, y \in \cal{V}ar; t' := t; q := y; q_s := x \circ y; P := \emptyset$.
(2) [Divide.] while $q \sigma \neq t'$ do
\{ $q := q \circ y; q_s := q_s \circ y;$ if no $p \in P$ matches $q$ then
  if $q \sigma = t'$ for some substitution $\sigma$ then $P := P \cup \{q\}; t' := \sigma(y)$ \}
\end{verbatim}
References


