REWRITING METHODS FOR WORD PROBLEMS*

NACHUM DERSHOWITZ
Department of Computer Science, University of Illinois at Urbana-Champaign,
1304 West Springfield Ave., Urbana, IL 61801–2987, U.S.A.

Abstract

This paper outlines various recent approaches to solving word problems. Term orderings are used to define a terminating rewrite relation. When confluent, that relation defines unique normal forms that can be used to decide word problems. Some results obtained by these methods are summarized.

1. Introduction

The central idea of rewriting is to impose directionality on the use of equations in proofs. A *rewrite rule* is an ordered pair of terms, written $l \to r$. Like equations, rules are used to replace instances of l by corresponding instances of r; unlike equations, rules are not used to replace instances of the right-hand side r. For any given set R of rules, the *rewrite relation* $\to_R$ is the closure of R (viewed as a binary relation) under the "replacement" property (within any context) and "fully invariant property" (under any substitution). In other words, $s \to_R t$ if s contains a subterm that is an instance $l\sigma$ of l, for some rule $l \to r$ in R, and t is s with that subterm replaced by $r\sigma$. The final result of a unextendible sequence of rewrites is called a *normal form.* In Section 2, we consider *ordered* rewriting, using, instead, rewrite relations that are defined in terms of a set of unordered equations and a well-founded partial ordering.

Systems of rewrite rules are used to check for equality in an equational theory by rewriting both sides of the identity in question to normal form. Hence, one of the most essential properties a rewrite system can enjoy is *unique normalization*, by which is meant that every term possesses exactly one normal form. If all derivations (sequences of rewrites) lead to a normal form, the system is called *terminating;* if all derivations lead to a unique normal form, the system is called *convergent* (or *canonical*, or *complete*). Computations with convergent systems, starting with equal terms, always terminate in the same unique normal form. Such a system serves as a procedure for deciding whether two terms are equal in the equational theory defined by the rules (viewed as equations), and, in particular, solves the word problem for that free theory.

*This research was supported in part by the U. S. National Science Foundation under Grant CCR-90-07195.

An equational proof of $s = t$ in the equational theory underlying an arbitrary rewrite system R (treating its rules as equational axioms) is any derivation of the form $s \leftrightarrow_R^* t$, where $\leftrightarrow_R^*$ is the reflexive-symmetric-transitive closure of $\rightarrow_R$. A *rewrite* proof of $s = t$ takes the specific "valley" form $s \rightarrow_R^* v \leftarrow_R^* t$, in which the same term v is reached by rewriting s and t any number of times. When such a "direct" proof exists for any consequence of the equations represented by a system R, the system is called *Church-Rosser*, or *confluent*. Convergent systems are Church-Rosser. Though the Church-Rosser property is undecidable, in 1970, Knuth[1] devised an effective *superposition* test (based on "critical overlaps") to decide whether a terminating system is convergent. But termination itself is undecidable.

The basic idea is to decide $s \leftrightarrow_R^* t$ (that is, provability of $s = t$ in the underlying theory of R) by computing the normal forms of s and t, and checking if the two results are identical—since, for convergent R, $s \leftrightarrow_R^* t$ iff they have identical normal forms. Normalization is a function for uniquely-normalizing systems, in general, and for convergent systems, in particular. For this function to be computable, we need reducibility (applicability of any rule) to be recursive, which is certainly the case when R is finite. Thus, if R is a finite convergent system, the variety it defines has a solvable free word problem. This rewrite-system method of deciding validity of identities is extended in Section 3 to ordered rewriting. For more on rewriting, see, for example, the recent survey by Dershowitz and Jouannaud.[2]

Rewriting methods have turned out to be among the more successful approaches to equational theorem proving. Of course, not all word problems can be solved by rewriting: some theories are not finitely based, and some finitely-based equational theories are undecidable. But, at least, any theory with decidable word problem can be solved by rewriting with an ordered system for some conservative extension of the theory.[3]

Many rewrite-system decision procedures are known; perhaps the first such procedure for a word problem was Trevor Evans', for "loops".[4] In their seminal paper,[1] Knuth and Bendix (building on the work of Evans) demonstrated how failure of the superposition test suggests additional rules that can be used to help "complete" a nonconvergent system. Completion utilizes an ordering on terms to provide guidance in the generation of new rules and to direct the simplification of equations. *Ordered completion*, a powerful extension of this method, developed by Brown,[5] Lankford,[6] Hsiang and Rusinowitch,[7] and Bachmair, Dershowitz, and Plaisted,[8] is the subject of Section 4.

When a rewriting decision procedure exists, it can be very effective. Completion itself can also sometimes be used to uniformly decide word problems for finitely-presented varieties, such as Abelian groups. Pedersen[9] has described the applications of rewriting techniques and completion methods to word problems; we concentrate here on the ordered-rewriting approach. Some successful applications of both approaches are listed in Section 5.

Lankford[6] proposed that completion-like methods be incorporated in resolution-

based theorem provers for the first-order predicate calculus, with paramodulation used for unorientable equations. Completion-based methods can also be used as first-order theorem-provers, by working, for example, in Boolean rings, as first suggested by Hsiang.[10] Using ordered completion, this gives a complete theorem proving strategy, based on term orderings, for first-order predicate calculus; this extension is covered briefly in Section 6.

2. Rewriting

The subterm of a term t at position π is denoted $t|_\pi$, and $t[s]_\pi$ is used to denote that term with $t|_\pi$ replaced by s. A well-ordering $>$ of ground terms is called a *complete simplification ordering* if it has the replacement property ($s > s'$ implies $t[s]_\pi > t[s']_\pi$). Such a ground-term ordering must also have the subterm property ($t \geq t|_\pi$ for all subterms $t|_\pi$ of t). For any complete simplification ordering $>$ on ground terms, one can define a partial ordering $\succ$ on terms with variables by interpreting $s \succ t$ as meaning $s\sigma > t\sigma$ for all ground substitutions σ. The resultant ordering is well-founded, and has the replacement and subterm properties. No such ordering can be total on all (first-order) terms, since no two distinct variables x and y can be ordered. (Were one to have $x \succ y$, then $y \succ x$, as well, the latter being an instance of the former.)

The polynomial and path orderings commonly used in rewrite-based theorem provers[11] can all be extended to complete simplification orderings. An important example is Kamin and Lévy's[12] lexicographic variant of the recursive path ordering;[13] it is a complete simplification ordering whenever the operators have a total precedence ordering. In this ordering, a term $s = f(s_1, \ldots, s_m)$ is greater than another term $t = g(t_1, \ldots, t_n)$ if any subterm of s is greater than t (in the same ordering), if f is greater than g in the precedence and s is greater than all subterms of t, or if $f = g$, $m = n$, and the n-tuple $\langle s_1, \ldots, s_m \rangle$ is lexicographically greater than $\langle t_1, \ldots, t_n \rangle$ (with subterms compared recursively in the same path ordering). This *lexicographic path ordering* has decidability properties that make it ideal for use in theorem proving. In particular, the above-described extension of the complete ground ordering to all terms is decidable.[14]

An *ordered-rewriting system* comes in two parts: a partial ordering and a set of equations. Let $\succ$ be a well-founded strict partial ordering on first-order terms, with the full-invariance and replacement properties; this means that $t[s\sigma]_\pi \succ t[s'\sigma]_\pi$ whenever $s \succ s'$ (for all terms s, s', t, positions π in t, and substitutions σ). Let e be an equation; the rewriting relation $\rightarrow_e$ is the intersection of the two relations, $\leftrightarrow_e$ (one single equation step via e) and $\succ$. Thus, an equation may be used to rewrite in whichever direction agrees with the given ordering. Similarly, for a set of (unordered) equations E, $\rightarrow_E = \leftrightarrow_E \cap \succ$. (The variables occurring on the two sides of an equation need not coincide for termination of $\rightarrow_E$, since the terms substituted for the variables

will have to be such that the rewritten term is smaller vis-a-vis the ordering $\succ$.) If the ordering $\succ$ is total on ground (variable-free) terms then each ground instance of an equation in E can be oriented one way or another. Suppose, for example, that "$\cdot$" is commutative and that $(a \cdot b) \cdot z$ is greater than $(b \cdot a) \cdot z$ in the ordering (using a, b, etc. for constants and x, y, and z for variables). Then one would rewrite $c \cdot ((a \cdot b) \cdot d)$ to $c \cdot ((b \cdot a) \cdot d)$, but not vice-versa.

To get a taste of the general approach, consider first how rewriting can be used to transform a (finite) set E of ground equations into a simple decision procedure for E.[6] This transformation can be expressed as the following set of two "inference rules" which use a complete simplification ordering to *replace* equations with simpler ones, without changing the theory:

$$\begin{array}{rcl} \{u = u\} & \vdash & \emptyset \\ \{e, r\} & \vdash & \{e', r\} \quad \text{if } e \ \rightarrow_r \ e' \end{array}$$

The first rule deletes trivial equations; the second rewrites an existing equation e using another equation r and the given complete simplification ordering (implicit in $\rightarrow_r$).

These inference rules apply to any subset of the current set of equations. Regardless of the order in which things are done, this process terminates. Done right, it "reduces" E in time proportional to $n \log n$, where n is the number of symbols in E.[15] The final system R reduces any term t to normal form in no more steps than symbols in t. An equation e holds in E iff the result R reduces e to a trivial equation of the form $u = u$.

For example, if 0 is a constant and s, a unary operator, these rules have the following effect:

$$\begin{array}{rcl} \{sss0 = 0,\ ss0 = 0,\ ssss0 = s0\} & \vdash & \{sss0 = 0,\ ss0 = 0,\ s0 = s0\} \\ & \vdash & \{sss0 = 0,\ ss0 = 0\} \\ & \vdash & \{s0 = 0,\ ss0 = 0\} \\ & \vdash & \{s0 = 0\} \end{array}$$

The result is the one-rule rewrite system, $s0 \rightarrow 0$, which reduces all terms s^i0 to normal form 0.

3. Confluence

Turning to the general case, when the axioms in E have variables, the same method, viz. reducing to normal form with an equivalent confluent system R, can be attempted. As long as reducibility is recursive (in particular when R is finite), a confluent ordered-rewriting system computes unique normal forms. *Skolemizing* the variables in an equation $s = t$ (that is, treating variables as constants for the purpose

of rewriting), and reducing both sides to normal form, gives identical normal forms iff $R \models s = t$.

For example, consider the following system for entropic groupoids:[7]

$$\begin{array}{rcl} (x \cdot y) \cdot x & \to & x \\ x \cdot (y \cdot z) & \to & x \cdot z \\ ((x \cdot y_1) \cdot y_2) \cdot z & \to & x \cdot z \\ (x \cdot y_1) \cdot z & \leftrightarrow & (x \cdot y_2) \cdot z \end{array}$$

The symbol $\to$ will be used for equations such that all instances of the left-hand side are greater than corresponding instances of the right-hand side, and $\leftrightarrow$, otherwise. The last equation is used to rewrite any product of the form $(x \cdot y_1) \cdot z$ to the same term with the occurrence of y_1 replaced by a sufficiently small term in the ordering $>$, assuming such a term can be found. (In this example, it matters little which complete simplification ordering is used.) It is as though the system were phrased as follows:

$$\begin{array}{rcll} (x \cdot a) \cdot x & \to & x & \\ x \cdot (y \cdot z) & \to & x \cdot z & \\ ((x \cdot a) \cdot a) \cdot z) & \to & x \cdot z & \\ (x \cdot y) \cdot z & \to & (x \cdot a) \cdot z & \qquad (y \neq a) \end{array}$$

where a is a new constant smaller than any term. (There is no need to consider instances other than $y = a$ of the original first and third rules, since terms to which they would apply can first be rewritten by the last rule.)

To prove validity of the identity $(x \cdot y_1) \cdot (y_2 \cdot z) = (x \cdot y_2) \cdot (y_1 \cdot z)$, say, the normal forms of $(a \cdot b) \cdot (c \cdot d)$ and $(a \cdot c) \cdot (b \cdot d)$ are computed. Suppose a is smaller than any other term; then we can compute normal forms as indicated above. From $(a \cdot b) \cdot (c \cdot d)$, one gets $(a \cdot b) \cdot d$ by the second rule and then $(a \cdot a) \cdot d$ by the last. Since the same normal form is obtained from $(a \cdot c) \cdot (b \cdot d)$, the identity holds. (There is a presumption here that $>$ is also a complete simplification ordering for ground terms in the expanded language, and also that one can find ground terms of a specified form, bounded by a given ground term.)

Adapting the Critical Pair Lemma of Knuth[1] gives a test for confluence of ordered-rewriting systems. Let $l = r$ and $s = t$ be two (not necessarily distinct) equations (with variables made disjoint) such that l "overlaps" s at nonvariable position π with most general unifying substitution μ; that is, $l\mu = s\mu|_\pi$. The equation $t\mu = s\mu[r\mu]_\pi$ is a *critical pair* of the two equations if $t\mu\gamma \leftarrow_{s=t} s\mu\gamma \to_{l=r} s\mu\gamma[r\mu\gamma]_\pi$ for some substitution γ. Let $cp(E)$ denote the set of all critical pairs between equations in a set E. Lankford[6] showed that for any set of equations E, complete simplification ordering $>$, and peak $s \leftarrow_E u \to_E t$ between ground terms s, t, u, there either exists a rewrite proof $s \to_E^* v \leftarrow_E^* t$ or a critical-pair proof $s \leftrightarrow_{cp(E)} t$. Hence, if each ground instance $l\sigma = r\sigma$ of a critical pair has a rewrite proof, then every peak $s \leftarrow_E u \to_E t$ between ground terms has one, and (by Newman's Lemma[16]) the rewrite relation $\to_E$ is Church-Rosser for ground terms.

Though for finite E, there are a finite number of critical pairs, the question is how, in general, can one check that all ground instances admit rewrite proofs. In the above example, the critical pair $x = (x \cdot y_2) \cdot x$ (one of several between the first two rules) always reduces, by another application of the first rule, to $x = x$. On the other hand, the critical pair $((x \cdot x_2) \cdot y_1) \cdot z = ((x \cdot x_1) \cdot y_2) \cdot z$ can be rewritten to $((x \cdot x_2) \cdot y_2) \cdot z = ((x \cdot x_2) \cdot y_2) \cdot z$ by the fourth rule, but only if $x_1 > x_2$ and $y_1 > y_2$. Fortunately, these inequalities must hold for the critical pair to have arisen in the first place. In this case, there is actually no problem at all finding a rewrite proof, since the third rule can be used instead to show that both sides have normal form $x \cdot z$. But more sophisticated techniques may be needed, in general.

4. Completion

Knuth's "standard" completion procedure[1] works on equations that can be oriented into rules that always apply in one direction. Therefore, it gets stuck when it encounters an unorientable equation, like commutativity. Lankford's theorem-proving method,[6] treats unorientable equations as bidirectional; Hsiang and Rusinowitch[7] use a complete simplification ordering to orient specific instances of unorientable equations whenever possible. This ordering is used both to limit deduction (generation of critical pairs) and to increase simplification (reduction of critical pairs). Bachmair, Dershowitz, and Plaisted[8] allow more simplification of equations and supply completion with any "approximation" (with the replacement property) that is contained (as a relation) in the complete simplification ordering. The notion of ordered rewriting that is used here is somewhat more general, when the ordering is partial, than in these previous works. In standard completion, by comparison, the ordering need not extend to a complete simplification ordering (though in practice it virtually always does).

Let $>$ be a complete simplification ordering on ground terms, $\succ$ be an ordering on all terms that approximates $>$, and $\trianglerighteq$ be the encompassment ordering in which a term is greater than its subterms and smaller than its instances. Consider the following set of inference rules, operating on an initial set of equations E_0:

$$
\begin{array}{rcll}
E & \vdash & E \cup \{e\} & \text{if } e \in cp(E) \\
\{u = u\} & \vdash & \emptyset & \\
\{l = r, u = v[l\sigma]_\pi\} & \vdash & \{l = r, u = v[r\sigma]_\pi\} & \text{if } v[l\sigma]_\pi \succ v[r\sigma]_\pi \text{ and} \\
 & & & u \succ v[r\sigma]_\pi \text{ or } v[l\sigma]_\pi \trianglerighteq l
\end{array}
$$

The first rule generates critical pairs; the other two simplify them. The equation $l = r$ is only allowed to simplify $u = v$ if it is applied to the smaller of u and v, or l (the larger side of $l = r$) is strictly more general (in the encompassment ordering) than the larger of u and v.

Ordered completion either constructs a finite ordered-rewriting system that is convergent for ground terms in finite time, or else generates an infinite system. With

a finite system, validity can be decided by rewriting; an infinite system serves as a semi-decision procedure. For example, given equational axioms for Abelian groups and a suitable ordering, ordered completion generates the following system:[17, 18]

$$\begin{array}{rcl@{\qquad}rcl} 1^- & \to & 1 & x \cdot y & \leftrightarrow & y \cdot x \\ x \cdot 1 & \to & x & x \cdot (y \cdot z) & \leftrightarrow & y \cdot (x \cdot z) \\ 1 \cdot x & \to & x & (x \cdot y) \cdot z & \to & x \cdot (y \cdot z) \\ (x^-)^- & \to & 1 & x \cdot (x^- \cdot z) & \to & z \\ x \cdot x^- & \to & 1 & (y \cdot x)^- & \to & x^- \cdot y^- \end{array}$$

It can be shown that if all critical pairs that persist forever are accounted for, then a rewrite proof between two ground terms will eventually be generated iff they are equal in the theory.[8] Thus, the word problem in arbitrary equational theories can always be semidecided by ordered completion. When completion terminates with a finite ordered-rewriting system, that system decides validity. Moreover, ordered completion must succeed in generating a convergent set of (one-way) rules whenever one exists with a rewrite relation conforming to a complete simplification ordering approximated by the ordering $\succ$ supplied to the completion procedure,[8] or if all axioms are linear (have at most one occurrence of a variable on each side).[19]

An alternative technique for handling "permutative" axioms like associativity and commutativity is to use a rule to rewrite any subterm that matches the left-hand side, where "match" is defined "up to" applications of those axioms. For example, if "$\cdot$" is commutative, then a rule with left-hand side $x \cdot 1$ would also apply to a product $1 \cdot b$. This, very successful approach (to the problem that Knuth left open) was initiated by Lankford and Ballantyne[17] and Peterson and Stickel.[20] More recent work by Jouannaud and Kirchner[21] and Bachmair and Dershowitz[22] extends this method further. There are a fair number of implementations of associative-commutative completion, including REVE,[23] KADS,[24] RRL,[25] and KB.[26] Current implementations of ordered completion are METIS,[27] SBREVE,[28] and HIPER;[29] the more sophisticated, "reasoning by cases" approach, alluded to at the end of the previous section, is being developed by Martin and Nipkow[18] and Peterson.[30]

Knuth[1] suggested treating unorientable equations by introducing new symbols into the vocabulary. For example, the equation $(x \cdot y_1) \cdot z = (x \cdot y_2) \cdot z$ might suggest a new rule $(x \cdot y) \cdot z \to x * z$, where the new operator "$*$" takes as its operands the variables x and z appearing on both sides of the problematic equation. This results in the following decision procedure for entropic groupoids, operating in the expanded vocabulary:[31]

$$\begin{array}{rcl@{\qquad}rcl} (x \cdot y) \cdot z & \to & x * z & x * x & \to & x \\ x * (y \cdot z) & \to & x * z & x \cdot (y * z) & \to & x \cdot z \\ (x \cdot y) * z & \to & x \cdot z & (x * y) \cdot z & \to & x \cdot z \\ (x * y) * z & \to & x * z & x * (y * z) & \to & x * z \\ x \cdot (y \cdot z) & \to & x \cdot z & & & \end{array}$$

This technique sometimes suggests meaningful "definitions", but often degenerates by coming up with infinitely many such operators. On the positive side, it is worth noting that any theory with decidable word problem can be solved by rewriting with some ordered system for some conservative extension of the theory.[3] (A similar result is obtainable for a two-phased version of rewriting, wherein normal forms of the first system are inputs to the second.[32] It is not, however, so for ordinary one-phase rewriting.[33])

5. Word Problems

Knuth and Bendix[1] used their original completion program to construct convergent systems for free groups, loops (this was the system found manually by Evans[4]), left and right groups, and central groupoids. Lankford and Ballantyne[17] extended completion to handle the most important nonterminating axioms and found an associative-commutative systems for free Abelian groups; Peterson and Stickel[20] used a similar process to derive systems for free commutative rings with unit and distributive lattices (and tried Boolean algebra without success). Hullot[34] used these procedures to derive systems for various quasigroups, associative and non-associative rings (but could not handle anticommutative or Lie rings), left and right A-modules, A-bimodules, A-rings, and A-algebras. Pedersen[31] used a variant of associative-commutative completion to generate an infinite system that "computes" Whitman normal form for lattices. The monograph by Benninghofen, Kemmerich and Richter[35] discusses rewriting techniques for semigroups, monoids, and groups. Book[36] surveys the monoid case, particularly when all rules are length-decreasing. The Gröbner basis approach to properties of polynomial ideals and solving the uniform word problem for finitely-presented commutative rings is similar to completion; see Buchberger's survey.[37]

Lescanne[38] experimented with various presentations of free groups. He applied Knuth's idea of handling unorientable equations by introducing new operators to Higman and Neumann's one-equation group presentation; completion then came up with definitions of identity and inverse. Pedersen[39, 31] used this technique to generate the extended system for entropic groupoids. His "morphocompletion" procedure, which automatically introduces new generators, solved many one relation monoid word problems.[40] The idea of deciding word problems by applying rewrite systems in phases, each rewriting normal forms of the previous one, appears in Bauer.[32]

Christian's[29] system incorporates completion for permutative equations with ordered completion, and also has heuristics for introducing new symbols; his experiments with Burnside groups and Grau's ternary Boolean algebra were only partially successful. Edelson[41] has used this system to (re-)discover syntactic proofs for properties of regular rings. Foret[42] has found rewrite-based decision procedures for several systems $(K, Q, T, S5)$ of propositional modal logic. Chou and Schelter[43] have

compared the power of the rewriting approach with other automated methods for geometry theorem proving.

An *absorbing* rule is one in which every nonvariable subterm on the left contains all the variables.[44] This ensures that its critical pairs with ground equations are still ground. With associative-commutative systems this is essentially never the case, but sometimes the nonground pairs are certain to reduce to ground ones. Ballantyne and Lankford[45] accordingly gave rewriting-based procedures for the uniform word problem in finitely-generated commutative semigroups; with Butler,[46] they did the same for finitely-generated Abelian groups; Lankford[47] used this method to show decidability of the uniform word problem in finitely-presented J-algebras. Pedersen[44] gave sufficient syntactic conditions for decidability of the uniform word problem for some absorbing systems, including finitely-presented loops, using a completion procedure that adds new symbols as needed. Even when the uniform word problem is undecidable, as for non-Abelian groups, this method frequently finds decision procedures for specific word problems; in this vein, Le Chenadec[48, 49] did substantial work on finitely-presented groups from topology and geometry, including the Coxeter groups (showing termination was sometimes difficult); Bündgen[50] used completion to generate systems for several other finitely-presented groups. The relation between completion and small cancellation theory was explored by Bücken,[51] Le Chenadec,[52] and Benninghofen, Kemmerich, and Richter.[35]

Burris and Lawrence[53] have presented systems for finite fields, rings with $x^n = x$ ($n = 2$ was done by Hsiang[54]), and such rings with n prime and $nx = 0$ (studied by Nipkow[55]). Kapur and Zhang[56] have used an enhanced associative-commutative completion procedure (which avoids many unnecessary critical pair computations) to prove commutativity for rings with $x^n = x$, for many specific n. Anatharaman and Hsiang[57] have used a combination of ordered and associative-commutative completion as a theorem prover to derive purely syntactic proofs of the Moufang identities for alternative rings.

The free algebra for n generators can be computed by generating critical pairs, looking at the normal forms of bigger and bigger words until they stabilize. Pedersen[9] did this for bands with up to three generators. Conditional rewriting, using conditional equations, can sometimes be used to capture an infinite number of unconditional rules in one conditional one. This approach was taken by Siekmann and Szabo[58] for bands. Pelin and Gallier[59] considered multiple-phase conditional rewriting. There is, however, no general-purpose mechanism for discovering conditional systems. There are some completion procedures for conditional equations,[60, 61, 62] which could in some instances provide decision procedures for quasi-varieties, but, in practice, they do not work well and further research is required.

6. First-Order Theories

The exclusive-or normal form of Zhegalkin[63] and Stone[64] for Boolean rings can be expressed as the following convergent system:[10, 30]

$$\begin{array}{rcl@{\qquad}rcl} -x & \to & x & x \cdot y & \leftrightarrow & y \cdot x \\ x \cdot 1 & \to & x & x + y & \leftrightarrow & y + x \\ 1 \cdot x & \to & x & (x \cdot y) \cdot z & \to & x \cdot (y \cdot z) \\ 0 \cdot x & \to & 0 & x \cdot (x \cdot y) & \to & x \cdot y \\ x \cdot x & \to & x & x \cdot (y \cdot x) & \to & x \cdot y \\ 0 + x & \to & x & (x + y) \cdot z & \to & (x \cdot z) + (y \cdot z) \\ x + x & \to & 0 & (x + y) + z & \to & x + (y + z) \\ x + (x + y) & \to & y & x + (y + x) & \to & y \end{array}$$

Ordered rewriting with a lexicographic path ordering (in which addition precedes multiplication and 0 is minimal) defines unique normal forms for ground terms. (Symmetric rules for $x \cdot 0$ and $x + 0$ are omitted, since commuting any such ground term gives terms that can be no bigger and to which the given rules apply.)

This system may be used with associative-commutative completion[54] or ordered completion[18] as a theorem prover in first-order predicate calculus, interpreting "$\cdot$" as conjunction and "$+$" as exclusive-or, and representing the negation of x by $x + 1$. (Burris[65] suggests using discriminator varieties in the rôle played here by Boolean rings.) Negating and Skolemizing an hypothesis, adding it to the above system, and completing—with ordered completion—is guaranteed (by Herbrand's Theorem) to produce a contradiction ($x = 0$), iff the theorem is valid.

Acknowledgement

I thank Leo Bachmair and François Bronsard for their comments.

References

[1] D. E. Knuth and P. B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297, Pergamon Press, Oxford, U. K., 1970. Reprinted in *Automation of Reasoning 2*, Springer, Berlin, pp. 342–376 (1983).

[2] N. Dershowitz and J. P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science B: Formal Methods and Semantics*, chapter 6, pages 243–320, North-Holland, Amsterdam, 1990.

[3] N. Dershowitz, L. Marcus, and A. Tarlecki. *Existence, Uniqueness, and Construction of Rewrite Systems.* Technical Report ATR-85(8354)-7, Computer Science Laboratory, The Aerospace Corporation, El Segundo, CA, December 1985.

[4] T. Evans. On multiplicative systems defined by generators and relations, I. *Proceedings of the Cambridge Philosophical Society*, 47:637–649, 1951.

[5] T. C. Brown, Jr. *A Structured Design-Method for Specialized Proof Procedures.* PhD thesis, California Institute of Technology, Pasadena, CA, 1975.

[6] D. S. Lankford. *Canonical Inference.* Memo ATP-32, Automatic Theorem Proving Project, University of Texas, Austin, TX, December 1975.

[7] J. Hsiang and M. Rusinowitch. On word problems in equational theories. In T. Ottmann, editor, *Proceedings of the Fourteenth EATCS International Conference on Automata, Languages and Programming*, pages 54–71, Karlsruhe, West Germany, July 1987. Vol. 267 of *Lecture Notes in Computer Science*, Springer, Berlin.

[8] L. Bachmair, N. Dershowitz, and D. A. Plaisted. Completion without failure. In H. Aït-Kaci and M. Nivat, editors, *Resolution of Equations in Algebraic Structures 2: Rewriting Techniques*, chapter 1, pages 1–30, Academic Press, New York, 1989.

[9] J. Pedersen. Computer solution of word problems in universal algebra. In *Computers in Algebra*, pages 103–128, 1988. Vol. 111 of *Lecture Notes in Pure and Applied Mathematics*, Marcel-Dekker, New York.

[10] J. Hsiang. *Topics in Automated Theorem Proving and Program Generation.* PhD thesis, Department of Computer Science, University of Illinois, Urbana, IL, December 1982. Report R-82-1113.

[11] N. Dershowitz. Termination of rewriting. *J. of Symbolic Computation*, 3(1&2):69–115, February/April 1987. Corrigendum: *4*, 3 (December 1987), 409–410.

[12] S. Kamin and J. J. Lévy. *Two Generalizations of the Recursive Path Ordering.* Unpublished note, Department of Computer Science, University of Illinois, Urbana, IL, February 1980.

[13] N. Dershowitz. Orderings for term-rewriting systems. *Theoretical Computer Science*, 17(3):279–301, March 1982.

[14] H. Comon. Solving inequations in term algebras (Preliminary version). In *Proceedings of the Fifth Annual IEEE Symposium on Logic in Computer Science*, pages 62–69, Philadelphia, PA, June 1990.

[15] W. Snyder. Efficient ground completion: An *O(n log n)* algorithm for generating reduced sets of ground rewrite rules equivalent to a set of ground equations *E*. In N. Dershowitz, editor, *Proceedings of the Third International Conference on Rewriting Techniques and Applications*, pages 419–433, Chapel Hill, NC, April 1989. Vol. 355 of *Lecture Notes in Computer Science*, Springer, Berlin.

[16] M. H. A. Newman. On theories with a combinatorial definition of 'equivalence'. *Annals of Mathematics*, 43(2):223–243, 1942.

[17] D. S. Lankford and A. M. Ballantyne. *Decision Procedures for Simple Equational Theories with Commutative-Associative Axioms: Complete Sets of Commutative-*

Associative Reductions. Memo ATP-39, Department of Mathematics and Computer Sciences, University of Texas, Austin, TX, August 1977.

[18] U. Martin and T. Nipkow. Ordered completion. In M. Stickel, editor, *Proceedings of the Tenth International Conference on Automated Deduction*, pages 366–380, Springer-Verlag, Kaiserslautern, West Germany, July 1990. *Lecture Notes in Computer Science*, Springer, Berlin.

[19] H. Devie. When ordered completion fails. In M. Okada, editor, *Proceedings of the Second International Workshop on Conditional and Typed Rewriting Systems*, Springer-Verlag, Montreal, Canada, 1990. To appear.

[20] G. E. Peterson and M. E. Stickel. Complete sets of reductions for some equational theories. *J. of the Association for Computing Machinery*, 28(2):233–264, April 1981.

[21] J. P. Jouannaud and H. Kirchner. Completion of a set of rules modulo a set of equations. *SIAM J. on Computing*, 15:1155–1194, November 1986.

[22] L. Bachmair and N. Dershowitz. Completion for rewriting modulo a congruence. *Theoretical Computer Science*, 67(2 & 3), October 1989.

[23] P. Lescanne. Computer experiments with the REVE term rewriting system generator. In *Proceedings of the Tenth ACM Symposium on Principles of Programming Languages*, pages 99–108, Austin, TX, January 1983.

[24] M. E. Stickel. A case study of theorem proving by the Knuth Bendix method discovering that $x^3 = x$ implies ring commutativity. In R. E. Shostak, editor, *Proceedings of the Seventh International Conference on Automated Deduction*, pages 248–259, Napa, CA, May 1984. Vol. 170 of *Lecture Notes in Computer Science*, Springer, Berlin.

[25] D. Kapur and H. Zhang. An overview of Rewrite Rule Laboratory (RRL). In N. Dershowitz, editor, *Proceedings of the Third International Conference on Rewriting Techniques and Applications*, pages 559–563, Chapel Hill, NC, April 1989. Vol. 355 *Lecture Notes in Computer Science*, Springer, Berlin.

[26] F. Fages. *Le système KB: manuel de référence: présentation et bibliographie, mise en oeuvre*. Report R. G. 10.84, Greco de Programmation, Bordeaux, France, 1984.

[27] A. Ohsuga and K. Sakai. *Metis: A Term Rewriting System Generator*. Technical Report, ICOT Research Center, Tokyo, Japan, 1986.

[28] S. Anantharaman, J. Hsiang, and J. Mzali. SbReve2: A term rewriting laboratory with (AC-)unfailing completion. In N. Dershowitz, editor, *Proceedings of the Third International Conference on Rewriting Techniques and Applications*, pages 533–537, Chapel Hill, NC, April 1989. Vol. 355 *Lecture Notes in Computer Science*, Springer, Berlin.

[29] J. Christian. Fast Knuth-Bendix completion: Summary. In N. Dershowitz, editor, *Proceedings of the Third International Conference on Rewriting Techniques and Applications*, pages 551–555, Chapel Hill, NC, April 1989. Vol. 355 *Lecture Notes in Computer Science*, Springer, Berlin.

[30] G. E. Peterson. Complete sets of reductions with constraints. In M. Stickel, editor, *Proceedings of the Tenth International Conference on Automated Deduction*, pages 381–395, Kaiserslautern, West Germany, July 1990. *Lecture Notes in Computer Science*, Springer, Berlin.

[31] J. Pedersen. Obtaining complete sets of reductions and equations without using special unification algorithms. In B. F. Caviness, editor, *Proceedings of the European Conference on Computer Algebra: Research Contributions*, pages 422–423, Linz, Austria, April 1985. Vol. 204 of *Lecture Notes in Computer Science*, Springer, Berlin.

[32] G. Bauer. n-level rewriting systems. *Theoretical Computer Science*, 40:85–99, 1985.

[33] C. Squier. Word problems and a homological finiteness condition for monoids. *J. of Pure and Applied Algebra*. To appear.

[34] J. M. Hullot. *A Catalogue of Canonical Term Rewriting Systems*. Technical Report CSL-113, SRI International, Menlo Park, CA, April 1980.

[35] B. Benninghofen, S. Kemmerich, and M. M. Richter. *Systems of Reductions*. Volume 277 of *Lecture Notes in Computer Science*, Springer, Berlin, 1987.

[36] R. V. Book. Thue systems as rewriting systems. *J. of Symbolic Computation*, 3(1&2):39–68, February/April 1987.

[37] B. Buchberger. Gröbner bases: An algorithmic method in polynomial ideal theory. In *Multidimensional Systems Theory*, chapter 6, Reidel, Dordrecht, 1985.

[38] P. Lescanne. Term rewriting systems and algebra. In R. E. Shostak, editor, *Proceedings of the Seventh International Conference on Automated Deduction*, pages 166–174, Napa, CA, May 1984. Vol. 170 of *Lecture Notes in Computer Science*, Springer, Berlin.

[39] J. Pedersen. *Confluence Methods and the Word Problem in Universal Algebra*. PhD thesis, Emory University, Australia, 1984.

[40] J. Pedersen. Morphocompletion for one-relation monoids. In N. Dershowitz, editor, *Proceedings of the Third International Conference on Rewriting Techniques and Applications*, pages 574–578, Chapel Hill, NC, April 1989. Vol. 355 *Lecture Notes in Computer Science*, Springer, Berlin.

[41] R. Edelson. April 1990. Private Communication.

[42] A. Foret. Rewrite rule systems for modal propositional logic. In *Proceedings on an International Workshop on Algebraic and Logic Programming*, pages 146–157, Akademie-Verlag, Gaussig, GDR, November 1988.

[43] S. C. Chou and W. F. Schelter. *Proving Geometry Theorems with Rewrite Rules*. Preprint, University of Texas, Austin, Texas, December 1985.

[44] J. Pedersen. The word problem in absorbing varieties. *Houston J. of Mathematics*, 11(4):575–590, 1985.

[45] A. M. Ballantyne and D. S. Lankford. New decision algorithms for finitely presented commutative semigroups. *J. Computational Mathematics with Applications*, 7:159–165, 1981.

[46] D. Lankford, G. Butler, and A. Ballantyne. A progress report on new decision algorithms for finitely presented Abelian groups. In R. E. Shostak, editor, *Proceedings of the Seventh International Conference on Automated Deduction*, pages 128–141, Napa, CA, 1984. Vol. 170 of *Lecture Notes in Computer Science*, Springer, Berlin.

[47] D. S. Lankford. *The Uniform Word Problem for J-Algebras is Decidable.* Memo MTP-10, Department of Mathematics, Louisiana Tech. University, Ruston, LA, May 1980.

[48] P. Le Chenadec. Canonical forms in finitely presented algebras. In R. E. Shostak, editor, *Proceedings of the Seventh International Conference on Automated Deduction*, pages 142–165, Napa, CA, 1984. Vol. 170 of *Lecture Notes in Computer Science*, Springer, Berlin.

[49] P. Le Chenadec. *Canonical Forms in Finitely Presented Algebras.* Pitman-Wiley, London, 1985.

[50] R. Bündgen. Applying term rewriting methods to finite groups. In H. Kirchner and W. Wechler, editors, *Proceedings of the Second Conference on Algebraic and Logic Programming*, pages 332–346, Nancy, France, October 1990. Vol. 463 of *Lecture Notes in Computer Science*, Springer, Berlin.

[51] H. Bücken. Reduction-systems and small cancellation theory. In *Proceedings of the Fourth Workshop on Automated Deduction*, pages 53–59, Austin, TX, February 1979.

[52] P. Le Chenadec. Analysis of Dehn's algorithm by critical pairs. *Theoretical Computer Science*, 51(1,2):27–52, 1987.

[53] S. Burris and J. Lawrence. Computing in finite fields. Unpublished draft, University of Waterloo, 1990.

[54] J. Hsiang and N. Dershowitz. Rewrite methods for clausal and non-clausal theorem proving. In *Proceedings of the Tenth International Colloquium on Automata, Languages and Programming*, pages 331–346, European Association of Theoretical Computer Science, Barcelona, Spain, July 1983. Vol. 154 of *Lecture Notes in Computer Science*, Springer, Berlin.

[55] T. Nipkow. Unification in primal algebras, their powers and their varieties. *J. of the Association for Computing Machinery.* To appear.

[56] D. Kapur and H. Zhang. A case study of the completion procedure: Proving ring commutativity problems. May 1989. Unpublished Draft.

[57] S. Anantharaman and J. Hsiang. Automated proofs of the Moufang identities in alternative rings. *J. of Symbolic Computation*, 6:79–109, 1990.

[58] J. Siekmann and P. Szabo. A Noetherian and confluent rewrite system for idempotent semigroups. *Semigroup Forum*, 25(1/2):83–110, 1982.

[59] A. Pelin and J. H. Gallier. Building exact computation sequences. *Theoretical Computer Science*, 53:125–150, 1987.

[60] S. Kaplan. Simplifying conditional term rewriting systems: Unification, termination and confluence. *J. of Symbolic Computation*, 4(3):295–334, December 1987.

[61] H. Ganzinger. A completion procedure for conditional equations. In S. Kaplan and J. P. Jouannaud, editors, *Proceedings of the First International Workshop on Conditional Term Rewriting Systems*, pages 62–83, Orsay, France, July 1987. Vol. 308 of *Lecture Notes in Computer Science*, Springer, Berlin (1988).

[62] E. Kounalis and M. Rusinowitch. On word problems in Horn theories. In E. Lusk and R. Overbeek, editors, *Proceedings of the Ninth International Conference on Automated Deduction*, pages 527–537, Argonne, Illinois, May 1988. Vol. 310 of *Lecture Notes in Computer Science*, Springer, Berlin.

[63] I. I. Zhegalkin. On a technique of evaluation of propositions in symbolic logic. *Matematicheskii Sbornik*, 34(1):9–27, 1927.

[64] M. Stone. The theory of representations for Boolean algebra. *Transactions of the American Mathematical Society*, 40:37–111, 1936.

[65] S. Burris. NOTES on discriminator varieties and symbolic computation. Unpublished draft, University of Waterloo, July 1989.