Normalized Rewriting: an unified view of Knuth-Bendix completion and Gröbner bases computation*

Claude Marché
Lab. de Recherche en Informatique
Université Paris-Sud et CNRS URA 410
91405 Orsay cedex
France

Abstract

Buchberger's algorithm and Knuth-Bendix completion are two fundamental algorithms used respectively in computer algebra and in equational logic. It has been remarked since 1981 that these two algorithms behave in the same way. We show that this similarity is made explicit by viewing them as two particular instances of a very general completion algorithm: normalized completion.

1 Introduction

In the past 30 years have been developped two fundamental algorithmic methods for solving problems in two different domains of computer science. In the domain of computer algebra, Buchberger's algorithm, introduced in 1965 (Buchberger, 1965), computes *Gröbner bases* of polynomial ideals. It allows to solve various kinds of algebraic problem and has many applications (Buchberger and Loos, 1982; Buchberger, 1987) where algebraic constraints are concerned. In the domain of automatic theorem proving, Knuth-Bendix completion algorithm, introduced in 1970 (Knuth and Bendix, 1970), computes canonical rewriting systems associated to equational axioms. It allows to prove identities in equational logic, and has several applications for example in program specification.

It has been further remarked by Loos and Buchberger (Loos, 1981; Buchberger and Loos, 1982) that these two algorithms behave in a similar way : they

*This work has been partly supported by the "GDR Programmation du CNRS", the ESPRIT Working Group "Compass", the ESPRIT Basic Research Action "TYPES", the ESPRIT Project CCL and ESPRIT Project Console

	Gröbner bases	Monoids and Groups	Knuth-Bendix
Expressions (Syntax)	polynomials $(X^2Y + 2X + \cdots)$	words $(a^2bc^{-1}\cdots)$	first-order terms $(f(x, g(a, \ldots)))$
Rules	$X^2Y \to ..$ $XY^2 \to ..$	$ab \to ..$ $bc \to ..$	$f(x, g(a, x)) \to \ldots$ $g(x, b) \to \ldots$
Superpositions	X^2Y^2 $\swarrow \quad \searrow$	abc $\swarrow \quad \searrow$	f(b,g(a,b)) $\swarrow \quad \searrow$

Figure 1: Similarities between completion-like algorithms

combine the notions of *simplification* by *reduction rules* and *deduction* by *superposition* of rules. It also appeared that similar algorithms where known for solving the word problem in finitely generated monoids and groups (Le Chenadec, 1986). The similarities between those algorithms are shown on figure 1.

Since that time, many researchers have tried to unify in some sense these algorithms. We can cite the works (Le Chenadec, 1986; Kandri-Rody et al., 1989; Pottier, 1989; Bündgen, 1991a; Bündgen, 1991b; Bachmair and Ganzinger, 1994) and we miss certainly many others.

Here we still propose another approach which tries to unify these algorithms not by describing an abstract behaviour, but by giving a really new and more general algorithm : the S-normalized completion, that appears to be used (for particular choices of S) both for computing Gröbner bases, canonical rewrite systems for arbitray equational presentations and for monoids and groups, among many other cases.

Let us remark that the discovery of this new algorithm has been possible thanks to recent advances in rewriting techniques: rewriting modulo a set of equations (Jouannaud and Kirchner, 1986; Bachmair and Dershowitz, 1989), the normalization proof method (Bachmair and Dershowitz, 1989; Bachmair et al., 1986), and contrained rewriting (Kirchner and Kirchner, 1989; Baird et al., 1989; Jouannaud and Marché, 1992).

All the details and proofs of results concerning normalized rewriting and completion can be found in (Marché, 1996). In this paper we introduce briefly normalized rewriting (Section 2) and normalized completion (Section 3). In Section 4, we apply the technique to monoids, groups and polynomials over $\mathbb{Z}$ and $\mathbb{F}_p$ for p prime, giving some practical results obtained by our implementation of normalized completion called C*i*ME[1]. Finally, we look further to termination issues (Section 5).

[1] `http://www.lri.fr/~marche/cime.html`, `ftp://lri.lri.fr/LRI/soft/cime/cime.tar.Z`

2 Normalized rewriting

Our notations and definitions are consistent with those given in the survey on rewriting techniques of Dershowitz and Jouannaud (Dershowitz and Jouannaud, 1990).

We assume we are given an first-order algebra by a signature $\mathcal{F}$ and a set of equations E defining some relations between constructors or operators in $\mathcal{F}$. The considered algebra is then the quotient of the free algebra $\mathcal{F}(\mathcal{F})$ (*i.e.* the term algebra) by the equational theory generated by E. There may be an arbitrary number of associative-commutative operators, denoted hereafter by AC.

The technique of normalized rewriting needs that each equivalence classes modulo E has a particular representative: a *normal form W*. For this reason, we will assume the equational theory E having an equivalent canonical rewriting system S, possibly modulo AC.

Our goal now is to define a rewriting relation on this quotient algebra.

Definition 2.1 *We denote by $s\downarrow_S$ the S-normal form of a term s. The S-*normalized rewrite relation *by a rule $l \to r$, is defined by*

$$s \xrightarrow[l\to r/S]{p} t \text{ iff } s\downarrow_S \xrightarrow[l\to r]{p} t$$

When R is a set of rules, we denote by $\to_{R/S}$ the S-normalized rewrite relation generated by R.

Example 2.2 *Assume we have an algebra containing an AC operator $+$ which has a unit 0. We take then $S = \mathrm{ACU}(+,0) = \{x+0 \to x\}$. Assume now we would like to rewrite by $R = \{-(x+y) \to (-x)+(-y)\}$. We have $-(a+b) \to_{R/S} (-a)+(-b)$ but we can not rewrite $-(0+b)$ to $(-0)+(-b)$ because the S-normal form of $-(0+b)$ is $(-b)$ which is not an instance of $-(x+y)$.*

We see on this example that the idea of normalized rewriting captures the notion of ACU-constrained rewriting (Baird et al., 1989; Jouannaud and Marché, 1992).

Example 2.3 *Assume our algebra is a monoid. We take for S the convergent rewrite system of monoids, that is*

$$(x.y).z \to x.(y.z) \qquad x.1 \to x \qquad 1.x \to x$$

Assume now we have $R = \{a.b \to b\}$ where a and b are some constants (representing generators), then $(a.a).b \to^*_{R/S} b$ since the S-normal form of $(a.a).b$ is $a.(a.b)$.

Assume our algebra is a group. We take for S the convergent rewrite system of groups (Knuth and Bendix, 1970), that is

$$\left\{\begin{array}{llll} x.1 \to x & (x.y).z \to x.(y.z) & (x.y)^{-1} \to y^{-1}.x^{-1} & x.(x^{-1}.y) \to y \\ 1.x \to x & x.x^{-1} \to 1 & (x^{-1})^{-1} \to x & x^{-1}.(x.y) \to y \\ 1^{-1} \to 1 & x^{-1}.x \to 1 & & \end{array}\right.$$

Assume now we have $R = \{a.b \to b\}$, then $(a.a).b \to^*_{R/S} b$, but $(a.b).b^{-1}$ does not rewrite to 1 since its S-normal form is a.

We see on these last example that S-normalized rewriting captures well the notion of reduction in finitely presented groups, whereas the usual notion of rewriting modulo S (Jouannaud and Kirchner, 1986; Bachmair and Dershowitz, 1989) would not.

Example 2.4 *Assume now our algebra is a ring of multivariate polynomials. We take for S the AC-convergent rewrite system of commutative rings theory, that is*

$$\left\{\begin{array}{lll} x+0 \to x & x+(-x) \to 0 & -(x+y) \to (-x)+(-y) \\ -0 \to 0 & -(-x) \to x & x*(y+z) \to (x*y)+(x*z) \\ x*1 \to x & x*0 \to 0 & x*(-y) \to -(x*y) \end{array}\right.$$

where $+$ and $*$ are AC. Assume now we have $R = \{X*X \to Y\}$ where X and Y are some constants (representing indeterminates, hence terms represent polynomials). Then $X*X*X \to_{R/S} X*Y$ but $X*(X+(-Y))+((-X)*X)$ can not be rewritten since the S-normal form of $X*(X+(-Y))+((-X)*X)$ is $-(X*Y)$ and is not reducible by R.

We see in this case that normalized rewriting captures the notion of polynomial reduction used in Gröbner basis computation, where the distributivity law is applied before the rules.

2.1 Termination of normalized rewriting

Proving termination of (usual) rewriting modulo an equational theory E requires an ordering compatible with E but such an ordering does not exist in general. One interesting property of normalized rewriting is that we only need a reduction ordering compatible with AC. Such an ordering can be defined in various ways. For general notions on orderings and termination, we refer to (Dershowitz, 1987). For definitions of AC-compatible orderings, see (Bachmair and Plaisted, 1985; Ben Cherifa and Lescanne, 1986; Narendran and Rusinowitch, 1991; Nieuwenhuis and Rubio, 1993; Delor and Puel, 1993).

From now on, we assume we are given an AC reduction ordering $\succeq$, such that the set of rules S satisfies $\to_S \subseteq \succ$ (that is the termination of S can be proved by $\succeq$). The following proposition is straightforward.

Proposition 2.5 *Let R be a set of rules such that for all $l \to r$ in R, $l \succ r$. Then the S-normalized rewrite relation $\to_{R/S}$ is well-founded.*

3 Normalized completion

We still assume we are given a signature containing arbitrarily many AC operators, a set of equations which has an equivalent convergent rewriting system S, together with a reduction ordering $\succeq$ such that $\rightarrow_S \subseteq \succ$. We assume now we have a new set of equations E_0 defining for example some new operations. Our aim is to complete E_0 into a set of rules R such that S-normalized rewriting by R is well-founded and confluent.

We first need to introduce the notion of *normalizing pairs*, which in some sense will replace the usual notion of orientation in completion procedures. We assume given a *admissible proof ordering* $\succ_P$, that is a well-founded and monotonic ordering on rewrite proofs (Marché, 1996).

Definition 3.1 *A function that maps a pair of terms (u,v) to a pair $(\Theta(u,v), \Psi(u,v))$ where $\Theta(u,v)$ is a set of equations and $\Psi(u,v)$ a set of rules, is called an S-*normalizing pair *(w.r.t. $\succ_P$) if for any terms u and v such that $u \succ v$:*

(i). for any elementary irreducible proof of the form

$$s \underset{u=v}{\longleftrightarrow} t$$

there exists a smaller proof (w.r.t. $\succ_P$) in $\Theta(u,v) \cup \Psi(u,v)$ between s and t;

(ii). for all $l \rightarrow r \in \Psi(u,v)$, for all set of rules R, for all r' such that $r \rightarrow^*_{R/S} r'$, for any elementary irreducible proof of the form

$$s \underset{l \rightarrow r'}{\longrightarrow} t$$

there exists a smaller proof (w.r.t. $\succ_P$) in $\Theta(u,v) \cup \Psi(u,v) \cup R$ between s and t;

As customary, we describe the completion process by a set of inference rules (Figure 2). $\succeq$ is a reduction ordering, E is a set of equations and R is a set of rules. In the rule Collapse, the ordering $\succ\!\!\succ$ is defined by $(u,v) \succ\!\!\succ (l, r\theta)$ iff $p \neq \Lambda$, or $p = \Lambda$ and θ is not a renaming, or $p = \Lambda$, θ renaming and $u \succ r\theta$.

Notice that the rule Orient does not act anymore as usual by simply turning an equation into a rule, it rather transforms an equation into its normalizing pair.

The rule Deduce computes critical pairs modulo some equational theory T that we can choose arbitrarily between AC and S (AC $\subseteq T$ for completeness, $T \subseteq S$ for soundness). This is a very important point for two reasons:

(i). S may not be decidable and finitary with respect to unification. For example, S may contain distributivity law, and we know that unification modulo ACD is undecidable (Kirchner, 1990). In such a case we should use $T =$ AC.

(ii). It is known that ACU-unification, unification modulo Abelian group theory and Boolean ring theory lead to complete sets of unifiers which are usually much smaller than AC unification (Kirchner, 1990).

ORIENT
$$E \cup \{u = v\}; R \vdash E \cup \Theta(u,v); R \cup \Psi(u,v) \quad \text{if } u = u{\downarrow_S},\ v = v{\downarrow_S},\ u \succ v.$$
DEDUCE
$$E; R \vdash E \cup \{u = v\}; R \quad \text{if } u = v \in \mathrm{CP}_T(R).$$
NORMALIZE
$$E \cup \{u = v\}; R \vdash E \cup \{u{\downarrow_S} = v{\downarrow_S}\}; R$$
DELETE
$$E \cup \{u = v\}; R \vdash E; R \quad \text{if } u =_{\mathrm{AC}} v.$$
COMPOSE
$$E; R \cup \{u \to v\} \vdash E; R \cup \{u \to v'\} \quad \text{if } v \xrightarrow[R/S]{} v'.$$
SIMPLIFY
$$E \cup \{u = v\}; R \vdash E \cup \{u' = v\}; R \quad \text{if } u \xrightarrow[R/S]{} u'.$$
COLLAPSE
$$E; R \cup \{u \to v\} \vdash E \cup \{u' = v\}; R \quad \text{if } l \to r \in R,\ u \xrightarrow[l \to r/S]{\theta, p} u',\ (u,v) \succ (l, r\theta).$$

Figure 2: Inference rules of normalized completion

Definition 3.2 *An S-*normalized completion algorithm *is an algorithm which takes as input a set of equations E_0 and a reduction ordering $\succeq$ and produces a (finite or infinite) sequence $(E_n; R_n)$ where $R_0 = \emptyset$ and for all i, $E_i; R_i \vdash E_{i+1}; R_{i+1}$. Let:*

$$E_\infty = \bigcup_{n=0}^{\infty} \left(\bigcap_{i=n}^{\infty} E_i \right) \;,\; R_\infty = \bigcup_{n=0}^{\infty} \left(\bigcap_{i=n}^{\infty} R_i \right)$$

E_∞ and R_∞ are respectively the set of persisting equations and the set of persisting rules. We say that the algorithm fails *if E_∞ is not empty and* succeeds *otherwise, it* diverges *if the sequence is infinite.*

A derivation $E_0; R_0 \vdash E_1; R_1 \vdash \ldots$ is fair *if all persisting critical pairs are computed, i.e.*

$$\mathrm{CP}_T(R_\infty) \subseteq \bigcup_{i=0}^{\infty} E_i$$

A completion algorithm is fair if all sequences that it produces are fair.

In practice, it is worth to use the simplification rules as much as possible. This yields sets of rules which are inter-reduced, an important property as far as the uniqueness of the completion result is concerned.

The main theorem is the following one, proved by the proof normalization method (Bachmair et al., 1986; Bachmair and Dershowitz, 1989).

theory S	convergent system	$\Theta_S(u,v)$
ACU$(+,0)$	$x+0\to x$	$\{u\theta=v\theta \mid \theta=x\mapsto 0, u\trianglerighteq x+w\}$
ACI$(+)$	$x+x\to x$	$\mathrm{CP}_{\mathrm{AC}}(u\to v, x+x\to x)$
ACUI$(+,0)$	$x+x\to x,\ x+0\to x$	$\Theta_{\mathrm{ACU}}(u,v)\cup\Theta_{\mathrm{ACI}}(u,v)$
AC0$(*,0)$	$x*0\to 0$	$\{u\theta=v\theta \mid \theta=x\mapsto 0, u\trianglerighteq x*w\}$
ACN$(+,0)$	$x+x\to 0$	$\mathrm{CP}_{\mathrm{AC}}(u\to v, x+x\to 0)$

Figure 3: Set Θ_S for some simple theories

Theorem 3.3 *Assume we have an S-normalizing pair (Θ,Ψ) (w.r.t. $\succ_P$). Assume that the completion is fair and succeeds. Then for all s and t, $s =_{E_0\cup\mathrm{AC}\cup S} t$ if and only if*

$$s \xrightarrow[R_\infty/S]{*} u \xleftrightarrow[S]{*} v \xleftarrow[R_\infty/S]{*} t$$

In particular, $\to_{R_\infty/S}$ is confluent.

4 Applications

To use S-normalized completion for a given S, one has to defined a S-normalising pair. It is possible to define a general S-normalizing pair, for an arbitrary S (Marché, 1996) but this definition leads to unefficient computation. It is better to define "optimized" normalising pairs for the applications we have in mind.

4.1 Some simple theories

Figure 3 (where $s \trianglerighteq t$ means t is a subterm of s) shows definitions of Θ_S where S is either ACU, ACI, ACUI, AC0 or ACN, and in all these cases $\Psi_S(u,v)=\{u\to v\}$. Notice also that in the cases ACI and ACN, $\mathrm{CP}_{\mathrm{AC}}$ is the AC-critical pairs of the second rule *inside* the first: the inverse is not necessary since rules defining ACI and ACN have depth 1.

Proposition 4.1 *If S is either ACU, ACI, ACUI, AC0 or ACN, the above defined mappings Θ_S, Ψ_S are S-normalizing.*

4.2 Associativity

Another interesting case is when S is the theory of associativity of one symbol (in that case we don't have AC symbols anymore, but of course what we have done before is still valid!). For the canonical rewrite system S we can choose an orientation for associativity (by use of a lexicographic path ordering for example), let's say for example $S=\{(x*y)*z\to x*(y*z)\}$.

In that case we can avoid many useless critical pairs. The idea is that deduction between S and a rule of the form $s_1 * \cdots * s_n \to t$ will only produce one useful deduction, which can be seen as an A-extension of the original rule: $s_1 * \cdots * s_n * x \to t * x$ where x is a new variable. We have of course also to compute an optimized set Θ: it can be done avoiding call to unification by simply looking for subterms of the form $x * s$ where x is a variable.

Definition 4.2 *Let* $\Psi_{\mathrm{A}}(u, v) = \{u \to v\}$ *if* $\mathrm{Head}(u) \neq *$ *and*

$$\Psi_{\mathrm{A}}(u_1 * \cdots * u_n, v) = \begin{cases} u_1 * \cdots * u_n \to v \\ u_1 * \cdots * u_n * x \to v * x \end{cases}$$

otherwise. Let

$$\Theta_{\mathrm{A}}(u, v) = \{u\theta = v\theta \mid \theta = x \mapsto x * y, u \trianglerighteq x * w\}$$

Proposition 4.3 *The above defined mapping* $(\Theta_{\mathrm{A}}, \Psi_{\mathrm{A}})$ *is* A*-normalizing.*

Example 4.4 *Let M be the monoid $\langle a, b, c, d; ab^2 = c, bc = d\rangle$. We can build a canonical rewriting system for M by AU-normalised completion, the result obtained by our implementation CiME is*

$$\begin{cases} b.c \to d & b.(c.x) \to d.x \\ a.(b.b) \to c & a.(b.(b.x)) \to c.x \\ a.(b.d) \to c.c & a.(b.(d.x)) \to c.(c.x) \end{cases}$$

Of course, one has to remove the additional rules with variables (needed to "simulate" string rewriting) to obtain the equivalent string rewriting system $bc \to d$, $abb \to c$ $abd \to cc$. Notice that the same completion can be achieve directly by Knuth-bendix completion by giving it the equations of monoids above, but this leads to a longer computation : 25 critical pairs are computed (CPU time 1.567 seconds on a SPARC station 5) instead of 8 (in 0.883s).

4.3 Groups and symmetrization

For computing canonical rewrite systems for groups, a special technique called *symmetrization* have been introduced (Le Chenadec, 1986). These technique can be seen as a particular normalising pair when $S = G$, the group theory. We do not have room enough to discuss in detail this technique here but we refer to (Le Chenadec, 1986).

Example 4.5 *Let G be the diedral group of order 5 $\langle a, b; a^2 = 1, b^5 = 1, a.b = b^{-1}.a\rangle$. We can build a canonical rewriting system for G by G-normalised completion, the result is*

$$\begin{cases} a^{-1} \to a & b^{-1} \to b.(b.(b.b)) \\ a.a \to 1 & a.(a.x) \to x \\ a.b \to b.(b.(b.(b.a))) & a.(b.x) \to b.(b.(b.(b.(a.x)))) \\ b.(b.(b.(b.b))) \to 1 & b.(b.(b.(b.(b.x)))) \to x \end{cases}$$

this computes 38 critical pairs (in 8.267s) , whereas the same example by Knuth-bendix completion computes 198 critical pairs (in 20.383s).

4.4 Commutative symmetrization

When S contains at least Abelian group theory, we can optimize much further the normalizing pair by using commutative symmetrization. The idea is that in an equation $u_1 + \cdots + u_n = v_1 + \cdots + v_m$, we may move one term from one side to the other changing its sign. We use the abbreviation nt for $\underbrace{t + \cdots + t}_{n \text{ times}}$.

Definition 4.6 *The* symmetrization *of a pair (u, v) is obtained in the following way: let w be the AG-normal form of $u + (-v)$, written as $w = n_1 w_1 + \cdots + n_k w_k$, with $\forall j \geq 2$, $w_1 \succ w_j$. Then $\mathrm{sym}(u, v) = (n_1, w_1, -n_2 w_2 - \cdots - n_k w_k)$. If there is no maximum w_i, $\mathrm{sym}(u, v)$ is undefined.*

4.5 Abelian group theory

Definition 4.7 *For a pair (u, v) that has a symmetrization (n, s, t), let $\Psi_{\mathrm{AG}}(u, v) = s \to t$ if $n = 1$ and*

$$\Psi_{\mathrm{AG}}(u, v) = \begin{cases} ns \to t \\ -s \to (n-1)s + (-t) \end{cases}$$

if $n \geq 2$. Let $\Theta_{\mathrm{AG}}(u, v) = \Theta_{\mathrm{ACU}}(ns, t) \cup \Sigma_1(ns, t) \cup \Sigma_2(ns, t)$ where $\Sigma_1(u, v) = \mathrm{CP}_{\mathrm{AC}}(u \to v, x + (-x) \to 0)$ where superpositions are computed only in strict subterms of u, and $\Sigma_2(u, v) = \{u\sigma = v\sigma \mid \sigma = x \mapsto 0$ or $-y$ or $y + z$ if $u \trianglerighteq -x\}$. If (u, v) does not have a symmetrization, the equation $u = v$ will be considered as not orientable.

Example 4.8 *If $u = a + a + (-b) + c + c$ and $v = a + b + b + c$, then $u + (-v)$ normalizes to $a + (-b) + (-b) + (-b) + c$. If the ordering makes a greater than $(-b)$ and c, then $\mathrm{sym}(u, v) = (1, a, b+b+b+(-c))$ and $\Psi_{\mathrm{AG}}(u, v) = \{a \to b+b+b+(-c)\}$. If the ordering makes $-b$ greater than a and c, then $\mathrm{sym}(u, v) = (3, b, (-a)+(-c))$ and $\Psi_{\mathrm{AG}}(u, v) = \{b + b + b \to (-a) + (-c), -b \to b + b + a + c\}$.*

The pair above defined is AG-normalizing only if the ordering $\succ$ satisfy particular property w.r.t. the operators $+$ and $-$.

Proposition 4.9 *Let us assume that the term ordering satisfies the following property: for all terms u, v and w which do not have $+$, 0 or $-$ at the top, if $u \succ v$ and $u \succ w$ then $u \succ (-v) + w$. Then the pair (Θ_{AG}, Ψ_{AG}) defined above is AG-normalizing.*

An ordering satisfying the property above could be for example a precedence-based ordering (like the recursive path ordering if there is only $+$ as AC symbol, or the associative path ordering and its extensions if there are other AC symbols), with a precedence $- > + > 0$ and all other symbols greater than $-$.

This symmetrization technique improves a lot over standard AC completion when the set of equations to complete contains Abelian group theory.

Example 4.10 *Consider the Abelian group G presented by $E = \{2a - 3b + c = 0, -3a + 2b + 3c = 0, 2a + 2b - 2c = 0\}$ (Lankford et al., 1984). We give the set of equations above to the AG-normalized completion algorithm, and the result is $\{b \to 9a, c \to 25a, 30a \to 0, -a \to 29a\}$. AG-normalized completion computes only 13 critical pairs (in 5.10s), whereas the AC completion of $E \cup \mathrm{AG}$ computes 1531 critical pairs (in 278.0s).*

It is possible to apply the symmetrization technique to normalized completion modulo commutative ring theory, Boolean ring theory, and also to theories defining finite fields. Unfortunately, there is no convergent system for fields theory, because of the negative conditional equation $x * x^{-1} = 1$ if $x \neq 0$.

4.6 Commutative ring theory

Definition 4.11 *For a pair (u, v) that has a symmetrization (n, s, t), let $\Psi_{\mathrm{CR}}(u, v) = \{s \to t\}$ if $n = 1$ and*

$$\Psi_{\mathrm{CR}}(u, v) = \left\{ \begin{array}{ll} ns \to t & n(x * s) \to x * t \\ -s \to (n-1)s + (-t) & -(x * s) \to (n-1)(x * s) + -(x * t) \end{array} \right.$$

if $n \geq 2$. Let

$$\begin{array}{rcl} \Theta_{\mathrm{CR}}(u, v) & = & \Theta_{\mathrm{ACU}(+,0)}(ns, t) \cup \Theta_{\mathrm{ACU}(*,1)}(ns, t) \cup \Theta_{\mathrm{AC0}(*,0)}(ns, t) \cup \\ & & \Sigma_1(ns, t) \cup \Sigma_2(ns, t) \cup \Sigma_3(ns, t) \end{array}$$

where $\Sigma_1(u, v)$ and $\Sigma_2(u, v)$ are the same as for AG, and $\Sigma_3(u, v) = \{u\sigma = v\sigma \mid \sigma = x \mapsto x + y \text{ or } -x \text{ if } u \trianglerighteq x * z\}$.

Proposition 4.12 *If we assume the term ordering satisfying the required property for symmetrization, then the pair $(\Theta_{\mathrm{CR}}, \Psi_{\mathrm{CR}})$ defined above is* CR*-normalizing.*

Now we show an example of Gröbner basis computation using normalized completion. When polynomials have integer coefficients, computing a Gröbner basis amounts to normalized completion modulo commutative ring theory.

Example 4.13 *To compute a Gröbner basis of the ideal $(2X^2Y - Y, 3XY^2 - X)$ over $\mathbb{Z}$ (Kandri-Rody and Kapur, 1984) we give to CR-normalized completion the* *set of equations* $\{2XXY - Y = 0, 3XYY - X = 0\}$ *where* X, Y *are two constants,* $Y > X$ *in the precedence. The completion will produce:*

$$\left\{\begin{array}{ll} 2XXY \to Y & 2XXYx \to Yx \\ -XXY \to XXY - Y & -XXYx \to XXYx - Yx \\ XXYY \to XX - YY & \\ 3YY \to 2XX & 3YYx \to 2XXx \\ -YY \to 2YY - 2XX & -YYx \to 2YYx - 2XXx \\ 2XXX \to X & 2XXXx \to Xx \\ -XXX \to XXX - X & -XXXx \to XXXx - Xx \end{array}\right.$$

which corresponds to the Gröbner basis $\{2X^2Y - Y, X^2Y^2 - X^2 + Y^2, 3Y^2 - 2X^2, 2X^3 - X\}$. *CR-normalized completion computes 90 critical pairs (in 36.95s) whereas AC-completion computes 1990 critical pairs (in 9mn 22s).*

4.6.1 Boolean ring theory

Boolean ring theory BR is defined by

$$\left\{\begin{array}{lll} x + 0 \to x & x + x \to 0 & x * (y + z) \to (x * y) + (x * z) \\ -x \to x & x * 0 \to 0 & x * 1 \to x \end{array}\right.$$

In this case, the definition of normalizing pair is simpler since rules $x + x \to 0$ and $-x \to x$ implies that after a symmetrization, the leading coefficient will always be 1. It is not necessary then to have more than one rule in Ψ_{BR}.

Definition 4.14 *For a pair* (u, v) *that has a symmetrization* $(1, s, t)$, *let* $\Psi_{\mathrm{BR}}(u, v) = \{s \to t\}$. *Let*

$$\begin{array}{rcl} \Theta_{\mathrm{BR}}(u, v) & = & \Theta_{\mathrm{ACU}(+,0)}(s, t) \cup \Theta_{\mathrm{ACU}(*,1)}(s, t) \cup \Theta_{\mathrm{AC0}(*,0)}(s, t) \cup \\ & & \Theta_{\mathrm{ACN}(+)}(s, t) \cup \Sigma_4(ns, t) \end{array}$$

where $\Sigma_4(u, v) = \{u\sigma = v\sigma \mid \sigma = x \mapsto x + y \text{ if } u \trianglerighteq x * z\}$.

Proposition 4.15 *The pair* $(\Theta_{\mathrm{BR}}, \Psi_{\mathrm{BR}})$ *defined above is* BR*-normalizing.*

In that case, normalized completion is suitable for computing Boolean Gröbner bases.

4.6.2 Finite fields theory

Finite field theory $\mathrm{FF}(p)$ for a prime number p is defined by

$$\left\{\begin{array}{lll} x + 0 \to x & x * (y + z) \to (x * y) + (x * z) & x * 0 \to 0 \\ x * 1 \to x & -x \to (p - 1)x & px \to 0 \end{array}\right.$$

We can use in these case an other kind of symmetrization, which will act similarly as taking the inverse of the leading coefficient: if we have an equation of the form $ns = t$ then we would like to symmetrize it into $s \to (n^{-1} \bmod p)t$ in some sense. In the finite field $\mathrm{FF}(p)$ we do not need an inverse operator to do that because $n^{-1} \equiv n^{p-2} \pmod p$. Formally speaking in terms of equational algebra, we know that for any terms s and t, and any positive integer n, the equation $s = \overline{n}t$ (where $\overline{n}$ denotes $n^{p-2} \bmod p$) is an equational consequence of $ns = t$ and $\mathrm{FF}(p)$.

Definition 4.16 *For a pair (u, v) that has a symmetrization (n, s, t), let $\Psi_{\mathrm{FF}}(u, v) = \{s \to \overline{n}t\}$. Let*

$$\begin{aligned} \Theta_{\mathrm{FF}}(u,v) &= \Theta_{\mathrm{ACU}(+,0)}(s,\overline{n}t) \cup \Theta_{\mathrm{ACU}(*,1)}(s,\overline{n}t) \cup \Theta_{\mathrm{AC0}(*,0)}(s,\overline{n}t) \cup \\ & \quad \Sigma_4(s,\overline{n}t) \cup \Sigma_5(s,\overline{n}t) \end{aligned}$$

where $\Sigma_4(u, v)$ is the same as for CR, and $\Sigma_5(u, v) = \mathrm{CP}_{\mathrm{AC}}(u \to v, px \to 0)$ where superpositions are computed only in strict subterms of u.

Proposition 4.17 *The pair $(\Theta_{\mathrm{FF}}, \Psi_{\mathrm{FF}})$ defined above is $\mathrm{FF}(p)$-normalizing.*

We see now an example which shows how to compute Gröbner bases of polynomial ideals over $\mathrm{FF}(p)$ with $\mathrm{FF}(p)$-normalized completion, more efficiently than with AC-completion (it has already been remarked by Bündgen that computation of such a Gröbner basis can be done by AC completion (Bündgen, 1991a)).

Example 4.18 *The same example as above over $\mathbb{F}_5$ produces*

$$XXX \to 3X \qquad YXX \to 3YY \qquad YY \to 4XX$$

which corresponds to the Gröbner basis $\{X^3 - 3X, X^2Y - 3Y^2, Y^2 - 4X^2\}$. FF(5)-normalized completion computes 12 critical pairs (in 6.85s) whereas AC-completion computes 562 critical pairs (in 2mn33s)

Of course, programs dedicated to computation of Gröbner bases are more efficient, but that's not surprising at all, since our method is supposed to deal with a much wider class of problems.

5 Termination issues

Now, we investigate termination issues of the completion process: it is well-known that Buchberger's algorithm always terminates, whereas Knuth-Bendix may not. Of course, the difference between the two algorithms is that when Buchberger's algorithm is viewed as a particular case of completion, the initial set of equations to complete is *ground*, *i.e.* has no first-order variables. In that particular case of ground completion, termination is already known for Knuth-Bendix completion but also for AC-completion (Narendran and Rusinowitch, 1991; Marché, 1991). We extend now this result to S-normalized ground completion for some interesting theories S. An interesting consequence is that now, the termination results of

Buchberger's algorithm and AC-completion are particular cases of this new termination result. For details about this proof, we refer to (Marché, 1996). Let us simply give now the essential.

First of all, we have to insure that completion will never fail, hence we have to use a total ordering. In the case of simple theories ACU, ACI, ACUI, AC0 and ACN, we can use the total AC ordering of Narendran and Rusinowitch (Narendran and Rusinowitch, 1991) or the one of Nieuwenhuis and Rubio (Nieuwenhuis and Rubio, 1993) (with the condition $+ > 0$ for ACN, in order to orient $x + x \to 0$). In the case of Abelian group theory (AG), we can use a recursive path ordering with a total precedence of the form $\mathcal{F} > - > + > 0$, and such that $+$ has multi-set status and all other operators have lexicographic status (for totality). In the case of commutative ring theory (CR), it is a bit more complicated since the total AC orderings above always orient distributivity in the wrong way! The solution is to use the lexicographic extension of the *modified associative path ordering* (Delor and Puel, 1993) with precedence $1 > * > - > + > 0$, and then any total AC-compatible ordering (Marché, 1995). Such an ordering can be used also for Boolean ring (BR) and finite fields theories FF(p).

Second, we have to assume a condition on the completion strategy: completion should always perform simplification in priority, that is the simplification rules Normalize, Delete, Simplify, Compose and Collapse have priority on Orient and Deduce. This condition on the strategy is essential, otherwise completion could diverge whereas R_∞ is finite indeed (Marché, 1991). The main result is then the following.

Theorem 5.1 *If the initial set of equations is ground, then the S-normalized completion terminates when S is either AC, ACU, ACI, ACUI, AC0, ACN, AG, CR, BR or FF(p). As a consequence, every equational theory presented by $C \cup S$, where C is a set of ground equations and S is one of the previous theories, has a finite S-normalized canonical rewriting system, in particular it has a decidable word problem.*

6 Conclusion

We obtained a new completion algorithm, which generalizes the already known completion modulo an equational theory. It has also the interesting property that it unifies Knuth-Bendix completion (and its extensions AC completion, ACU-constrained completion) and Buchberger algorithm for computing Gröbner bases.

Some interesting questions are now open for future work:

- Is it possible to embed the computation of a Gröbner basis of a polynomial ideal with coefficients in an infinite field like $\mathbb{Q}$, in an S-normalized completion for a well-chosen S?
- How is it possible to carry known critical pair criteria from Buchberger's algorithm to equational completion, or in the other way?

- Non-commutative Gröbner bases can also be computed by R-normalized completion, where R is the canonical rewrite system for non-commutative rings. Are there some interesting contributions to the problem of non-commutative Gröbner bases by use of this technique of normalized rewriting?

References

Bachmair, L. and Dershowitz, N. (1989). Completion for rewriting modulo a congruence. *Theoretical Computer Science*, 67(2&3):173–201.

Bachmair, L., Dershowitz, N., and Hsiang, J. (1986). Orderings for equational proofs. In *Proc. 1st IEEE Symp. Logic in Computer Science, Cambridge, Mass.*, pages 346–357.

Bachmair, L. and Ganzinger, H. (1994). Buchberger's algorithm: A constraint-based completion procedure. In Jouannaud, J.-P., editor, *First International Conference on Constraints in Computational Logics*, volume 845 of *Lecture Notes in Computer Science*, pages 285–301, München, Germany. Springer-Verlag.

Bachmair, L. and Plaisted, D. A. (1985). Termination orderings for associative-commutative rewriting systems. *Journal of Symbolic Computation*, 1(4):329–349.

Baird, T., Peterson, G., and Wilkerson, R. (1989). Complete sets of reductions modulo Associativity, Commutativity and Identity. In *Proc. 3rd Rewriting Techniques and Applications, Chapel Hill, LNCS 355*, pages 29–44. Springer-Verlag.

Ben Cherifa, A. and Lescanne, P. (1986). An actual implementation of a procedure that mechanically proves termination of rewriting systems based on inequalities between polynomial interpretations. In Siekmann, J. H., editor, *8th International Conference on Automated Deduction*, Lecture Notes in Computer Science vol. 230, pages 42–51, Oxford, England. Springer-Verlag.

Book, R. V., editor (1991). *4th International Conference on Rewriting Techniques and Applications*, volume 488 of *Lecture Notes in Computer Science*, Como, Italy. Springer-Verlag.

Buchberger, B. (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).

Buchberger, B. (1987). History and basic features of the critical pair / completion procedure. *Journal of Symbolic Computation*, 3(1).

Buchberger, B. and Loos, R. (1982). Algebraic simplification. In *Computer Algebra, Symbolic and Algebraic Computation. Computing Supplementum 4*. Springer-Verlag.

Bündgen, R. (1991a). Simulating Buchberger's algorithm by a Knuth-Bendix completion procedure. In (Book, 1991).

Bündgen, R. (1991b). *Term Completion versus Algebraic Completion*. PhD thesis, Universität Tübingen.

Delor, C. and Puel, L. (1993). Extension of the associative path ordering to a chain of associative-commutative symbols. In (Kirchner, 1993).

Dershowitz, N. (1987). Termination of rewriting. *Journal of Symbolic Computation*, 3(1):69–115.

Dershowitz, N. and Jouannaud, J.-P. (1990). Rewrite systems. In van Leeuwen, J., editor, *Handbook of Theoretical Computer Science*, volume B, pages 243–309. North-Holland.

Jouannaud, J.-P. and Kirchner, H. (1986). Completion of a set of rules modulo a set of equations. *SIAM Journal on Computing*, 15(4):1155–1194.

Jouannaud, J.-P. and Marché, C. (1992). Termination and completion modulo associativity, commutativity and identity. *Theoretical Computer Science*, 104:29–51.

Kandri-Rody, A. and Kapur, D. (1984). An algorithm for computing the Gröbner basis of a polynomial ideal over an Euclidean ring. Technical Report 84CRD045, CRD, General Electric Company, Schenectady, New-York.

Kandri-Rody, A., Kapur, D., and Winkler, F. (1989). Knuth-Bendix procedure and Buchberger algorithm — a synthesis. In *Proc. of the 20th Int. Symp. on Symbolic and Algebraic Computation, Portland, Oregon*, pages 55–67.

Kirchner, C., editor (1990). *Unification*. Academic Press.

Kirchner, C., editor (1993). *5th International Conference on Rewriting Techniques and Applications*, volume 690 of *Lecture Notes in Computer Science*, Montreal, Canada. Springer-Verlag.

Kirchner, C. and Kirchner, H. (1989). Constraint equational reasoning. In *Proc. of the 20th Int. Symp. on Symbolic and Algebraic Computation, Portland, Oregon*.

Knuth, D. E. and Bendix, P. B. (1970). Simple word problems in universal algebras. In Leech, J., editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press.

Lankford, D., Butler, G., and Ballantyne, A. (1984). A progress report on new decision algorithms for finitely presented abelian groups. In Shostak, R. E., editor, *7th International Conference on Automated Deduction*, Lecture Notes in Computer Science vol. 170, Napa, California, USA. Springer-Verlag.

Le Chenadec, P. (1986). *Canonical forms in finitely presented algebras.* Pitman, London.

Loos, R. (1981). Term reduction systems and algebraic algorithms. In *Proceedings of the Fifth GI Workshop on Artificial Intelligence*, pages 214–234, Bad Honnef, West Germany. Available as *Informatik Fachberichte*, Vol. 47.

Marché, C. (1991). On ground AC-completion. In (Book, 1991).

Marché, C. (1995). Associative-commutative reduction orderings via head-preserving interpretations. Technical Report 95–2, LIFAC, E.N.S. de Cachan.

Marché, C. (1996). Normalized rewriting: an alternative to rewriting modulo a set of equations. *Journal of Symbolic Computation.* to appear.

Narendran, P. and Rusinowitch, M. (1991). Any ground associative-commutative theory has a finite canonical system. In (Book, 1991).

Nieuwenhuis, R. and Rubio, A. (1993). A precedence-based total AC-compatible ordering. In (Kirchner, 1993).

Pottier, L. (1989). Algorithmes de complétion et généralisation en logique du premier ordre. Thèse de doctorat. Université de Nice.