D-Bases for Polynomial Ideals over Commutative Noetherian Rings*

Leo Bachmair and Ashish Tiwari

Department of Computer Science, SUNY at Stony Brook,
Stony Brook, NY 11794, U.S.A,
{leo,astiwari}@cs.sunysb.edu

Abstract. We present a completion-like procedure for constructing D-bases for polynomial ideals over commutative Noetherian rings with unit. The procedure is described at an abstract level, by transition rules. Its termination is proved under certain assumptions about the strategy that controls the application of the transition rules. Correctness is established by proof simplification techniques.

1 Introduction

Buchberger's algorithm for constructing a *Gröbner basis* for polynomial ideals over a field is one of the central methods in computational algebra [7, 9]. It has been extended in two directions: to polynomials over coefficient domains other than fields and to non-commutative polynomial rings, cf. [6]. For instance, Kandri-Rody and Kapur [11] and Pan [14] considered polynomials over Euclidian rings and principal ideal domains, respectively. A Gröbner basis for an ideal I induces a terminating rewrite relation on polynomials, such that two polynomials have the same normal form if, and only if, they belong to the same remainder class of I. The algorithm proposed by Pan [14] constructs a D-basis, whereby any polynomial in the ideal I (and only these polynomials) can be reduced to zero. (Given a D-basis, one can determine whether p and q are equivalent modulo I by checking whether their difference $p - q$ can be rewritten to zero.) In this paper, we present a procedure for constructing D-bases for polynomials over more general Noetherian rings.

We describe our method in proof-theoretic terms, as an abstract completion procedure in the sense of Bachmair and Dershowitz [4, 1]. The presentation is by transition (or inference) rules operating on sets of equations and rewrite rules. Transitions describe basic steps for changing the presentation of the underlying equational theory. If a suitable control strategy is applied, this transformation process yields a D-basis. The correctness of the approach is proved by proof simplification arguments: we show that any proof of $p \approx 0$ can be normalized to a rewrite proof. Formally, our method is a constraint-based associative-commutative completion procedure, where all algebraic computations on the coefficient domain are embedded via constraints. Constraints thus provide a clean

* The research described in this paper was supported in part by the National Science Foundation under grant CCR-9510072.

way of separating the two modes of computation that are employed in the construction of polynomial ideal bases: rewriting (or reduction) on the polynomial ring and computations with ideals in the coefficient domain (e.g., computation of the intersection of finitely generated ideals).

Our work is motivated in part by our interest in extending general-purpose automated deduction methods by specialized mathematical procedures. Procedures for constructing polynomial ideal bases are examples of methods that combine abstract, syntactical operations (such as matching and unification) with specialized domain-dependent operations. The current paper is a continuation of previous work in this direction in which Buchberger's original algorithm for constructing a Gröbner basis for a polynomial ideal over a field was described in a proof-theoretic setting similar to the one we use, cf. [2]. But in the case of polynomials over fields, a considerably optimized and simplified procedure is possible that, for instance, does not require the flexibility of the full constraint formalism. The addition of constraints gave rise to a number of technical problems, which required us to redesign some transition rules and to add new ones, while also changing the term signature slightly.

We assume that the reader is familiar with the theory of term rewriting and completion [4, 1]. The integration of constraints is fairly straightforward Kirchner, Kirchner and Rusinowitch [12] and fits into the framework proposed by for constraint-based completion.

2 Polynomials

Buchberger's algorithm manipulates (multi-variate) polynomials, i.e., expressions typically written as sums $\sum_{1 \leq i \leq n} c_i t_i$, where the c_i are elements from some coefficient domain and the t_i are products of distinguished constants called *indeterminates*. We first outline how polynomials may be represented as (variable-free) first-order terms over a many-sorted signature with sorts *Coef* (coefficients), *PProd* (power products) and *Poly* (polynomial expressions), see also Bündgen [10] and Bachmair and Ganzinger [2].

Coefficients

We assume that the coefficient domain is a commutative ring with unit, specified by some algebra $\mathcal{B}$ over a given signature so that every element of $\mathcal{B}$ is represented by some ground term (i.e., $\mathcal{B}$ is term-generated). The specific term representation of coefficients is not significant for our purposes and may vary with the coefficient domain, but we do assume that the signature contains at least the declarations

$$\begin{array}{rr}
0, 1: & \to \mathit{Coef} \\
-: & \mathit{Coef} \to \mathit{Coef} \\
+, \cdot: & \mathit{Coef} \times \mathit{Coef} \to \mathit{Coef}
\end{array}$$

Variables of sort *Coef*, also called *coefficient variables* are denoted by the letters x, y, and z (possibly with subscripts or other annotation). We also use (meta-variables) a, b, c, and d to denote terms (usually ground terms) of sort *Coef*, also called *coefficient terms*.

We will simplify the notation, writing, for instance, ax instead of $a \cdot x$, or using summations, $\sum_{j=1}^{k} a_j x_j$. We also use vector notation: $\mathbf{a}$ and $\mathbf{b}$ denote vectors of coefficient terms, e.g., $(a_1, \ldots, a_k)$, whereas $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ denote *transposed* vectors (of dimension $k \times 1$) of coefficient variables. Thus $\mathbf{a} \cdot \mathbf{x}$ also denotes a sum $a_1 \cdot x_1 + a_2 \cdot x_2 + \cdots a_k \cdot x_k$.

We define B (the "base theory") as the set of all ground equations $a \approx b$ such that a and b denote the same element in $\mathcal{B}$. We need to be able to decide whether a given equation $a \approx 0$ belongs to B, and hence the ground theory B must be decidable. The construction of ideal bases depends on additional properties of the coefficient domain, which will be given below.

Polynomial expressions

Polynomial expressions are constructed from coefficients, power products and function symbols representing ring operations. We use the following signature:

$$\begin{array}{rr}
I, X_1, \ldots, X_n : & \rightarrow \mathit{PProd} \\
\cdot : & \mathit{PProd} \times \mathit{PProd} \rightarrow \mathit{PProd} \\
M : & \mathit{Coef} \times \mathit{PProd} \rightarrow \mathit{Poly} \\
\Omega : & \rightarrow \mathit{Poly} \\
\oplus, \otimes : & \mathit{Poly} \times \mathit{Poly} \rightarrow \mathit{Poly}
\end{array}$$

Variables of sort *Poly* are denoted by the letters α, β, and γ; variables of sort *PProd*, by ν and μ. The constants X_i are called *indeterminates*. Terms of sort *PProd* are called *power products* and denoted by s, t and u; terms of sort *Poly*, which are also called *polynomial expressions*, are denoted by p, q, and r. Terms of the form $M(a, t)$ are called *monomials*.

We will use standard algebraic notation for power products, monomials, and polynomial expressions. For instance, for our purposes it is sufficient to characterize power products up to associativity and commutativity, and hence we write, say, X^3Y. The symbol M will usually also be represented by juxtaposition. For instance, we write $3X^2Y$ instead of $M(3, X \cdot (X \cdot Y))$. We may even write $c \oplus p$, taking a coefficient term c as an abbreviation for the power product $M(c, I)$. The constant Ω will also be denoted by 0.

Constraints

The construction of ideal bases requires different modes of computation for coefficients and polynomial expressions. Computation with coefficients requires algebraic operations such as intersection and union of ideals, whereas the manipulation of the non-coefficient parts of polynomials involves syntactical operations such as (associative-commutative) matching and unification and term replacement. A constraint-based formalism allows for a clean separation of the two parts.

The *simple constraints* we use are either *equality constraints* of the form $x = a$, $x = y + z$, $x = y \cdot z$, or $x = \sum_{j=1}^{k} a_j y_j$, where all terms a and a_j are ground; or *negated* equality constraints of the form $x \neq 0$. A *constraint* is a conjunction of (zero or more) simple constraints.[2] We use the matrix notation $\mathbf{x} = \mathbf{A} \cdot \mathbf{y}$ to represent a conjunction of constraints $x_i = \mathbf{a_i} \cdot \mathbf{y}$, for $i = 1, \ldots, l$, where $\mathbf{A}$ contains the vectors $\mathbf{a_i}$ as rows. The letters C and D are used to denote constraints.

By a *solution* of a constraint we mean a substitution σ that assigns ground terms to coefficient variables, such that, in the case of an equality constraint, both terms denote the same element of the coefficient domain $\mathcal{B}$ or, in the case of a negated constraint $x \neq 0$, the term $x\sigma$ denotes a non-zero element. Any substitution is a solution of the empty constraint.

A *constrained equation* is an expression $p \approx q$ `if` C, where p and q are polynomial expressions and C is a constraint. If C is empty, we simply write $p \approx q$. By an (*unconstrained*) *instance* of $p \approx q$ `if` C we mean any equation $p\sigma \approx q\sigma$, where the restriction of σ to coefficient variables is a solution of the constraint C. By the ground theory represented by a set of constrained equations we mean the equational theory induced by the set of all its unconstrained ground instances.

We shall enforce a sharp distinction between coefficient and non-coefficient terms by transforming expressions via *variable abstraction*. That is, we replace a non-variable coefficient subterm a in a polynomial expression by a new coefficient variable x defined by a simple constraint $x = a$.

For example, the polynomial equation $3X^2Y \oplus (-2)XY^2 \approx 0$ is transformed to

$$uX^2Y \oplus vXY^2 \approx 0 \quad \texttt{if} \;\; u = 3,\; v = -2.$$[3]

Evidently, the ground instances of the new equation are equivalent (up to the representation of coefficients) to the ground instances of the original equation.

Polynomials

Let us next discuss the equational theory of polynomials. By H we denote the set of equations

$$\begin{aligned} M(x,\nu) \oplus M(y,\nu) &\approx M(x+y,\nu) \\ M(x,\mu) \otimes M(y,\nu) &\approx M(x \cdot y, \mu \cdot \nu) \end{aligned}$$

that relate addition and multiplication of polynomials to the corresponding operations on coefficients and power products; by AC we denote the set of associativity and commutativity axioms

$$\begin{array}{cc} \alpha \oplus (\beta \oplus \gamma) \approx (\alpha \oplus \beta) \oplus \gamma & \alpha \oplus \beta \approx \beta \oplus \alpha \\ \alpha \otimes (\beta \otimes \gamma) \approx (\alpha \otimes \beta) \otimes \gamma & \alpha \otimes \beta \approx \beta \otimes \alpha \\ \alpha \cdot (\beta \cdot \gamma) \approx (\alpha \cdot \beta) \cdot \gamma & \alpha \cdot \beta \approx \beta \cdot \alpha \end{array}$$

and by R the set of ring axioms. The set $R \cup AC \cup H$ is an equational specification of polynomial monoid rings over commutative rings. We next present a rewrite system that defines unique normal forms for ground terms of sort *Poly*.

[2] Thus, as special cases we have (i) the empty constraint and (ii) constraints consisting of a single simple constraint.

[3] Conjunctions of simple constrained are usually written as sequences.

Let R be a rewrite system and A be a set of equations. By $R\backslash A$ we denote the rewrite system consisting of all rules $u \to v$ such that $u \leftrightarrow_A^* u'\sigma$ and $v = v'\sigma$, for some rule $u' \to v'$ in R and some substitution σ. In other words, rewriting by $R\backslash A$ corresponds to rewriting by R via A-matching. We say that R is *convergent modulo* A if (i) R is terminating modulo A and (ii) for all terms u and v with $u \leftrightarrow_{R\cup A}^* v$, there exist terms u' and v' such that $u \to_{R\backslash A}^* u' \leftrightarrow_A^* v' \leftarrow_{R\backslash A}^* v$. We speak of *ground convergence* if condition (ii) is satisfied for all ground terms.

We will consider rewriting with AC-matching, where AC is the above set of associativity and commutativity axioms. In this case, part of condition (ii) above can be satisfied by the inclusion of so-called *extended rules* (cf. Peterson and Stickel [15]). If $f(e, e') \to e''$ is a rule, where f is an AC-symbol, then the corresponding extended rule is of the form $f(f(e, e'), v) \to f(e'', v)$, where v is a new variable (of suitable type). (Extensions of constrained rules are defined in the same way.) For any rewrite sytem R we denote by R^e the set R plus all corresponding extended rules. The key property of extended rules is that whenever an expression e is AC-reducible by R^e and $e \leftrightarrow_{AC}^* e'$, then e' is also AC-reducible by R^e.

Let P be the (constrained) rewrite system

$$
\begin{aligned}
\alpha \cdot I &\to \alpha && && (1)\\
M(x,\nu) &\to \Omega && \texttt{if } x = 0 && (2)\\
\Omega \oplus \alpha &\to \alpha && && (3)\\
\Omega \otimes \alpha &\to \Omega && && (4)\\
M(x,\nu) \oplus M(y,\nu) &\to M(z,\nu) && \texttt{if } z = x + y && (5)\\
M(x,\mu) \otimes M(y,\nu) &\to M(z,\mu\cdot\nu) && \texttt{if } z = x \cdot y && (6)\\
\alpha \otimes (\beta \oplus \gamma) &\to (\alpha \otimes \beta) \oplus (\alpha \otimes \gamma) && && (7)
\end{aligned}
$$

Then P^e consists of P plus the extended versions of all rules in P except (2). For instance, the extensions of rules (5) and (6) are

$$
\begin{aligned}
M(x,\nu) \oplus M(y,\nu) \oplus \alpha &\to M(z,\nu) \oplus \alpha && \texttt{if } z = x + y\\
M(x,\mu) \otimes M(y,\nu) \otimes \alpha &\to M(z,\mu\cdot\nu) \otimes \alpha && \texttt{if } z = x \cdot y
\end{aligned}
$$

(The extensions of rules (1), (3), and (4) are actually not needed, as these rules may serve as their own extensions, so to say.)

Proposition 1. *The rewrite system $P^e\backslash AC$ is ground convergent modulo $B \cup AC$ and the ground equational theories induced by $B \cup P \cup AC$ and $B \cup R \cup H \cup AC$ are the same.*

Termination of P^e can be established by showing that all rules are decreasing in a suitable associative path ordering (Bachmair and Plaisted [3]). Any precedence $\succeq$ will work in which (i) $a \sim b$, for all coefficients a and b and (ii) $I \succ \otimes \succ M \succ \oplus \sim \cdot \succ \Omega$ and all constants $X_1, \ldots, X_n$ precede I.[4] We denote this associative path ordering by $\succ_{apo}$ or simply $\succ$.

[4] Any precedence on the indeterminates will do. If all indeterminates are distinguished, we obtain an extension of a "lexicographic ordering," whereas an extension of the "total degree ordering" is obtained by declaring all indeterminates to be equivalent.

Proposition 2. *The ordering $\succ$ is a $(AC \cup B)$-compatible reduction ordering that contains all (ground instances of) rewrite rules in P.*

Ground terms (of sort *Poly*) in normal form correspond to (and will be called) *polynomials*. They can be written as sums of monomials $\sum_j a_j t_j$.

3 Polynomial Rules and Equations

By a *standardized (constrained) equation* we mean an equation of the form

$$\sum_i x_i t_i \approx 0 \;\; \texttt{if} \;\; \mathbf{x} = \mathbf{A} \cdot \mathbf{y}$$

where $\mathbf{A}$ is a matrix of ground coefficient terms and the coefficient variables in $\mathbf{x}$ and $\mathbf{y}$ are all distinct. In other words, the coefficient variables x_i in a standardized equation are all constrained by (and only by) linear equations.

Polynomial equations can easily be converted to standardized form. For example, take the equation $4X^2Y^3 \approx -7X^4Y$ between polynomials in one indeterminate Y (with polynomials in X as coefficients). An equivalent equation is $4X^2Y^3 \oplus 7X^4Y \approx 0$, which we transform to

$$x_1Y^3 \oplus x_2Y \approx 0 \;\; \texttt{if} \;\; x_1 = 4X^2, \; x_2 = 7X^4$$

by variable abstraction. The latter equation can be generalized to

$$x_1Y^3 \oplus x_2Y \approx 0 \;\; \texttt{if} \;\; x_1 = 4X^2y, \; x_2 = 7X^4y.$$

This constrained rule obviously represents more (unconstrained) ground instances than the original equation (because of the additional variable y), but all of these instances can be derived from the original equation and the general axioms of the polynomial ring.

By a (constrained) *polynomial rule* we mean an equation of the form $x_0t_0 \to \sum_i x_i t_i$ $\texttt{if}$ C, where $x_0t_0 \succ x_it_i$, for all i. Such a rule is said to be in *standardized form* if it can be written as

$$x_0t_0 \to \sum_i x_i t_i \;\; \texttt{if} \;\; x_0 \neq 0, \; x_0 = \mathbf{a} \cdot \mathbf{y}, \; \mathbf{x} = \mathbf{A} \cdot \mathbf{y}$$

where $\mathbf{a}$ is a vector of ground coefficient terms, $\mathbf{A}$ a matrix of ground coefficient terms, and the variables $x_0, x_1, \ldots, x_n$ and $y_1, \ldots, y_k$ are all distinct. We also call x_0 the *leading coefficient* and x_0t_0 the *leading monomial* of the rule.

Suppose we want to apply such a rewrite rule to a monomial ct_0. This is possible if x_0 can be "matched" with c, i.e., if the constraint $c = \mathbf{a} \cdot \mathbf{y}$ is solvable, which is the case precisely if c is an element of the ideal $Id(a_1, \ldots, a_k)$ generated by the elements of $\mathbf{a} = (a_1, \ldots, a_k)$. Therefore, we call the constraint $x_0 = \mathbf{a} \cdot \mathbf{y}$ the *ideal membership constraint* of the polynomial rule, and sometimes write $x_0 \in Id(\mathbf{a})$. Once a solution for the ideal membership constraint has been obtained, the variables in $\mathbf{y}$ get instantiated, so that the remaining equality constraints can be trivially satisfied by instantiating the variables in $\mathbf{x}$ appropriately. The negated constraint $x_0 \neq 0$ ensures that monomials with coefficient zero can not be rewritten.

Definition 3. Let E be a set of ground polynomial equations. A set of polynomial rules D is called a *D-basis* (for E) if for every ground polynomial expression p such that $p \leftrightarrow^*_{B \cup R \cup H \cup AC \cup E} 0$ we have $p \rightarrow^*_{(P^e \cup D) \setminus AC} 0$.

Computations on Coefficients

Any method for constructing D-bases presupposes certain assumptions about the given coefficient domain. We assume that the coefficient domain $\mathcal{B}$ is a commutative ring with unit that satisfies the following conditions:

- B1 Decidability and solvability of ideal membership) Given coefficients a and $a_1, \cdots, a_k$, it is decidable whether $a \in Id(a_1, \cdots, a_k)$ and, if so, elements $b_1, \cdots, b_k$ can be computed such that $a = \sum_{i=1}^{k} b_i a_i$.
- B2 The intersection of any two finitely generated ideals in $\mathcal{B}$ is finitely generated.
- B3 (Computation of intersection ideals) Given two finitely generated ideals I_1 and I_2, the generators of $I_1 \cap I_2$ can be computed.
- B4 The ideal $\{x \in \mathcal{B} | ax = 0\}$ is finitely generated, for all $a \in \mathcal{B}$.
- B5 (Computation of zero divisors) Given $a \in \mathcal{B}$, the generators of the ideal $\{x | ax = 0\}$ can be computed.

Conditions (B2)-(B5) can also be replaced by the requirement that syzygies be effectively computable, see the discussion on constraint manipulation below.

We have seen that condition (B1) is needed for the matching part of rewriting with polynomial rules. The computation of intersections of ideals will be needed when left-hand sides of rules have to be unified during critical pair computation. Finally, the condition about zero divisors enables us to "split off" certain parts of equations so as to obtain a suitable notion of rewriting.

4 Polynomial Completion

We use a completion approach for constructing D-bases. We follow the methodology of Bachmair and Dershowitz [4, 1]. and describe completion by transition rules operating on sets of (polynomial) equations and rules. Each set of equations E defines a set of (ground) proofs. Transition rules are designed to enable proof transformations that eventually yield certain normal-form proofs. More specifically, we have to transform ground proofs $p \leftrightarrow^*_{E \cup B \cup P \cup AC} 0$ to *rewrite proofs* of the form $p \rightarrow^*_{E' \cup B \cup P \cup AC} 0$. (The transition rules are also sound, in that the underlying equational theory does not change.)

We classify transition rules into four groups: (i) the *orientation rule* turns equations into directed rewrite rules; (ii) *deduction rules* eliminate subproofs that are not allowed in normal-form proofs; (iii) *constraint manipulation rules* are needed to standardize equations and to simplify constraints; and (iv) *simplification rules* induce additional proof transformations that are critical for the efficiency and termination of the completion process.

Orientation

The transformation of unoriented equations into directed rewrite rules is one of the key steps in completion. In the context of polynomial equations the idea is simply to shift the leading monomial to the left-hand side of a rule and the remaining monomials to the right-hand side. The presence of zero divisors causes technical complications, which we handle via constraints. More specifically, we use negated constraints to ensure that all ground instances of a rule have a non-zero leading coefficient, and "split off" those instances for which the leading coefficient is zero.

$$\textbf{Orientation:} \quad \frac{N \cup \{x_0 t_0 \oplus \sum x_i t_i \approx 0 \;\; \texttt{if} \;\; x_0 = \mathbf{a} \cdot \mathbf{y} \wedge \mathbf{x} = \mathbf{A} \cdot \mathbf{y}\}}{N \cup \{x_0 t_0 \to \sum x_i t_i \;\; \texttt{if} \;\; C_1\} \cup \{\sum x_i t_i \approx 0 \;\; \texttt{if} \;\; C_2\}}$$

where $x_0 t_0 \succ \sum x_i t_i$, C_1 is the constraint

$$x_0 \neq 0, \; x_0 = \mathbf{a} \cdot \mathbf{y}, \; \mathbf{x} = -\mathbf{A} \cdot \mathbf{y},$$

C_2 is the constraint

$$\mathbf{a} \cdot \mathbf{y} = 0, \; \mathbf{x} = \mathbf{A} \cdot \mathbf{y}$$

and $-\mathbf{A}$ denotes the matrix obtained from $\mathbf{A}$ by replacing each entry $a_{i,j}$ by $-a_{i,j}$. (One may also use a more conventional orientation rule without splitting, and describe splitting by a separate transition rule.)

Deduction

The deduction rules form the core of completion. They ensure that undesirable subproofs can be eliminated. Since we are dealing with a special case of (constrained) associative-commutative completion, subproofs to be eliminated include "peaks" of the form $q \leftarrow_{\mathcal{R}} p \rightarrow_{\mathcal{R}\backslash AC} r$ and "cliffs" of the form $q \leftrightarrow_{AC} p \rightarrow_{\mathcal{R}\backslash AC} r$, where $\mathcal{R}$ is the set of rewrite rules (including P rules) under consideration. The standard way of getting rid of these proof patterns is by critical pair computation (in the case of peaks) and by extended rules (in the case of cliffs), cf. Bachmair and Dershowitz [5]. Roughly speaking, critical pairs are equations $p \approx q$ obtained from "minimal" peaks called "overlaps."

The (ground) convergent rewrite system P^e, for instance, contains the extensions of rules in P. We also have to add, for any polynomial rule

$$x_0 t_0 \to \sum x_i t_i \;\; \texttt{if} \;\; x_0 \neq 0, \; x_0 = \mathbf{a} \cdot \mathbf{y}, \; \mathbf{x} = \mathbf{A} \cdot \mathbf{y}$$

an *extended polynomial rule*

$$x_0(t_0 \nu) \to \sum x_i(t_i \nu) \;\; \texttt{if} \;\; x_0 \neq 0, \; x_0 = \mathbf{a} \cdot \mathbf{y}, \; \mathbf{x} = \mathbf{A} \cdot \mathbf{y}$$

with an "extension variable" ν. If $\mathcal{R}$ is a set of polynomial rules, we denote by $\mathcal{R}^e$ the set $\mathcal{R}$ plus all corresponding extended rules.

Furthermore, an overlap between a polynomial rule $x_0 t_0 \to \sum x_i t_i \;\; \texttt{if} \;\; C$ and rule (5) of P yields an *extended equation*

$$(x_0 + z_0) t_0 \approx z_0 t_0 \oplus \sum x_i t_i \;\; \texttt{if} \; C.$$

Ideally, these equations ought to be used as left-to-right rules, but according to our ordering they would be oriented in the wrong way, from right to left.[5] Extended rules and equations will not be explicitly deduced via transition rules, but instead are implicitly used in the deduction of critical pairs (see below).

In describing critical pairs, we will make use of some algebraic terminology. We say that s *properly divides* t if $s\nu$ AC-matches t, and that s *divides* t, written $s|t$, if either $s \leftrightarrow^*_{AC} t$ or s properly divides t. The *least common multiple* of s and t is a shortest power product divided by both s and t.

An analysis of peaks involving polynomial rules (and their extensions) shows that two kinds of critical pairs need to be computed.

Definition 4. Take the two polynomial rules

$$x_0 s_0 \to \sum_i x_i s_i \ \ \texttt{if} \ \ x_0 \neq 0,\ x_0 = \mathbf{a}\cdot\mathbf{z},\ \mathbf{x} = \mathbf{A}\cdot\mathbf{z}$$

$$y_0 t_0 \to \sum_j y_j t_j \ \ \texttt{if} \ \ y_0 \neq 0,\ y_0 = \mathbf{b}\cdot\mathbf{z}',\ \mathbf{y} = \mathbf{B}\cdot\mathbf{z}'$$

and let u be the least common multiple of the two power products s_0 and t_0, and s' and t' be power products such that $u \leftrightarrow^*_{AC} s_0 s' \leftrightarrow^*_{AC} t_0 t'$.[6] Then the equation

$$\sum_i x_i(s_i s') \oplus \sum_j y_j(t_j t') \approx 0 \quad \texttt{if} \ \ \mathbf{a}\cdot\mathbf{z} = \mathbf{b}\cdot\mathbf{z}',\ \mathbf{x} = \mathbf{A}\cdot\mathbf{z},\ \mathbf{y} = -\mathbf{B}\cdot\mathbf{z}'$$

is called an *intersection critical pair*; and, if $\mathit{Id}(\mathbf{a}) \not\subseteq \mathit{Id}(\mathbf{b})$ and $\mathit{Id}(\mathbf{b}) \not\subseteq \mathit{Id}(\mathbf{a})$,

$$x_0(s_0 s') \oplus \sum_i x_i(s_i s') \oplus \sum_j y_j(t_j t') \approx 0$$
$$\texttt{if} \ \ x_0 = \mathbf{a}\cdot\mathbf{z} + \mathbf{b}\cdot\mathbf{z}',\ \mathbf{x} = -\mathbf{A}\cdot\mathbf{z},\ \mathbf{y} = -\mathbf{B}\cdot\mathbf{z}'$$

is called a *union critical pair*.

We speak of an "intersection" critical pair because the condition on the leading coefficients of the two rules is that they belong to the ideal $\mathit{Id}(\mathbf{a}) \cap \mathit{Id}(\mathbf{b})$. The computation of the least common multiple of two power products formally corresponds to (a special case of) AC-unification. Strictly speaking, the intersection critical pair is obtained from the *extended rules* corresponding to the above two rules. The extension variables enable the computation of the AC-unifiers of power products. (Note that intersection critical pairs may have to be computed even if the two power products s_0 and t_0 share no common indeterminates.) Union critical pairs are obtained from overlaps between a (extended) polynomial rule and an *extended equation*.

[5] Technically, this is the main reason why we obtain D-bases rather than more general Gröbner bases. If one wants to obtain Gröbner bases, a possible remedy might be to modify the ordering by taking into account coefficients. The drawback would be that the existence of a suitable ordering imposes additional conditions on the coefficient domain.

[6] If $s' = I$, then $s_0 s'$ is understood to denote s_0; and similarly for $t_0 t'$.

The computation of critical pairs is achieved by a deductive transition rule:

Superposition: $$\frac{N}{N \cup \{\rho\}}$$

if ρ is either an intersection or a union critical pair (from rules in N).

Constraint Manipulation

The computation of critical pairs, as defined above, involves only syntactic constraint transformations, but results in non-standardized equations. Standardization essentially amounts to solving certain constraints, and is required before the orientation rule can be applied. In other words, our formalism allows for "lazy evaluation" of constraints. (While we have chosen to require constraint solving before orientation, other ways of constraint handling are conceivable.) Let us now briefly outline the constraint solving process.

First note that the combination of constraints $\mathbf{x} = \mathbf{A} \cdot \mathbf{z}$ and $\mathbf{y} = \mathbf{B} \cdot \mathbf{z'}$, can be transformed to $\mathbf{x} = \mathbf{A} \cdot \mathbf{z} + \mathbf{0} \cdot \mathbf{z'}$ and $\mathbf{y} = \mathbf{0} \cdot \mathbf{z} + \mathbf{B} \cdot \mathbf{z'}$, which use the same coefficient variables on the right hand side.

Now suppose we add a constraint $\mathbf{b} \cdot \mathbf{y} = 0$ to a standardized constraint $\mathbf{x} = \mathbf{A} \cdot \mathbf{y}$, where $\mathbf{y}$ contains the variables $y_1, \ldots, y_k$. To transform the extended constraint to standardized form we *change variables* as follows.

If $k = 1$ (for instance, if the leading coefficient is constrained by membership in a principal ideal), then we simply replace each constraint $x_i = a_{i,1} y_1$ in C by $x_i = \sum_{j=1}^{l} d_{i,j} z_j$, where $d_{i,j} = a_{i,1} c_j$ and the coefficients $c_1, \ldots, c_l$ are the generators of the ideal $\{x : b_1 x = 0\}$.[7]

If $k > 1$, then we first compute the intersection of the two ideals $I_1 = Id(a_1, \ldots, a_{k-1})$ and $I_2 = Id(a_k)$, say $I = I_1 \cap I_2 = Id(c_1, \ldots, c_l)$. Then, for each i with $1 \leq i \leq l$, there exist coefficients $d_{i,j}$, such that $c_i = a_1 d_{i,1} + \ldots + a_{k-1} d_{i,k-1} = a_k(-d_{i,k})$.[8] Replace in C each variable y_i by $d_{1i} z_1 + \ldots + d_{li} z_l$.

In this way we obtain for any non-standardized constraint C (that can be produced during completion) a standardized constraint $\Theta(C)$, such that $p \approx 0 \texttt{ if } C$ and $p \approx 0 \texttt{ if } \Theta(C)$ have the same (unconstrained) ground instances.

Constraint solving: $$\frac{N \cup \{p \approx 0 \;\; \texttt{if} \;\; C\}}{N \cup \{p \approx 0 \;\; \texttt{if} \;\; \Theta(C)\}}$$

In addition to constraint solving, there are various mechanisms for constraint simplification, including

- deletion of equations with unsolvable constraints,
- simplification of membership constraints, and
- deletion of redundant parts of constraints.

[7] The coefficients c_i, $1 \leq i \leq l$, are the generators of the syzygies of (b_1).

[8] The tuples $(d_{i,1}, \cdots, d_{i,k-1}, d_{i,k})$, $1 \leq i \leq l$ are the generators of the syzygies of $(a_1, \cdots, a_k)$.

For example, consider the two rules $3X \to 1$ and $2X \to 3$ in the polynomial ring $\mathcal{Z}[X]$ over the integers. In our notation, the rules are written as

$$x_0 X \to x_1 \;\; \texttt{if} \;\; x_0 \neq 0,\; x_0 = 3z,\; x_1 = z$$
$$y_0 X \to y_1 \;\; \texttt{if} \;\; y_0 \neq 0,\; y_0 = 2z,\; y_1 = 3z$$

The union critical pair of these two rules is

$$z_0 X \oplus z_1 \oplus z_2 \approx 0 \;\; \texttt{if} \;\; z_0 \neq 0,\; z_0 = 3z_1' + 2z_2',\; z_1 = -z_1',\; z_2 = -3z_2'.$$

Since the integers are a principal ideal domain we can express the ideal membership constraint in terms of a principal ideal. We have $Id(2,3) = Id(1)$ or, more specifically, $3z_1' + 2z_2' = 1$ if $z_1' = 1$ and $z_2' = -1$. Thus we substitute z' for z_1' and $-z'$ for z_2' and obtain a simplified constrained equation

$$z_0 X \oplus z_1 \oplus z_2 \approx 0 \;\; \texttt{if} \; z_0 \neq 0,\; z_0 = z',\; z_1 = -z',\; z_2 = 3z'.$$

Simplification

Mechanisms for simplifying equation and rules are crucial for the efficiency of completion. In the present case, the termination of completion also depends on them.

We say that a set $\mathcal{R}$ of polynomial rules is *left-reduced* if $\mathcal{R}$ contains no two rules

$$\rho_1 : \; x_0 s_0 \to \sum_i x_i s_i \;\; \texttt{if} \;\; x_0 \neq 0,\; x_0 = \mathbf{a} \cdot \mathbf{z},\; \mathbf{x} = \mathbf{A} \cdot \mathbf{z}$$

and

$$\rho_2 : \; y_0 t_0 \to \sum_j y_j t_j \;\; \texttt{if} \;\; y_0 \neq 0,\; y_0 = \mathbf{b} \cdot \mathbf{z'},\; \mathbf{y} = \mathbf{B} \cdot \mathbf{z'}$$

such that s_0 divides t_0 and $Id(\mathbf{b}) \subseteq Id(\mathbf{a})$.

If a set $\mathcal{R}$ is not left-reduced, then either it contains two rules with the "same" left-hand side, or else some rule can be "collapsed" to an equation, as expressed in the following transition rule.

$$\textbf{Collapse:} \quad \frac{N \cup \{\rho_1, \rho_2\}}{N \cup \{\rho_1\} \cup \{\sum_i x_i s_i t' \oplus \sum_j y_j t_j \approx 0 \;\texttt{if}\; C\}}$$

where the rules ρ_1 and ρ_2 are as defined above, C is the constraint $\mathbf{b}^{\mathbf{T}} = \mathbf{X} \cdot \mathbf{a}, \mathbf{x} = \mathbf{X}^{\mathbf{T}} \cdot \mathbf{z'}, \mathbf{y} = \mathbf{B} \cdot \mathbf{z'}$, where X is a matrix of (new) coefficient variables, and (i) $s_0 t' = t_0$ and (ii) either s_0 properly divides t_0 or else $Id(\mathbf{b})$ is a proper subset of $Id(\mathbf{a})$.

By $\mathbf{A}^{\mathbf{T}}$ we denote the transpose of a matrix $\mathbf{A}$. To standardize the constraint of the new equation one has to compute a coefficient matrix C such that $\mathbf{b}^{\mathbf{T}} = \mathbf{C} \cdot \mathbf{a}$. In other words, one has to solve several ideal membership constraints (which are solvable because $Id(\mathbf{b}) \subseteq Id(\mathbf{a})$). Other common simplification techniques include

- deletion of trivial equations,
- simplification of polynomial equations by rules in P,

– simplification of polynomial equations by polynomial rules, and
– simplification of right-hand sides of polynomial rules.

These can be formulated as transition rules in a similar way as the collapse rule. We leave out details, but emphasize that simplification of equations is necessary so that left-reduced rewrite systems can be produced.

Consider the two equations $XZ \approx 1$ and $YZ \approx 1$ between polynomials in the ring $(\mathcal{Z}[X,Y])[Z]$, where $\mathcal{Z}$ denotes the set of integers. Corresponding constrained equations are

$$x_0 Z \oplus x_1 \approx 0 \ \texttt{ if } \ x_0 = Xz_1,\ x_1 = -z_1$$
$$y_0 Z \oplus y_1 \approx 0 \ \texttt{ if } \ y_0 = Yz_1',\ y_1 = -z_1'$$

By orientation and simplification we get rules

$$x_0 Z \rightarrow x_1 \ \texttt{ if } \ x_0 \neq 0,\ x_0 = Xz_1,\ x_1 = z_1$$
$$y_0 Z \rightarrow y_1 \ \texttt{ if } \ y_0 \neq 0,\ y_0 = Yz_1',\ y_1 = z_1'$$

(The equations that are split off during orientation have unsatisfiable constraints, and hence can be deleted.) The intersection critical pair between these two rules is, $x_1 \oplus x_2 \approx 0 \texttt{ if } Xy_1 = Yy_2 \wedge x_1 = y_1 \wedge x_2 = -y_2$. Using the fact that $Id(X) \cap Id(Y) = Id(XY)$ we get a standardized equation

$$x_1 \oplus x_2 \approx 0 \ \texttt{ if } \ x_1 = Yy_1,\ x_2 = -Xy_1$$

which can be simplified to $x_0 \approx 0 \texttt{ if } x_0 = (Y - X)y_1$. Orientation yields a rule

$$x_0 \rightarrow 0 \texttt{ if } x_0 \neq 0,\ x_0 = (Y-X)y_1.$$

The union critical pair corresponding to the first two rules is

$$x_0 Z \oplus x_1 \approx 0 \ \texttt{ if } \ x_0 = Xy_1 + Yy_2 \wedge x_1 = -y_1 - y_2.$$

Orienting this equation yields a rule with which the original rules can be collapsed. Any equations that remain at that point can be simplified to trivial equations and deleted. We end up with a D-basis of two rules:

$$x_0 \rightarrow 0 \ \texttt{ if } \ x_0 \neq 0,\ x_0 = (Y-X)y_1$$
$$x_0 Z \rightarrow x_1 \ \texttt{ if } \ x_0 \neq 0,\ x_0 = Xy_1 + Yy_2 \wedge x_1 = y_1 + y_2$$

Correctness

A *derivation* is a (possibly infinite) sequence of transition steps $N_0 \vdash N_1 \vdash N_2 \vdash \cdots$ where N_0 is a set of polynomial equations in standardized form. The *limit* N_∞ of the derivation is the set $\bigcup_i \bigcap_{j \geq i} N_j$ of all *persisting* equations and rules.

Definition 5. A derivation is said to be *fair* if (i) N_∞ is a left-reduced set of polynomial rules, and (ii) $\bigcup_i N_i$ contains all intersection and union critical pairs between (extended) rules in N_∞.

The following technical lemma will be used in the proofs of termination and correctness.

Lemma 6. *If N_∞ is the limit of a fair derivation and $x_0t_0 \oplus \sum x_it_i \approx 0$ is an equation in $\bigcup_i N_i$ such that $x_0t_0 \succ x_it_i$, for all i, and x_0 is constrained by an ideal membership constraint $x_0 \in I$, then there is a rule in N_∞ with left-hand side y_0u_0 and ideal membership constraint $y_0 \in I'$, such that u_0 divides t_0 and and $I \subseteq I'$.*

Intuitively, the lemma states that all leading monomials of polynomial equations can be reduced by persisting rules. This is a consequence of part (i) of the fairness conditions. (Polynomial equations do not persist. Thus the leading monomial is either reduced at some point, or else it will become the left-hand side of a rule when the equation is oriented.)

Theorem 7 (Termination). *Suppose the coefficient domain $\mathcal{B}$ is Noetherian and satisfies conditions (B1)-(B5). Then the limit N_∞ of any fair derivation is a finite set of polynomial rules.*

Proof. (By contradiction) Suppose N_∞ contains infinitely many polynomial rules. Let x_nt_n be the corresponding left-hand sides, for $n \in \mathbf{N}$, where x_n is constrained by an ideal membership constraint $x_n \in I_n$. Since the divisibility relation on power products is a Dickson partial order, there exists an infinite subsequence $t_{n_1}, t_{n_2}, \ldots$ such that t_{n_i} properly divides t_{n_j} for all $i < j$ (cf. [6], pp. 162 & 189). Since N_∞ is left-reduced, we have $I_{n_{i+1}} \not\subseteq I_{n_i}$, for all i. Suppose we have $I_{n_j} \not\subseteq I_{n_{j+1}}$, for some j. Then there exists a union critical pair with maximal monomial $xt_{n_{j+1}}$, where x is constrained by $x \in I_{n_j} \cup I_{n_{j+1}}$. By Lemma 6, the monomial $xt_{n_{j+1}}$ is reducible by some rule in N_∞, which contradicts the assumption that N_∞ is left-reduced. We conclude that $I_{n_j} \subset I_{n_{j+1}}$, for all j, which contradicts the assumption that the coefficient domain is Noetherian. □

Theorem 8 (Correctness). *Suppose the coefficient domain $\mathcal{B}$ is Noetherian and satisfies conditions (B1)-(B5). If N_∞ is the limit of a fair derivation, then N_∞^e is a D-basis.*

This theorem can be proved using proof simplification techniques for associative-commutative completion (and similar proof orderings), as described by Bachmair and Dershowitz [5, 4]. There is one important difference, though. We do not deduce *extended equations* and consequently are not able to eliminate corresponding peaks of the form $q \leftarrow_{P\setminus AC} p \rightarrow_{\mathcal{R}} r$. However, it can be shown that in (ground) proofs of $p' \approx 0$ (which is all we need to consider) such a peak only needs to be eliminated if it is part of a subproof $q \leftarrow_{P\setminus AC} p \rightarrow_{\mathcal{R}} r \rightarrow_{P\cup\mathcal{R}} r'$. We eliminate the latter proofs using union critical pairs (which derive from extended equations, of course). We thus normalize (ground) proofs of $p' \approx 0$, but not proofs of arbitrary polynomial equations $p' \approx p''$, and therefore obtain a D-basis, and not a Gröbner basis.

As a corollary, we also obtain the following result.

Theorem 9. *If $\mathcal{B}$ is a commutative Noetherian ring with unit that satisfies the conditions $(B1) - (B5)$, then the same is true for the polynomial ring $\mathcal{B}[X]$.*

A similar result was proved by Richman [16].

5 Summary

Algorithms for constructing polynomial ideal bases have been extensively studied, cf. Becker and Weispfenning [6]. Our work generalizes previous results for commutative polynomials in that we consider a larger class of coefficient domains (including domains with zero divisors). The basic conditions we impose on a coefficient domain are (i) decidability of ideal membership (for finitely generated ideals) and (ii) computability of syzygy generators. In addition, termination of the completion process is guaranteed only if the coefficient domain is Noetherian. If one has a D-basis for finitely generated ideals in $\mathcal{B}[X]$, then ideal membership in $\mathcal{B}$ is decidable and syzygy generators are computable for any finite set S of elements in $\mathcal{B}$. This suggests that our assumptions are in a sense minimal ones needed to carry out a D-basis computation.

We believe that most of the rewrite-based methods for constructing ideal bases for commutative polynomial rings can be described within our formalism. In particular, this is the case for Buchberger's original algorithm and for the method proposed by Pan [14] for polynomials over principal ideal domains. Buchberger's original algorithm works for coefficient domains that are fields, in which case we also obtain Gröbner bases (and not just D-bases), because extended equations and union critical pairs become irrelevant. We conjecture, that our method can be shown to produce a Gröbner basis whenever a partial order on the coefficient domain is given, such that for any ideal I, every remainder class of I has a unique minimal element. It remains to be investigated to what extent our techniques can be carried over to non-commutative polynomials.

There have also been various attempts at characterizing the class of rings that admit a Gröbner basis algorithm. These characterizations typically involve as one key parameter a (axiomatically characterized) reduction relation; see Buchberger [8] and Stifter [17] for one notion of "reduction rings" and Madlener and Reinert [13] for another. We have taken a different approach that employs only standard associative-commutative rewriting (with constraints). We believe our results indicate that in the case of commutative polynomial rings standard rewriting is sufficient. Non-standard reduction relations have been used, for instance, by Trinks [18]. There several rewrite rules may be simultaneously applied to a single monomial. We actually obtain a similar effect, to some extent, via constraints. For example, a D-basis for the two polynomial equations $2X = 0$ and $3X = 0$ over the $\mathcal{Z}$ consists of just one rule $xX \to 0$ `if` $x \neq 0,\ x = 2y_1 + 3y_2$ (which is obtained via a union critical pair and renders the other two rules redundant). From a computational point of view (and probably also conceptually) constrained rewriting appears to be preferable to arbitrary "simultaneous reduction" by several rules. Other features of constraints, such as the possibility of delaying constraint solving, might also be of interest for implementation purposes.

Acknowledgements. We would like to thank Deepak Kapur and Klaus Madlener for fruitful discussions on the topic of this paper and the anonymous reviewers for their helpful comments.

References

1. L. Bachmair and N. Dershowitz. Equational inference, canonical proofs, and proof orderings. *J. of the Association for Computing Machinery*, 41:236–276, 1994.
2. L. Bachmair and H. Ganzinger. Buchberger's algorithm: a constraint-based completion procedure. In *1st Int. Conference on Constraints in Computational Logics*, volume 845 of *Lecture Notes in Computer Science*, pages 285–301. Springer-Verlag, 1994.
3. L. Bachmair and D. Plaisted. Termination orderings for associative-commutative rewriting systems. *J. Symbolic Computation*, 1:329–349, 1985.
4. Leo Bachmair. *Canonical Equational Proofs*. Birkhäuser, Boston, 1991.
5. Leo Bachmair and Nachum Dershowitz. Completion for rewriting modulo a congruence. *Theoretical Computer Science*, 67(2 & 3):173–201, October 1989.
6. T. Becker and V. Weispfenning. *Gröbner bases: a computational approach to commutative algebra*. Springer-Verlag, Berlin, 1993.
7. B. Buchberger. *An algorithm for finding a basis for the residue class ring of a zero-dimensional ideal*. PhD thesis, University of Innsbruck, Austria, 1965.
8. B. Buchberger. A critical-pair completion algorithm for finitely generated ideals in rings. In *Proc. Logic and Machines: Decision Problems and Complexity*, volume 171 of *Lect. Notes in Comput. Sci.*, pages 137–161. Springer-Verlag, Berlin, 1983.
9. B. Buchberger. Gröbner bases: An algorithmic method in polynomial ideal theory. In N. K. Bose, editor, *Recent Trends in Multidimensional Systems Theory*, pages 184–232. Reidel, 1985.
10. R. Bündgen. Buchberger's algorithm: The term rewriter's point of view. *Theoretical Computer Science*, 1996.
11. A. Kandri-Rody and D. Kapur. Computing a Gröbner basis of a polynomial ideal over a Euclidean domain. *J. Symbolic Computation*, 6:37–57, 1988.
12. C. Kirchner, H. Kirchner, and M. Rusinowitch. Deduction with symbolic constraints. Revue Française d'Intelligence Artificielle, 4:9–52, 1990.
13. K. Madlener and B. Reinert. Computing Gröbner bases in monoid and group rings. In *Proc. ISSAC 93*, pages 254–263. 1993.
14. L. Pan. On the D-bases of polynomial ideals over principal ideal domains. *J. Symbolic Computation*, 7:55–69, 1988.
15. G. E. Peterson and M. E. Stickel. Complete sets of reductions for some equational theories. *J. of the Association for Computing Machinery*, 28:233–264, 1981.
16. F. Richman. Constructive aspects of Noetherian rings. *Proc. of the American Mathematical Society*, 44(2):436–441, 1974.
17. S. Stifter. A generalization of reduction rings. *J. Symbolic Computation*, 4:351–364, 1987.
18. W. Trinks. On B. Buchberger's method for solving algebraic equations. *J. Number Theory*, 10(4):475–488, 1978. (German).