Theoretical Computer Science 17 (1982) 279–301
North-Holland Publishing Company

ORDERINGS FOR TERM-REWRITING SYSTEMS*

Nachum DERSHOWITZ

Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, U.S.A.

Communicated by Z. Manna
Received August 1979
Revised September 1980

Abstract. Methods of proving that a term-rewriting system terminates are presented. They are based on the intuitive notion of 'simplification orderings', orderings in which any term that is syntactically simpler than another is smaller than the other. As a consequence of Kruskal's Tree Theorem, any nonterminating system must be self-embedding in the sense that it allows for the derivation of some term from a simpler one; thus termination is guaranteed if every rule in the system is a reduction in some simplification ordering.

Most of the orderings that have been used for proving termination are indeed simplication orderings; using this notion often allows for much easier proofs. A particularly useful class of simplification orderings, the 'recursive path orderings', is defined. Examples of the use of simplification orderings in termination proofs are given.

1. Introduction

It is sometimes convenient to express programs in the form of term-rewriting systems. Such programs are easy to understand and have a simple, elegant syntax and semantics. For example, the following system of five rewrite rules transforms logical formulae [containing the operators $\vee$ (disjunction), $\wedge$ (conjunction), and $\neg$ (negation)] into equivalent formulae in disjunctive normal form:

$$\begin{aligned}
&\neg\neg\alpha \to \alpha \\
&\neg(\alpha \vee \beta) \to (\neg\alpha \wedge \neg\beta) \\
&\neg(\alpha \wedge \beta) \to (\neg\alpha \vee \neg\beta) \qquad\qquad (A) \\
&\alpha \wedge (\beta \vee \gamma) \to (\alpha \wedge \beta) \vee (\alpha \wedge \gamma) \\
&(\beta \vee \gamma) \wedge \alpha \to (\beta \wedge \alpha) \vee (\gamma \wedge \alpha).
\end{aligned}$$

The first rule indicates that double negations may be eliminated; the second and third rules apply DeMorgan's laws to push negations inward; the last two rules apply the distributivity of conjunction over disjunction. Such systems are becoming

* Research supported in part under NSF grants MCS 77-22830 and MCS 79-04897. A previous version of this paper appeared in the Proceedings of the Twentieth IEEE Symposium on Foundations of Computer Science, October 1979.

0304-3975/82/0000–0000/$2.75

increasingly popular in automated simplification and theorem-proving applications; some examples are Iturriaga [11], Moses [20], Griesmer and Jenks [7], Hearn [8], Ballantyne and Bledsoe [1], Boyer and Moore [2], Carter et al. [3], Weyhrauch [28], and Musser [21].

The above program is executed for a given input term by repeatedly replacing subterms of the form of the left-hand side of some rule with the corresponding right-hand side, until no further rewrites are possible. Thus, the second rule in the above system may be applied to the input term $a \wedge \neg\neg(b \vee c)$ by replacing $\neg(b \vee c)$ with $(\neg b \wedge \neg c)$, thereby obtaining $a \wedge \neg(\neg b \wedge \neg c)$. The computation iterates in this manner, at each stage choosing some applicable rule and applying it to some subterm. Continuing with our example: By applying the third rule, we get $a \wedge (\neg\neg b \vee \neg\neg c)$. Two applications of the first rule then yield $a \wedge (b \vee c)$. Finally, an application of the fourth rule gives $(a \wedge b) \vee (a \wedge c)$ which is in disjunctive normal form. At this point, no rule is applicable and the system is said to have 'terminated' with the final result $(a \wedge b) \vee (a \wedge c)$.

To verify the correctness of such a program, one must show

(1) that it always terminates, i.e. given any input term, execution will always reach a stage for which there is no way to continue applying rules, and

(2) that it is 'partially correct', in the sense that if it does terminate, then the final result is what was desired.

In this paper, we deal only with the termination aspect of correctness.

The difficulty in proving the termination of a system such as the one for disjunctive normal form above stems from the fact that while some rules may decrease the size of a term, other rules may increase its size and duplicate occurrences of subterms. Furthermore, applying a rule to a subterm not only affects the structure of that subterm, but also changes the structure of its superterms. Any proof of termination must take into consideration the many different possible rewrite sequences generated by the nondeterministic choice of rules and subexpressions. Various methods for proving termination of term-rewriting systems have been suggested in recent years, including Iturriaga [11], Knuth and Bendix [13], Manna and Ness [19], Lankford [15], Lipton and Snyder [18], Plaisted [23], Plaisted [24], Dershowitz and Manna [6], and Lankford [16]. In this paper we present new methods of proving termination. One can show (Huet and Lankford [10]) that termination is in general an undecidable property of such systems.

The partial correctness of term-rewriting systems, on the other hand, is often easy to verify. One usually shows that each rule is 'value-preserving', i.e. if $l \to r$ is a rule in the system, then $l = r$ in the intended interpretation. (In the above example, each rule preserves logical equivalence.) Furthermore, one must verify that all possible final results have the desired properties, for example by showing that were a final result not of the desired form, then some rule could still be applied to it. (By the definition of disjunctive normal form, no compound formula may be negated, nor may a disjunction be conjoined with another formula.) Hence, proving partial correctness is in many cases formally quite simple.

Another property of term-rewriting systems that is often desirable is 'confluence', i.e. that there be a unique final result for all rewrite sequences beginning with the same term. The above system, for example, is not confluent: applying DeMorgan's laws to $\neg(a \wedge (b \vee c))$ yields $(\neg a \vee (\neg b \wedge \neg c))$, while first distributing leads to $(((\neg a \wedge \neg a) \vee (\neg a \wedge \neg c)) \vee ((\neg b \wedge \neg a) \vee (\neg b \wedge \neg c)))$. Confluent term-rewriting systems are used as decision procedures for equational theories: see, among others, Knuth and Bendix [13], Slagle [25], Lankford [15], Huet [9], Lankford and Ballantyne [17], and Stickel and Peterson [26]. Termination is frequently a prerequisite for demonstrating confluence.

To illustrate the difficulty of determining if and why a system terminates we present four variations on system (A):

The first variation is

$$\begin{aligned} &\neg\neg\alpha \to \alpha \\ &\neg(\alpha \vee \beta) \to (\neg\neg\neg\alpha \wedge \neg\neg\neg\beta) \\ &\neg(\alpha \wedge \beta) \to (\neg\neg\neg\alpha \vee \neg\neg\neg\beta) \\ &\alpha \wedge (\beta \vee \gamma) \to (\alpha \wedge \beta) \vee (\alpha \wedge \gamma) \\ &(\beta \vee \gamma) \wedge \alpha \to (\beta \wedge \alpha) \vee (\gamma \wedge \alpha). \end{aligned} \tag{B}$$

Here the second and third rules have been modified to introduce additional double negations (that can be eliminated by the first rule).

The next variation is the same as System (B) with the two rules for distribution removed:

$$\begin{aligned} &\neg\neg\alpha \to \alpha \\ &\neg(\alpha \vee \beta) \to (\neg\neg\neg\alpha \wedge \neg\neg\neg\beta) \\ &\neg(\alpha \wedge \beta) \to (\neg\neg\neg\alpha \vee \neg\neg\neg\beta). \end{aligned} \tag{C}$$

This system pushes negations into disjunctions or conjunctions and eliminates double negations.

The third variation is

$$\begin{aligned} &\neg\neg\alpha \to \alpha \\ &\neg(\alpha \vee \beta) \to ((\neg\neg\neg\alpha \wedge \neg\neg\neg\beta) \wedge (\neg\neg\neg\alpha \wedge \neg\neg\neg\beta)) \\ &\neg(\alpha \wedge \beta) \to ((\neg\neg\neg\alpha \vee \neg\neg\neg\beta) \vee (\neg\neg\neg\alpha \vee \neg\neg\neg\beta)) \\ &(\alpha \wedge \alpha) \to \alpha \\ &(\alpha \vee \alpha) \to \alpha. \end{aligned} \tag{D}$$

Here the second and third rules have been further complicated to duplicate conjuncts and disjuncts. To compensate, two rules for their elimination have been added.

The last variation is the same as System (D), except that the extra negations have been removed from the second and third rules:

$$\begin{aligned}
&\neg\neg\alpha \to \alpha\\
&\neg(\alpha \vee \beta) \to ((\neg\alpha \wedge \neg\beta) \wedge (\neg\alpha \wedge \neg\beta))\\
&\neg(\alpha \wedge \beta) \to ((\neg\alpha \vee \neg\beta) \vee (\neg\alpha \vee \neg\beta))\\
&(\alpha \wedge \alpha) \to \alpha\\
&(\alpha \vee \alpha) \to \alpha.
\end{aligned} \tag{E}$$

The reader is invited to determine which of these five systems do terminate and which do not.

In the next section we characterize nontermination and show how 'simplification orderings' may be used to prove termination. (This extends the result reported in Dershowitz [5].) We explain why most of the orderings previously used for proving termination have in fact been simplification orderings. In Section 3 similar methods are described for using quasi-orderings to prove termination or the weaker concept 'quasi-termination'. Then, in Section 4, we apply these methods to several orderings; in particular, we define a class of 'recursive path orderings' and show that they are simplification orderings. Finally, in Section 5, the use of these orderings in several termination proofs is illustrated.

2. Termination and nontermination

Given a set of operators F, we consider the set $T(F)$ of all terms constructed from operators in F. In general, we shall assume that all operators have variable arity; thus, if f is an operator in F and $t_1, \ldots, t_n$ $(n \geq 0)$ are terms in $T(F)$, then $f(t_1, \ldots, t_n)$ is also a term in $T(F)$. The results of this paper apply to any subset T of $T(F)$ with the property that $f(t_1, \ldots, t_n)$ is a term in T, only if f is an operator in F and $t_1, \ldots, t_n$ are also terms in T. For example, T may restrict an operator f to a fixed arity, in which case $f(t_1, \ldots, t_n) \in T$ only if f is of arity n.

A *term-rewriting system* P over such a set of terms T is a finite set of rewrite rules, each of the form $l_i(\bar{\alpha}) \to r_i(\bar{\alpha})$, where $l_i(\bar{\alpha})$ and $r_i(\bar{\alpha})$ are 'open terms', i.e. terms constructed from operators in F and from variables $\bar{\alpha}$ (ranging over T). Such a rule may be applied to a term $t \in T$ if t contains a subterm $l_i(\bar{a})$ with the terms $\bar{a} \in T$ substituted for the variables $\bar{\alpha}$. The rule is applied by replacing the subterm $l_i(\bar{a})$ in t with the term $r_i(\bar{a})$. (The variables appearing in r_i must therefore be a subset of those in l_i.) The choice of which rule to apply is made nondeterministically from amongst the applicable rules; similarly, the choice of which subterm to apply a rule to is nondeterministic. We write $t \Rightarrow t'$ (and say 't derives t'') to indicate that the term $t' \in T$ may be obtained from the term $t \in T$ by a single application of some rule in P.

For example, the one-rule system

$$(\alpha \wedge \beta) \wedge \gamma \rightarrow \alpha \wedge (\beta \wedge \gamma) \tag{F}$$

reparenthesizes a conjunction by associating to the right. Applying that rule to the term $t = ((a \wedge b) \wedge c) \wedge (d \wedge e)$, we get

$$t \Rightarrow (a \wedge (b \wedge c)) \wedge (d \wedge e) \Rightarrow a \wedge ((b \wedge c) \wedge (d \wedge e)) \Rightarrow a \wedge (b \wedge (c \wedge (d \wedge e))),$$

or alternatively,

$$t \Rightarrow (a \wedge b) \wedge (c \wedge (d \wedge e)) \Rightarrow a \wedge (b \wedge (c \wedge (d \wedge e))).$$

In either case, no further applications of the rule are possible. We say that a term-rewriting system P *terminates* for a set of terms T, if there exist no infinite sequences of terms $t_i \in T$ such that $t_1 \Rightarrow t_2 \Rightarrow t_3 \Rightarrow \cdots$; conversely, a system is *nonterminating* if there exists any such infinite derivation.

The *homeomorphic embedding* ('syntactically simpler') relation $\trianglelefteq$ on terms in $T(F)$ is defined as follows (viewing terms as ordered trees):

$$s = f(s_1, s_2, \ldots, s_m) \trianglelefteq g(t_1, t_2, \ldots, t_n) = t,$$

if and only if

(a) $f = g$ and $s_i \leq t_{j_i}$, for all i, $1 \leq i \leq m$, where $1 \leq j_1 < j_2 < \cdots < j_m \leq n$, or

(b) $s \trianglelefteq t_j$ for some j, $1 \leq j \leq n$.

Thus, $s \trianglelefteq t$ if s may be obtained from t by deletion of operators. For example, $\neg\neg(a \wedge (a \vee b)) \trianglelefteq (c \wedge \neg\neg\neg((\neg a \vee \neg a) \wedge (\neg a \vee \neg b)))$.

We shall say that a derivation $t_1 \Rightarrow t_2 \Rightarrow \cdots$ is *self-embedding* if $t_j \trianglelefteq t_k$ for some $j < k$. With this notion, we can characterize nontermination in the following manner:

Nontermination Theorem. *If a term-rewriting system P does not terminate, then there exists an infinite self-embedding derivation.*

Proof. If P does not terminate, then by definition there exists at least one infinite derivation $t_1 \Rightarrow t_2 \Rightarrow \cdots$. There can be only a finite number of operators appearing in the derivation (those in t_1 and in P) and, by the Tree Theorem (Kruskal [14], see next section), in any infinite sequence of terms $t_1, t_2, \ldots$ with a finite number of operations, $t_j \leq t_k$ for some $j < k$.

Note that homeomorphic self-embedding does not, however, imply nontermination. For example, the term-rewriting system consisting of the single rule $f(f(\alpha)) \rightarrow f(g(f(\alpha)))$ is both self-embedding and terminating. But we can use homeomorphic embedding to give a sufficient condition for termination. First, we will need the following concepts:

A *partially-ordered* set $(S, >)$ consists of a set S and a transitive and irreflexive binary relation $>$ defined on elements of S. (Asymmetry of a partial ordering follows from transitivity and irreflexivity.) A partially ordered set is said to be *totally ordered* if for any two distinct elements s and s' of S, either $s > s'$ or $s' > s$. For example, both the set $\mathbf{Z}$ of integers and the set $\mathbf{N}$ of natural numbers are totally ordered by the 'greater-than' relation $>$. The set $\mathcal{P}(\mathbf{Z})$ of all subsets of the integers is partially ordered by the subset relation $\subset$.

A partial ordering $>$ on a set S is said to be *well-founded* if it admits no infinite descending sequences $s_1 > s_2 > s_3 > \cdots$ of elements of S. Thus, $>$ is a well-founded ordering of $\mathbf{N}$, since no sequence can descend beyond 0, but $>$ is not a well-founded ordering of $\mathbf{Z}$, since $-1 > -2 > -3 > \cdots$ is an infinite descending sequence.

The following definition and theorem (see [19], and also [15]) are often used to prove the termination of term-rewriting systems:

Definition 1. A relation R over a set of terms T is *monotonic* if

$$tRt' \text{ implies } f(\ldots t \ldots) R f(\ldots t' \ldots) \quad \text{(replacement)}$$

for any terms $f(\ldots t \ldots), f(\ldots t' \ldots) \in T$.

Theorem 1 (Manna and Ness [19]). *A term-rewriting system $P = \{l_i \to r_i\}_{i=1}^{p}$ over a set of terms T terminates, if there exists a monotonic well-founded ordering $>$ over T such that*

$$l_i > r_i, \quad i = 1, \ldots, p, \qquad \textit{(reduction)}$$

for any substitution of terms in T for the variables of l_i.

The reduction condition asserts that applying any rule reduces the subterm to which the rule is applied in the well-founded ordering. The replacement condition allows for this 'local' measure by guaranteeing that reducing subterms also reduces the top-level term. Thus, $t \Rightarrow t'$ implies $t > t'$. Since by the nature of a well-founded ordering there can be no infinite descending sequences, there can also be no infinite derivations.

Our method for proving termination is based on the following

Definition 2. A transitive and irreflexive relation $>$ (a partial ordering) is a *simplification ordering* for a set of terms T if it possesses the following three properties:

(1) $t > t'$ implies $f(\ldots t \ldots) > f(\ldots t' \ldots)$, (replacement)
(2) $f(\ldots t \ldots) > t$, (subterm)
(3) $f(\ldots t \ldots) > f(\ldots \ \ldots)$ (deletion)

for any terms $f(\ldots t \ldots), f(\ldots t' \ldots), f(\ldots \ \ldots) \in T$.

By iterating the subterm property, any term is also greater than any of the (not necessarily immediate) subterms contained within it. The deletion condition asserts that deleting subterms of a (variable arity) operator reduces the term in the ordering; if the operators f have fixed arity, the deletion condition is superfluous. Together these conditions imply that 'syntactically simpler' terms are smaller in the ordering:

Embedding Lemma. *Let s and t be terms in T. If $s \trianglelefteq t$, then $s \preceq t$ in any simplification ordering $\succ$ over T.*

In other words, the relation $\trianglelefteq$ is contained in the relation $\preceq$. As usual, $s \preceq t$ means $t \succ s$ or $t = s$.

Proof. The proof is by induction on the *size* (number of occurrences of operators) of t. Assume that $s' \trianglelefteq t'$ implies $s' \preceq t'$ for any t' smaller than t and for any s'. By the definition of $\trianglelefteq$, if $s = f(s_1, \ldots, s_m) \trianglelefteq g(t_1, \ldots, t_n) = t$ (m or n may be zero), then either

(a) $f = g$ and $s_i \trianglelefteq t_{j_i}$ for all i, $1 \leqslant i \leqslant m$, in which case $s_i \preceq t_{j_i}$ by the induction hypothesis and therefore $s \preceq f(t_{j_1}, \ldots, t_{j_m}) \preceq t$ by the replacement and deletion properties; or else

(b) $s \trianglelefteq t_j$ for some j, $1 \leqslant j \leqslant n$, in which case $s \preceq t_j \prec g(\ldots t_j \ldots) = t$ by the induction hypothesis and the subterm property.

The following theorem gives a sufficient criterion for proving that a term-rewriting system terminates for all inputs.

First Termination Theorem. *A term-rewriting system $P = \{l_i \to r_i\}_{i=1}^{p}$ over a set of terms T terminates if there exists a simplification ordering $\succ$ over T such that*

$$l_i \succ r_i, \quad i = 1, \ldots, p, \qquad (\textit{reduction})$$

for any substitution of terms in T for the variables of l_i.

Proof. If P does not terminate, then by the Nontermination Theorem there exists a derivation $t_j \Rightarrow \cdots \Rightarrow t_k$ $(j < k)$ such that $t_j \trianglelefteq t_k$ and by the Embedding Lemma $t_j \preceq t_k$ in the given simplification ordering $\succ$. On the other hand, if $l_i \succ r_i$, then it follows by the replacement property that $t_j \succ \cdots \succ t_k$ and by transitivity that $t_j \succ t_k$. This contradicts the asymmetry of the partial ordering $\succ$.

Most of the well-founded orderings that have been used to prove the termination of term-rewriting systems are in fact simplification orderings. The following proposition explains why.

Proposition. *Any total monotonic ordering $\succ$ on a set $T(F)$ of terms over a finite set F of fixed-arity operators is well-founded, if and only if it possesses the subterm property.*

Proof. If $\succ$ is monotonic and has the subterm property, then it is a simplification ordering (the deletion property is vacuously true for fixed-arity operators). As is implicit in the preceding proof, a simplification ordering is well-founded when the set of operators is finite.

On the other hand, were the subterm property not to hold, i.e. $t \succ f(\ldots t \ldots)$ for some term $f(\ldots t \ldots) \in T(F)$, then (by monotonicity) there would exist an infinite descending sequence $t \succ f(\ldots t \ldots) \succ f(\ldots f(\ldots t \ldots) \ldots) \succ \cdots$ of terms in $T(F)$.

At the end of the next section a sufficient condition for the well-foundedness of a simplification ordering is given.

3. Quasi-orderings

In this section we investigate methods for proving termination that use quasi-orderings. A *quasi-ordered* set $(S, \succsim)$ consists of a set S and a transitive and reflexive binary relation $\succsim$ defined on elements of S. For example, the set $\mathbf{Z}$ of integers is quasi-ordered under the relation 'greater or congruent modulo 10'. Given a quasi-ordering $\succsim$ on a set S, define the equivalence relation $\approx$ as both $\succsim$ and $\precsim$, and the partial ordering $\succ$ as $\succsim$ but not $\precsim$.

We say that a term-rewriting system P is *quasi-terminating* for a set of terms T, if all (infinite) derivations contain only a finite number of different terms. Equivalently, any infinite derivation must contain some term twice. Thus, termination of a quasi-terminating system for a given input term is decidable (construct all derivations initiated by that term until they terminate or repeat). The following theorem may be used to prove termination for all inputs.

Theorem 2. *A quasi-terminating term-rewriting system $P = \{l_i \rightarrow r_i\}_{i=1}^{p}$ over a set of terms T terminates if there exists a monotonic quasi-ordering $\succsim$ such that*

$$l_i \succ r_i, \quad i = 1, \ldots, p \qquad (\textit{reduction})$$

for any substitution of terms in T for the variables of l_i.

Proof. In any infinite derivation of a quasi-terminating system there must be a segment $t_i \Rightarrow t_{i+1} \Rightarrow \cdots \Rightarrow t_j = t_i$ $(i < j)$. Now, it must be that for some (not necessarily proper) subterm s_i of t_i, there is a self-derivation $s_i \Rightarrow s_{i+1} \Rightarrow \cdots \Rightarrow s_k \Rightarrow s_{k+1} \Rightarrow \cdots \Rightarrow s_j = s_i$ in which a rule is applied to a top-level term s_k. By the monotonicity property, if $t \succsim t'$ then $f(\ldots t \ldots) \succsim f(\ldots t' \ldots)$, and thus $t \Rightarrow t'$ implies $t \succsim t'$. But then $s_i \succsim s_{i+1} \succsim \cdots \succsim s_k \succ s_{k+1} \succsim \cdots \succsim s_j = s_i$, and by transitivity $s_k \succ s_{k+1} \succsim s_k$, which is a contradiction.

To prove that a system is quasi-terminating, one can use the following

Quasi-termination Theorem. *Let the quasi-ordering $\succsim$ be a monotonic extension of a simplification ordering $\succ$ on a set of terms T. A term-rewriting system $P = \{l_i \rightarrow r_i\}_{i=1}^{p}$*

over T *is quasi-terminating if*

$$l_i \succsim r_i, \quad i = 1, \ldots, p,$$

for any substitution of terms in T *for the variables of* l_i.

Proof. By the Nontermination Theorem and the Embedding Lemma, if P does not terminate, then in any infinite derivation $t_1 \Rightarrow t_2 \Rightarrow \cdots$, $t_j \preceq t_k$ for some $j < k$. On the other hand, if $l_i \succsim r_i$, then it follows by monotonicity and transitivity that $t_j \succsim t_k$, and consequently $t_j \not\prec t_k$. It must be then that $t_j = t_k$; the system therefore quasi-terminates.

Analogous to the definition of a simplification ordering, we have

Definition 3. A transitive and reflexive relation $\succsim$ (a quasi-ordering) is a *quasi-simplification ordering* for a set of terms T if it possesses the following three properties:

(1) $t \succsim t'$ implies $f(\ldots t \ldots) \succsim f(\ldots t' \ldots)$, (replacement)
(2) $f(\ldots t \ldots) \succsim t$, (subterm)
(3) $f(\ldots t \ldots) \succsim f(\ldots \; \ldots)$ (deletion)

for any terms $f(\ldots t \ldots), f(\ldots t' \ldots), f(\ldots \; \ldots) \in T$.

The Embedding Lemma also holds for quasi-simplification orderings, i.e. $s \trianglelefteq t$ implies $s \precsim t$.

We generalize the termination theorem of the previous section with a

Second Termination Theorem. *A term-rewriting system* $P = \{l_i \rightarrow r_i\}_{i=1}^{p}$ *over a set of terms* T *terminates if there exists a quasi-simplification ordering* $\succsim$ *such that*

$$l_i \succ r_i, \quad i = 1, \ldots, p, \qquad (reduction)$$

for any substitution of terms in T *for the variables of* l_i.

Proof. If no rules are applied to the top-level terms t_i of an infinite derivation $t_1 \Rightarrow t_2 \Rightarrow \cdots$, then some proper subterm of t_1 must also initiate an infinite derivation. Thus, for any infinite derivation $t_1 \Rightarrow t_2 \Rightarrow \cdots$, some (not necessarily proper) subterm s_1 of t_1 must initiate an infinite derivation $s_1 \Rightarrow s_2 \Rightarrow \cdots \Rightarrow s_i \Rightarrow s_{i+1} \Rightarrow \cdots$ in which a rule is applied to a top-level term s_i. Under the assumptions of the theorem, $t_1 \succsim s_1$ (subterm property), $s_1 \succsim s_2 \succsim \cdots \succsim s_i$ (replacement property), and $s_i \succ s_{i+1}$ (reduction). By transitivity, then, $t_1 \succ s_{i+1}$.

Accordingly, were P not to terminate, then an infinite descending sequence of terms u_i would exist, beginning with $u_1 = t_1$ and $u_2 = s_{i+1}$, and then continuing with the descending sequence extracted from the remaining infinite derivation $s_{i+1} \Rightarrow s_{i+2} \Rightarrow \cdots$. Since this sequence $u_1 \succ u_2 \succ \cdots \succ u_j \succ \cdots \succ u_k \succ \cdots$ is constructed from a finite number of operators, $u_j \precsim u_k$ for some $j<k$ (Tree Theorem and Embedding Lemma). But $u_j \succ u_k$ and $u_j \precsim u_k$ is a contradiction.

We conclude this section with a sufficient condition for the restriction $\succ$ of a quasi-simplification ordering $\succsim$ to be well-founded.

Definition 4 (Kruskal [14]). A set S is *well-quasi-ordered* under a quasi-ordering $\precsim$ if every infinite sequence $s_1, s_2, \ldots$ of elements of S contains a pair of elements s_j and s_k, $j<k$, such that $s_j \precsim s_k$.

Note that any finite set is well-quasi-ordered under any quasi-ordering (including equality). It follows from the definitions that a set is well-founded under the partial ordering $\succ$ when it is well-quasi-ordered under $\precsim$; the converse is true for total orderings, i.e. if a set is well-founded under a total ordering $\succ$, then it is well-quasi-ordered under $\precsim$.

Well-foundedness Theorem. *Let $\succsim$ be a quasi-simplification ordering for a set of terms $T(F)$. If there exists any well-quasi-ordering $\precsim$ of the set of operators F such that*

$$f \succsim g \text{ implies } f(t_1, \ldots, t_n) \succsim g(t_1, \ldots, t_n) \quad \text{(operator replacement)}$$

for all terms $f(t_1, \ldots, t_n), g(t_1, \ldots, t_n) \in T(F)$, then $T(F)$ is well-quasi-ordered under $\precsim$ and well-founded under the partial ordering $\succ$.

Corollary. *If $\succsim$ is a quasi-simplification ordering for a set of terms $T(F)$ over a finite set of operators F, then $T(F)$ is well-founded under the partial ordering $\succ$.*

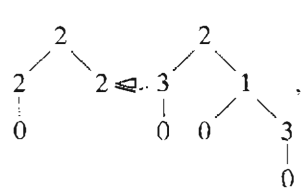
To prove this theorem, we first need the full version of Kruskal's Tree Theorem. A quasi-ordering $\precsim$ of a set of operators F can be extended to a homeomorphic embedding relation $\trianglelefteq_{\precsim}$ on the terms $T(F)$, as follows:

$$s = f(s_1, s_2, \ldots, s_m) \trianglelefteq_{\precsim} g(t_1, t_2, \ldots, t_n) = t,$$

if and only if

(a) $f \precsim g$ and $s_i \trianglelefteq_{\precsim} t_{j_i}$, for all i, $1 \leqslant i \leqslant m$, where $1 \leqslant j_1 < j_2 < \cdots < j_m \leqslant n$, or
(b) $s \trianglelefteq_{\precsim} t_j$ for some j, $1 \leqslant j \leqslant n$.

For example,



where $\leqslant$ is the 'less than or equal' ordering of numbers. Note that $\trianglelefteq$, as defined in the previous section, is the homeomorphic extension of equality.

Tree Theorem (Kruskal [14]). *A set F of operators is well-quasi-ordered under a quasi-ordering $\lesssim$, if and only if the set of terms $T(F)$ is well-quasi-ordered under the embedding relation $\trianglelefteq_{\lesssim}$.*

As a special case: if F is finite, then $T(F)$ is well-quasi-ordered under $\trianglelefteq$, since F is well-quasi-ordered under $=$. A simple proof of the general theorem may be found in Nash-Williams [22].

Proof of Well-foundedness Theorem. If F is well-quasi-ordered under $\lesssim$, then $T(F)$ is well-quasi-ordered under $\trianglelefteq_{\lesssim}$ (Tree Theorem). It is easy to see (along the lines of the Embedding Lemma) that $s \trianglelefteq_{\lesssim} t$ implies $s \lesssim t$. Thus, $T(F)$ is well-quasi-ordered under $\lesssim$ and is therefore well-founded under $\succ$.

4. Applications

In this section, we give a recursive definition of an ordering on terms and show that it is a simplification ordering and also that (under suitable conditions) it is well-founded.

Given a partial ordering $\succ$ on a set S, it may be extended to a partial ordering $\gg$ on finite multisets of elements of S, wherein a multiset is reduced by removing one or more elements and replacing them with any finite number of elements, each of which is smaller than one of the elements removed. For example, if $>$ is the 'greater than' ordering on the natural numbers, then $\{3, 3, 4, 0\} \gg \{3, 2, 2, 1, 1, 1, 4\}$ in the multiset ordering, since an occurrence of 3 has been replaced by five smaller numbers and in addition an occurrence of 0 has been removed (i.e. replaced by zero elements). Such a multiset ordering $\gg$ is well-founded, if and only if S is well-founded under $\succ$ (see [6]). We use this multiset ordering in the following

Definition 5. Let $\succ$ be a partial ordering on a set of operators F. The *recursive path ordering* $\succ^*$ on the set $T(F)$ of terms over F is defined recursively as follows:

$$s = f(s_1, \ldots, s_m) \succ^* g(t_1, \ldots, t_n) = t,$$

if and only if

$$f = g \text{ and } \{s_1, \ldots, s_m\} \gg^* \{t_1, \ldots, t_n\},$$

or

$$f \succ g \text{ and } \{s\} \gg^* \{t_1, \ldots, t_n\},$$

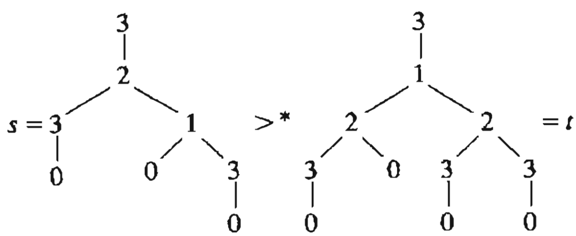
or

$$f \not\geq g \text{ and } \{s_1, \ldots, s_m\} \ \underline{\gg}^* \{t\},$$

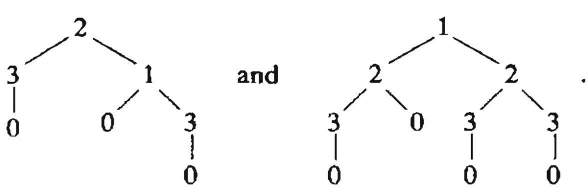
where $\gg^*$ is the extension of $>^*$ to multisets and $\underline{\gg}^*$ means $\gg^*$ or $=$.

Two terms shall be considered equal if they are the same except for permutations among subterms. This definition is similar to a characterization of the 'path of subterms' ordering given in [24].

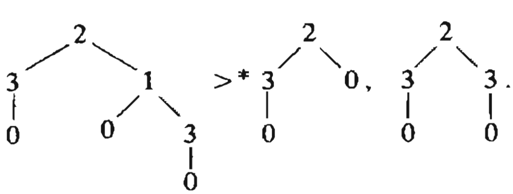
To determine, then, if a term s is greater in this ordering than a term t, the outermost operators of the two terms are compared first. If the operators are equal, then those (immediate) subterms of t that are not also subterms of s must each be smaller (recursively in the term ordering) than some subterm of s. If the outermost operator of s is greater than that of t, then s must be greater than each subterm of t; while if the outermost operator of s is neither equal to nor greater than that of t, then some subterm of s must be greater than or equal to t. For example, representing terms as trees, we have



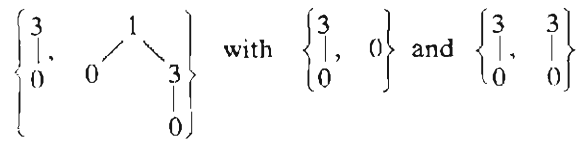
in the recursive path ordering over $T(\mathbf{N})$ with the operators ordered by $>$: By the definition of $>^*$, to compare two terms with the same outermost operator, in our case 3, we must compare (the multisets of) their subterms, viz.



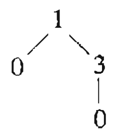
Since $2 > 1$, for the former to be greater than the latter we must have



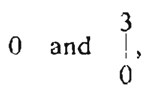
Since $2 = 2$, we must now compare



in the multiset ordering $\gg^*$. Finally, since



is greater than both



we indeed have $s >^* t$.

We have

Theorem 3. *The recursive path ordering $>^*$ is a simplification ordering.*

Proof. We must show that the relation $>^*$ is irreflexive and transitive and that it satisfies the replacement, subterm, and deletion conditions of simplification orderings.

Irreflexivity: We wish to prove that $t \not>^* t$ for any term t. The proof is by induction on the size (number of operators) of t. If t is of the form $f(t_1, \ldots, t_n)$, then by the inductive hypothesis, the relation $>^*$ is irreflexive for its subterms t_j. It follows from the definition of the multiset ordering that $\{t_1, \ldots, t_n\} \not\gg^* \{t_1, \ldots, t_n\}$. Thus, by the definition of the recursive path ordering, $f(t_1, \ldots, t_n) \not>^* f(t_1, \ldots, t_n)$.

Subterm: We show instead that if $s \geq^* t$ for two terms s and t, then

(a) $s >^* t_j$ for any immediate subterm t_j of t and

(b) $f(\ldots s \ldots) >^* t$ for any superterm $f(\ldots s \ldots)$ of s.

Since $s \geq^* s$, it follows from (b) that $f(\ldots s \ldots) >^* s$, as desired.

Let g and h be the outermost operators of s and t, respectively. We prove (a) and (b) simultaneously by induction on the (combined) size of s and t.

For (a), $s >^* t_j$, consider three cases:

(1) $g = h$. By the definition of $>^*$, if $s \geq^* t$ then $s_i \geq^* t_j$ for some subterm s_i of s, and by the inductive hypothesis (b) it follows that $s >^* t_j$.

(2) $g > h$. In this case, it follows directly from the definition of $>^*$ that $s >^* t_j$.

(3) $g \not\geq h$. From the definition of $>^*$, we have $s_i \geq^* t$ for some subterm s_i of s. By the inductive hypothesis (a), $s_i >^* t_j$, and by hypothesis (b), we get $s >^* t_j$.

For (b), $f(\ldots s \ldots) \succ^* t$, we again consider three cases:

(1) $f = h$. We already know (a) that $s \succ^* t_j$ for any subterm t_j of t. Thus, by the definition of the multiset ordering, $\{\ldots s \ldots\} \gg^* \{\ldots t_j \ldots\}$ and by the definition of $\succ^*$, $f(\ldots s \ldots) \succ^* t$.

(2) $f \succ h$. Since $s \succ^* t_j$, it follows from the inductive hypothesis (b) that $f(\ldots s \ldots) \succ^* t_j$, and therefore $\{f(\ldots s \ldots)\} \gg^* \{\ldots t_j \ldots\}$ in the multiset ordering. Thus, by the definition of $\succ^*$, $f(\ldots s \ldots) \succ^* t$.

(3) $f \not\succeq h$. We are given that $s \succeq^* t$. It follows from the definition of the multiset ordering that $\{\ldots s \ldots\} \ggeq^* \{t\}$ and from the definition of $\succ^*$ that $f(\ldots s \ldots) \succ^* t$.

Transitivity: We must show that $s \succ^* t$ and $t \succ^* u$ together imply $s \succ^* u$. Note that by the subterm condition, $s \succ^* t_j$ and $t \succ^* u_k$ for any immediate subterms t_j of t and u_k of u. Let f, g, and h be the outermost operators of s, t, and u, respectively. The proof is by induction on the size of s, t, and u and considers five cases:

(1) $f \succ h$. By the definition of $s \succ^* u$, we must show that $s \succ^* u_k$ for all subterms u_k of u. But we are given that $s \succ^* t \succ^* u_k$ and the result follows by the induction hypothesis, since u_k is smaller than u.

(2) $f \not\succeq g, h$. We are given that $s_i \succeq^* t \succ^* u$ for some subterm s_i of s. By the induction hypothesis, $s_i \succ^* u$, since s_i is smaller than s, and by the definition of $\succ^*$, $s \succ^* u$.

(3) $f = h \not\succeq g$. We must show that $\{\ldots s_i \ldots\} \gg^* \{\ldots u_k \ldots\}$ and are given that $s_i \succeq^* t \succ^* u_k$ for some s_i and for all u_k. The result follows by the induction hypothesis.

(4) $g \not\succeq h$. We are given that $s \succ^* t_j$ for any subterm t_j of t, while by the definition of $t \succ^* u$, we have $t_j \succeq^* u$ for some t_j. Thus, $s \succ^* t_j \succeq^* u$, and $s \succ^* u$ follows from the induction hypothesis, since t_j is smaller than t.

(5) $f = g = h$. We must show that $\{\ldots s_i \ldots\} \gg^* \{\ldots u_k \ldots\}$ and are given that $\{\ldots s_i \ldots\} \gg^* \{\ldots t_j \ldots\} \gg^* \{\ldots u_k \ldots\}$. By the induction hypothesis, $s_i \succ^* t_j \succ^* u_k$ implies $s_i \succ^* u_k$ for all s_i, t_j, and u_k, and since the extension of a transitive relation to multisets is also transitive, it follows that $\{\ldots s_i \ldots\} \gg^* \{\ldots u_k \ldots\}$.

These five cases cover all possible relations between f, g, and h (if $f \not\succeq g$, then cases 1, 2, and 3 cover $f \succ h$, $f \not\succeq h$, and $f = h$, respectively; if $f \succeq g$, then cases 1, 4, and 5 cover $g \succeq h \neq f$, $g \not\succeq h$, and $g = h = f$, respectively). Thus, our proof of transitivity is complete.

Replacement: By the definition of a multiset ordering, $\{\ldots s \ldots\} \gg^* \{\ldots s' \ldots\}$ if $s \succ^* s'$. Therefore, by the definition of the recursive path ordering, $f(\ldots s \ldots) \succ^* f(\ldots s' \ldots)$.

Deletion: By the definition of a multiset ordering, $\{\ldots s \ldots\} \gg^* \{\ldots \; \ldots\}$. Thus, by the definition of the recursive path ordering, $f(\ldots s \ldots) \succ^* f(\ldots \; \ldots)$.

Since the recursive path ordering is a simplification ordering, it may be used in conjunction with the First Termination Theorem to prove the termination of term-rewriting systems. The following theorem gives a necessary and sufficient condition for the ordering to be well-founded.

Theorem 4. *The recursive path ordering* $>^*$ *on the set of terms* $T(F)$ *is well-founded, if and only if the partial ordering* $>$ *on the set of operators* F *is well-founded.*

Proof. The 'only-if' direction follows trivially from the fact that for $f, g \in F$, $f > g$ implies $f >^* g$.

The proof of the 'if' direction is an application of the Well-foundedness Theorem: If $>$ is well-founded, then (using Zorn's Lemma) it can be extended to some total well-founded ordering $>_-$ on F. Since F is then well-quasi-ordered under $\leq_+$ and the recursive path ordering $>^*_-$ satisfies the operator replacement condition, i.e. $f \geq_- g$ implies $f(t_1, \ldots, t_n) \geq^*_- g(t_1, \ldots, t_n)$ (by the subterm property), it follows (by the Well-foundedness Theorem) that $>^*_-$ is well-founded. But $>^*_-$ contains $>^*$ ($t >^* t'$ implies $t >^*_- t'$ by a straightforward induction); therefore, $>^*$ must be well-founded as well.

It turns out that when $>$ is a total ordering, the recursive path ordering $>^*$ is in effect the same as the 'path of subterms' ordering defined in Plaisted [24] in a more complex manner. When $>$ is partial, the recursive path ordering is contained in (an obvious extension of) the 'path of subterms' ordering. Our proof of well-foundedness extends to the latter as well; Plaisted's proof is considerably longer and requires that $>$ be total.

The 'multiset' and 'nested multiset' orderings in [6] and the 'simple path' ordering in [23] are special cases of this recursive path ordering in which the multiset constructor is greater than other operators. Their well-foundedness follows as a corollary of this theorem.

One way of extending the recursive path ordering is to allow some function of a term $f(t_1, \ldots, t_n)$ to serve the role of the operator f. For example, we can consider the kth operand t_k to be the operator, and compare two terms by first recursively comparing their kth operands. This yields a simplification ordering for the same reasons that the original definition does. Furthermore, this new ordering satisfies the operator replacement condition, i.e. $t_k \geq^* t'_k$ implies $f(t_1, \ldots, t_k, \ldots, t_n) \geq^* f(t_1, \ldots, t'_k, \ldots, t_n)$.

To prove that this extended ordering is well-founded, we appeal to the Well-foundedness Theorem. Define the depth $d(t)$ of a term t to be the maximum nesting of kth operands. It is easy to show that $s >^* t$, for two terms s and t, if $d(s) > d(t)$. Thus, it suffices to show for all i that terms of depth i are well-quasi-ordered and consequently well-founded. This follows immediately from the Well-foundedness Theorem by induction on the depth i.

Further extensions of the recursive path ordering may be found in Kamin and Levy [12].

Other examples of simplification orderings are the ('linear') ordering in [13] and the 'polynomial' ordering in [16]. Whereas these methods require that terms be mapped onto the well-founded nonnegative integers, using simplification orderings allows the methods to be extended to domains that are not themselves well-founded.

For example, in Dershowitz [5] we suggest associating a polynomial $F(x_1, \ldots, x_n)$ over the reals with each n-ary operator f. This mapping extends to a morphism φ on terms by letting $\varphi(f(t_1, \ldots, t_n)) = F(\varphi(t_1), \ldots, \varphi(t_n))$. For any choice of polynomials F, one must have that $x_i \geqslant x_i'$ implies $F(x_1, \ldots, x_i, \ldots, x_n) \geqslant F(x_1, \ldots, x_i', \ldots, x_n)$ and $F(x_1, \ldots, x_i, \ldots, x_n) \geqslant x_i$ for all positions i and for all real-valued x's, and that $\varphi(l_i) > \varphi(r_i)$ for all rules $l_i \to r_i$ and for all real value assignments $\varphi(\alpha)$ to the variables α in l_i. (Allowing the x's to take on any real value is usually too strong a requirement; instead one may show that terms always map into some subset R' of the reals, i.e. $x_1, \ldots, x_n \in R'$ implies $F(x_1, \ldots, x_n) \in R'$. Then one need only show that the conditions hold for all x's in R'.)

These conditions are all decidable (albeit in superexponential time) for polynomials over the reals (Tarski [27]; see Cohen [4] for a much briefer decision procedure). Thus, the polynomial ordering can be effectively 'lifted' to open terms. It is similarly decidable if there exist polynomials F (and predicate R') of a given maximum degree that satisfy the conditions and thereby prove termination. (The procedure, however, cannot point to the appropriate polynomials.) For polynomials over the natural numbers, these conditions are not decidable (see [16]).

5. Examples

We return, in this section, to the six examples (A–F) of term-rewriting systems that have been presented in the previous sections. We prove that four of them terminate, as do two additional examples (G–H).

(A) Our first example was the following system for computing the disjunctive normal form of a logical formula:

$$\neg\neg\alpha \to \alpha$$

$$\neg(\alpha \vee \beta) \to (\neg\alpha \wedge \neg\beta)$$

$$\neg(\alpha \wedge \beta) \to (\neg\alpha \vee \neg\beta)$$

$$\alpha \wedge (\beta \vee \gamma) \to (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$$

$$(\beta \vee \gamma) \wedge \alpha \to (\beta \wedge \alpha) \vee (\gamma \wedge \alpha).$$

We wish to prove that this system terminates for all inputs. It can be shown that no polynomial ordering reduces for all five rules. We can, however, use a recursive path ordering on terms with operators $\neg$, $\wedge$, and $\vee$ ordered by $\neg > \wedge > \vee$. Since this is a simplification ordering on terms, by the First Termination Theorem, we need only show that

$$\neg\neg\alpha >^* \alpha,$$

$$\neg(\alpha \vee \beta) >^* (\neg\alpha \wedge \neg\beta),$$

$$\neg(\alpha \wedge \beta) >^* (\neg\alpha \vee \neg\beta),$$

$$\alpha \wedge (\beta \vee \gamma) >^* (\alpha \wedge \beta) \vee (\alpha \wedge \gamma),$$

$$(\beta \vee \gamma) \wedge \alpha >^* (\beta \wedge \alpha) \vee (\gamma \wedge \alpha),$$

for any terms α, β, and γ.

The first inequality follows from the subterm condition of simplification orderings. By the definition of the recursive path ordering, to show that $\neg(\alpha \vee \beta) >^* (\neg\alpha \wedge \neg\beta)$ when $\neg > \wedge$, we must show that $\neg(\alpha \vee \beta) >^* \neg\alpha$ and $\neg(\alpha \wedge \beta) >^* \neg\beta$. Now, since the outermost operators of $\neg(\alpha \vee \beta)$, $\neg\alpha$, and $\neg\beta$ are the same, we must show that $\alpha \vee \beta >^* \alpha$ and $\alpha \vee \beta >^* \beta$. But this is true by the subterm condition. Thus the second inequality holds. By an analogous argument, the third inequality also holds.

For the fourth inequality, we must show $\alpha \wedge (\beta \vee \gamma) >^* (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$. Since $\wedge > \vee$, we must show $\alpha \wedge (\beta \vee \gamma) >^* \alpha \wedge \beta$ and $\alpha \wedge (\beta \vee \gamma) >^* \alpha \wedge \gamma$. By the definition of the recursive path ordering for the case when two terms have the same outermost operator, we must show that $\{\alpha, \beta \vee \gamma\} \gg^* \{\alpha, \beta\}$ and $\{\alpha, \beta \vee \gamma\} \gg^* \{\alpha, \gamma\}$. These two inequalities between multisets hold, since the element $\beta \vee \gamma$ is greater than both β and γ with which it is replaced. Thus the fourth inequality holds. Similarly the fifth inequality may be shown to hold. Therefore, by the First Termination Theorem, this system terminates for all inputs.

(B) The variant

$$\neg\neg\alpha \rightarrow \alpha$$

$$\neg(\alpha \vee \beta) \rightarrow (\neg\neg\neg\alpha \wedge \neg\neg\neg\beta)$$

$$\neg(\alpha \wedge \beta) \rightarrow (\neg\neg\neg\alpha \vee \neg\neg\neg\beta)$$

$$\alpha \wedge (\beta \vee \gamma) \rightarrow (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$$

$$(\beta \vee \gamma) \wedge \alpha \rightarrow (\beta \wedge \alpha) \vee (\gamma \wedge \alpha)$$

of System (A) does not in fact terminate for all inputs, though whenever it does terminate, the resulting expression is in disjunctive normal form.

To see that it does not terminate, consider the following derivation:

$$\begin{aligned}
&\neg\neg(a \wedge (a \vee b)) \Rightarrow \neg\neg((a \wedge a) \vee (a \wedge b))\\
&\quad \Rightarrow \neg(\neg\neg\neg(a \wedge a) \wedge \neg\neg\neg(a \wedge b)) \Rightarrow \cdots \Rightarrow \neg(\neg(a \wedge a) \wedge \neg(a \wedge b))\\
&\quad \Rightarrow \cdots \Rightarrow \neg((\neg\neg\neg a \vee \neg\neg\neg a) \wedge (\neg\neg\neg a \vee \neg\neg\neg b))\\
&\quad \Rightarrow \cdots \Rightarrow \neg((\neg a \vee \neg a) \wedge (\neg a \vee \neg b))\\
&\quad \Rightarrow \neg((\neg a \wedge (\neg a \vee \neg b)) \vee (\neg a \wedge (\neg a \vee \neg b)))\\
&\quad \Rightarrow (\neg\neg\neg(\neg a \wedge (\neg a \vee \neg b)) \wedge \neg\neg\neg(\neg a \wedge (\neg a \vee \neg b))) \Rightarrow \cdots .
\end{aligned}$$

Thus, beginning with a term of the form $\neg\neg(\alpha \wedge (\alpha \vee \beta))$, a term containing a subterm of the same form is derived, and the process may continue ad infinitum.

(C) Our third example was

$$\neg\neg\alpha \rightarrow \alpha$$

$$\neg(\alpha \vee \beta) \rightarrow (\neg\neg\neg\alpha \wedge \neg\neg\neg\beta)$$

$$\neg(\alpha \wedge \beta) \rightarrow (\neg\neg\neg\alpha \vee \neg\neg\neg\beta).$$

We cannot order the operators so as to enable the use of a recursive path ordering to prove the termination of this system. Instead, we use the Second Termination Theorem and define the following quasi-simplification ordering: $t \gtrsim t'$ for two terms t and t', if and only if

$$[t] \geqslant [t'] \quad \text{and}$$

$$\{[\alpha]\colon \neg\alpha \text{ appears in } t\} \geqq \{[\alpha]\colon \neg\alpha \text{ appears in } t'\},$$

where $[\alpha]$ denotes the number of occurrences of operators other than $\neg$ in α, and $\geqq$ means either $\gg$ in the multiset extension of the ordering $>$ on numbers, or else $=$.

It is easy to see that this quasi-ordering satisfies the replacement and subterm properties of quasi-simplification orderings on fixed-arity terms. It remains to show that each rule reduces the subterm it is applied to under the ordering $\succ$. For all three rules the number of operators other than $\neg$ is the same on both sides. To see that

$$\neg\neg\alpha \succ \alpha,$$

note that there are two less elements in the multiset of numbers of operators for the right-hand side than for the left-hand side. To see that

$$\neg(\alpha \vee \beta) \succ (\neg\neg\neg\alpha \wedge \neg\neg\neg\beta) \quad \text{and} \quad \neg(\alpha \wedge \beta) \succ (\neg\neg\neg\alpha \vee \neg\neg\neg\beta),$$

note that the number of operators other than $\neg$ in $\alpha \vee \beta$ and $\alpha \wedge \beta$ is greater than that of $\neg\neg\alpha$, $\neg\alpha$, α, $\neg\neg\beta$, $\neg\beta$, and β. Thus the multisets corresponding to the left-hand sides are strictly greater than those for the right-hand sides.

(D) The system

$$\neg\neg\alpha \rightarrow \alpha$$

$$\neg(\alpha \vee \beta) \rightarrow ((\neg\neg\neg\alpha \wedge \neg\neg\neg\beta) \wedge (\neg\neg\neg\alpha \wedge \neg\neg\neg\beta))$$

$$\neg(\alpha \wedge \beta) \rightarrow ((\neg\neg\neg\alpha \vee \neg\neg\neg\beta) \vee (\neg\neg\neg\alpha \vee \neg\neg\neg\beta))$$

$$(\alpha \wedge \alpha) \rightarrow \alpha$$

$$(\alpha \vee \alpha) \rightarrow \alpha,$$

however, does not terminate. The following derivation demonstrates this:

$$\neg\neg(a \wedge b) \Rightarrow \neg((\neg\neg\neg a \vee \neg\neg\neg b) \vee (\neg\neg\neg a \vee \neg\neg\neg b)) \Rightarrow \cdots$$

$$\Rightarrow \neg((\neg a \vee \neg b) \vee (\neg a \vee \neg b))$$

$$\Rightarrow((\neg\neg\neg(\neg a \vee \neg b) \wedge \neg\neg\neg(\neg a \vee \neg b))$$
$$\wedge (\neg\neg\neg(\neg a \vee \neg b) \wedge \neg\neg\neg(\neg a \vee \neg b))) \Rightarrow \cdots$$
$$\Rightarrow \neg\neg\neg(\neg a \vee \neg b) \Rightarrow \cdots \Rightarrow \neg\neg\neg\neg(\neg\neg a \wedge \neg\neg b) \Rightarrow \cdots .$$

(E) The proof of the termination of the system

$$\neg\neg\alpha \rightarrow \alpha$$
$$\neg(\alpha \vee \beta) \rightarrow ((\neg\alpha \wedge \neg\beta) \wedge (\neg\alpha \wedge \neg\beta))$$
$$\neg(\alpha \wedge \beta) \rightarrow ((\neg\alpha \vee \neg\beta) \vee (\neg\alpha \vee \neg\beta))$$
$$(\alpha \wedge \alpha) \rightarrow \alpha$$
$$(\alpha \vee \alpha) \rightarrow \alpha.$$

is similar to that of System (A). We use the recursive path ordering with the operators partially ordered by $\neg > \wedge$ and $\neg > \vee$.

We have

$$\neg\neg\alpha >^* \alpha, \qquad (\alpha \wedge \alpha) >^* \alpha \quad \text{and} \quad (\alpha \vee \alpha) >^* \alpha,$$

by the subterm condition; we have

$$\neg(\alpha \vee \beta) >^* ((\neg\alpha \wedge \neg\beta) \wedge (\neg\alpha \wedge \neg\beta)) \quad \text{and} \quad \neg(\alpha \wedge \beta) >^* ((\neg\alpha \vee \neg\beta) \vee (\neg\alpha \vee \neg\beta)),$$

since $\neg$ is greater than both $\wedge$ and $\vee$, and the subterms $\alpha \vee \beta$ and $\alpha \wedge \beta$ are greater than either α or β by the subterm condition.

Using the recursive path ordering to prove the termination of systems in this manner, generalizes the conditions for termination in [11]. The cases where Iturriaga's method works are those for which the operators are partially ordered so that the outermost ('virtual') operators of the left-hand side of the rules are greater than any other ('complementary') operators on the left-hand side, which in turn are greater than any other operators.

(F) To prove the termination of the one-rule system

$$(\alpha \wedge \beta) \wedge \gamma \rightarrow \alpha \wedge (\beta \wedge \gamma),$$

we again use the Second Termination Theorem. We define the quasi-ordering $t \gtrsim t'$, if and only if

$$|t| > |t'|$$

or else (t and t' are conjunctions and)

$$|t| = |t'| \quad \text{and} \quad |t_1| \geq |t_1'|,$$

where $|\alpha|$ denotes the total number of occurrences of operators in α and t_1 and t_1' are the left conjuncts of t and t', respectively.

To see that this is a quasi-simplification ordering, note that $t \gtrsim t'$ implies $|t| \geq |t'|$. Replacing a right conjunct t_2 with a smaller or equivalent one (under $\gtrsim$) can only decrease the total size of a conjunction $t = t_1 \wedge t_2$ and cannot change the size of t_1; replacing t_1 with a smaller or equivalent left conjunct cannot increase the size of t or t_1. The subterm condition $t_1 \wedge t_2 \gtrsim t_1, t_2$ obviously holds since $|t_1 \wedge t_2| > |t_1|, |t_2|$.

It remains to show that

$$(\alpha \wedge \beta) \wedge \gamma > \alpha \wedge (\beta \wedge \gamma).$$

But $|(\alpha \wedge \beta) \wedge \gamma| = |\alpha \wedge (\beta \wedge \gamma)|$, while $|\alpha \wedge \beta| > |\alpha|$, and the proof is complete.

This example illustrates how the conditions for termination required by the methods of Knuth and Bendix [13] and Lankford [16] may be relaxed: Given a quasi-ordering $\gtrsim_F$ on (fixed arity) operators and a quasi-simplification ordering $\gtrsim_T$ on terms, such that

$$f(\ldots t \ldots) \approx_T t \text{ implies } f \text{ unary and}$$
$$f \gtrsim_F g \text{ for all operators } g, \qquad (*)$$

we define the quasi-simplification ordering

$$s = f(s_1, \ldots, s_m) \gtrsim g(t_1, \ldots, t_n) = t,$$

if and only if

$$(s, f, s_1, \ldots, s_m) \gtrsim (t, g, t_1, \ldots, t_n),$$

where the two tuples are compared lexicographically, first according to the terms $s \gtrsim_T t$, then according to the operators $f \gtrsim_F g$, and finally according to the subterms $s_i \gtrsim_T t_i$ (or, alternatively, $s_i \gtrsim t_i$ recursively). The condition $(*)$ ensure that $\gtrsim$ possesses the subterm property. To prove termination, one must find appropriate quasi-orderings $\gtrsim_F$ and $\gtrsim_T$ for which $l_i > r_i$ for all rules $l_i \to r_i$ in the given system. In the above example: $s \gtrsim_T t$ if and only if $|s| \geqslant |t|$, and $\gtrsim_F$ is equality. (This method applies also to example (C) with $s >_T t$ if and only if $[s] \geqslant [t]$, $\neg >_F g$ for all other g, and subterms compared recursively.

The method of Knuth and Bendix assigns a positive integer weight to each zeroary operator and a nonnegative integer weight to each other operator, with $\gtrsim_T$ comparing terms according to the sum of the weights of their respective operators, $\gtrsim_F$ a total ordering of operators, and subterms compared recursively. Thus, $(*)$ requires that a unary operator have zero weight only if it is the largest operator under $>_F$. Lankford replaces the linear sum of weights function with monotonic polynomials having nonnegative integer coefficients. Since both these methods use total monotonic orderings, the subterm condition is both necessary and sufficient for the orderings to be well-founded; the integer requirements are not themselves necessary. Thus, instead of using a specific linear or polynomial ordering to orient rules generated by the Knuth–Bendix 'completion' algorithm [13], one could use a decision procedure for real polynomials to determine, at each step of the algorithm and for both possible orientations, whether there exists any ordering of a specific degree that reduces for all the rules obtained.

This example also illustrates the use of quasi-termination: The quasi-ordering $\gtrsim$, where $t \gtrsim t'$ if and only if $|t| \geqslant |t'|$, is a monotonic extension of the simplification ordering $t > t'$. Since $(\alpha \wedge \beta) \wedge \gamma \approx \alpha \wedge (\beta \wedge \gamma)$, the system quasi-terminates. To complete the proof of termination, the monotonic quasi-ordering $\gtrsim'$, where $t \gtrsim' t'$ if and only if $|t| = |t'|$ and $|t_1| \geqslant |t'_1|$, may be used.

The method of Lipton and Snyder [18] is somewhat similar in its use of quasi-termination. Whereas they require that $>$ be a well-founded ω-ordering, we require it to be monotonic; without monotonicity they must insist that $l_i \approx r_i$ and not $l_i \gtrsim r_i$. For $t \gtrsim' t'$ they use $|t| \leqslant |t'|$.

(G) To illustrate the use of an operand as an operator in a recursive path ordering, consider the one-rule system

$$if(if(\alpha, \beta, \gamma), \delta, \varepsilon) \rightarrow if(\alpha, if(\beta, \delta, \varepsilon), if(\gamma, \delta, \varepsilon)).$$

The conditional expression "$if(\alpha, \beta, \gamma)$" stands for "if α then β else γ" and this system 'normalizes' conditional expressions by repeatedly removing nested *if*'s from the condition α.

To see that this system terminates we consider the condition to be the operator. The condition $if(\alpha, \beta, \gamma)$ of the left-hand side is greater (by the subterm property) than the condition α of the right-hand side. Thus, we need to show that the left-hand side is greater than both right-hand-side operands $if(\beta, \delta, \varepsilon)$ and $if(\gamma, \delta, \varepsilon)$. Again, $if(\alpha, \beta, \gamma)$ is greater than both operators β and γ, and now the left-hand side is clearly greater than the remaining operands δ and ε.

This method would work for system (F) as well.

(H) Finally, consider the system

$$\neg\neg\alpha \rightarrow \alpha$$

$$\neg(\alpha \vee \beta) \rightarrow (\neg\alpha \wedge \neg\beta)$$

$$\neg(\alpha \wedge \beta) \rightarrow (\neg\alpha \vee \neg\beta)$$

$$\alpha \wedge (\beta \vee \gamma) \rightarrow (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$$

$$(\beta \vee \gamma) \wedge \alpha \rightarrow (\beta \wedge \alpha) \vee (\gamma \wedge \alpha)$$

$$(\alpha \wedge \beta) \wedge \gamma \rightarrow \alpha \wedge (\beta \wedge \gamma)$$

$$\alpha \vee (\beta \vee \gamma) \rightarrow (\alpha \vee \beta) \vee \gamma$$

$$(\alpha \vee \alpha) \rightarrow \alpha$$

$$(\alpha \wedge \alpha) \rightarrow \alpha,$$

combining the rules for disjunctive normal form of system (A) with associativity of conjunction and disjunction as in system (F). Unfortunately, the orderings used for each of those two systems can increase for the other system.

Nevertheless, we can combine the recursive path idea used for (A) with the lexicographic idea of (F) by using operands as operators in a recursive path ordering. We let $\neg > \wedge > \vee$ as for (A), but use the first operand as the operator when comparing two conjunctions and the second operand as operator when comparing disjunctions. (This is similar to the use of lexicographic recursive path orderings in Kamin and Levy [12].)

Acknowledgment

I gratefully acknowledge the valuable discussions with David Plaisted, Tom Brown, and Gérard Huet, that helped lead to the results reported here.

References

[1] A.M. Ballantyne and W.W. Bledsoe, Automatic proofs of theorems in analysis using nonstandard techniques, *J. ACM* **24** (3) (1977) 353–374.

[2] R.S. Boyer and J S. Moore, A lemma driven automatic theorem prover for recursive function theory, *Proc. 5th International Joint Conference on Artificial Intelligence* (MIT, Cambridge, MA, 1977) 511–519.

[3] W.C. Carter, H.A. Ellozy, W.H. Joyner, Jr. and G.B. Leeman, Jr., Techniques for microprogram validation, Memo RC6361, IBM T.J. Watson Research Center, Yorktown Heights, NY (1977).

[4] P.J. Cohen, Decision procedures for real and p-adic fields, *Comm. Pure Appl. Math.* **22** (2) (1969) 131–151.

[5] N. Dershowitz, A note on simplification orderings, *Information Processing Lett.* **9** (5) (1979) 212–215.

[6] N. Dershowitz and Z. Manna, Proving termination with multiset orderings, *Comm. ACM* **22** (8) (1979) 465–476.

[7] J.H. Griesmer and R.D. Jenks, SCRATCHPAD/1—An interactive facility for symbolic mathematics, *Proc. 2nd Symposium on Symbolic and Algebraic Manipulation*, Los Angeles, CA (ACM, New York, 1971) 42–48.

[8] A.C. Hearn, REDUCE 2—A system and language for algebraic manipulation, *Proc. 2nd Symposium on Symbolic and Algebraic Manipulation*, Los Angeles, CA (ACM, New York, 1971) 128–133.

[9] G. Huet, Confluent reductions: Abstract properties and applications to term rewriting systems, *J. ACM* **27** (4) (1980) 797–821.

[10] G. Huet and D.S. Lankford, On the uniform halting problem for term rewriting systems, Report 283, INRIA, Le Chesnay, France (1978).

[11] R. Iturriaga, Contributions to mechanical mathematics, Ph.D. thesis, Carnegie–Mellon University, Pittsburgh, PA (1967).

[12] S. Kamin and J.-J. Levy, Two generalizations of the recursive path orderings, Unpublished note, Department of Computer Science, University of Illinois, Urbana, IL (1980).

[13] D.E. Knuth and P.B. Bendix, Simple word problems in universal algebras, in: J. Leech, Ed., *Computational Problems in Abstract Algebra* (Pergamon Press, Oxford, 1969) 263–297.

[14] J.B. Kruskal, Well-quasi-ordering, the tree theorem, and Vazsonyi's conjecture, *Trans. Amer. Math. Soc.* **95** (1960) 210–225.

[15] D.S. Lankford, Canonical algebraic simplification in computational logic, Memo ATP-25, Automatic Theorem Proving Project, University of Texas, Austin, TX (1975).

[16] D.S. Lankford, On proving term rewriting systems are Noetherian, Memo MTP-3, Mathematics Department, Louisiana Technical University, Ruston, LA (1979).

[17] D.S. Lankford and A.M. Ballantyne, Decision procedures for simple equational theories with permutative axioms: Complete sets of permutative reductions, Report ATP-37, Departments of Mathematics and Computer Sciences, University of Texas, Austin, TX (1977).

[18] R.J. Lipton and L. Snyder, On the halting of tree replacement systems, *Proc. Conference on Theoretical Computer Science*, University of Waterloo, Waterloo, Ontario (1977) 43–46.

[19] Z. Manna and S. Ness, On the termination of Markov algorithms, *Proc. 3rd Hawaii International Conference on System Sciences*, Honolulu, HI (1970) 789–792.

[20] J. Moses, Algebraic simplification, a guide for the perplexed, *Comm. ACM* **14** (8) (1971) 527–537.

[21] D.R. Musser, A data type verification system based on rewrite rules, Memo, Information Sciences Institute, University of Southern California, Marina del Ray, CA (1978).

[22] C.St.J.A. Nash-Williams, On well-quasi-ordering finite trees, *Proc. Cambridge Philos. Soc.* **59** (1963) 833–835.

[23] D. Plaisted, Well-founded orderings for proving termination of systems of rewrite rules, Report R-78-932, Department of Computer Science, University of Illinois, Urbana, IL (1978).

[24] D. Plaisted, A recursively defined ordering for proving termination of term rewriting systems, Report R-78-943, Department of Computer Science, University of Illinois, Urbana, IL (1978).

[25] J.R. Slagle, Automated theorem-proving for theories with simplifiers, commutativity and associativity, *J. ACM* **21** (4) (1974) 622–642.

[26] M.E. Stickel and G.E. Peterson, Complete sets of reductions for some equational theories, *J. ACM* **28** (2) (1981) 233–264.

[27] A. Tarski, A Decision Method for Elementary Algebra and Geometry (University of California Press, Berkeley, CA, 1951).

[28] R.W. Weyhrauch, A users manual for FOL, Memo AIM-235.1, Artificial Intelligence Laboratory, Stanford University, Stanford, CA (1977).