Buchberger's algorithm: A constraint-based completion procedure*

Leo Bachmair[1] and Harald Ganzinger[2]

[1] Department of Computer Science, SUNY at Stony Brook,
Stony Brook, NY 11794, U.S.A,
leo@sbcs.sunysb.edu
[2] Max-Planck-Institut für Informatik,
Im Stadtwald, D-66123 Saarbrücken, Germany,
hg@mpi-sb.mpg.de

Abstract. We present an extended completion procedure with built-in theories defined by a collection of associativity and commutativity axioms and additional ground equations, and reformulate Buchberger's algorithm for constructing Gröbner bases for polynomial ideals in this formalism. The presentation of completion is at an abstract level, by transition rules, with a suitable notion of fairness used to characterize a wide class of correct completion procedures, among them Buchberger's original algorithm for polynomial rings over a field.

1 Introduction

Buchberger's algorithm is one of the central methods in computational algebra (Buchberger 1965, Buchberger 1985). It constructs a canonical simplifier for ideals in polynomial rings over a field, and hence provides a basis for computational solutions to various problems in polynomial ideal theory. The representation of canonical simplifiers is by polynomial rewrite systems that define a well-founded reduction relation. In other words, the computational paradigm underlying the algorithm is reduction to normal form.

The concept of computation by normalization is widely used in computing applications, e.g., in the many variants of the Knuth-Bendix completion procedure. The similarity between Buchberger's algorithm and completion appears to have first been observed by Loos (1981) and Buchberger and Loos (1982). Since then a number of researchers have attempted to clarify, and formalize, the connection between the two methods; see Buchberger (1987) for a discussion of the fundamental issues and further references. The typical approach has been to generalize rewrite-based procedures so that both Buchberger's algorithm and

* The research described in this paper was supported in part by the German Science Foundation (Deutsche Forschungsgemeinschaft) under grant Ga 261/4-1, by the German Ministry for Research and Technology (Bundesministerium für Forschung und Technologie) under grant ITS 9102/ITS 9103, by the ESPRIT Basic Research Working Group 6028 (Construction of Computational Logics), and by the National Science Foundation under grant INT-9314412. The first author was also supported by the Alexander von Humboldt Foundation.

(certain variants of) completion can be derived as special cases. Most of these results apply only to variants of Buchberger's algorithm for polynomials over a commutative ring, though, and not to the original algorithm with a field as the coefficient domain. Only recently has Bündgen (1991) shown how to simulate polynomial computations over *finite* fields by associative-commutative completion.

One key observation that serves as our starting point is that Buchberger's algorithm is based not only on rewriting, but employs *two* different modes of computation. While rewriting indeed provides the fundamental mechanism for polynomial computation, there are also operations on coefficients the realization of which depends on the given coefficient domain, but typically does not involve any form of rewriting. Thus we believe that attempts aimed at describing Buchberger's algorithm exclusively by rewrite operations are misguided. The appropriate concept for modelling domain-dependent computation are hierarchical theories and *constraints*. Constraint techniques have been developed in logic programming, but can also be integrated into rewrite methods. In automated theorem proving, the replacement of syntactic unification by E-unification is a standard method for dealing with equational constraints over certain equational theories. Refutational theorem proving for hierarchical first-order theories with equality and arbitrary base theories was studied by Bachmair, Ganzinger and Waldmann (1994) and a method based on superposition with abstraction has been developed. In this paper we show that Buchberger's algorithm is a specific instance of the latter method and hence can be viewed as a completion procedure for hierarchical equational theories.

This reformulation requires a representation of polynomials as first-order terms, which we outline in Section 3. We use a two-sorted (hierarchical) signature (with sorts for coefficients and polynomials). Ideals of rings are in one-to-one correspondence with kernels of surjective ring homomorphisms which are in turn in one-to-one correspondence with equational theories over the ring elements (in the case of polynomial rings the indeterminates and the coefficients) that contain the axioms of a commutative ring. The problem we consider is the word problem for finitely presented such theories, i.e., theories that are defined by the union of (i) finitely many equations between polynomials, (ii) the axioms of a commutative ring, (iii) the equations defining equality of coefficient ground terms, and (iv) the axioms which specify the embedding of the coefficients into the polynomials. The method is to apply associative-commutative (AC) completion to such systems in a way such that termination is uniformly guaranteed.

As a first step in that direction we present in section 3 a convergent rewrite system (modulo AC) for commutative rings plus the axioms for embedding the coefficients. Polynomials correspond to ground normal forms defined by this rewrite system.

A second subproblem is how to deal with the usually infinitely many equations between coefficient ground terms. Let B denote this ground theory. We represent Buchberger's algorithm as an (extended) completion procedure with a built-in theory $AC \cup B$. Deduction requires AC-unification as well as similar unification operations on coefficients that can conveniently be described by constraints over the base theory. (We assume the coefficient theory to be recursive.) We show that in the case of fields rewrite rules can be suitably transformed so that deduction requires no coefficient operations, but only associative-commutative unification. In this case one may therefore dispense with constraints altogether.

Like any other decision method for logical theories, Buchberger's algorithm is essentially a *simplification procedure* on (sets of) formulas. The central simplification step apart from normal-form reduction is the *isolation of maximal monomials*, i.e., the transformation of a polynomial equation $p \approx q$ into an equivalent equation $M \approx r$ where M is the maximal monomial in (the normal form) of $p - q$ and where r is the normal form of $q - p + M$. In Section 2 we describe completion by a set of transition rules in a way that abstracts from specific notions of critical pairs and simplifications. We specify corresponding fairness conditions which a procedure based on these transition rules has to satisfy in order to be correct (i.e., compute a convergent rewrite system). A detailed exposition of this approach is contained in Bachmair and Dershowitz (1994). In Section 4 we describe the specific instances of completion that correspond to the various versions of Buchberger's algorithm. We demonstrate that the eager application of certain simplification transitions renders most critical pairs redundant, so that termination of the completion process is guaranteed. An interesting technical point is that we provide as the basis for our completeness proof an AC-compatible reduction ordering that is *total* on ground terms. Therefore this reasoning about redundancy of certain equations and critical pairs can be carried over to the context of refutational theorem proving for arbitrary first-order theories over commutative rings via the concept of saturation by superposition up to redundancy as developed in Bachmair and Ganzinger (1994). This extends, for instance, results by Marché (1994) to unfailing completion and first-order theorem proving.

In summary, we show how to view Buchberger's algorithm as an instance of associative-commutative completion for a hierarchical equational theory. Constraints prove useful for representing and abstracting from the computations in the underlying base theory of coefficients. In the case of fields, rewrite rules can be standardized, so that deduction requires no manipulation of coefficients. In the general case the necessary operations on coefficients can be described by constraints. Previous attempts at modelling Buchberger's algorithm failed to deal with fields, whereas in our setting this turns out to be the simplest case. We take this as an indication that our approach better captures the essence of Buchberger's original algorithm and its variants. Furthermore, our results allow us to incorporate the fundamental techniques employed by Buchberger's algorithm into the more general context of first-order theorem proving.

We assume that the reader is familiar with the theory of term rewriting. In our terminology we follow Dershowitz and Jouannaud (1990). The necessary theoretical background about completion can be found in Bachmair (1991). The integration of constraints is fairly straightforward and fits into the framework proposed by Kirchner, Kirchner and Rusinowitch (1990) for constraint-based completion.

2 Extended completion

Completion is a deductive method for reasoning about equational theories. The variant that is relevant for applications to polynomials is completion with a built-in theory, or *extended completion*. We outline the basic ideas underlying this method and refer to Bachmair (1991) for further details.

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 $R\backslash A$ (with standard matching) corresponds to rewriting by R with A-matching. We say that R is *convergent modulo* A if (i) R is terminating modulo A and (ii) R is confluent modulo A in the sense that 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.

Extended completion attempts to construct, given sets of equations E and A, a rewrite system R, such that $R\backslash A$ is ground convergent modulo A and the congruence relation induced by $E \cup A$ and $R \cup A$ on (ground) terms are the same. A well-founded reduction ordering $\succ$ compatible with A is supplied and used to determine in which direction an equation is to be oriented into a one-way rewrite rule. The rewrite systems constructed by extended completion are therefore terminating modulo A.

Let us analyze the confluence property. An (*equational*) *proof* is a finite sequence of equational replacement steps. A proof of the form $u \to_{R\backslash A}^* u' \leftrightarrow_A^* v' \leftarrow_{R\backslash A}^* v$ is called a *rewrite proof modulo* A (by $R\backslash A$). Obviously, a rewrite system is ground confluent if every ground equation that is provable at all, is also provable by a rewrite proof. A proof in $R \cup A$, on the other hand, is a rewrite proof modulo A if and only if it contains neither a *cliff* (i.e., a proof $s \leftrightarrow_A t \to_{R\backslash A} u$ or the inverse thereof) nor a *peak* (i.e., a proof $s \leftarrow_R t \to_{R\backslash A} u$ or the inverse thereof). In completion new equations are deduced with which cliffs and peaks can be eliminated, so that any proof is eventually simplified to a rewrite proof.

The elimination of cliffs is straightforward for many equational theories A, including those relevant for application to polynomials. Suppose $u \approx v$ is an equation in A and $l \to r$ is a rule in R, such that l is A-unifiable with some proper non-variable subterm s of u. Then $u[l] \to u[r]$ is called an *extended rule* of $l \to r$. (Here $u[l]$ and $u[r]$ are obtained by replacing s in u by l and r, respectively.) For example, if A contains the associativity axiom $(x \oplus y) \oplus z \approx x \oplus (y \oplus z)$, then the rule $p \oplus q \to r$ can be extended to $(p \oplus q) \oplus z \to r \oplus z$, where z is a new variable, also called an *extension variable*. By R^e we denote the set R plus all its extended rules. Extended rules were introduced by Peterson and Stickel (1981) and generalized by Jouannaud and Kirchner (1986).

Elimination of peaks can be achieved by computation of critical pairs modulo A. By $CP_A(R)$ we denote the set of all critical pairs modulo A between rules in R. Completion can now be described in terms of transition (or inference) rules that operate on sets of equations and rules, cf. Bachmair (1991), Bachmair and Dershowitz (1994).

Deduction: $\quad N \vdash N \cup \{s \approx t\}$

if every ground instance of $s \approx t$ is an equational consequence of $N \cup A$.

In addition, completion employs a variety of simplification techniques based on elimination of redundant formulas. We formalize this notion of redundancy in terms of *proof simplification* and a suitable well-founded ordering on equational proofs.

We distinguish between (unoriented) equations and (oriented) rules, so that each set N can be written as $E \cup R$. We define a complexity measure for proofs in $A \cup E \cup R^e$. Let $t = t[u\sigma]_p \leftrightarrow t[v\sigma]_p = t'$ be an equational replacement step with an equation $u \approx v$ applied to the subterm of t at position p. The complexity of the proof step is the tuple $(\{t, t'\}, \bot, \bot, \bot)$, if $u \approx v$ is in E; it is $(\{t\}, \{u\sigma\}, \top, \{t'\})$ if $u \to v$ is in R, and $(\{t'\}, \{v\sigma\}, \top, \{t\})$ if $v \to u$ is in R; it is $(\{t\}, \bot, |t| - |u\sigma|, \{t'\})$ if $u \to v$ is in $R^e \setminus R$, and $(\{t'\}, \bot, |t'| - |v\sigma|, \{t\})$ if $v \to u$ is in $R^e \setminus R$; and it is $(\{t\}, \bot, \bot, \bot)$ if $u \approx v$ is in A. (Here $|t|$ denotes the size of a term t.) Such tuples are compared using the multiset extension of the given reduction ordering in the first, second, and last component; and the greater-than relation on the natural numbers in the third component. (The constants $\top$ and $\bot$ denote additional maximum and minimum elements.) The complexity of a proof is the multiset of complexities of its proof steps. The multiset extension of the ordering on tuples yields a proof ordering, denoted by the symbol $\succ$.

Deletion: $\quad N \cup \{s \approx t\} \vdash N$

if the rule or equation $s \approx t$ is *redundant* in N, that is, for every ground instance $s\sigma \approx t\sigma$ there exists an equational proof Π (of $s\sigma \approx t\sigma$) in $N \cup A$, such that $(s\sigma \leftrightarrow t\sigma) \succ \Pi$.

A derived inference of the form $N \cup \{s \approx t\} \vdash N \cup \{u \approx v, s \approx t\} \vdash N \cup \{u \approx v\}$ in which the newly deduced equation or rule $u \approx v$ makes $s \approx t$ become redundant represents a *simplification* of $s \approx t$ to $u \approx v$.

In Section 4 we apply extended completion to polynomials and present a collection of derived inference rules based on deduction and deletion.

A *derivation* is a (possibly infinite) sequence of transition steps $N_0 \vdash N_1 \vdash N_2 \vdash \cdots$. The *limit* N_∞ of the derivation is the set $\bigcup_i \bigcap_{j \geq i} N_j$ of all *persisting* equations. By an *non-failing derivation* we mean one in which the limit N_∞ contains no unoriented equations. We call a derivation *fair* if for every ground proof Π in $N_\infty^e \cup A$ that is a cliff or peak, there exists a proof Σ (of the same equation) in $\bigcup_i N_i$, such that $\Pi \succ \Sigma$.

Theorem 1. *If N_∞ is the limit of a fair, non-failing derivation, then $N_\infty^e \setminus A$ is ground convergent modulo A.*

We will demonstrate that Buchberger's algorithm is an extended completion procedure for combinations A of a set AC of associativity and commutativity axioms and a set B of ground equations in a hierarchical setting. B will be the base theory which is reflected syntactically in that polynomials are not unifiable with subterms of equations in B. Hence extensions of rules will result only from overlaps with the (two) associativity axioms.

Lemma 2. *Let N_∞ be the limit of an unfailing derivation by extended completion modulo $AC \cup B$. The derivation is fair if $CP_{AC \cup B}(N_\infty^e) \subseteq \bigcup_i N_i$.*

3 Polynomials

Buchberger's algorithm manipulates (multi-variate) polynomials, i.e., expressions typically written as sums $\sum_{1 \leq i \leq n} c_i \mathbf{x}_i$, where the c_i are elements from some coefficient domain and the $\mathbf{x}_i$ are products of distinguished constants called *indeterminates*. We shall represent polynomials as (variable-free) first-order terms over a many-sorted signature with sorts *Coef* (coefficients) and *Poly* (polynomial expressions). More specifically, polynomials are representatives of equivalence classes on ground terms. The relevant equivalence relation is described in this section.

Coefficients

Coefficients form the base domain in our approach. They are either built into the deductive mechanism via matching and unification, or else are dealt with via constraints within a hierarchical deductive system. We shall outline both approaches.

Buchberger's algorithm has originally been formulated for polynomials over a field, though variants have been developed for other coefficient domains. The feasibility of these algorithms depends on certain properties of the coefficient domain. In all cases, the coefficient domain is a commutative ring with unit. We assume that it is specified by some algebra $\mathcal{B}$ over a given signature and 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 and may vary with the coefficient domain, but we do assume that the signature contains at least the declarations

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

The symbols c and d (possibly with subscripts or other annotation) are used to denote terms of sort *Coef*.

We define B as the set of all ground equations $c \approx d$ such that c and d denote the same elements in $\mathcal{B}$. We need to be able to decide whether a given equation $c \approx 0$ belongs to B. Hence the ground theory B must be decidable. In the case of fields, one typically considers structures in which the inverse of any non-zero coefficient can be computed. In general, the solvability of certain division problems is a key requirement for Buchberger's algorithm. For instance, the result c/d of the division of c by a non-zero coefficient d may be described as a solution of the equation $c = d \cdot v$, where v is a variable of sort *Coef*.

Polynomial expressions

Polynomial expressions are constructed from coefficients, function symbols representing ring operations and the indeterminates as free generators. We use the following signature:

$$\begin{aligned} \Omega, I, X_1, \ldots, X_n &: \quad \to \mathit{Poly} \\ \ominus &: \mathit{Poly} \to \mathit{Poly} \\ \oplus, \otimes &: \mathit{Poly} \times \mathit{Poly} \to \mathit{Poly} \end{aligned}$$

The constants X_i are assumed to denote the indeterminates. Sometimes we also use metavariables X, Y, and Z to denote indeterminates. By AC we denote the set of associativity and commutativity axioms for addition and multiplication of polynomials,

$$\begin{aligned} x \oplus y &\approx y \oplus x \\ x \oplus (y \oplus z) &\approx (x \oplus y) \oplus z \\ x \otimes y &\approx y \otimes x \\ x \otimes (y \otimes z) &\approx (x \otimes y) \otimes z \end{aligned}$$

and by R the set of equations

$$\begin{aligned} x \oplus \Omega &\approx x \\ \ominus x \oplus x &\approx \Omega \\ x \otimes I &\approx I \\ x \otimes (y \oplus z) &\approx (x \otimes y) \oplus (x \otimes z) \end{aligned}$$

The set $R \cup AC$ is an equational specification of the variety of commutative rings with a unit element.

Coefficient terms are regarded as a subsort of polynomial expressions. We formalize the subsort relation in a many-sorted framework via an injection function

$$\iota : \mathit{Coef} \to \mathit{Poly}$$

cf. Ganzinger (1989). The set H, consisting of the equations

$$\begin{aligned} \iota(0) &\approx \Omega \\ \iota(1) &\approx I \\ \iota(-c) &\approx \ominus\iota(c) \\ \iota(c + d) &\approx \iota(c) \oplus \iota(d) \\ \iota(c \cdot d) &\approx \iota(c) \otimes \iota(d), \end{aligned}$$

expresses the requirement that the coefficient ring be homomorphically embedded in the polynomial ring.[3]

The equational theory induced by $B \cup R \cup H \cup AC$ on *ground* terms of sort *Poly* represents the embedding of one ring (the coefficient ring) in another one (the polynomial ring). We next design a rewrite system for this theory that is convergent modulo $B \cup AC$. Thus every ground term has a normal form that is unique up to associativity and commutativity and representation of coefficients. These normal forms correspond to polynomials in the usual sense.

[3] The first and the third equation are consequences of the other equations and the ring axioms for coefficients and polynomials.

Polynomials

The equations in H are directly turned into rewrite rules, with the exception of the third equation, which is replaced by a slightly more general rule:

$$\Omega \to \iota(0) \tag{1}$$
$$I \to \iota(1) \tag{2}$$
$$\ominus x \to \iota(-1) \otimes x \tag{3}$$
$$\iota(v) \oplus \iota(v') \to \iota(v + v') \tag{4}$$
$$\iota(v) \otimes \iota(v') \to \iota(v \cdot v') \tag{5}$$

These rewrite rules reduce some polynomial ring operations to operations on coefficients.

Additional rules are obtained either directly from R by simplification with the above rules, or from critical pair computations:

$$\iota(0) \oplus x \to x \tag{6}$$
$$\iota(0) \otimes x \to \iota(0) \tag{7}$$
$$\iota(1) \otimes x \to x \tag{8}$$
$$x \otimes (y \oplus z) \to (x \otimes y) \oplus (x \otimes z) \tag{9}$$
$$x \oplus x \to \iota(1 + 1) \otimes x \tag{10}$$
$$(\iota(v) \otimes x) \oplus x \to \iota(v + 1) \otimes x \tag{11}$$
$$(\iota(v) \otimes x) \oplus (\iota(v') \otimes x) \to \iota(v + v') \otimes x \tag{12}$$

By P we denote the set of the above twelve rewrite rules.

Proposition 3. *The rewrite system $P^e \backslash (B \cup AC)$ is convergent modulo $B \cup AC$. Its equational theory is the same as the one induced by $B \cup R \cup H \cup AC$.*

We outline a proof of the termination property below. The uniqueness of normal forms up to equivalence under $B \cup AC$ can be shown easily.

Ground terms in normal form correspond to (and will be called) *polynomials*. Normal-form terms built only from indeterminates and polynomial multiplication, are called *monomials*. For example, $(X \otimes Y) \otimes (X \otimes X)$ is a monomial. For our purposes it is sufficient to characterize monomials up to associativity and commutativity, and hence we use the common algebraic notation and write, for example, X^3Y to denote the above monomial.

Termination

The termination property of P (modulo $AC \cup B$) can be established by defining a suitable $AC \cup B$-compatible reduction ordering on ground terms of sort *Poly*. The ordering is constructed by the lexicographic combination of several associative path orderings (Bachmair and Plaisted 1985) and is total on ground terms. We briefly recall some of the basic properties of the associative path ordering (APO) for our particular case of two AC-operators $\oplus$ and $\otimes$. The APO is based on a precedence on operators. Admissible precedences are total, well-founded quasi-orderings $\succeq$ for which $f \succ \otimes \succ \oplus$ or $f \succ \otimes \succ \oplus \succ e$, where e is a minimal constant and f is any of the other function symbols. The associative path ordering is an ordering on ACD-equivalence classes. That is, it orders transformed terms, where transformation means normalizing by flattening with respect to $\oplus$ and $\otimes$ and by distributing $\otimes$ over $\oplus$. (Let us denote this rewrite system by F.) In this particular application we order terms after normalization by additional rewrite rules. Hence we will have to explicitly prove the compatibility of the ordering with contexts. Moreover, when we apply such an ordering to prove the termination of a rewrite system we will have to explicitly prove the compatibility of the orientation with ground substitutions. (We refer the reader to Bachmair and Dershowitz 1986 where these issues of transformation orderings are discussed in more detail.)

The first step in ordering the ring axioms (1)–(12) consists in ignoring coefficients altogether. This is achieved by a (confluent) transformation T_1 consisting of the following rules

$$\begin{array}{lll} \Omega \to e & \qquad & e \otimes x \to x \\ I \to e & & x \otimes e \to x \\ \iota(v) \to e & & \end{array}$$

where e is a new constant. The transformation basically deletes all occurences of coefficients from polynomials. Let $\succeq_1$ be a precedence in which $f \succ_1 \ominus \succ \otimes \succ_1 \oplus \succ_1 e$. In particular we are free to choose the (quasi-)precedence among the indeterminates X_i. If all indeterminates are distinguished, the ordering corresponds to the usual "lexicographic ordering," whereas the "total degree ordering" is obtained by declaring all indeterminates to be equivalent.

Let $\trianglerighteq_1$ denote the T_1-transform of the associative path ordering over this precedence, i.e., $s \trianglerighteq_1 t$ if and only if $s{\downarrow_{TF_1}} \succeq_1^{apo} t{\downarrow_{TF_1}}$ where $\succeq_1^{apo}$ is the associative path ordering over $\succeq_1$ and where TF_1denotes $T_1 \cup F$.

Proposition 4. *$\trianglerighteq_1$ is a total reduction quasi-ordering on ground terms of sort Poly.*

Proposition 5. *For any of the ground instances $l \to r$ of the rules (3), (4), (6), (10)–(12) we have $l \triangleright_1 r$. For any ground instance $l \to r$ of any of the remaining rules we have $l \trianglerighteq_1 r$.*

To orient the remaining rules except distributivity we use an associative path ordering in which all coefficients are identified but not deleted. More precisely, let T_2 be the transform consisting of the rule $\iota(v) \to e$ and let $\succ_2$ be a strict total precedence of the form $f \succ_2 \ominus \succ_2 \otimes \succ_2 \oplus \succ_2 e$. Let $\trianglerighteq_2$ denote the T_2-transform of the associative path ordering over $\succ_2$.

Proposition 6. *$\trianglerighteq_2$ is a total reduction quasi-ordering on ground terms of sort Poly. Moreover, for any of the ground instances $l \to r$ of the rules (1), (2), (5), (7), (8) we have $l \triangleright_2 r$. For any ground instance $l \to r$ of (9) we have $l \trianglerighteq_2 r$.*

Let $\trianglerighteq_3$ be an associative path ordering over an admissible total precedence which refines the given ordering (modulo B) on the coefficients. The purpose of $\trianglerighteq_3$ is to order terms which only differ in the (congruence classes of) coefficient subterms which they contain. Finally, let D denote some $AC \cup B$-compatible reduction ordering which orients distributivity (9) from left to right. We define $\succeq$ as the lexicographic combination $(\trianglerighteq_1, \trianglerighteq_2, \trianglerighteq_3, D)$ of these four orderings .

Theorem 7. *$\succ$ is a total, $AC \cup B$-compatible reduction ordering on ground terms of sort Poly containing the (ground instances of the) rewrite rules in P^e.*

4 Polynomial completion

The theories to which we apply extended completion are congruence relations on ground terms induced by the set $P \cup B \cup AC$ plus polynomial equations. We design derived inference rules based on extended completion modulo $B \cup AC$, that allow one to simplify equations and thereby decrease the number of deductions needed to construct a convergent system. We call the derived inference system *polynomial completion.*

Derived inference rules

By a *polynomial rule* we mean an equation $t \to r$, where $t \succ r$ and t is of the form t' or $\iota(c) \otimes t'$, where t' is a monomial and c is a term of sort *Coef* (possibly containing variables). One of the key features of Buchberger's algorithm is that any ground equation $p \approx q$ can be transformed into a polynomial rule.

Equations may be normalized to polynomial form:

Normalization: $\qquad N \cup \{p \approx q\} \vdash N \cup \{p' \approx q'\}$

if p' and q' are normal forms of p and q, respectively, with respect to $P^e \backslash (B \cup AC)$, and P is a subset of N.

Trivial and built-in equations can be deleted:

Elimination: $\qquad N \cup \{p \approx q\} \vdash N$

if $p \leftrightarrow^*_{B \cup AC} q$.

We may also cancel equal terms on both sides:

Cancellation: $\qquad N \cup \{p \approx q\} \vdash N \cup \{p' \approx q'\}$

if $p \leftrightarrow^*_{B \cup AC} p' \oplus r$ and $q \leftrightarrow^*_{B \cup AC} q' \oplus r$.

We have the following form of subsumption:

Subsumption: $\qquad N \cup \{u \approx v, p \approx q\} \vdash N \cup \{p \approx q\}$

if $u \leftrightarrow^*_{B \cup AC} w[p]$ and $v \leftrightarrow^*_{B \cup AC} w[q]$, for some term w different from p.

Finally, equations may be turned into rules:

Orientation: $\qquad N \cup \{p \approx q\} \vdash N \cup \{t \to \iota(-1)r\}$

if t is a product with $t \succ r$ and $t \oplus r$ is a normal form of $\iota(-1)p \oplus q$ with respect to $P^e \backslash (B \cup AC)$.

For example, the polynomial equation $2XY \oplus X \approx 3XY \oplus Y$ can be oriented into a polynomial rule $XY \to -(-X \oplus Y)$. (For simplicity, we denote polynomial multiplication by juxtaposition and write c instead of $\iota(c)$ and $-u$ instead of $\iota(-1)u$.)

These inference rules are sufficient to eliminate almost all polynomial equations.

Lemma 8. *If none of the inference rules for normalization, deletion, cancellation, and orientation can be applied to a ground equation (between terms of sort Poly), then the equation is of the form $\iota(c) \approx \iota(d)$, where c and d denote different coefficients.*

The significance of the lemma is that it allows one to avoid failure in polynomial completion. For if the above transition rules are applied whenever possible, then the only equations that may persist are of the form $\iota(c) \approx \iota(d)$, where c and d denote different coefficients. But if such an equation is produced, then $x \approx \iota(0)$ may be deduced, which means that all terms of sort *Poly* are equal. In other words, this case characterizes the equational theory in which all polynomial expressions are equal (the trivial polynomial ideal). In all non-trivial cases, it is possible to construct non-failing derivations.

Fairness

We already know that a non-failing derivation yields a (possibly infinite) convergent set of rules whenever it is fair. Fairness requires the elimination of peaks involving persisting rules and extensions thereof. By Lemma 2 all peaks can be eliminated if all extended critical pairs modulo $B \cup AC$ are deduced. Critical pair computation is thus the main form of deduction in polynomial completion:

Superposition: $\quad N \vdash N \cup \{p \approx q\}$

if $p \approx q$ is an extended critical pair modulo $B \cup AC$ between rules in N^e. No extensions result from B.

Let us analyze which equations may be deduced by superposition. Persisting rules are either polynomial rules or rules in P. Since P^e is convergent modulo $B \cup AC$, the corresponding critical pairs are not needed, as the respective peaks can be eliminated. Let us look at critical pairs involving polynomials rules.

We first assume that the coefficient domain is a field and that the inverse of any non-zero element can be computed. In that case we can further simplify the left-hand sides of polynomial rules.

Standardization: $\quad N \cup \{\iota(c)t \to r\} \vdash N \cup \{t \to \iota(d)r\}$

where d is any coefficient term equal to $1/c$.

We also call $t \to r$ a *standardized rule* if t is a monomial (and $t \succ r$).

Let $t \to r$ and $t' \to r'$ be standardized rules, such that $t = uv$ and $t' = vu'$, where the two monomials u and u' contain no common indeterminates. Then $ru' \approx ur'$ is an extended critical pair between the two rules. Formally, the critical pair is obtained by superposition with the two extended rules $tx \to rx$ and $t'y \to$

$r'y$. (The AC-unification of the extended left-hand sides yields the overlapped term uvu', which can be rewritten either to ru' or ur'.) Note that unification modulo B is not required, as coefficients do not appear on the left-hand sides of standardized rules. The AC-unification of products tx and $t'y$ corresponds to the computation of the least common multiple of t and t'.

For example, from (the extensions of) the two rules $X^2Y \to X \oplus Y$ and $XY^2 \to X \oplus Y$ we obtain a critical pair $XY \oplus Y^2 \approx X^2 \oplus XY$.

It is fairly easy to show that the indicated critical pairs are the only ones that need to be deduced from (extensions of) standardized rules. (The AC-unification of the extended left-hand sides uvz and $vu'z$ also yields an overlapped term $uvu'z$ and a corresponding critical pair $ru'z \approx ur'z$. This additional critical pair is subsumed by the ground equation $ru' \approx ur'$, and therefore can be deleted.) Also, no critical pairs need to be computed if the two left-hand sides t and t' share no common indeterminate. (Corresponding peaks can be eliminated by rearranging their subproofs.) Furthermore, there are no non-trivial critical pairs between standardized rules and P^e. The non-redundant critical pairs are again ground polynomial equations so that no non-ground equation is ever generated.

Let us now briefly consider critical pairs between P^e and non-standardized rules. For example, there is a critical pair between the extension $\iota(3)X^3z \to XYz$ of the rule

$$\iota(3)X^3 \to XY$$

and the extended rule $\iota(v)\iota(v')x \to \iota(vv')x$. The overlapped term $\iota(3)\iota(v')X^3$ can be rewritten either to $\iota(3v')X^3$ or to $\iota(v')XY$, resulting in a critical pair

$$\iota(3v')X^3 \approx \iota(v')XY.$$

The critical pair can *almost* be oriented into a rule, except that both sides are equal if the variable v' is instantiated by zero. In other words, we have a conditional rule that is subject to the constraint that v' be instantiated by a non-zero coefficient term.

The use of constraints in polynomial completion will be discussed in the next section. But first we have to mention another key feature of completion procedures, the use of newly deduced rules for simplification by rewriting.

Simplification

The simplification mechanisms used by completion procedures are indispensable for efficiency, and polynomial completion is no exception in this regard. The following simplification rules are typically used in completion.

Simplification: $\quad N \cup \{p \approx q, t \to r\} \vdash N \cup \{p'[r\sigma] \approx q, t \to r\}$

if $p \leftrightarrow^*_{B\cup AC} p'[t\sigma]$.

Composition: $\quad N \cup \{u \to q[v], t \to r\} \vdash N \cup \{u \to q[r\sigma], t \to r\}$

if $v \leftrightarrow^*_{B\cup AC} t\sigma$.

Collapse: $\quad N \cup \{u \to q, t \to r\} \vdash N \cup \{u' \otimes r\sigma \approx q, t \to r\}$

if $u \leftrightarrow^*_{B \cup AC} u' \otimes t\sigma$.

We emphasize that simplification rules crucially depend on the given proof ordering. (The reader is invited to verify that the above rules indeed represent combinations of deduction and deletion steps.) A slightly stronger collapse rule may be obtained by refining the proof ordering, so that rules may be kept "interreduced" in the sense that no left-hand side of a (persisting) rule can be rewritten by any other (persisting) rule. The same effect is achieved, though, if the maximal product in an equation is fully simplified before the equation is oriented into a rule;[4] see Bachmair (1991), Bachmair and Dershowitz (1994) for a discussion of these issues.

Buchberger's algorithm

By polynomial completion we mean the inference system consisting of the above transition rules for normalization, elimination, cancellation, subsumption, orientation, superposition, standardization, simplification, composition, and collapse. The term "Buchberger's algorithm" refers to a class of polynomial completion procedures that accept as input a set N of polynomial equations of the form $p \approx \Omega$, and use the above derived inference rules to produce a derivation $N_0 \vdash N_1 \vdash \cdots$ from the initial set $N_0 = N \cup P$, such that (i) P is a subset of all sets N_i, (ii) $N_\infty \setminus P$ contains only ground polynomial rules, and (iii) the rules in N_∞ are interreduced. The coefficient domain is assumed to be a field, so that all polynomial rules can be standardized.

The computation of sufficiently many extended critical pairs ensures fairness and therefore, by our general completeness result, implies that the final set of rules is convergent modulo $B \cup AC$.[5] Since the final set of rules N_∞ is interreduced, one may apply Dickson's Lemma (cf. Becker and Weispfenning 1993) to infer that N_∞ is finite, so that these procedures indeed *terminate*.

5 Constraints

Polynomial completion is extended completion with a built-in theory $B \cup AC$, where building in a theory essentially means extending the relevant unification and matching procedures. In recent years the trend in logic programming and in automated deduction has been to describe unification problems, and hence built-in theories, in a more abstract form via *constraints*. Manipulation of coefficients can be naturally described in this way.

A *simple constraint* is either an equation $c \approx d$ between coefficient terms or the negation thereof, $c \not\approx d$; a *constraint* is a conjunction of simple constraints. A *solution* of a constraint is any substitution that assigns ground terms to variables of sort *Coef*, such that all the resulting identities are satisfied in $\mathcal{B}$.

For example, the substitution σ that maps the variables v, v_1, and v_2 to 2 and the variable v_0 to 8 is a solution of the constraint $v_0 \approx 4v \wedge v_1 \approx v \wedge v_2 \approx v \wedge v \not\approx 0$.

[4] All known completion procedures construct derivations obeying this restriction.

[5] We should mention that our notion of fairness also covers the various criteria that have been proposed for avoiding computation of redundant critical pairs.

A *constrained equation* is an expression $p \approx q$ `if` C, where $p \approx q$ is an equation and C a constraint. 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 variables of sort *Coef* is a solution of 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. Completion procedures can readily be extended to constrained equations, see Kirchner, Kirchner and Rusinowitch (1990).

For instance, we have seen that for every polynomial rule

$$\iota(c)t \to r$$

we obtain a critical pair $\iota(cv)t \approx \iota(v)r$, where v is a variable of sort *Coef*. The latter equation can be turned into a constrained rule

$$\iota(cv)t \to \iota(v)r \ \texttt{if} \ v \not\approx 0.$$

(The transformation is sound for polynomial completion because $\iota(cv)t \approx \iota(v)r$ is equivalent to $v \approx 0 \vee \iota(cv)t \approx \iota(v)r$ in all models of $B \cup AC \cup P$.) Note that the new variables of sort *Coef* play a similar role as the extension variables of sort *Poly*, in that they enable the computation of non-trivial unifiers modulo the built-in theory B during superposition.

For example, take the two rules

$$\iota(4v)X^2Y \to \iota(v)X \oplus \iota(v)Y \ \texttt{if} \ v \not\approx 0$$

and

$$\iota(6v')XY^2 \to \iota(v')X \oplus \iota(v')Y \ \texttt{if} \ v' \not\approx 0$$

and the integers as coefficient domain. The unification problem $4v \leftrightarrow_B^* 6v'$ is solvable under the constraint $v \not\approx 0$: instantiate v by a multiple of 3 and v' by the corresponding multiple of 2. The term $\iota(12)X^2Y^2$, for example, can be rewritten modulo $B \cup AC$ to $Y(\iota(3)X \oplus \iota(3)Y)$ by the first rule, and to $X(\iota(2)X \oplus \iota(2)Y)$ by the second rule; we may deduce the corresponding critical pair by superposition.

We shall next formulate unification of coefficient terms via constraints. First observe that the ground equational theory induced by $AC \cup B$ can be decomposed into its two components; i.e., that is, the congruence relation $\leftrightarrow_{B \cup AC}^*$ is identical to $\leftrightarrow_B^* \circ \leftrightarrow_{AC}^*$ and to $\leftrightarrow_{AC}^* \circ \leftrightarrow_B^*$. Using this observation, we explicitly separate terms of sort *Coef* from terms of sort *Poly* in polynomial equations by replacing each subterm $\iota(c)$ by a term $\iota(v)$, where v is a new variable defined by a constraint $v \approx c$.

For example, the constrained rule

$$\iota(4v)X^2Y \to \iota(v)X \oplus \iota(v)Y \ \texttt{if} \ v \not\approx 0$$

is transformed to a rule

$$\iota(v_0)X^2Y \to \iota(v_1)X \oplus \iota(v_2)Y \ \texttt{if} \ v_0 \approx 4v \wedge v_1 \approx v \wedge v_2 \approx v \wedge v \not\approx 0.$$

(This transformation is a special case of "variable abstraction" as used, for instance, in Bachmair, Ganzinger and Waldmann 1994.)

Critical pair computation with constrained rules is simple. Let $\iota(v)t \to p$ `if` C and $\iota(v')t' \to p'$ `if` C' be constrained rules. If $q \approx r$ is a critical pair between the standardized rules $t \to p$ and $t' \to p'$, then

$$q \approx r \texttt{ if } C \wedge C' \wedge v \approx v'$$

is a critical pair between the constrained rules. Constrained equations can be transformed into constrained rules.

In summary, constraints may be used to describe unification problems over the base domain $\mathcal{B}$. We emphasize that for deduction it is not necessary to actually compute solutions (i.e., unifiers) of these constraints. It suffices to check for solvability, equations with an unsolvable constraint may be deleted. The possibility of delaying the solving of constraints may be useful in practice, especially when coefficients tend to become large. It is sufficient to compute the coefficients at the end, for the final convergent system, and use the implicit description via constraints for intermediate polynomial equations. Lazy constraint solving is usually combined with domain-dependent mechanisms for constraint simplification.

Buchberger's algorithm is applied, among others, for proving theorems in geometry. In this context, theorems to be proved are usually true only if certain non-degeneracy conditions are met; and it is considered the task of the prover to find these conditions automatically. Technically this often means that indeterminates are classified into independent and dependent ones. If one considers the independent ones as (existentially quantified) variables over the coefficient structure so that they become part of the constraints, the final constraint associated with an "inconsistency" $x \approx 0$ `if` C will represent one such non-degeneracy condition. This idea of generating non-degeneracy conditions as constraints should be investigated further.

We conclude this section by pointing out that the critical pairs mentioned above are sufficient if the coefficient domain is a field. If the coefficient domain is not a field, superposition between non-standardized (constrained) rules and rules 11 and 12 of P produces further non-trivial critical pairs.

For example, the term $\iota(v)X^3 \oplus \iota(3)X^3$ can be rewritten either to $\iota(v+3)X^3$ by rule 12, or to $\iota(v)X^3 \oplus XY$ by the rule $\iota(3)X^3 \to XY$. If the coefficient domain is a field, say the rational numbers, the resulting critical pair,

$$\iota(v+3)X^3 \approx \iota(v)X^3 \oplus XY,$$

can be simplified to a trivial equation and deleted. (Several cases have to be considered. If $v \not\approx 0$ and $v+3 \not\approx 0$, then the rule $\iota(3v')X^3 \to \iota(v')XY$ may be used to rewrite the term $\iota(v+3)X^3$ to $\iota((v+3)/3)XY$, and the term $\iota(v)X^3 \oplus XY$ to $\iota(v/3)XY \oplus XY$. Normalizing by P^e results in a trivial equation. The other cases are similar.) If the coefficient domain is not a field, the necessary simplifications may not be possible, and the critical pairs have to processed further, cf. Kandri-Rody and Kapur (1988).

6 Summary

We have presented polynomial completion, a variant of extended completion with a built-in theory $AC \cup B$. This variant of completion covers Buchberger's algorithm for constructing Gröbner bases for ideals in polynomial rings over fields. A major obstacle in previous attempts at reformulating Buchberger's algorithm as a completion procedure has been the specification of the required operations on the coefficient domain. The rewrite mechanisms that were employed for that purpose proved to be either inadequate or required an overly complicated technical machinery. We have demonstrated that constraints provide a conceptually simple alternative for an abstract specification of coefficient domains. The approach we have proposed is also flexible enough to be adapted to variants and extensions of Buchberger's algorithm, e.g., to Boolean ring-based theorem proving procedures (Hsiang 1985, Kapur and Narendran 1985, Bachmair and Dershowitz 1987, Zhang 1992).

One motivation for this work has come from the experience we had with our implementation of a chaining-based theorem prover for (dense) total orderings (Bachmair and Ganzinger 1993). Chaining is an inference mechanism similar to superposition, but applies to non-symmetric binary relations, such as orderings, rather than equivalence or congruence relations. In many applications orderings appear in combination with operations forming a abelian group or a ring. Experiments indicate that chaining by itself is too weak an inference mechanism for handling theorem proving problems in such richer algebraic structures efficiently. We believe that improved theorem proving systems can be obtained by combining inference systems such as chaining with more specialized techniques, such as those for polynomial ring structures. In this context it is of interest that polynomial completion as formalized above is essentially a hierarchical theorem proving system in the sense of Bachmair, Ganzinger and Waldmann (1994), as this provides us with a way of generalizing the techniques inherent in Buchberger's algorithm to first-order theorem proving.

References

Leo Bachmair, 1991. *Canonical equational proofs.* Birkhäuser, Boston.

Leo Bachmair and Nachum Dershowitz, 1986. Commutation, transformation, and termination. In *Proc. 8th Conf. on Automated Deduction, Oxford, LNCS 230*, pp. 5–20. Springer-Verlag, Berlin.

Leo Bachmair and Nachum Dershowitz, 1987. Inference rules for rewrite-based first-order theorem proving. In *Proc. Second IEEE Symp. Logic in Computer Science (Ithaca, New York)*, pp. 331–337. IEEE Comp. Soc. Press.

Leo Bachmair and Nachum Dershowitz, 1994. Equational inference, canonical proofs, and proof orderings. *Journal of the ACM*, Vol. 41, pp. 236–276.

Leo Bachmair and Harald Ganzinger, 1993. Ordered chaining for total orderings. Technical Report MPI-I-93-250, Max-Planck-Institut für Informatik, Saarbrücken. To appear in *Proc. CADE'94*.

Leo Bachmair and Harald Ganzinger, 1994. Rewrite-based equational theorem proving with selection and simplification. *J. Logic and Computation*, Vol. 4, No. 3, pp. 217–247.

Leo Bachmair, Harald Ganzinger and Uwe Waldmann, 1994. Theorem proving for hierarchic first-order theories. *Applicable Algebra in Engineering, Communication and Computing*, Vol. 5, pp. 193–212.

Leo Bachmair and David Plaisted, 1985. Termination orderings for associative-commutative rewriting systems. *J. Symbolic Computation*, Vol. 1, pp. 329–349.

THOMAS BECKER AND VOLKER WEISPFENNING, 1993. *Gröbner bases: a computational approach to commutative algebra.* Springer-Verlag.
BRUNO BUCHBERGER, 1965. *An Algorithm for Finding a Basis for the Residue Class Ring of a Zero-Dimensional Ideal.* PhD thesis, University of Innsbruck, Austria. (In German).
BRUNO BUCHBERGER, 1985. Gröbner bases : an algorithmic method in polynomial ideal theory. In N.K. Bose, editor, *Recent Trends in Multidimensional Systems theory*, pp. 184–232. Reidel.
BRUNO BUCHBERGER, 1987. History and Basic Features of the Critical Pair / Completion Procedure. *J. Symbolic Computation*, Vol. 3, pp. 3–38.
BRUNO BUCHBERGER AND RÜDIGER LOOS, 1982. Algebraic simplification. In *Computer Algebra*, pp. 14–43. Springer-Verlag, Berlin.
REINHARD BÜNDGEN, 1991. Simulating Buchberger's Algorithm by a Knuth-Bendix Completion Procedure. In *Proc. Fourth International Conference on Rewriting Techniques and Applications (Como, Italy), LNCS 488*, Berlin, Springer-Verlag.
NACHUM DERSHOWITZ AND JEAN-PIERRE JOUANNAUD, 1990. Rewrite Systems. In , Handbook of Theoretical Computer Science, vol. B, pp. 243–309. North-Holland.
HARALD GANZINGER, 1989. Order-Sorted Completion: The Many-Sorted Way (Extended Abstract). In J. Díaz, F. Orejas, editors, *Proc. TAPSOFT '89, Barcelona 1989, volume II*, Lecture Notes in Computer Science, vol. 351, pp. 244–258, Berlin, Springer-Verlag. Full version in TCS, volume 89, 1991.
JIEH HSIANG, 1985. Refutational theorem proving using term-rewriting systems. *Artificial Intelligence*, Vol. 25, pp. 255–300.
JEAN-PIERRE JOUANNAUD AND HÉLÈNE KIRCHNER, 1986. Completion of a Set of Rules Modulo a Set of Equations. *SIAM Journal on Computing*, Vol. 15, No. 4, pp. 1155–1194.
ABDELIGAH KANDRI-RODY AND DEEPAK KAPUR, 1988. Computing a Gröbner basis of a polynomial ideal over a Euclidean domain. *J. Symbolic Computation*, Vol. 6, pp. 37–57.
DEEPAK KAPUR AND PALIATH NARENDRAN, 1985. An equational approach to theorem proving in first-order predicate calculus. In *Proc. Ninth International Joint Conference on Artificial Intelligence*, pp. 1146–1153, Los Angeles, CA.
CLAUDE KIRCHNER, HELENE KIRCHNER AND MICHAEL RUSINOWITCH, 1990. Deduction with symbolic constraints. *Revue Française d'Intelligence Artificielle*, Vol. 4, No. 3, pp. 9–52.
RÜDIGER LOOS, 1981. Term reduction systems and algebraic algorithms. In *Proceedings of the Fifth GI Workshop on Artificial Intelligence*, pp. 214–234, Berlin, Springer-Verlag. Available as *Informatik Fachberichte*, Vol. 47.
CLAUDE MARCHÉ, 1994. Normalised rewriting and normalised completion. In *Proc. IEEE Symposium on Logic in Computer Science.* IEEE Comp. Soc. Press. To appear.
GERALD E. PETERSON AND MARK E. STICKEL, 1981. Complete sets of reductions for some equational theories. *Journal of the ACM*, Vol. 28, pp. 233–264.
HANTAO ZHANG, 1992. A new method for the Boolean ring based theorem proving. In *Proc. Second Int. Symposium on Artificial Intelligence and Mathematics.*