# TERMINATION*

Nachum Dershowitz

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

## ABSTRACT

This survey describes methods for proving that systems of rewrite rules terminate. Illustrations of the use of path orderings and other simplification orderings in termination proofs are given. The effect of restrictions, such as linearity, on the form of rules is considered. In general, though, termination is an undecidable property of rewrite systems.

## 1. INTRODUCTION

A *term-rewriting (rewrite) system* $R$ over a set of terms $\mathcal{T}$ is a set of *rewrite rules,* each of the form $l \rightarrow r$, where $l$ and $r$ are terms in $\mathcal{T}$ or are terms containing *variables* ranging over $\mathcal{T}$. Such a rule applies to a term $t$ in $\mathcal{T}$ if a subterm $s$ of $t$ matches the left-hand side $l$ with some substitution $\sigma$ of terms in $\mathcal{T}$ for variables appearing in $l$ (i.e. $s = l\sigma$). The rule is applied by replacing the subterm $s$ in $t$ with the corresponding right-hand side $r\sigma$ of the rule, within which the same substitution $\sigma$ of terms for variables has been made. We write $t \Rightarrow_R u$, or just $t \Rightarrow u$, to indicate that a term $u$ in $\mathcal{T}$ is *derivable* in this way from the term $t$ in $\mathcal{T}$ by a single application of some rule in $R$. If $t \Rightarrow \cdots \Rightarrow u$ in zero or more steps, abbreviated $t \Rightarrow^* u$, then we say that $t$ *reduces* to $u$; if no rule can be applied to $t$, we say that $t$ is *irreducible;* when $t$ reduces to an irreducible term $u$, we say that $u$ is a *normal form* of $t$.

There are five properties involved in the verification of rewrite systems:

1) *termination*—no infinite derivations are possible,

2) *confluence*—each term has at most one normal form,

3) *soundness*—terms are only rewritten to equal terms,

4) *completeness*—equal terms have the same normal form,

5) *correctness*—all normal forms satisfy given requirements.

---

*The preparation of this survey was supported in part by the National Science Foundation under Grant MCS 83-07755.

This survey is devoted to a discussion of the first aspect, namely termination, generally a prerequisite for demonstrating other properties. Two related concepts, only briefly discussed, are "quasi-termination" and "weak termination." A quasi-terminating rewrite system is one for which only a *finite* number of different terms are derivable from any given term. A weakly-terminating system is one for which every term has at least one normal form.

Consider, for example, the following simple system consisting of three rules:

$$\begin{array}{rcl} \textit{white},\textit{red} & \rightarrow & \textit{red},\textit{white} \\ \textit{blue},\textit{red} & \rightarrow & \textit{red},\textit{blue} \\ \textit{blue},\textit{white} & \rightarrow & \textit{white},\textit{blue} \end{array} \tag{0}$$

This program plays the "Dutch National Flag" game. Given a sequence of marbles, colored *red*, *white*, or *blue* and placed side by side in no particular order, the program rearranges the marbles so that all *red* ones are on the left, all *blue* ones are on the right, and all *white* ones are in the middle. The first rule, for example, states that if anywhere in the series there is an adjacent pair of marbles, the left one *white* and the right one *red*, then they should be exchanged so that the *red* marble is on the left and the *white* one is on the right. It is not hard to prove that, regardless of the initial arrangement of marbles, applying the above rules in any order always results in a sequence of correctly arranged marbles. As we will see, a termination proof can be based on the ordering

*blue* is greater than *white* and *white* is greater than *red*.

Each rule replaces two marbles, the one on the left with "greater" color is exchanged with the "smaller" one to its right.

To illustrate the difficulty often encountered when attempting to determine if, and why, a rewrite system terminates, consider the following system (for disjunctive normal form):

$$\begin{array}{rcl} --\alpha & \rightarrow & \alpha \\ -(\alpha+\beta) & \rightarrow & -\alpha\times-\beta \\ -(\alpha\times\beta) & \rightarrow & -\alpha+-\beta \\ \alpha\times(\beta+\gamma) & \rightarrow & (\alpha\times\beta)+(\alpha\times\gamma) \\ (\beta+\gamma)\times\alpha & \rightarrow & (\beta\times\alpha)+(\gamma\times\alpha) \end{array} \tag{1}$$

The first rule eliminates double negations; the second and third rules apply DeMorgan's laws to push negations inward; the last two apply the distributivity of $\times$ over $+$. The difficulty in proving termination for systems such as this stems from the fact that while some rewrites may decrease the size of a term, other rewrites 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. And a proof of termination must take into consideration the many different possible rewrite sequences generated by the nondeterministic choice of rules and subterms.

Various methods for proving termination of rewrite systems have been suggested, including [Gorn-67, Iturriaga-67, Knuth-Bendix-70, Manna-Ness-70, Gorn-73, Lankford-75, Lipton-Snyder-77, Plaisted-78, Plaisted-78b, Dershowitz-Manna-79, Lankford-79, Kamin-Levy-80, Pettorossi-81, Dershowitz-82, Jouannaud,*etal.*-82, Dershowitz,*etal.*-83, Lescanne-84, Jouannaud-Munoz-84, Kapur,*etal.*-85, Bachmair-Plaisted-85, Bachmair-Dershowitz-85, Rusinowitch-85]. Termination is in general an undecidable property of rewrite systems (as it is for Markov systems on strings; see [Huet-Lankford-78]). For a lively discussion of tasks that are difficult to show terminating, see [Gardner-83].

In the next section we prove that termination is undecidable. In Section 3 we show how *well-founded orderings* are used in termination proofs, and in Section 4 we show how *simplification orderings* are used. Similar methods are described in Section 5 for using *quasi-orderings* to prove termination (or quasi-termination). Section 6 presents *multiset orderings.* Then, in Section 7, we define *path orderings* based on an underlying operator "precedence". This is followed in the last two sections with methods for determining if rewrite systems of restricted form terminate (or weakly-terminate). Examples are provided throughout; proofs are generally omitted.

## 2. NONTERMINATION

Given a set of operators $F$, we consider the set $\mathcal{T}(F)$ of all terms constructed from operators in $F$. Operators in $F$ may be *varyadic,* i.e. have variable arity, in which case if $f$ is an operator and $t_1, \ldots, t_n$ $(n \geq 0)$ are terms in $\mathcal{T}(F)$, then $f(t_1, \ldots, t_n)$ is also a term in $\mathcal{T}(F)$. Or an operator $f$ may be restricted to a fixed arity, in which case $f(t_1, \ldots, t_n) \in \mathcal{T}$ only if $f$ is of arity $n$.

**Definition 1.** A rewrite system $R$ is *terminating* for a set of terms $\mathcal{T}$, if there exist no infinite sequence of terms $t_i \in \mathcal{T}$ such that $t_1 \Rightarrow t_2 \Rightarrow t_3 \Rightarrow \cdots$. A system is *nonterminating* if there exists any such infinite derivation. A system is *weakly-terminating* if for each term $t \in \mathcal{T}$ there is an irreducible term derivable from $t$.

Terminating systems are variously called *finitely terminating, uniformly terminating, strongly terminating,* and *noetherian.* Unless indicated otherwise, when we speak of termination, we mean with respect to *all* terms constructed from a given set of (fixed or variable) operators $F$. Rules of a terminating system are called *reductions.*

*Example.* A trivial example of a terminating system is

$$--\alpha \rightarrow \alpha. \tag{2}$$

An equally trivial example of a nonterminating system is

$$-\alpha \rightarrow ---\alpha. \tag{3}$$

A less trivial example (of what?) is

$$-(\alpha+\beta) \rightarrow (--\alpha+\beta)+\beta. \tag{4}$$

An example of a non-weakly-terminating system is

$$f(g(\alpha)) \rightarrow g(g(f(f(\alpha)))). \tag{5}$$

**Theorem 1.** *Termination of rewrite systems is undecidable, even if the system has only two rules.*

*Proof.* Turing machines can be simulated by rewrite systems. Given any Turing machine $M$, there exists a two-rule system $R_M$ such that $R_M$ terminates for all initial terms if, and only if, $M$ halts for all input tapes. Since it is undecidable (not even semi-decidable) if a Turing machine halts uniformly, it is also undecidable if rewrite systems terminate.

Each state symbol and tape symbol of the machine will be a constant in the system. Additionally, we need three operators: a binary operator (which we will denote by adjacency and assume associates to the right), a unary operator $\partial$ (the *erase* function), and a ternary operator $C$.[1] We use an additional constant $\Box$ to denote the end of the tape. Corresponding to a machine in state $q$ with nonblank left portion of the tape $a_1a_2 \cdots a_m$ (from the left end until the symbol preceding the read head) and right portion $b_1b_2 \cdots b_n$ (from the symbol being scanned to the end), is the term

$$C(a_m \cdots a_2a_1\Box, qb_1b_2 \cdots b_n\Box, machine),$$

where *machine* is a term encoding transitions as subterms of the form

$$\partial(\alpha\sigma\beta\ \gamma\gamma'\sigma'\delta'\delta)$$

signifying "if the machine is in state $\sigma$ reading the symbol $\beta$ and the symbol immediately to left of $\beta$ is $\alpha$, then replace the tape segment $\alpha\beta$ with $\gamma\gamma'\delta'\delta$, position the head on $\delta'$, and go into state $\sigma'$." Any extra tape symbols introduced in this way, are placed within an "erase" term $\partial$. Thus, for each left-moving instruction of the form "if in state $q$ reading $a$, write $a'$, move left, and go into state $q'$," there are subterms of the form

$$\partial(sqa\ \partial(\#)\partial(\#)q'sa')$$

for *every* tape symbol $s$, as well as an extra subterm of the form

$$\partial(\Box qa\ \Box\partial(\#)q'\#a')$$

(where $\#$ is the blank symbol) to handle the left end of the tape. For each right-moving instruction of the form "if in state $q$ reading $a$, write $a'$, move right, and go into state $q'$," there are subterms of the form

$$\partial(sqa\ sa'q'\partial(\#)\partial(\#))$$

for *every* tape symbol $s$, as well as extra subterm of the form

$$\partial(sq\Box\ sa'q'\partial(\#)\Box)$$

when $a$ is the blank symbol $\#$ (to handle the right end of the tape). The term *machine* is

[1]Cf. [Bergstra-Tucker-80], where it is shown that *six* "hidden" functions suffice for the specification of com-

the concatenation of all transitions.

The rewrite system $R_M$ consists of exactly two rules:

$$\begin{array}{rcl} \partial(\xi)\tau & \rightarrow & \tau \\ C(\alpha\lambda,\sigma\beta\rho,\partial(\alpha\sigma\beta\ \gamma\gamma'\sigma'\delta'\delta)\tau) & \rightarrow & C(\gamma'\gamma\lambda,\sigma'\delta'\delta\rho,machine). \end{array}$$

The first rule erases transitions from the machine description until an applicable one reaches the beginning of the description, at which time the second rule can be applied to simulate a move. Though there are rewrite sequences that erase all applicable transitions and therefore do not correspond to a machine computation, those sequences all terminate. Clearly, if the machine $M$ does not terminate for some input tape, then the system $R_M$ does not terminate for the corresponding input term. Note that no rewrite step can increase the number of occurrences of the operator $C$ in a term. Thus, the only way for $R_M$ not to terminate is for one of the occurrences of $C$ to be infinitely rewritten, in a manner corresponding to an infinite computation of $M$. □

An alternative proof of undecidability of termination is given in [Huet-Lankford-78]; see Section 9. The number of rules in that proof depends on the number of machine transitions.[2]

Though termination of a rewrite system means that all (infinitely many) possible derivations are finite, one need only consider derivations that begin with certain terms:

> **Lemma 1.** *A rewrite system is terminating (for all terms) if, and only if, it terminates for all instances of its left-hand sides.*

By an *instance* of a left-hand side $l$ we mean a term $l\sigma$ with terms substituted for the variables of the left-hand side. Certainly, if a derivation repeats a term, the system is nonterminating. We say that

> **Definition 2.** A derivation $t_1 \Rightarrow t_2 \Rightarrow \cdots \Rightarrow t_j \Rightarrow \cdots \Rightarrow t_k \Rightarrow \cdots$ *cycles* if $t_j = t_k$ for some $j < k$. A rewrite system *cycles* if it has a cycling derivation.

Cycling is a special case of "looping":

> **Definition 3.** A derivation $t_1 \Rightarrow t_2 \Rightarrow \cdots \Rightarrow t_j \Rightarrow \cdots \Rightarrow t_k \Rightarrow \cdots$ *loops* if $t_j$ is a (not necessarily proper) subterm of $t_k$ for some $j < k$. A rewrite system *loops* if it has a looping derivation.

It is also obvious that looping systems do not terminate. But a system need not be looping to be nonterminating.

*Example.* System (4) does not terminate. The following infinite derivation begins with an instance of its left-hand side, but the system is nonlooping:

---

putable data types. In fact, *three* do.

[2]Perhaps a proof along the lines of the one given above was intended by [Lipton-Snyder-77] when they asserted, *sans* proof, that *three* rules suffice for undecidability.

$$
\begin{aligned}
--(0+1) &\Rightarrow -((--0+1)+1) \\
&\Rightarrow (--(--0+1)+1)+1 \\
&\Rightarrow (-((----0+1)+1)+1)+1 \\
&\Rightarrow (((--(----0+1)+1)+1)+1)+1 \\
&\Rightarrow \cdots
\end{aligned}
$$

To characterize nontermination, therefore, a notion weaker than looping is needed. Viewing terms as ordered trees suggests the following definition:

**Definition 4.** A term $s$ is *homeomorphically embedded* in a term $t$ written $s \trianglelefteq t$, if, and only if, $s$ is of the form $f(s_1,s_2,\ldots,s_m)$, $t$ is of the form $g(t_1,t_2,\ldots,t_n)$, and either
(a) $f=g$ and $s_i \trianglelefteq t_{j_i}$ for all $i$, $1 \le i \le m$, where $1 \le j_1 < j_2 < \cdots < j_m \le n$, or
(b) $s \trianglelefteq t_j$ for some $j$, $1 \le j \le n$.

Thus, this relation embodies a notion of "syntactic simplicity": $s \trianglelefteq t$ if $s$ may be obtained from $t$ by deletion of selected operators and operands. If $s$ is embedded in $t$, but $s \neq t$, then we write $s \triangleleft t$. For example,

$$--(0+1) \triangleleft (((--(----0+1)+1)+1)+1)+1.$$

**Definition 5.** A derivation $t_1 \Rightarrow t_2 \Rightarrow \cdots \Rightarrow t_j \Rightarrow \cdots \Rightarrow t_k \Rightarrow \cdots$ is *self-embedding* if $t_j \trianglelefteq t_k$ for some $j<k$. A rewrite system is *self-embedding* if it allows a self-embedding derivation.

**Theorem 2** [Dershowitz-82]. *If a rewrite system is nonterminating, then it is self-embedding.*

The proof of this is based on the Tree Theorem [Higman-52, Kruskal-60, Nash-Williams-63].[3]

This theorem means that, to show termination of a system, one can prove it to be non-self-embedding. The converse, however, does not hold: self-embedding does not imply nontermination.

*Example.* The rewrite system

$$f(f(\alpha)) \rightarrow f(g(f(\alpha))) \tag{6}$$

is both self-embedding and terminating.

Unfortunately, even this sufficient condition for termination is undecidable:

**Theorem 3** [Plaisted-85]. *It is undecidable whether a rewrite system is self-embedding.*

Of course, self-embedding is partially decidable: just search through all derivations until an embedding is discovered. It is similarly undecidable if a system cycles or loops. (For details,

[3]A weaker form of "embedding" and correspondingly weaker results appear as an exercise in [Knuth-73], where it was suggested that embedding has applications to proofs of termination.

see [Plaisted-85].)

## 3. TERMINATION

To express proofs of termination, we need the following concepts: A *partially-ordered* set $(S,\succ)$ consists of a set $S$ and a transitive and irreflexive binary relation $\succ$ defined on elements of $S$.[4] As usual, $s \succeq t$ means that either $s \succ t$ or $s=t$, $s \prec t$ means the same as $t \succ s$, and $s \preceq t$ means $t \succeq s$. A partially ordered set is said to be *totally ordered* if for any two distinct elements $s$ and $s'$ of $S$, either $s \succ s'$ or $s' \succ s$. For example, both the set of integers and the set of natural numbers are totally ordered by the "greater-than" relation $>$. The set of all subsets of the integers is partially ordered by the "proper subset" relation $\subsetneq$. An *extension* of a partial ordering $\succ$ on $S$ is a partial ordering $\succ'$ also on $S$ such that $s \succ s'$ implies $s \succ' s'$ for all $s, s' \in S$. Partial orderings of component elements can also be extended to a partial ordering of tuples of elements: a tuple $(s_1, s_2, \ldots, s_n)$ in $(S_1,\succ_1)\times(S_2,\succ_2)\times \cdots (S_n,\succ_n)$ is *lexicographically* greater than another tuple $(t_1,t_2, \ldots, t_n)$ if for some $i$ $(1 \le i \le n)$ $s_i \succ_i t_i$ while $s_j = t_j$ for all $j < i$.

A partially ordered set $(S,\succ)$ is said to be *well-founded* if there are no infinite descending sequences $s_1 \succ s_2 \succ s_3 \succ \cdots$ of elements of $S$. Thus, the natural numbers **N** under their "natural" ordering $>$ is well-founded, since no sequence of natural numbers can descend beyond 0. But $>$ is not a well-founded ordering of all the integers, since, for example, $-1>-2>-3> \cdots$ is an infinite descending sequence. Nor is $>$ a well-founded ordering of the reals. If $(S_1,\succ_1)$ and $(S_2,\succ_2)$ are two well-founded sets, then their lexicographically ordered cross-product $(S_1 \times S_2, \succ^*)$ is also well-founded, where a pair $(s_1,s_2)$ in $S_1 \times S_2$ is greater than another pair $(t_1,t_2)$ in $S_1 \times S_2$ if either $s_1 \succ_1 t_1$ or else $s_1 = t_1$ and $s_2 \succ_2 t_2$. Similarly, a lexicographic ordering of tuples of any fixed length is well-founded, if the orderings of the components are. For example, the tuple (2,5,1,6) is greater than (2,4,9,8) in the well-founded lexicographic ordering of tuples of naturally ordered natural numbers. (See, e.g., [Manna-74].)

The notion of well-foundedness suggests the following straightforward method of proving termination:

**Theorem 4.** *A rewrite system $R$ over a set of terms $\mathcal{T}$ is terminating if, and only if, there exists a well-founded ordering $\succ$ over $\mathcal{T}$ such that*

$$t \Rightarrow u \text{ implies } t \succ u$$

*for all terms $t$ and $u$ in $\mathcal{T}$.*

*Example.* System (0) terminates, since the lexicographic ordering of tuples of colors (with *blue* > *white* > *red*) is well-founded and the tuple of colors corresponding to a sequence of marbles is reduced with each rule application. By the nature of the lexicographic ordering, one

[4]Asymmetry of a partial ordering follows from transitivity and irreflexivity.

need only consider the change in the leftmost of the two affected components: if it was *white* before, then it is *red* after; if it was *blue* before, then it is either *red* or *white* after.

The following equivalent formulation (see[Kamin-Levy-80]) takes advantage of the structure of terms:

> **Corollary .** *A rewrite system R over a set of terms $\mathcal{T}$ is terminating if, and only if, there exists a well-founded ordering $\succ$ over $\mathcal{T}$ such that*
>
> $$l \succ r$$
>
> *for each rule $l \rightarrow r$ in R and for any substitution of terms in $\mathcal{T}$ for the variables of the rule, and such that*
>
> $$t \Rightarrow u \text{ and } t \succ u \text{ imply } f(\cdots t \cdots) \succ f(\cdots u \cdots)$$
>
> *for all terms in $\mathcal{T}$.*

*Example.* The system

$$f(f(\alpha)) \quad \rightarrow \quad f(g(f(\alpha))) \tag{6}$$

is terminating, since the number of adjacent $f$'s is reduced with each application. Note that counting the number of adjacencies makes $g(f(f(a))) > f(a)$, though $f(g(f(f(a)))) \not\succ f(f(a))$.

The following definition and theorem eliminate the need to consider all derivations $t \Rightarrow u$ and are often used to prove termination:

> **Definition 6.** A partial ordering $\succ$ over a set of terms $\mathcal{T}$ is *monotonic* (with respect to term structure) if it has the *replacement property,*
>
> $$t \succ u \text{ implies } f(\cdots t \cdots) \succ f(\cdots u \cdots),$$
>
> for all terms in $\mathcal{T}$.

In other words, reducing a subterm, reduces any superterm containing it.

> **Theorem 5** [Manna-Ness-70]. *A rewrite system R over a set of terms $\mathcal{T}$ is terminating if, and only if, there exists a monotonic well-founded ordering $\succ$ over $\mathcal{T}$ such that*
>
> $$l \succ r$$
>
> *for each rule $l \rightarrow r$ in R and for any substitution of terms in $\mathcal{T}$ for the variables of the rule.*

Note that the ordering $\succ$ is defined on $\mathcal{T}$, not on terms like $l$ and $r$ containing variables. That is why we require that $l \succ r$ for all *substitutions* that yields terms in $\mathcal{T}$. With monotonicity, this ensures that $t \succ u$ whenever $t$ reduces to $u$. As we will see, it is sometimes possible to "lift" an ordering on $\mathcal{T}$ to an orderings on terms with variables so that $l \succ r$ in the lifted ordering guarantees that in fact $l \succ r$ for all substitutions.

*Example.* The system

$$f(g(\alpha)) \rightarrow g(f(\alpha)) \tag{7}$$

terminates. To see this, consider the following well-founded monotonic ordering on monadic terms: given an ordering on operators, a term $s$ is greater than a term $t$ if $s$ has more operators than does $t$, or if they have the same number of operators, but the outermost operator of $s$ is greater than that of $t$, or if they are of the same length and their outermost operators are identical, but the operand of $s$ is (recursively) greater than that of $t$. Choosing an operator ordering $f > g$, the above rule is a reduction.

It is frequently convenient to separate a well-founded ordering on terms into two parts: a *termination function* $\tau$ that maps terms in $\mathcal{T}$ to a set $\mathcal{W}$ and a "standard" well-founded ordering $\succ$ on $\mathcal{W}$.

**Definition 7.** A *termination function* $\tau:\mathcal{T} \rightarrow \mathcal{W}$ is composed of a set of functions $f_\tau:\mathcal{W} \rightarrow \mathcal{W}$, one for each operator $f$, and is defined by

$$\tau(f(t_1, \ldots, t_n)) = f_\tau(\tau(t_1), \ldots, \tau(t_n))$$

for every term $f(t_1, \ldots, t_n)$ in $\mathcal{T}$, and for which

$$x \succ x' \text{ implies } f_\tau(\cdots x \cdots) \succ f_\tau(\cdots x' \cdots)$$

for all $x, x', \cdots$ in $\mathcal{W}$.

In other words a termination function is a monotonic morphism on terms.

**Theorem 6** [Manna-Ness-70]. *A rewrite system $R$ over a set of terms $\mathcal{T}$ is terminating if, and only if, there exists a well-founded set $(\mathcal{W},\succ)$ and termination function $\tau:\mathcal{T} \rightarrow \mathcal{W}$, such that*

$$\tau(l) \succ \tau(r)$$

*for each rule $l \rightarrow r$ in $R$ and for any substitution of terms in $\mathcal{T}$ for the variables of the rule.*

The use of monotonic *polynomial interpretations* was suggested in [Manna-Ness-70, Lankford-75, Lankford-79]. Using this method, an integer polynomial $F(x_1, \ldots, x_n)$ of degree $n$ is associated with each $n$-ary operator $f$. The choice of coefficients must ensure monotonicity and that terms are mapped into nonnegative integers only; this is the case if all coefficients are positive. (A number of examples may be found in [Dershowitz-Manna-79]; some work on automated polynomial proofs is in progress [BenCherifa-84].) The use of rewrite systems as termination functions and the formulation of abstract monotonicity conditions are explored in [Bachmair-Dershowitz-85, Gnaedig-85].

*Example.* Consider the following system (for symbolic differentiation with respect to $x$):

$$\begin{array}{rcl}
D_x\, x & \rightarrow & 1 \\
D_x\, a & \rightarrow & 0 \\
D_x\,(\alpha+\beta) & \rightarrow & D_x\,\alpha+D_x\,\beta \\
D_x\,(\alpha-\beta) & \rightarrow & D_x\,\alpha-D_x\,\beta \\
D_x\,(-\alpha) & \rightarrow & -D_x\,\alpha \\
D_x\,(\alpha\times\beta) & \rightarrow & \beta\times D_x\,\alpha+\alpha\times D_x\,\beta \\
D_x\,(\frac{\alpha}{\beta}) & \rightarrow & \frac{D_x\,\alpha}{\beta}-\alpha\frac{D_x\,\beta}{\beta^2} \\
D_x\,(\ln\ \alpha) & \rightarrow & \frac{D_x\,\alpha}{\alpha} \\
D_x\,(\alpha^\beta) & \rightarrow & \beta\times\alpha^{\beta-1}D_x\,\alpha+\alpha^\beta\times(\ln\ \alpha)\times D_x\,\beta
\end{array} \qquad (8)$$

where $a$ is any constant symbol other than $x$. Let the termination function $\tau:\mathcal{T}\rightarrow\mathbf{N}$ be defined as follows:

$$\begin{array}{rcl}
\tau(\alpha+\beta) & = & \tau(\alpha)\ +\ \tau(\beta) \\
\tau(\alpha\times\beta) & = & \tau(\alpha)\ +\ \tau(\beta) \\
\tau(\alpha-\beta) & = & \tau(\alpha)\ +\ \tau(\beta) \\
\tau(\frac{\alpha}{\beta}) & = & \tau(\alpha)\ +\ \tau(\beta) \\
\tau(\alpha^\beta) & = & \tau(\alpha)\ +\ \tau(\beta) \\
\tau(D_x\,\alpha) & = & \tau(\alpha)^2 \\
\tau(-\alpha) & = & \tau(\alpha)\ +\ 1 \\
\tau(\ln\ \alpha) & = & \tau(\alpha)\ +\ 1 \\
\tau(u) & = & 4
\end{array}$$

where $u$ is any constant (including $x$). For each of the nine rules $l\rightarrow r$, the value of $\tau$ decreases, i.e. $\tau(l)>\tau(r)$. For example,

$$\tau(D_x\,(\frac{\alpha}{\beta})) = \tau(\frac{\alpha}{\beta})^2 = (\tau(\alpha)+\tau(\beta))^2 = \tau(\alpha)^2+\tau(\beta)^2+2\tau(\alpha)\tau(\beta),$$

while

$$\tau(\frac{D_x\,\alpha}{\beta}-\alpha\frac{D_x\,\beta}{\beta^2}) = \tau(\alpha)^2+\tau(\beta)^2+\tau(\alpha)+2\tau(\beta)+4.$$

This is a decrease, since $\tau(\alpha)$ and $\tau(\beta)$ are at least 4 and therefore

$$2\tau(\alpha)\tau(\beta) \geq 4\tau(\alpha)+4\tau(\beta) > \tau(\alpha)+2\tau(\beta)+4.$$

Integer polynomials cannot, however, suffice for termination proofs in general, since that would place a polynomial bound on computations (see, e.g., [Huet-Oppen-80]).

*Example.* It seems that System (1) cannot be proved to terminate with any monotonic polynomial interpretation [Dershowitz-83]. But termination can be proved using exponentials [Filman-78], defining $\tau:\mathcal{T}\rightarrow\mathbf{N}$ as follows:

$$\begin{aligned}
\tau(\alpha+\beta) &= \tau(\alpha) + \tau(\beta)+1 \\
\tau(\alpha\times\beta) &= \tau(\alpha)\tau(\beta) \\
\tau(-\alpha) &= 3^{\tau(\alpha)} \\
\tau(u) &= 3,
\end{aligned}$$

where $u$ is any constant. Since the value of any term is at least 3, each rule is a reduction.

Proving termination of *rewriting modulo equations* is, in practice, considerably more difficult than for plain rewrite systems. Here, given an equational theory (congruence relation) $E$, a rule $l \rightarrow r$ in $R$ applies to a term $t \in \mathcal{T}$ if there is a substitution $\sigma$ such that $l\sigma = s$ for some subterm $s$ of a term $v$ such that $v =_E t$ in the theory $E$. If $l \rightarrow r$ applies in this sense, then we write $t \Rightarrow_{R/E} u$, where $u$ is any term equal (in $E$) to that $v$ with $s$ replaced by $r\sigma$. The question then is: for given $R$ and $E$, does there exist an infinite sequence of terms $t_i \in \mathcal{T}$ such that $t_1 \Rightarrow_{R/E} t_2 \Rightarrow_{R/E} \cdots$?

*Example.* Let $I$ denote the equational theory (idempotence):

$$\alpha+\alpha = \alpha.$$

For any nonempty $R$, $R/I$ cannot be terminating, since there must be an infinite derivation $l =_I l+l \Rightarrow_R l+r =_I (l+l)+r \Rightarrow_R \cdots$ for any $l \rightarrow r \in R$.

The equational theory $AC$, consisting of the associative and commutative axioms,

$$\begin{aligned}
f(\alpha,f(\beta,\gamma)) &= f(f(\alpha,\beta),\gamma) \\
f(\alpha,\beta) &= f(\beta,\alpha),
\end{aligned}$$

is particularly important in practice. Let $\bar{t}$ denote the flattened version of a term $t$, with all nested occurrences of associative-commutative operators stripped, and where the order of arguments of such operators is not significant, and let $\bar{\mathcal{T}} = \{\bar{t} : t \in \mathcal{T}\}$. Two terms $u$ and $v$ are equal in $AC$ if, and only if, $\bar{u}$ and $\bar{v}$ are the same. It is natural, therefore, to consider orderings on flattened terms.

**Theorem 7** [Dershowitz,*etal.*-83]. *Let R be a rewrite system over some set of terms $\mathcal{T}$ and F a set of associative-commutative operators. The rewrite relation R/AC is terminating if, and only if, there exists a well-founded ordering $\succ$ on $\overline{\mathcal{T}}$ such that*

$$\overline{l} \succ \overline{r}$$

*for each rule $l \rightarrow r$ in R and for any substitution of terms for the variables of the rule, and*

$$\overline{f(l,\xi)} \succ \overline{f(r,\xi)}$$

*for each rule $l \rightarrow r$ in R whose left-hand side l or right-hand side r has outermost associative-commutative symbol $f \in F$ or whose right-hand side is just a variable (where $\xi$ is a variable otherwise not occurring in the rule), and such that*

$$u \Rightarrow_{R/AC} v \text{ and } \overline{u} \succ \overline{v} \text{ imply } f(\cdots \overline{u} \cdots) \succ f(\cdots \overline{v} \cdots)$$

*for all terms u and v in $\mathcal{T}$ and $f(\cdots \overline{u} \cdots)$ and $f(\cdots \overline{v} \cdots)$ in $\overline{\mathcal{T}}$.*

Since addition and multiplication are themselves associative and commutative, monotonic polynomial interpretations are frequently helpful. To provide an ordering for flattened terms, a polynomial interpretation of a term should preserve its value under associativity and commutativity. The interpretations, $F(x,y)=xy$ and $F(x,y)=x+y+1$, for example, preserve value, whereas $F(x,y)=xy+1$, though symmetric, does not.

*Example.* Consider the following system (for Boolean rings):

$$\begin{array}{rcl} \alpha\cdot 1 & \rightarrow & \alpha \\ \alpha\cdot 0 & \rightarrow & 0 \\ \alpha\cdot\alpha & \rightarrow & \alpha \\ \alpha+0 & \rightarrow & \alpha \\ \alpha+\alpha & \rightarrow & 0 \\ (\alpha+\beta)\cdot\gamma & \rightarrow & (\alpha\cdot\gamma)+(\beta\cdot\gamma) \end{array} \tag{9}$$

One can use the following polynomial interpretation to prove its termination:

$$\begin{array}{rcl} \tau(\alpha+\beta) & = & \tau(\alpha)+\tau(\beta)+1 \\ \tau(\alpha\cdot\beta) & = & \tau(\alpha)\cdot\tau(\beta) \\ \tau(u) & = & 2, \end{array}$$

where $u$ is any constant.

## 4. SIMPLIFICATION ORDERINGS

In proving termination, one can use any ordering $\succ$ that is well-founded over all terms that could appear in any one derivation; the ordering need not be well-founded over all terms that appear in all derivations. We call an ordering for which $\succ \cap \Rightarrow^*$ is always well-founded, regardless of what rules are in $R$, *well-founded for derivations*. Thus, to apply Theorem 4, we need only that $\succ$ be a well-founded ordering for derivations. In particular, Theorem 2 implies the following:

**Theorem 8.** *A partial ordering $\succ$ is well-founded for derivations if it has an extension that contains the embedding relation $\triangleright$.*

To apply Theorem 5, we need $\succ$ to be monotonic, as well as well-founded for derivations. The following definition describes monotonic extensions of $\triangleright$:

**Definition 8** [Dershowitz-82]. A monotonic partial ordering $\succ$ is a *simplification ordering* for a set of terms $\mathcal{T}$ if it possesses the ***subterm property,***

$$f(\cdots t \cdots) \succ t,$$

and the *deletion property,*

$$f(\cdots t \cdots) \succ f(\cdots \quad \cdots),$$

for all terms in $\mathcal{T}$.

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.

**Theorem 9** [Dershowitz-79]. *Any simplification ordering is a monotonic well-founded ordering for derivations.*

In the previous section, we saw the use of polynomial interpretations for termination proofs. That method requires that terms be mapped onto the well-founded nonnegative integers; using simplification orderings, on the other hand, allows the methods to be extended to domains that are not themselves well-founded. For example, one can associate a monotonic polynomial $F(x_1, \ldots, x_n)$ over the *reals* with each $n$-ary operator $f$ [Dershowitz-79]. For any given choice of polynomials $F$ to provide a simplification ordering, we must have that

$$x_i > x_i' \textit{ implies } F(\cdots x_i \cdots) > F(\cdots x_i' \cdots)$$

and

$$F(\cdots x_i \cdots) > x_i$$

for all positions $i$ and for all real-valued $x$s.[5] For termination, we need

$$\tau(l) > tau(r),$$

for all rules $l \rightarrow r$ and for all real value assignments to the variables $\tau(\alpha)$ in $\tau(l)$. 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. $x_1$, ..., $x_n$ in $R'$ implies $F(x_1, \ldots, x_n)$ in $R'$. Then one need only show that the conditions hold for all $x$ in $R'$. The above conditions are all decidable (albeit in superexponential time), since they are logical combinations of multivariate polynomial inequalities over the reals [Tarski-51] (see [Cohen-69] for a much

[5]The methods of the next section allow the strict inequalities $>$ in these two conditions to be replaced by $\geq$.

briefer decision procedure and [Collins-75] for a more efficient one). Thus, the polynomial ordering can be effectively "lifted" to open (i.e. nonground) terms. It is similarly decidable if there exists polynomials (and a suitable definition of $R'$) of a given maximum degree that satisfy the conditions and thereby prove termination. (The decision procedure, however, cannot point to the appropriate polynomials). For polynomials over the natural numbers, these conditions are not decidable (see [Lankford-79]).

*Example.* Consider the set of expressions $\mathcal{T}$ constructed from some set of constants and the single operator $\times$ and the system (for semigroups)

$$(\alpha\times\beta)\times\gamma \quad \rightarrow \quad \alpha\times(\beta\times\gamma) \tag{10}$$

Terms $t$ and $u$ are compared by comparing their real value interpretations, $\tau(t)$ and $\tau(u)$. The real polynomials used are

$$\tau(\alpha\times\beta) = d\cdot\tau(\alpha) + \tau(\beta)$$

for some real $d>1$, for products, and

$$\tau(u) = e$$

for some $e>0$, for constants $u$. The value of the function $\tau$ decreases for the subexpression that the rule is applied to: for any terms $\alpha$, $\beta$, and $\gamma$,

$$\tau((\alpha\times\beta)\times\gamma) = d\cdot\tau(\alpha\times\beta)\times\tau(\gamma) = d^2\cdot\tau(\alpha) + d\cdot\tau(\beta) + \tau(\gamma),$$

while

$$\tau(\alpha\times(\beta\times\gamma)) = d\cdot\tau(\alpha)+\tau(\beta\times\gamma) = d\cdot\tau(\alpha)+d\cdot\tau(\beta)+\tau(\gamma).$$

This is a reduction, i.e.

$$\tau((\alpha\times\beta)\times\gamma) > \tau(\alpha\times(\beta\times\gamma)),$$

since $d^2>d$ and $\tau(\alpha)>0$.

Most orderings used in conjunction with Theorem 5 to prove termination of rewrite systems are simplification orderings. In fact:

**Theorem 10.** *Any total monotonic ordering $\succ$ is well-founded for derivations if, and only if, it is a simplification ordering.*

In general, however, total monotonic orderings, and hence simplification orderings, do not suffice for termination proofs.

*Example.* Consider the system

$$\begin{array}{lcl} f(a) & \rightarrow & f(b) \\ g(b) & \rightarrow & g(a). \end{array} \tag{11}$$

If an ordering $>$ is total, then either $a>b$ or $b>a$. If $a>b$, then we would also have $g(a)>g(b)$, and the second rule would not be a reduction; analogously, if $b>a$, the first rule would not be.

We have seen above (Theorem 1) that termination is undecidable for two-rule systems; for one-rule systems, the question of decidability is open. On the other hand,

**Theorem 11** [Jouannaud-Kirchner-82]. *It is decidable if a system of only one rule reduces under any simplification ordering.*

## 5. QUASI-ORDERINGS

This section describes methods for proving termination using 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 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 $\precsim$. A quasi-order $\succsim$ on $S$ is *total* if, for any two elements $s$ and $s'$ in $S$, either $s\succsim s'$ or else $s\precsim s'$. Note that the strict part $\succ$ is well-founded if, and only if, all infinite *quasi-descending* sequences $s_1\succsim s_2\succsim s_3\succsim\cdots$ of elements of $S$ contain a pair $s_j\precsim s_k$ for some $j<k$. In other words, if $\succ$ is well-founded, then from some point on, in any infinite quasi-descending sequence, all elements are equivalent.

A stronger notion than well-foundedness is accordingly the following:

**Definition 9** [Kruskal-60]. A set $S$ is *well-quasi-ordered* under a quasi-ordering $\precsim$ if every infinite sequence $s_1,s_2,\cdots$ of elements of $S$ contains a pair of elements $s_j$ and $s_k$, $j<k$, such that $s_j\precsim s_k$.

Thus, the strict part of any well-quasi-ordering is well-founded. Well-quasi-ordered sets are said to have the *finite basis property* in [Higman-52]; for a survey of the history and applications of well-quasi-orderings, see [Kruskal-72]. A generalization, limiting the contexts in which an embedding may occur, and possibly having applications to proofs of termination, can be found in [Ehrenfeucht,*etal.*-83, Bucher,*etal.*-84, Puel-85]. A even stronger notion than well-quasi-ordering, namely *better-quasi-ordering,* is exploited in [Laver-78].

Note that any finite set is well-quasi-ordered under any quasi-ordering (including equality). It follows from the definitions that if a set is well-quasi-ordered under $\precsim$, then it is well-founded under (any extension of) the partial ordering $\succ$; 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 $\precsim$.

**Theorem 12.** *A rewrite system $R$ over a set of terms $T$ is terminating if there exists a quasi-ordering $\succsim$, which extends a well-founded ordering $\succ$ and has the strict subterm property*

$$f(\cdots t \cdots) \succ t,$$

*such that*

$$l \succ r$$

*for each rule $l \to r$ in $R$ and for any substitution of terms in $T$ for the variables of the rule, such that*

$$s \Rightarrow t \text{ and } s \succsim t \text{ imply } f(\cdots s \cdots) \succsim f(\cdots t \cdots).$$

(Cf. [Kamin-Levy-80].)

The quasi-ordering used in the above theorem can be a combination of two quasi-orderings, one used to show that eventually all terms in a derivation are equivalent and the second to show that there can only be a finite number of equivalent terms in any such derivation.

**Definition 10.** A rewrite system $R$ is *quasi-terminating* for a set of terms $T$, if all (infinite) derivations contain only a finite number of different terms. Equivalently (for finite systems), any infinite derivation must cycle.

Quasi-terminating systems are also referred to as *globally finite*. To prove that a system is quasi-terminating, one can use quasi-orderings in the obvious way:

**Theorem 13.** *A rewrite system $R$ over a set of terms $T$ is quasi-terminating if there exists a quasi-ordering $\succsim$, which extends a well-founded ordering $\succ$ and whose equivalence relation $\approx$ admits only finite equivalence classes, such that*

$$t \Rightarrow u \text{ implies } t \succsim u$$

*for all terms $t$ and $u$ in $T$.*

The following theorem gives one method for establishing finiteness of equivalence classes:

**Theorem 14.** *If the strict part $\succ$ of a quasi-ordering $\succsim$ on a set of terms $T$ is an extension of the embedding relation $\rhd$, then $\approx$ admits only finite equivalence classes.*

*Example.* Consider the polynomial interpretation

$$\tau(if(\alpha,\beta,\delta)) = \tau(\alpha)\times(\tau(\beta)+\tau(\gamma))$$

with constants assigned the value 2. The partial ordering $t \succ u$ if, and only if, $\tau(t) > \tau(u)$ does contain the embedding relation. Since, for the system (for normalizing conditionals)

$$if(if(\alpha,\beta,\gamma),\delta,\epsilon) \to if(\alpha, if(\beta,\delta,\epsilon), if(\gamma,\delta,\epsilon)) \tag{12}$$

$\tau(l)=\tau(r)$, the system is quasi-terminating.

Another method is the following:

**Theorem 15.** *If the strict part $\succ$ of a quasi-ordering $\succsim$ on a set of terms $\mathcal{T}$ is well-founded and has the subterm property*

$$f(\cdots t \cdots) \succ t,$$

*the deletion property*

$$f(\cdots t \cdots) \succ f(\cdots \quad \cdots),$$

*and admits only a finite number of terms smaller than any given one, then $\approx$ admits only finite equivalence classes.*

Note that the ordering need not be monotonic. (Cf. [Lipton-Snyder-77].)

*Example.* Consider the following system (for distributivity):

$$\begin{array}{rcl} \alpha\times(\beta+\gamma) & \rightarrow & (\alpha\times\beta)+(\alpha\times\gamma) \\ (\beta+\gamma)\times\alpha & \rightarrow & (\beta\times\alpha)+(\gamma\times\alpha) \\ \alpha\times 1 & \rightarrow & \alpha \\ 1\times\alpha & \rightarrow & \alpha \end{array} \tag{13}$$

Under the natural interpretation (+ as addition and × as multiplication, but constants as 2) terms can be mapped into natural numbers (and hence the term ordering has order-type $\omega$) while satisfying the subterm property. Since $l \approx r$ under this interpretation the system quasi-terminates.

Of course:

**Theorem 16** [Guttag,*etal.*-83]. *Quasi-termination of rewrite systems is undecidable.*

On the other hand, nontermination of any quasi-terminating system is clearly semi-decidable. Also, termination of a quasi-terminating system for a *given* input term is decidable (construct all derivations initiated by that term until they terminate or cycle).

*Example.* System (10) quasi-terminates, as does any (finite) system that never increases the size of terms.

Another notion that has been investigated is *fair termination* (of quasi-terminating systems), in which all infinite derivations must include an application of each rule that is infinitely often applicable. See [Porat-Francez-85].

Using a definition of monotonicity, we can give a local condition for quasi-termination:

**Definition 11.** A quasi-ordering $\succsim$ over a set of terms $\mathcal{T}$ is *monotonic* if

$$t \succsim u \textit{ implies } f(\cdots t \cdots) \succsim f(\cdots u \cdots)$$

for all terms in $\mathcal{T}$.

**Theorem 17** [Dershowitz-82]. *A rewrite system $R$ over a set of terms $\mathcal{T}$ is quasi-terminating if there exists a monotonic quasi-ordering $\succsim$, which extends a simplification ordering $\succ$, such that*

$$l \succsim r$$

*for each rule $l \rightarrow r$ in $R$ and for any substitution of terms in $\mathcal{T}$ for the variables of the rule.*

*Example.* System (13) can be shown to be quasi-terminating using the "natural" interpretation which preserves the value of a term under rewriting, i.e. $\tau(l)=\tau(r)$ for both rules. By letting constants have a positive value, the quasi-ordering $\geq$ is an extension of the simplification ordering $>$.

Given quasi-termination, the following method may be used to prove full termination:

**Theorem 18.** *A quasi-terminating rewrite system $R$ over a set of terms $\mathcal{T}$ is terminating if there exists a monotonic quasi-ordering $\succsim$ such that*

$$l \succ r$$

*for each rule $l \rightarrow r$ in $R$ and for any substitution of terms in $\mathcal{T}$ for the variables of the rule.*

Thus, to prove termination one can first find a monotonic quasi-ordering $\succsim$ guaranteeing quasi-termination, and then find any monotonic quasi-ordering $\succsim'$ under which each rule is a reduction.[6]

*Example.* The proof of termination of System (12) may be completed using the monotonic quasi-ordering $t \succsim' u$ if, and only if, $|t| \leq |u|$, which "decreases" with application of the length-increasing rules.

*Example.* To complete a proof of termination for the quasi-terminating System (10), a monotonic quasi-ordering $\succsim$ can be used, under which $t_1 \times t_2 \succsim t_1' \times t_2'$ if, and only if, $|t_1 \times t_2| = |t_1' \times t_2'|$ and $|t_1| \geq |t_1'|$.

Extending the results of the previous section, we have

**Definition 12** [Dershowitz-82]. A monotonic quasi-ordering $\succsim$ is a *quasi-simplification ordering* for a set of terms $\mathcal{T}$ if it possesses the subterm property

$$f(\cdots t \cdots) \succsim t,$$

and deletion property,

$$f(\cdots t \cdots) \succsim f(\cdots \quad \cdots),$$

for all terms in $\mathcal{T}$.

That is, a quasi-simplification ordering is a monotonic extension of the embedding relation $\trianglerighteq$. A quasi-simplification ordering for fixed-arity operators is called a *divisibility order* in [Higman-52]. The strict part $\succ$ of any quasi-simplification ordering $\succsim$ is well-founded for

[6] [Lipton-Snyder-77, Guttag, *etal.*-83] use "increasing length" where any monotonic quasi-ordering would do.

derivations. For proving termination, it is enough that $\succsim$ be monotonic:

**Theorem 19** [Dershowitz-82]. *A rewrite system $R$ over a set of terms $\mathcal{T}$ is terminating if there exists a quasi-simplification ordering $\succsim$ such that*

$$l \succ r$$

*for each rule $l \rightarrow r$ in $R$ and for any substitution of terms in $\mathcal{T}$ for the variables of the rule.*

## 6. MULTISET ORDERINGS

*Multisets,* or *bags,* are like sets, but allow multiple occurrences of identical elements. A partial ordering $\succ$ on any given set $S$ can be extended to form an ordering $\gg$ on finite multisets over $S$. In this extended ordering, $M \gg M'$, for two finite multisets $M$ and $M'$ over $S$, if $M'$ can be obtained from $M$ by replacing one or more elements in $M$ by any (finite) number of elements taken from $S$, each of which is smaller than one of the replaced elements. More formally, let $\mathcal{M}(S)$ denote the set of finite multisets of elements of $S$. Then:

**Definition 13** [Dershowitz-Manna-79]. For a partially-ordered set $(S,\succ)$, the *multiset ordering* $\gg$ on $\mathcal{M}(S)$ is defined as follows:

$$M \gg M'$$

if, and only if, for some multisets $X,Y \in \mathcal{M}(S)$, where $X$ is a nonempty subset of $M$,

$$M' = (M - X) \cup Y$$

and for all $y \in Y$ there is an $x \in X$ such that

$$x \succ y.$$

**Definition 14.** For a quasi-ordered set $(S,\succsim)$, the *multiset quasi-ordering* $\ggcurly$ on $\mathcal{M}(S)$ is defined as follows:

$$M \ggcurly M'$$

if, and only if, for some multisets $X,Y \in \mathcal{M}(S)$,

$$M' \approx (M - X) \cup Y$$

and for all $y \in Y$ there is an $x \in X$ such that

$$x \succ y,$$

where two multisets are considered equivalent if the equivalence classes of their elements (under $\approx$) are the same.

For example, the multiset {3,3,3,4,0,0} of natural numbers is identical to the multiset {0,3,3,0,4,3}, but distinct from {3,4,0}. If **N** is the set of natural numbers 0, 1, 2, ... with the $>$ ordering, then under the corresponding *multiset ordering* $\gg$ over **N**, the multiset {3,3,4,0} is greater than each of the three multisets {3,4}, {3,2,2,1,1,1,4,0}, and {3,3,3,3,2,2}. In the first case, two elements have been removed (i.e. replaced by zero elements); in the second case, an occurrence of 3 has been replaced by two occurrences of 2 and three occurrences of 1; and in the third case, the element 4 has been replaced by two occurrences each of 3 and 2, and in addition the element 0 has been removed. (See also [Smullyan-79, Gardner-83].)

This ordering on multisets enjoys the following minimality property:

**Theorem 20** [Lescanne-Jouannaud-82]. *For a given partial ordering $\succ$ on a set $S$, any partial ordering $\gg'$ on $M(S)$ that satisfies the property*

$$s \succ s' \text{ implies } \{\cdots s \cdots\} \gg' \{\cdots s' \cdots\}$$

*is contained in the multiset ordering $\gg$.*

Multiset orderings are used in termination proofs on account of the following:

**Theorem 21** [Dershowitz-Manna-79]. *The multiset ordering $\gg$ is well-founded if, and only if, $\succ$ is.*

*Example.* To prove termination of System (8), we use a *simple path ordering* of [Plaisted-78]. Terms are mapped into multisets of sequences of operators; sequences are compared in the *monadic path ordering* $>_{mpo}$, as we did for System (7). The monotonic termination function used for the simple path ordering is

$$\tau(t) = \{(f_1, f_2, \ldots, f_k) \mid (f_1, f_2, \ldots, f_k) \text{ is a path in } t\},$$

where a *path* is a sequence of operators, starting at the outermost one of the whole term (the root, viewing terms as trees) and taking subterms until a constant (leaf) is reached. For the operator ordering, we take $D$ to be greater than all else. For example, consider the expression

$$t = D_x D_x (D_x y \times (y + D_x D_x x)),$$

or with the $D$'s numbered for expository purposes,

$$t = D_1 D_2 (D_3 y \times (y + D_4 D_5 x)).$$

There are three paths, and

$$\tau(t) = \{(D_1, D_2, \times, D_3, y), (D_1, D_2, \times, +, y), (D_1, D_2, \times, +, D_4, D_5, x)\}.$$

Applying the rule

$$D_x(\alpha \times \beta) \quad \rightarrow \quad \beta \times D_x \alpha \ + \ \alpha \times D_x \beta$$

to $t$ yields

$$u = D_1(((y + D_4 D_5 x) \times D_2 D_3 y) + (D_3 y \times D_2(y + D_4 D_5 x)))$$

(with the labeling of the $D_x$'s retained), and accordingly

$$\tau(u) = \{(D_1, +, \times, +, y), (D_1, +, \times, +, D_4, D_5, x), (D_1, +, \times, D_2, D_3, y),$$
$$(D_1, +, \times, D_3, y), (D_1, +, \times, D_2, +, y), (D_1, +, \times, D_2, +, D_4, D_5, x)\}.$$

We have $\tau(t) \gg_{mpo} \tau(u)$, since

$$\begin{array}{rcl}(D_1,D_2,\times,D_3,y) & >_{mpo} & (D_1,+,\times,+,y)\\ (D_1,+,\times,D_2,+,D_4,D_5,x) & >_{mpo} & (D_1,+,\times,+,D_4,D_5,x)\\ (D_1,D_2,\times,D_3,y) & >_{mpo} & (D_1,+,\times,D_2,D_3,y)\\ (D_1,D_2,\times,D_3,y) & >_{mpo} & (D_1,+,\times,D_3,y)\\ (D_1,D_2,\times,D_3,y) & >_{mpo} & (D_1,+,\times,D_2,+,y)\\ (D_1,+,\times,D_2,+,D_4,D_5,x) & >_{mpo} & (D_1,+,\times,D_2,+,D_4,D_5,x).\end{array}$$

In the monadic path ordering, sequences are compared left-to-right: At each step, any operator or constant less than or equal to the corresponding one in the other sequence is skipped over. Whichever sequence is finished first is smaller; if both finish together, whichever last had a smaller operator is larger.[7]

If $(S,\succ)$ is *totally* ordered, then for any two multisets $M,M' \in \mathcal{M}(S)$, one may determine whether $M \gg M'$ by first sorting the elements of both $M$ and $M'$ in descending order (with respect to the relation $\succ$) and then comparing the two sorted sequences lexicographically.[8] For example, to compare the multisets {3,3,4,0} and {3,2,1,2,0,4}, one may compare the sorted sequences (4,3,3,0) and (4,3,2,2,1,0). Since (4,3,3,0) is lexicographically greater than (4,3,2,2,1,0), it follows that {3,3,4,0}≫{3,2,1,2,0,4}. [Lescanne-Jouannaud-82] describes an implementation of multiset orderings for the nontotal case.

Consider the case where there is a bound $k$ on the number of replacement elements. Any termination proof using this bounded multiset ordering over **N** may be translated into a proof using natural numbers. This may be done using the termination function

$$\psi(M) = \sum_{n \in M} \frac{k^n - 1}{k - 1}$$

which maps multisets over the natural numbers into the natural numbers. When exactly $k$ elements $n-1$ replace one element $n$, the above function gives the exact number of replacements until termination.

In general, if $(S,\succ)$ is of order type $\alpha$, then the multiset ordering $(\mathcal{M}(S),\gg)$ over $(S,\succ)$ is of order type $\omega^\alpha$. This follows from the fact that there exists a mapping $\psi$ from $\mathcal{M}(S)$ onto $\omega^\alpha$ that is one-to-one and *order-preserving*, i.e. if $M \gg M'$ for $M,M' \in \mathcal{M}(S)$, then the ordinal $\psi(M)$ is greater than $\psi(M')$. That mapping is

$$\psi(M) = \sum_{m \in M} \omega^{\phi(m)}$$

where $\sum$ denotes the natural (i.e. commutative) sum of ordinals and $\phi$ is the one-to-one order-preserving mapping from $S$ onto $\alpha$.

[7] [Gorn-73] uses a "stepped" lexicographic ordering (under which longer sequences are larger) to prove termination of differentiation, but without using multisets, his proof applies only when $D$'s are not nested.

[8] This is the ordering $I_\omega^+$ in [Manna-69].

*Example.* The simple path ordering does not work for the system:

$$\begin{array}{lcl} --\alpha & \rightarrow & \alpha \\ -(\alpha+\beta) & \rightarrow & ---\alpha \times ---\beta \\ -(\alpha\times\beta) & \rightarrow & ---\alpha + ---\beta \end{array} \tag{14}$$

Instead, we use Theorem 19 and define the following quasi-simplification ordering: $t \gg u$ for two terms $t$ and $u$ if, and only if,

$$|t|_{+\times} \geq |u|_{+\times} \ and \ \{|\alpha|_{+\times}: -\alpha \ in \ t\} \geq \{|\alpha|_{+\times}: -\alpha \ in \ u\},$$

where the multisets contain the value $|\alpha|_{+\times}$ (the number of occurrences of operators other than $-$ in $\alpha$) for each subterm of the form $-\alpha$, and multisets are compared using $\geq$. 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. For all three rules, the number of operators other than $-$ is the same on both sides. To see that

$$--\alpha \succ \alpha,$$

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

$$-(\alpha+\beta) \succ ---\alpha\times---\beta$$

$$-(\alpha\times\beta) \succ ---\alpha+---\beta$$

note that the number of operators other than $-$ in $\alpha+\beta$ and $\alpha\times\beta$ is greater than that of $--\alpha$, $-\alpha$, $\alpha$, $--\beta$, $-\beta$, and $\beta$.

## 7. PRECEDENCE ORDERINGS

We use the multiset ordering in the following:

**Definition 15** [Dershowitz-82]. Let $\succ$ be a partial ordering on a set of operators $F$. The *recursive path ordering* $\succ_{rpo}$ on the set $\mathcal{T}(F)$ of terms over $F$ is defined recursively as follows:

$$s = f(s_1, \ldots, s_m) \succ_{rpo} g(t_1, \ldots, t_n) = t$$

if

$$s_i \succsim_{rpo} t \quad \text{for } some \ i=1,\ldots,m$$

or

$$f \succ g \ and \ s \succ_{rpo} t_j \quad \text{for } all \ j=1,\ldots,n$$

or

$$f = g \ and \{s_1, \ldots, s_m\} \gg_{rpo} \{t_1, \ldots, t_n\},$$

where $\gg_{rpo}$ is the extension of $\succ_{rpo}$ to multisets and $\succsim_{rpo}$ means $\succ_{rpo}$ or permutatively congruent (equivalent up to permutations of subterms).

This definition is similar to a characterization of the *path of subterms ordering* given in

[Plaisted-78b].[9] The idea is that a term is decreased by replacing a subterm with any number of smaller (recursively) subterms connected by any structure of operators smaller (in the operator ordering) than the outermost operator of the replaced subterm.

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 nor greater that that of $t$, then some subterm of $s$ must be greater or equal to $t$. For example, suppose $- > +$, and let $s = -(1 \times (1+0))$ and $t = -1 + -(0 \times 1)$. The term $s$ is greater than $t$ under the corresponding recursive path ordering $>_{rpo}$ by the following line of reasoning:

$$
\begin{aligned}
&s >_{rpo} t \text{ since } - > + \text{ and } s >_{rpo} -1, -(0\times 1) \\
&\quad s >_{rpo} -1 \text{ since } 1\times(1+0) >_{rpo} 1 \\
&\qquad 1\times(1+0) >_{rpo} 1 \text{ since } 1 \geq_{rpo} 1 \\
&\quad s >_{rpo} -(0\times 1) \text{ since } 1\times(1+0) >_{rpo} 0\times 1 \\
&\qquad 1\times(1+0) >_{rpo} 0\times 1 \text{ since } 1=1 \text{ and } 1+0 >_{rpc} 0 \\
&\qquad\quad 1+0 >_{rpo} 0 \text{ since } 0 \geq 0.
\end{aligned}
$$

**Theorem 22** [Dershowitz-82]. *The recursive path ordering is a simplification ordering.*

Using the recursive path ordering to prove the termination of rewrite systems generalizes the (exponential interpretation) method in [Iturriaga-67].[10]

*Example.* We can use a recursive path ordering to prove termination of System (1). Let the operators be ordered by $- > \times > +$. Since this is a simplification ordering on terms, by Theorem 9, we need only show that

$$
\begin{array}{rcl}
--\alpha & >_{rpo} & \alpha \\
-(\alpha+\beta) & >_{rpo} & -\alpha \times -\beta \\
-(\alpha\times\beta) & >_{rpo} & -\alpha + -\beta \\
\alpha\times(\beta+\gamma) & >_{rpo} & (\alpha\times\beta)+(\alpha\times\gamma) \\
(\beta+\gamma)\times\alpha & >_{rpo} & (\beta\times\alpha)+(\gamma\times\alpha)
\end{array}
$$

for any terms $\alpha$, $\beta$, and $\gamma$. The first inequality follows from the subterm condition of simplification orderings. By the definition of the recursive path ordering, to show that

[9] This ordering addresses the problem posed in [Levy-80].

[10] 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 operators.

$-(\alpha+\beta)>_{rpo}(-\alpha\times-\beta)$ when $->\times$, we must show that $-(\alpha+\beta)>_{rpo}-\alpha$, and $-(\alpha+\beta)>_{rpo}-\beta$. Now, since the outermost operators of $-(\alpha+\beta)$, $-\alpha$, and $-\beta$ are the same, one must show that $\alpha+\beta>_{rpo}\alpha$ and $\alpha+\beta>_{rpo}\beta$. 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, since $\times>+$, we must show $\alpha\times(\beta+\gamma)>_{rpo}\alpha\times\beta$ and $\alpha\times(\beta+\gamma)>_{rpo}\alpha\times\gamma$. By the definition of the recursive path ordering for the case when two terms have the same outermost operator, we must show that $\{\alpha,\beta+\gamma\}\gg_{rpo}\{\alpha,\beta\}$ and $\{\alpha,\beta+\gamma\}\gg_{rpo}\{\alpha,\gamma\}$. These two inequalities between multisets hold, since the elements $\beta+\gamma$ is greater than both $\beta$ and $\gamma$ with which it is replaced. Similarly the fifth inequality may be shown to hold. Therefore, by Theorem 9, this system terminates for all inputs.

The *multiset ordering* described above, *nested multiset ordering* [Dershowitz-Manna-79], and *simple path ordering* may all be thought of as special cases of the *recursive path ordering,* in which the multiset constructor $\{\cdots\}$ is greater than other operators. The nested multiset ordering is just a recursive path ordering on all terms constructed from one varyadic operator, and (with just that one operator) is of order type $\epsilon_0$.[11] Gentzen used such an ordering to show termination of his "normalization procedure" [Gentzen-38]. Two other interesting examples of $\epsilon_0$ termination arguments may be found in [Kirby-Paris-82].

The above definition of the *recursive path ordering* is not particularly well-suited for computation. The *recursive decomposition ordering* $\succ_{rdo}$ (defined in [Lescanne-84, Plaisted-79] for the case when the ordering $\succ$ on operators is total) "preprocesses" terms in an attempt to improve efficiency. Suppose $\succ$ is total, and let $\bar{t}$ denote the term $t=g(t_1,\ldots,t_n)$ with all subterms sorted according to $\succ_{rdo}$, i.e. $\bar{t}=g(\bar{t}_{j_1},\ldots,\bar{t}_{j_n})$, where $\bar{t}_{j_1}\succeq_{rdo}\cdots\succeq_{rdo}\bar{t}_{j_n}$ and $\{t_{j_1},\ldots,t_{j_n}\}$ is permutatively congruent to $\{t_1,\ldots,t_n\}$. Consider two sorted terms $\bar{s}=u[f(s_1,\ldots,s_m)]$ and $\bar{t}=v[g(t_1,\ldots,t_n)]$, where $f$ and $g$ are the greatest operators in $s$ and $t$, and $u$ and $v$ are the "contexts" surrounding the leftmost (maximal) occurrences of $f$ and $g$ in $s$ and $t$, respectively. Then,

$$s \succ_{rdo} t$$

if, and only if, the decomposition of $\bar{s}$,

$$(f,(s_1,\ldots,s_m),u[\circ]),$$

is greater than the decomposition of $\bar{t}$,

$$(g,(t_1,\ldots,t_n),v[\circ]),$$

where the three components are compared lexicographically, the operators $f$ and $g$ according to $\succ$, the subterms $s_i$ and $t_j$ lexicographically (using $\succ_{rdo}$ recursively), and the contexts $u$ and $v$ recursively. In comparing contexts, the operator $\circ$ is considered to be greater than any term not containing $\circ$; in choosing greatest $f$ and $g$, circles are ignored. For example, suppose $0>->\times>+>1$, $s=-(1\times(1+0))$, and $t=-1+-(0\times1)$. Their sorted terms are

[11] That the nested multiset ordering has the properties of simplification orderings was pointed out in

$\bar{s}=-((0+1)\times 1)$ and $\bar{t}=-(0\times 1)+-1$. The full decomposition of $\bar{s}$ is

$$\langle 0,(\ ),\langle -,(\langle \times,(\langle +,(\circ,1),\circ\rangle,1),\circ\rangle),\circ\rangle\rangle;$$

that of $\bar{t}$ is

$$\langle 0,(\ ),\langle -,(\langle \times,(\circ,1),\circ\rangle),\langle +,(\circ,\langle -,(\ ),\circ\rangle),\circ\rangle\rangle\rangle.$$

The first decomposition is greater, since $\langle +,(\circ,1),\circ\rangle$ is greater than just $\circ$.

With the above definition, the comparison of two sorted terms is essentially lexicographic. Sorting a list of sorted terms and building the decomposition are believed to be relatively inexpensive [Dershowitz-Zaks-81, Lescanne-Steyaert-83]. The definition of "decomposition" can be extended to the nontotal case [Jouannaud,*etal.*-82, Rusinowitch-85]. The *recursive decomposition ordering* as well as the *path of subterms ordering* [Plaisted-78b] and *path ordering* [Kapur-Sivakumar-83], extend the recursive path ordering somewhat when the ordering $\succ$ on operators is partial. The four are are equivalent in the total case. For example, the path of subterms ordering makes $h(f(\alpha),f(\beta))>_{pso}h(g(\alpha,\beta),g(\alpha,\beta))$ if $f>g$, but the two are incomparable under $>_{rpo}$. With a total ordering on operators, terms are also totally ordered. Thus, one can determine that $h(f(\alpha),f(\beta))>_{rpo}h(g(\alpha,\beta),g(\alpha,\beta))$ in all three possible cases: $\alpha>\beta$, $\beta>\alpha$, and $\alpha=\beta$. The exact relation between them is investigated in [Rusinowitch-85]. These orderings are also equivalent for monadic terms, even when the operator ordering is partial; an efficient implementation of the monadic case is given in [Lescanne-81].

These precedence orderings may be conveniently lifted to apply to nonground terms (containing variables) by considering variables as (zeroary) constant symbols, unrelated to any other symbol. For the *recursive path ordering* this idea is illustrated in [Dershowitz-82] and formalized in [Huet-Oppen-80]; for the *recursive decomposition ordering* this is done in [Jouannaud,*etal.*-82]; for the *path of subterms ordering,* see [Plaisted-78b]. For example, we have $-(\alpha+\beta)>_{rpo}-\alpha\times-\beta$, where $\alpha$ and $\beta$ are variables, since $-$ is greater than $\times$ (under $>$) and $-(\alpha+\beta)$ is greater than both $-\alpha$ and $-\beta$ (under $>_{rpo}$). For $-(\alpha+\beta)>_{rpo}-\alpha$, it must be that $\alpha+\beta>_{rpo}\alpha$, which is true since $+\not\gtrless\alpha$ and $\alpha\geq_{rpo}\alpha$. Given a partial ordering $\succ$ of operators $F$, the following lifted ordering can also be used:

$$\bigcap_{\succ^+\supseteq\succ}\succ^+_{rpo},$$

where orderings are viewed as relations and all possible total extensions of the given precedence are considered. (See, for example, [Forgaard-84].)

These orderings are also *incremental.* That is, one can start with an empty ordering on operators, and add to it only as necessary to satisfy given inequalities between terms. How this may be done with the *recursive decomposition ordering* is described in [Jouannaud,*etal.*-82] (see also [AitKaci-83]). When comparing two terms, the comparison may stop when two decompositions have incomparable symbols, say $f$ and $g$, as their first components. The idea is to add $f>g$ to the ordering at that point. (This method has been implemented in the

[Scherlis-80]. For a "constructive" discussion of this ordering, see [Paulson-84].

REVE system [Lescanne-83]. Details may be found in [Choque-83, Detlefs-Forgaard-85].) For instance, in order for $\alpha\times(\beta+\gamma)>_{rdo}(\alpha\times\beta)+(\alpha\times\gamma)$ to hold, one needs $\times>+$; if $\times>+$, then for $-(\alpha+\beta)>_{rdo}-\alpha\times-\beta$ to hold, it must be that $->\times$. But choosing an ordering on operators so that two terms are comparable under the *recursive path ordering* is NP-complete [Krishnamoorthy-Narendran-84] in the number of different operators.

It is sometimes necessary to transform terms before comparing them in the recursive path ordering. As long as the ordering on the operators of the transformed terms is well-founded, the recursive path ordering on transformed terms will also be:

**Theorem 23** [Dershowitz-82]. *The recursive path ordering $\succ_{rpo}$ on the set of terms $\mathcal{T}(F)$ is well-founded if, and only if, the partial ordering $\succ$ on the set of operators $F$ is well-founded.*

But the transform $\tau$, which acts as termination function, needs to satisfy the monotonicity condition

$$\tau(t)\succ_{rpo}\tau(u) \text{ implies } \tau(f(\cdots t\cdots))\succ_{rpo}\tau(f(\cdots u\cdots)).$$

Depending on the particular $\tau$, this condition may or may not hold. One way in which terms may be transformed is to let the $k$th operand of a term act as its operator. Then to compare two terms one must first *recursively* compare their $k$th operands and then use the recursive path ordering. With this transform, the result is a monotonic simplification ordering. (See [Dershowitz-82].)

*Example.* To prove that System (12) 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 $\alpha$ of the right-hand side. Thus, we need to show that the left-hand side is greater than both right-hand-side operands $if(\beta,\delta,\epsilon)$ and $if(\gamma,\delta,\epsilon)$. Again, $if(\alpha,\beta,\gamma)$ is greater than both operators $\beta$ and $\gamma$, and now the left-hand side is clearly greater than the remaining operands $\delta$ and $\epsilon$.

*Example.* The following system (for a combinator $C$) terminates:

$$(C\cdot((\alpha\cdot\beta)\cdot\gamma)\cdot\delta \quad\rightarrow\quad (\alpha\cdot\gamma)\cdot((\beta\cdot\gamma)\cdot\delta). \tag{15}$$

One way to see that is to consider the left operand of $\cdot$ to be the operator.[12]

This particular ordering, considering the first operand to be the operator and applied to terms constructed only from one varyadic operator $f$, is of order type $\Gamma_0$ (see [Veblen-08, Feferman-68].) This can be shown with the following order-preserving mapping [Dershowitz-80] $\psi$ from $\mathcal{T}(\{f\})$ onto $\Gamma_0$:

---

[12]This kind of proof is possible when the combinator has a *non-ascending property* described in [Pettorossi-78, Pettorossi-81].

$$\psi(f) = 0$$

$$\psi(f(f, \ldots, f)) = n \text{ where } n \text{ is the number of operands } f$$

$$\psi(f(\alpha, \beta_1, \beta_2, \ldots, \beta_n) = \phi^{\psi(\alpha)}(\sum_{i=1}^{n} \omega^{\psi(\beta_i)}) + \delta(t)$$

where $\phi^0(\beta)=\beta$, $\phi^1(\beta)=\epsilon_\beta$ (the $\beta$-th epsilon number), $\phi^\alpha(\beta)$ is the $\beta$th fixpoint $\xi$ of $\phi^\mu(\xi)=\xi$ common to $\phi^\mu$ for all ordinals $\mu<\alpha$, $\sum$ is the natural (commutative) sum of ordinals, and $\delta$ is 1 if $\psi(\alpha)=1$, $n=1$, and $\psi(\beta_1)$ is an epsilon number and is 0 otherwise. (The purpose of $\delta$ is to ensure that $\psi(f(f,\beta))>\psi(\beta)$ even if $\psi(\beta)$ is an epsilon number.) That this mapping is order-preserving follows from the fact ([Feferman-68, Weyhrauch-78]) that $\phi^\alpha(\beta)>\phi^{\alpha'}(\beta')$ if and only if $\alpha=\alpha'$ and $\beta>\beta'$, or else $\alpha>\alpha'$ and $\phi^\alpha(\beta)>\beta'$, or else $\alpha<\alpha'$ and $\beta>\phi^{\alpha'}(\beta')$.

More generally, terms may be mapped by replacing their operators with the whole term itself, where the new operator is the whole term itself ordered by some *other* well-founded ordering:

**Definition 16** [Kamin-Levy-80, Plaisted-79]. Let $\succsim$ be a quasi-ordering on a set of terms $\mathcal{T}$. The *semantic path ordering* $\succ_{spo}$ on $\mathcal{T}$ is defined recursively as follows:

$$s = f(s_1, \ldots, s_m) \succ_{spo} g(t_1, \ldots, t_n) = t$$

if

$$s_i \succsim_{spo} t \quad \text{for } some\ i=1,\ldots,m$$

or

$$s \succ t \text{ and } s \succ_{spo} t_j \quad \text{for } all\ j=1,\ldots,n$$

or

$$s \approx t \text{ and } \{s_1, \ldots, s_m\} \succ\!\!\succ_{spo} \{t_1, \ldots, t_n\},$$

where $\succ\!\!\succ_{spo}$ is the extension of $\succ_{spo}$ to multisets and $\succsim_{spo}$ means $\succ_{spo}$ or permutatively congruent (equivalent up to permutations of subterms).

To use this semantic path ordering in a termination proof, the monotonicity condition

$$t \Rightarrow u \text{ implies } f(\cdots t \cdots) \succ f(\cdots u \cdots)$$

must hold.

*Example.* Consider the system

$$\begin{array}{ccc} g(\alpha,\beta) & \rightarrow & h(\alpha,\beta) \\ h(f(\alpha),\beta) & \rightarrow & f(g(\alpha,\beta)). \end{array} \tag{16}$$

The first rule suggests $g>h$; the second requires $h>f$ and $h \gtrsim g$. This conflict can be overcome by letting $>$ be a lexicographic combination of a recursive path ordering with $g \equiv h > f$ and one with $g>h$. Comparing terms under the corresponding $>_{spo}$ shows a reduction for both rules.

The recursive path ordering has also been adapted to handle associative-commutative operators by flattening and transforming terms (distributing large operators over small ones) before comparing them [Plaisted-83, Bachmair-Plaisted-85]. Here, too, the difficulty is in ensuring monotonicity. Flattening alone would not be monotonic. For instance, if $f > g$ then $f(a,a) >_{rpo} g(a,a)$, but $\overline{f(f(a,a),a)} = f(a,a,a) <_{rpo} f(g(a,b),c) = \overline{f(g(a,b),c)}$.

Another well-founded ordering is the following lexicographic version of the recursive path ordering:

**Definition 17** [Kamin-Levy-80]. Let $\succ$ be a partial ordering on a set of operators $F$. The *lexicographic path ordering* $\succ_{lpo}$ on the set $\mathcal{T}(F)$ of terms over $F$ is defined recursively as follows:

$$s = f(s_1, \ldots, s_m) \succ_{lpo} g(t_1, \ldots, t_n) = t$$

if

$$s \succ_{lpo} t_j \quad \text{for } all\ j=1,\ldots,n$$

and either

$$s_i \succsim_{lpo} t \quad \text{for } some\ i=1,\ldots,m$$

or

$$f = g\ and\ (s_1, \ldots, s_m) \succ_{lpo}^{*} (t_1, \ldots, t_n),$$

where $\succ_{lpo}^{*}$ is the lexicographic extension of $\succ_{lpo}$.

By the same token, some operators may have their operands compared lexicographically, while others are compared using multisets. [13] Multiset and lexicographic versions of these path orderings have been implemented in REVE [Lescanne-84, Detlefs-Forgaard-85] and RRL [Kapur-Sivakumar-83]. In [Kamin-Levy-80] it is pointed out that any well-founded manner of comparing operands that depends only on recursive comparisons of subterms would work as well.

*Example.* The following system (for Ackermann's function) can easily be seen to terminate with a lexicographic path ordering with empty precedence:

$$a(s(\alpha),s(\beta)) \rightarrow a(\alpha,a(s(\alpha),\beta)) \tag{17}$$

Sometimes, it is possible to adapt one of the above path orderings to work where otherwise it would not.

*Example.* The lexicographic path ordering cannot directly handle

---

[13] The same lexicographic path ordering has been described in [Sakai-84], where it is erroneously claimed to be an extension of the recursive path ordering; in fact, the two orderings are incomparable. How one might transform terms so that $t \succ_{lpo} u$ if, and only if, $\tau(t) \succ rpo \tau(u)$ is examined in [Pettorossi-81].

$$\begin{array}{lcl} (\alpha\cdot\beta)\cdot\gamma & \rightarrow & \alpha\cdot(\beta\cdot\gamma) \\ (\alpha+\beta)\cdot\gamma & \rightarrow & (\alpha\cdot\gamma)+(\beta\cdot\gamma) \\ \gamma\cdot(\alpha+f(\beta)) & \rightarrow & g(\gamma,\beta)\cdot(\alpha+a). \end{array} \tag{18}$$

One needs to differentiate between $\cdot$ in general and $g(\cdots)\cdot\cdots$, making the operator larger in the former case.

Suppose we are given a quasi-ordering $\succsim_F$ on (fixed arity) operators and a quasi-simplification ordering $\succsim_T$ on terms, such that $f(\cdots t\cdots)\succ_T t$ only when $f$ is unary and $f\succsim_F g$ all operators $g$. Then we can define a quasi-simplification ordering $\succsim$ in the following manner:

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

if, and only if,

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

where the two tuples are compared lexicographically, first according to the terms $s\succsim_T t$, then according to the operators $f\succsim_F g$, and finally according to the subterms $s_i\succsim_T t_i$, (or, alternatively, $s_i\succsim t_i$ recursively). The condition on the operator ordering $\succsim_F$ ensures that $\succsim$ possesses the subterm property. To prove termination, one must find appropriate quasi-orderings $\succsim_F$ and $\succsim_T$ for which $l\succ r$ for all rules $l\rightarrow r$ in the given system.

Other examples of simplification orderings are the *recursive lexicographic ordering* in [Knuth-Bendix-70] and the *polynomial ordering* in [Lankford-79]. The method of [Knuth-Bendix-70] assigns a positive integer weight to each zeroary operator and a nonnegative integer weight to each other operator, with $\succsim_T$ comparing terms according to the sum of the weights of their respective operators, $\succsim_F$ a total ordering of operators, and subterms compared recursively. Thus, the condition on $\succsim_F$ requires that a unary operator have zero weight only if it is the largest operator under $\succsim_F$. [Lankford-79] replaces the linear sum of weight 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.

*Example.* For System (10) we can use the Knuth-Bendix ordering, taking $t\succsim_T u$ to be $|t|\geq|u|$ and $\succsim_F$ to be equality, and comparing subterms recursively.

*Example.* This method applies also to System (14) with $t\succsim_T u$ if, and only if, $|t|_{+\times}\geq|u|_{+\times}$, the largest operator under $>_F$ is $-$, and subterms compared recursively.

## 8. COMBINED SYSTEMS

In this section we consider the termination of combinations of rewrite systems. If $R$ and $S$ are two (strongly or weakly) terminating systems, we wish to know under what conditions the system $R\cup S$, containing all the rules of both $R$ and $S$, also (strongly or weakly) terminates.

**Definition 18.** A rewrite relation $R$ *commutes over* another relation $S$, if whenever $t \Rightarrow_S u \Rightarrow_R v$, there is an alternative derivation of the form $t \Rightarrow_R w \Rightarrow^*_{R \cup S} v$.

With it we can reduce termination of the union of $R$ and $S$ to termination of each:

**Theorem 24** [Bachmair-Dershowitz-85]. *Let $R$ and $S$ be two rewrite systems over some set of terms $\mathcal{T}$. Suppose that $R$ commutes over $S$. Then, the combined system $R \cup S$ is terminating if, and only if, $R$ and $S$ both are.*

For rewriting modulo equations, we have the following analogous results:

**Theorem 25** [Jouannaud-Munoz-84]. *If the rewrite relation $R$ commutes over the congruence relation $E$, then $R/E$ is terminating if, and only if, $R$ is terminating.*

Furthermore,

**Theorem 26.** *Let $E$ be a congruence relation and $R$ and $S$ two $E$-terminating rewrite systems (over some set of terms $\mathcal{T}$). If whenever $t \Rightarrow_S u \Rightarrow_{R/E} v$, there is an alternative derivation of the form $t \Rightarrow_{R/E} w \Rightarrow^*_{(R \cup S)/E} v$, then the combined system $(R \cup S)/E$ is also terminating.*

Some suggestions of how noncommuting $R$ and $E$ might be handled are given in [Jouannaud-Munoz-84].

To show that two relations commute, we can make use of the following properties:

**Definition 19.** A system is *left-linear* if no variable occurs more than once on the left-hand side of a rule; it is *right-linear* if no variable has more than one occurrence on the right-hand side. We say that a system is *linear* if it is both left- and right-linear.

**Definition 20.** A term $u$ is said to *overlap* (or *superposes*) a term $t$ if $u$ can be unified with some (not necessarily proper) subterm $s$ of $t$, i.e. if the two can be made the same by substituting terms for the variables in $t$ and $u$ (Whenever we speak in this section of unifying two terms, we consider their variables to be disjoint and insist that neither of the terms be just a variable.) We say that there is no overlap between two terms $t$ and $u$ if neither $t$ overlaps $u$ nor $u$ overlaps $t$. A rewrite system $R$ is said to be *non-overlapping* if there is no overlap among the left-hand sides of $R$, i.e. no left-hand side $l_i$ overlaps a different left-hand side $l_j$ and no left-hand side $l_i$ overlaps a proper subterm of itself.

*Example.* The linear system

$$(\alpha \times \beta) \times \gamma \quad \rightarrow \quad \alpha \times (\beta \times \gamma) \tag{10}$$

is overlapping since $(\alpha \times \beta) \times \gamma$ is unifiable with $\alpha \times \beta$. The system

$$\alpha \times (\beta + \gamma) \quad \rightarrow \quad (\alpha \times \beta) + (\alpha \times \gamma) \tag{19}$$

is left-linear but not right-linear; the system

$$(\alpha \times \beta) + (\alpha \times \gamma) \quad \rightarrow \quad \alpha \times (\beta + \gamma) \tag{20}$$

is right-linear but not left-linear. Both are non-overlapping.

Other investigations of some of commuting systems include [Rosen-73, O'Donnell-77, Huet-Levy-79, Huet-80, Raoult-Vuillemin-80]

Using these properties to establish commutation, we have the following results:

**Corollary** [Dershowitz-81]. *Let R and S be two rewrite systems (over some set of terms $\mathcal{T}$). Suppose that R is left-linear, S is right-linear, and there is no overlap between left-hand sides of R and right-hand sides of S. Then, the combined system $R \cup S$ is terminating if, and only if, R and S both are.*

This generalizes the case exploited in [Bidoit-].

*Example.* The systems

$$\begin{array}{rcl} \alpha\times(\beta+\gamma) & \rightarrow & (\alpha\times\beta)+(\alpha\times\gamma) \\ (\beta+\gamma)\times\alpha & \rightarrow & (\beta\times\alpha)+(\gamma\times\alpha) \\ \alpha\times 1 & \rightarrow & \alpha \\ 1\times\alpha & \rightarrow & \alpha \end{array} \tag{13}$$

and

$$\begin{array}{rcl} \alpha\times\alpha & \rightarrow & \alpha \\ \alpha+\alpha & \rightarrow & \alpha \end{array} \tag{21}$$

each terminate; therefore their union also does.

Each of the three requirements of the above theorem is necessary, as evidenced by the following examples of nonterminating systems.

*Example.* The system

$$\begin{array}{rcl} f(\alpha,\alpha) & \rightarrow & f(a,b) \\ b & \rightarrow & a \end{array} \tag{22}$$

has the infinite derivation $f(a,a) \Rightarrow f(a,b) \Rightarrow f(a,a) \Rightarrow \cdots$, though each rule terminates, the first is right-linear, the second is linear, and there is no overlap (but the first is not left-linear).

*Example.* The system

$$\begin{array}{rcl} b & \rightarrow & a \\ f(a,b,\alpha) & \rightarrow & f(\alpha,\alpha,\alpha) \end{array} \tag{23}$$

has the infinite derivation $f(a,b,b) \Rightarrow f(b,b,b) \Rightarrow f(a,b,b) \Rightarrow \cdots$, though each rule terminates, the first is linear, the second is left-linear, and there is no overlap (but the second is not right-linear).

*Example.* The system

$$\begin{array}{lcl} b & \rightarrow & g(a) \\ a & \rightarrow & g(b) \end{array} \tag{24}$$

has the infinite derivation $b \Rightarrow g(a) \Rightarrow g(g(b)) \Rightarrow \cdots$, though each rule terminates and both are linear (but there is overlap).

Similarly for rewriting modulo equations, we have:

**Theorem 27.** *If $R/E$ is left-linear, $S$ is right-linear, and there is no overlap between left-hand sides of $R/E$ and right-hand sides of $S$, then the combined system $(R \cup S)/E$ is also terminating.*

*Example.* Let $E$ be $AC$, $S$ be

$$\begin{array}{rcl} \alpha\cdot 1 & \rightarrow & \alpha \\ \alpha\cdot 0 & \rightarrow & 0 \\ \alpha\cdot\alpha & \rightarrow & \alpha \\ \alpha+0 & \rightarrow & \alpha \\ \alpha+\alpha & \rightarrow & 0 \end{array} \tag{9a}$$

and $R$ be

$$(\alpha+\beta)\cdot\gamma \quad \rightarrow \quad (\alpha\cdot\gamma)+(\beta\cdot\gamma) \tag{9b}$$

The system $S$ is right-linear; the relation $R/AC$ is left-linear, since $R$ is left-linear and $AC$ is linear. There are no occurrences of $0$ on the left-hand sides of $R$, so there is no overlap. Therefore, $R/AC$ commutes over $S$. If, say,

$$(d\cdot(a\cdot a))\cdot(b+c) \Rightarrow_S (d\cdot a)\cdot(b+c) \Rightarrow_{R/AC} d\cdot(a\cdot b+a\cdot c)$$

then by the same token

$$(d\cdot(a\cdot a))\cdot(b+c) \Rightarrow_{R/AC} d\cdot((a\cdot a)\cdot b+(a\cdot a)\cdot c) \Rightarrow_S \Rightarrow_S d\cdot(a\cdot b+a\cdot c).$$

Recall that a system is *weakly* terminating if every term rewrites to an irreducible term.

*Example.* The following system [cf. System (1)] does not always terminate but is weakly-terminating and its irreducible terms are in disjunctive normal form:

$$\begin{array}{lcl} --\alpha & \rightarrow & \alpha \\ -(\alpha+\beta) & \rightarrow & ---\alpha\times---\beta \\ -(\alpha\times\beta) & \rightarrow & ---\alpha+---\beta \\ \alpha\times(\beta+\gamma) & \rightarrow & (\alpha\times\beta)+(\alpha\times\gamma) \\ (\beta+\gamma)\times\alpha & \rightarrow & (\beta\times\alpha)+(\gamma\times\alpha) \end{array} \tag{25}$$

To see that it does not terminate, consider the derivation

$$\begin{aligned} &--(0\times(0+1)) \Rightarrow --((0\times0)+(0\times1)) \Rightarrow -(---(0\times0)\times---(0\times1)) \\ &\quad\Rightarrow \cdots \Rightarrow -(-(0\times0)\times-(0\times1)) \\ &\quad\Rightarrow \cdots \Rightarrow -((---0+---0)\times(---0+---1)) \\ &\quad\Rightarrow \cdots \Rightarrow -((-0+-0)\times(-0+-1)) \Rightarrow -((-0\times(-0+-1))+(-0\times(-0+-1))) \\ &\quad\Rightarrow ---(-0\times(-0+-1))\times---(-0\times(-0+-1)) \Rightarrow \cdots . \end{aligned}$$

Thus, beginning with a term of the form $--(\alpha\times(\alpha+\beta))$, a term containing a subterm of the same form is derived, and the process may continue *ad infinitum*. On the other hand, any application of the second or third rule can be followed immediately by two applications of the first rule, thus simulating a derivation of System (1) and guaranteeing termination.

To prove that a system is weakly terminating, one can choose a particular evaluation strategy and show that the value of a term is reduced in some well-founded ordering for those rewrites allowed by the chosen strategy. Thus, for the union of two weakly terminating systems $R$ and $S$, one can choose to first reduce to an $R$-normal form and only then apply $S$. Then, if one can show that applying $S$ to an $R$-normal form results in an $R$-normal form, weak termination of $R \cup S$ follows.

*Example.* The nonterminating System (25) is weakly terminating by the following line of reasoning: The first three rules alone are weakly terminating, since applying one of those rules to an *outermost* occurrence of $-$ reduces the multiset of sizes of arguments of $-$. (Note that this is not, and need not be, a monotonic ordering.) Similarly, the last two rules can be shown weakly terminating. Since the first three rules eliminate all negations of nonconstants and the two distributivity rules cannot introduce other negations, weak termination is proved.

## 9. RESTRICTED SYSTEMS

In this section, we consider how linearity and nonoverlapping of rules make it possible to restrict the derivations that must be considered when proving termination or nontermination of a rewrite system. Unfortunately:

**Theorem 28** [Huet-Lankford-78]. *Termination of a rewrite system is undecidable, even if the system is linear and nonoverlapping and has only monadic operators and constants.*

In the extreme case of a single monadic rule with right-hand side no longer than left-hand side, deciding termination is trivial. [Metivier-83, Calladine-85] provide upper bounds on the length of a derivation in that case. Similarly:

**Theorem 29** [Guttag,*etal.*-83]. *Quasi-termination of a rewrite system is undecidable, even if the system is linear and nonoverlapping and has only monadic operators and constants.*

We need the following definitions:

**Definition 21** [Lankford-Musser-78]. The set of *forward closures* for a given rewrite system $R$ may be inductively defined as follows: Every rule in $R$ is a forward closure. Let

$$c_1 \Rightarrow c_2 \Rightarrow \cdots \Rightarrow c_m$$

and

$$d_1 \Rightarrow d_2 \Rightarrow \cdots \Rightarrow d_n$$

be two forward closures already included. If $c_m$ has a (nonvariable) subterm $s$ within some context $u$ such that $s$ unifies with $d_1$ via most general unifier $\sigma$, then

$$c_1\sigma \Rightarrow c_2\sigma \Rightarrow \cdots \Rightarrow c_m\sigma = u\sigma[d_1\sigma] \Rightarrow u\sigma[d_2\sigma] \Rightarrow \cdots \Rightarrow u\sigma[d_n\sigma]$$

is also a forward closure. (Two forward closures are considered equal if they can be obtained one from the other by variable renaming.)

This definition is related to the *narrowing process,* as defined in [Slagle-74, Hullot-80].

**Definition 22** [Guttag,*etal.*-83]. The set of *overlap closures* for a given rewrite system $R$ may be inductively defined as follows: Every rule in $R$ is a forward closure. Let

$$c_1 \Rightarrow c_2 \Rightarrow \cdots \Rightarrow c_m$$

and

$$d_1 \Rightarrow d_2 \Rightarrow \cdots \Rightarrow d_n$$

be two overlap closures already included. If $c_m$ has a (nonvariable) subterm $s$ within some context $u$ such that $s$ unifies with $d_1$ via most general unifier $\sigma$, then

$$c_1\sigma \Rightarrow c_2\sigma \Rightarrow \cdots \Rightarrow c_m\sigma = u\sigma[d_1\sigma] \Rightarrow u\sigma[d_2\sigma] \Rightarrow \cdots \Rightarrow u\sigma[d_n\sigma]$$

is also an overlap closure. If $d_1$ has a (nonvariable) subterm $t$ within some context $v$ such that $t$ unifies with $c_m$ via most general unifier $\tau$, then

$$v\tau[c_1\tau] \Rightarrow v\tau[c_2\tau] \Rightarrow \cdots \Rightarrow v\tau[c_m\tau] = d_1\tau \Rightarrow d_2\tau \Rightarrow \cdots \Rightarrow d_n\tau$$

is also an overlap closure. (Two overlap closures are considered equal if they can be obtained one from the other by variable renaming.)

*Example.* Consider the system

$$\begin{array}{lcl} --\alpha & \rightarrow & \alpha \\ -(\alpha+\beta) & \rightarrow & -\alpha+-\beta \end{array} \qquad (26)$$

The derivation

$$-((\alpha+-\beta)+-\gamma) \Rightarrow -(\alpha+-\beta)+--\gamma \Rightarrow$$
$$(-\alpha+--\beta)+--\gamma \Rightarrow (-\alpha+\beta)+--\gamma \Rightarrow (-\alpha+\beta)+\gamma$$

is a forward closure for that system; the derivation

$$-(\alpha+--\beta) \Rightarrow -\alpha+---\beta \Rightarrow -\alpha+-\beta$$

is an overlap closure, but not a forward one; the derivation

$$-(\alpha+--\beta) \Rightarrow -(\alpha+\beta) \Rightarrow -\alpha+-\beta$$

is neither.

**Theorem 30** [Dershowitz-81]. *A right-linear rewrite system is terminating if, and only if, it has no infinite forward closures.*

*Example.* The self-embedding rewrite system

$$f(h(\alpha)) \rightarrow f(g(h(\alpha))) \tag{27}$$

is right-linear and has only one forward closure:

$$f(h(\alpha)) \Rightarrow f(g(h(\alpha)))$$

Since this forward closure is finite, the system must terminate. Note that, by Theorem 10, no total monotonic ordering could prove termination of this system.

*Example.* The forward closures of

$$f(g(\alpha)) \rightarrow g(g(f(\alpha))) \tag{28}$$

are all of the form

$$f(g(g^i(\alpha))) \Rightarrow g(g(f(g^i(\alpha)))) \Rightarrow \cdots \Rightarrow g^{2i}(f(\alpha))$$

where $i \geq 0$. Since the system is right-linear and all its forward closures are finite, by the theorem, it must terminate for all inputs.

*Example.* The forward closures of

$$f(g(\alpha)) \rightarrow g(g(f(f(\alpha)))) \tag{29}$$

include

$$f(g(\alpha)) \Rightarrow g(g(f(f(\alpha))))$$

and the infinite forward closures

$$f(g(g(g^i(\alpha)))) \Rightarrow g(g(f(f(g(g^i(\alpha)))))) \Rightarrow g(g(f(g(g(f(f(g^i(\alpha)))))))) \Rightarrow \cdots$$

for all $i \geq 0$. Thus, the system does not terminate.

**Theorem 31** [Dershowitz-81]. *A non-overlapping left-linear rewrite system is terminating if, and only if, it has no infinite forward closures.*

It has been conjectured [Dershowitz-81] that left-linearity is unnecessary.

*Example.* None of the forward closures of the non-overlapping left-linear System (8) have nested $D$ operators. (This can be shown by induction.) Thus, the finiteness of those forward closures—and consequently the termination of the system—can be easily proved by considering the multiset of the sizes of the arguments of the $D$'s. Any rule application reduces that value under the multiset ordering.

In general, though, a term-rewriting system need not terminate even if all its chains do:

*Example.* The non-right-linear and overlapping system

$$\begin{array}{rcl} f(a,b,\alpha) & \rightarrow & f(\alpha,\alpha,b) \\ b & \rightarrow & a \end{array} \qquad (30)$$

has two finite forward closures. Nevertheless, the system does not terminate. To wit,

$$f(a,b,b) \Rightarrow f(b,b,b) \Rightarrow f(a,b,b).$$

**Theorem 32** [Guttag,*etal.*-83]. *A quasi-terminating left-linear rewrite system is terminating if, and only if, it has no infinite overlap closures.*

*Example.* System (30) has the following infinite overlap closure:

$$f(b,b,b) \Rightarrow f(a,b,b) \Rightarrow f(b,b,b) \Rightarrow \cdots$$

It is unknown whether quasi-termination and/or left-linearity are necessary in the above theorem.

The above theorems give necessary and sufficient conditions for a left-linear or right-linear system to terminate. One of the advantages in using closures is that nontermination is more easily detectable, as the next theorem will demonstrate. First, we must extend the definition of "looping."

**Definition 23.** A derivation $t_1 \Rightarrow t_2 \Rightarrow \cdots \Rightarrow t_i \Rightarrow \cdots$ *loops* if for some $j > i$ $t_j$ has a subterm that is an *instance* of $t_i$.

**Theorem 33** [Dershowitz-81]. *A right-linear or non-overlapping left-linear rewrite system is nonterminating if, and only if, it has infinitely many nonlooping infinite forward closures or it has a looping forward closure.*

*Example.* The system

$$\begin{array}{rcl} g(\alpha) & \rightarrow & h(\alpha) \\ f(\alpha,\alpha) & \rightarrow & f(a,\alpha) \\ b & \rightarrow & a \\ a & \rightarrow & b \end{array} \qquad (31)$$

has two finite forward closures $b \Rightarrow a$ and $b \Rightarrow c$, one infinite looping forward closure $f(\alpha,\alpha) \Rightarrow f(a,b) \Rightarrow f(a,a) \Rightarrow \cdots$, and an infinite number of finite forward closures

$f(\alpha,\alpha) \Rightarrow f(a,b) \Rightarrow f(a,a) \Rightarrow \cdots \Rightarrow f(a,b) \Rightarrow f(a,c)$ with the same initial term.

*Example.* Consider again the right-linear System (26). Since forward closures cannot begin with a term having $-$ other than as outermost or innermost operator, the termination of all closures can be easily proved using a multiset ordering on the sizes of the arguments to $-$.

**Corollary .** *The termination of a right-linear or non-overlapping left-linear rewrite system is decidable if the number of forward closures issuing from different initial terms is finite.*

*Example.* The non-overlapping left-linear system

$$\begin{array}{rcl} f(a,\alpha) & \rightarrow & f(\alpha,g(\alpha)) \\ g(a) & \rightarrow & a \end{array} \tag{32}$$

has three forward closures:

$$\begin{array}{l} g(a) \Rightarrow a \\ f(a,\alpha) \Rightarrow f(\alpha,g(\alpha)) \\ f(a,a) \Rightarrow f(a,g(a)) \Rightarrow f(a,a) \Rightarrow \cdots \end{array}$$

Since its third forward closure cycles, it does not terminate. On the other hand, the system

$$\begin{array}{rcl} f(a,\alpha) & \rightarrow & f(\alpha,g(\alpha)) \\ g(a) & \rightarrow & b \end{array} \tag{33}$$

has the forward closures:

$$\begin{array}{l} g(a) \Rightarrow b \\ f(a,\alpha) \Rightarrow f(\alpha,g(\alpha)) \\ f(a,a) \Rightarrow f(a,g(a)) \Rightarrow f(a,b) \Rightarrow f(b,g(b)) \end{array}$$

Since none of its three forward closures loops, it does terminate.

*Example.* The forward closures of

$$\begin{array}{rcl} f(\alpha,\alpha) & \rightarrow & f(a,b) \\ b & \rightarrow & c \end{array} \tag{34}$$

are

$$b \Rightarrow c$$

and

$$f(\alpha,\alpha) \Rightarrow f(a,b) \Rightarrow f(a,c)$$

Since the forward closures do not loop, the system terminates.

In particular,

**Corollary** [Huet-Lankford-78]. *The termination of a rewrite system containing no variables (a ground system) is decidable.*

Quasi-termination of ground systems is similarly decidable [Dauchet-Tison-84].

## REFERENCES

[AitKaci-83] Ait-Kaci, H. "An algorithm for finding a minimal recursive path ordering", Report MS-CIS-83-7, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, 1983.

[Bachmair-Dershowitz-85] Bachmair, L., and Dershowitz, N. "Commutation, transformation, and termination". (1985) (submitted).

[Bachmair-Plaisted-85] Bachmair, L., and Plaisted, D. A. "Associative path ordering". *Proceedings of the First International Conference on Rewriting Techniques and Applications*, Dijon, France (May 1985).

[BenCherifa-84] Ben Cherifa, A. "Preuve de la terminaison finie d'un système de reécriture par l'ordre polynômial", Unpublished manuscript, Centre de Recherche en Informatique de Nancy, Nancy, France, 1984.

[Bergstra-Tucker-80] Bergstra, J. A., and Tucker, J. V. "Equational specifications for computable data types: Six hidden functions suffice and other sufficiency bounds", Preprint IW 128/80, Mathematisch Centrum, Amsterdam, The Netherlands, January 1980.

[Bidoit-] Bidoit, M. "Thesis", Thèse.

[Bucher,*etal.*-84] Bucher, W., Ehrenfeucht, A., and Haussler, D. "On total regulators generated by derivation relations", University of Denver, 1984.

[Calladine-85] Calladine, P. "Personal communication", Laboratoire d'Informatique, Université de Poitiers, May 1985.

[Choque-83] Choque, G. "Calcul d'un ensemble complet d'incrementations minimales pour l'ordre recursif de decomposition", Technical report, Centre de Recherche en Informatique de Nancy, Nancy, France, 1983.

[Cohen-69] Cohen, P. J. "Decision procedures for real and $p$-adic fields". *Communications of Pure and Applied Mathematics*, Vol. 22, No. 2 (1969), pp. 131-151.

[Collins-75] Collins, G. "Quantifier elimination for real closed fields by cylindrical algebraic decomposition". *Proceedings Second GI Conference on Automata Theory and Formal Languages* (1975), pp. 134-183.

[Dauchet-Tison-84] Dauchet, M., and Tison, S. "Decidability of confluence for ground term rewriting systems", Unpublished report, Université de Lille I, Lille, France, 1984.

[Dershowitz,*etal.*-83] Dershowitz, N., Hsiang, J., Josephson, N. A., and Plaisted, D. A. "Associative-commutative rewriting". *Proceedings of the Eighth International*

*Joint Conference on Artificial Intelligence*, Karlsruhe, West Germany (August 1983), pp. 940-944.

[Dershowitz-79] Dershowitz, N. "A note on simplification orderings". *Information Processing Letters*, Vol. 9, No. 5 (November 1979), pp. 212-215.

[Dershowitz-80] Dershowitz, N. "On representing ordinals up to $\Gamma_0$", Unpublished note, Department Computer Science, University of Illinois, Urbana, IL, June 1980.

[Dershowitz-81] Dershowitz, N. "Termination of linear rewriting systems". *Proceedings of the Eighth EATCS International Colloquium on Automata, Languages and Programming*, Vol. 115, Acre, Israel (July 1981), pp. 448-458.

[Dershowitz-82] Dershowitz, N. "Orderings for term-rewriting systems". *J. Theoretical Computer Science*, Vol. 17, No. 3 (March 1982), pp. 279-301 (previous version appeared in Proceedings of the Symposium on Foundations of Computer Science, San Juan, PR, pp. 123-131 [October 1979]).

[Dershowitz-83] Dershowitz, N. "Well-founded orderings", Technical Report ATR-83(8478)-3, Information Sciences Research Office, The Aerospace Corporation, El Segundo, CA, May 1983.

[Dershowitz-Manna-79] Dershowitz, N., and Manna, Z. "Proving termination with multiset orderings". *Communications of the ACM*, Vol. 22, No. 8 (August 1979), pp. 465-476 (also in Proceedings of the International Colloquium on Automata, Languages and Programming, Graz, 188-202 [July 1979]).

[Dershowitz-Zaks-81] Dershowitz, N., and Zaks, S. "Applied tree enumerations". *Proceedings of the Sixth Colloquium on Trees in Algebra and Programming*, Vol. 112, Genoa, Italy (March 1981), pp. 180-193.

[Detlefs-Forgaard-85] Detlefs, D., and Forgaard, R. "A procedure for automatically proving the termination of a set of rewrite rules". *Proceedings of the First International Conference on Rewriting Techniques and Applications*, Dijon, France (May 1985).

[Ehrenfeucht,*etal.*-83] Ehrenfeucht, A., Haussler, D., and Rozenberg, G. "On regularity of context-free languages". *Theoretical Computer Science*, Vol. 27 (1983), pp. 311-32.

[Feferman-68] Feferman, S. "Systems of predicative analysis II: Representation of ordinals". *J. Symbolic Logic*, Vol. 33 (1968), pp. 193-220.

[Filman-78] Filman, R. E. "personal communication", 1978.

[Forgaard-84] Forgaard, R. "A program for generating and analyzing term rewriting systems", Master's thesis, Laboratory for Computer Science, Massachusetts Institute of

Technology, Cambridge, MA, September 1984.

[Gardner-83] Gardner, M. "Mathematical games: Tasks you cannot help finishing no matter how hard you try to block finishing them". *Scientific American*, Vol. 24, No. 2 (August 1983), pp. 12-21.

[Gentzen-38] Gentzen, G. "New version of the consistency proof for elementary number theory". In: *Collected Papers of Gerhard Gentzen*, M. E. Szabo, ed (1938). North-Holland, 1969, pp. 252-286 (1938).

[Gnaedig-85] Gnaedig, I. "personal communication", 1985.

[Gorn-67] Gorn, S. "Handling the growth by definition of mechanical languages". *Proceedings of the Spring Joint Computer Conference* (Spring 1967), pp. 213-224.

[Gorn-73] Gorn, S. "On the conclusive validation of symbol manipulation processes (How do you know it has to work?)". *J. of the Franklin Institute*, Vol. 296, No. 6 (December 1973), pp. 499-518.

[Guttag,*etal.*-83] Guttag, J. V., Kapur, D., and Musser, D. R. "On proving uniform termination and restricted termination of rewriting systems". *SIAM Computing*, Vol. 12, No. 1 (February 1983), pp. 189-214.

[Higman-52] Higman, G. "Ordering by divisibility in abstract algebras". *Proceedings of the London Mathematical Society (3)*, Vol. 2, No. 7 (September 1952), pp. 326-336.

[Huet-80] Huet, G. "Confluent reductions: Abstract properties and applications to term rewriting systems". *J. of the Association for Computing Machinery*, Vol. 27, No. 4 (1980), pp. 797-821 (previous version in Proceedings of the Symposium on Foundations of Computer Science, Providence, RI, pp. 30-45 [1977]).

[Huet-Lankford-78] Huet, G., and Lankford, D. S. "On the uniform halting problem for term rewriting systems", Rapport Laboria 283, IRIA, March 1978.

[Huet-Levy-79] Huet, G., and Levy, J. J. "Call by need computations in non-ambiguous linear term rewriting systems", Rapport laboria 359, INRIA, Le Chesnay, France, August 1979.

[Huet-Oppen-80] Huet, G., and Oppen, D. C. "Equations and rewrite rules: A survey". In: *Formal Language Theory: Perspectives and Open Problems*, R. Book, ed. Academic Press, New York, 1980, pp. 349-405.

[Hullot-80] Hullot, J. M. "Canonical forms and unification". *Proceedings of the Fifth Conference on Automated Deduction*, Les Arcs, France (July 1980), pp. 318-334.

[Iturriaga-67] Iturriaga, R. "Contributions to mechanical mathematics", Ph.D. Thesis, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 1967.

[Jouannaud,*etal.*-82] Jouannaud, J. P., Lescanne, P., and Reinig, F. "Recursive decomposition ordering". *Proceedings of the Second IFIP Workshop on Formal Description of Programming Concepts*, Garmisch-Partenkirchen, West Germany (June 1982), pp. 331-348.

[Jouannaud-Kirchner-82] Jouannaud, J. P., and Kirchner, H. "Construction d'un plus petit order de simplification", Unpublished note, Centre de Recherche en Informatique de Nancy, Nancy, France, 1982.

[Jouannaud-Munoz-84] Jouannaud, J. P., and Muñoz, M. "Termination of a set of rules modulo a set of equations". *Proceedings of the Seventh International Conference on Automated Deduction*, Napa, CA (May 1984), pp. 175-193.

[Kamin-Levy-80] Kamin, S., and Levy, J. J. "Two generalizations of the recursive path ordering", Unpublished note, Department of Computer Science, University of Illinois, Urbana, IL, February 1980.

[Kapur,*etal.*-85] Kapur, D., Narendran, P., and Sivakumar, G. "A path ordering for proving termination of term rewriting systems". *Proceedings of the Tenth Colloquium on Trees in Algebra and Programming* (1985).

[Kapur-Sivakumar-83] Kapur, D., and Sivakumar, G. "Experiments with and architecture of RRL, a rewrite rule laboratory". *Proceedings of an NSF Workshop on the Rewrite Rule Laboratory*, Schenectady, NY (September 1983), pp. 33-56 (available as Report 84GEN008, General Electric Research and Development [April 1984]).

[Kirby-Paris-82] Kirby, L., and Paris, J. "Accessible independence results for Peano arithmetic". *Bulletin London Mathematical Society*, Vol. 14 (1982), pp. 285-293.

[Knuth-73] Knuth, D. E. *Fundamental algorithms*. Addison-Wesley, Reading, MA, 1973 (second edition).

[Knuth-Bendix-70] Knuth, D. E., and Bendix, P. B. "Simple word problems in universal algebras". In: *Computational Problems in Abstract Algebra*, J. Leech, ed. Pergamon Press, 1970, pp. 263-297.

[Krishnamoorthy-Narendran-84] Krishnamoorthy, M. S., and Narendran, P. "A note on recursive path ordering", Unpublished note, General Electric Corporate Research and Development, Schenectady, NY, 1984.

[Kruskal-60] Kruskal, J. B. "Well-quasi-ordering, the Tree Theorem, and Vazsonyi's conjecture". *Transactions of the American Mathematical Society*, Vol. 95 (May 1960), pp. 210-225.

[Kruskal-72] Kruskal, J. B. "The theory of well-quasi-ordering: A frequently discovered concept". *J. Combinatorial Theory Ser. A*, Vol. 13 (1972), pp. 297-305.

[Lankford-75] Lankford, D. S. "Canonical algebraic simplification in computational logic", Memo ATP-25, Automatic Theorem Proving Project, University of Texas, Austin, TX, May 1975.

[Lankford-79] Lankford, D. S. "On proving term rewriting systems are Noetherian", Memo MTP-3, Mathematics Department, Louisiana Tech. University, Ruston, LA, May 1979.

[Lankford-Musser-78] Lankford, D. S., and Musser, D. R. "A finite termination criterion", Unpublished draft, 1981.

[Laver-78] Laver, R. "Better-quasi-orderings and a class of trees". *Studies in Foundations and Combinatorics* (1978), pp. 31-48.

[Lescanne-81] Lescanne, P. "Two implementations of the recursive path ordering on monadic terms". *Proceedings of the Nineteenth Allerton Conference on Communication, Control, and Computing*, Monticello, IL (September 1981), pp. 634-643.

[Lescanne-83] Lescanne, P. "Computer experiments with the REVE term rewriting system generator". *Proceedings of the Tenth Symposium on Principles of Programming Languages*, Austin, TX (January 1983), pp. 99-108.

[Lescanne-84] Lescanne, P. "Some properties of decomposition ordering, A simplification ordering to prove termination of rewriting systems". *RAIRO Theoretical Informatics*, Vol. 16, No. 4, pp. 331-347.

[Lescanne-84] Lescanne, P. "Uniform termination of term-rewriting systems with status". *Proceedings of the Ninth Colloquium on Trees in Algebra and Programming*, Bordeaux, France (March 1984).

[Lescanne-Jouannaud-82] Lescanne, P., and Jouannaud, J. P. "On multiset orderings". *Information Processing Lett.*, Vol. 15, No. 2 (September 1982), pp. 57-62.

[Lescanne-Steyaert-83] Lescanne, P., and Steyaert, J. M. "On the study of data structures: Binary tournaments with repeated keys". *Proceedings of the Tenth EATCS International Colloquium on Automata, Languages and Programming*, Barcelona, Spain (July 1983), pp. 466-475.

[Levy-80] Levy, J. J. "Problem 80-5". *J. of Algorithms*, Vol. 1, No. 1 (March 1980), pp. 108-109.

[Lipton-Snyder-77] Lipton, R., and Snyder, L. "On the halting of tree replacement systems". *Proceedings of the Conference on Theoretical Computer Science*, Waterloo, Canada (August 1977), pp. 43-46.

[Manna-69] Manna, Z. "Termination of algorithms", Ph.D. thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, April 1969.

[Manna-74] Manna, Z. *Mathematical Theory of Computation*. McGraw-Hill, New York, 1974.

[Manna-Ness-70] Manna, Z., and Ness, S. "On the termination of Markov algorithms". *Proceedings of the Third Hawaii International Conference on System Science*, Honolulu, HI (January 1970), pp. 789-792.

[Metivier-83] Metivier, Y. "About the rewriting systems produced by the Knuth-Bendix completion algorithm". *Information Processing Letters*, Vol. 16 (January 1983), pp. 31-34.

[Nash-Williams-63] Nash-Williams, C. S. J. A. "On well-quasi-ordering finite trees". *Proceedings of the Cambridge Philosophical Society*, 59 (1963), pp. 833-835.

[O'Donnell-77] O'Donnell, M. J. "Computing in systems described by equations". *Lecture Notes in Computer Science*, Vol. 58, Berlin (1977).

[Paulson-84] Paulson, L. C. "Constructing recursion operations in intuitionistic type theory", Technical Report 57, Computer Laboratory, University of Cambridge, Cambridge, UK, October 1984.

[Pettorossi-78] Pettorossi, A. "A property which guarantees termination in weak combinatory logic and subtree replacement systems", Report R.78-23, Instituto di Automatica, Università di Roma, Rome, Italy, November 1978.

[Pettorossi-81] Pettorossi, A. "Comparing and putting together recursive path orderings, simplification orderings and non-ascending property for termination proofs of term rewriting systems". *Proceedings of the Eighth International Colloquium on Automata, Languages and Programming*, Acre, Israel (July 1981), pp. 432-447.

[Plaisted-78] Plaisted, D. A. "Well-founded orderings for proving termination of systems of rewrite rules", Report R78-932, Department of Computer Science, University of Illinois, Urbana, IL, July 1978.

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

[Plaisted-79] Plaisted, D. A. "personal communication", 1979.

[Plaisted-83] Plaisted, D. A. "An associative path ordering". *Proceedings of an NSF Workshop on the Rewrite Rule Laboratory*, Schenectady, NY (September 1983), pp. 123-136 (available as Report 84GEN008, General Electric Research and Development [April 1984]).

[Plaisted-85] Plaisted, D. A. "The undecidability of self embedding for term rewriting systems". *Information Processing Lett.*, Vol. 20, No. 2 (February 1985), pp. 61-64.

[Porat-Francez-85] Porat, S., and Francez, N. "Fairness in term rewriting systems".

*Proceedings of the First International Conference on Rewriting Techniques and Applications*, Dijon, France (May 1985).

[Puel-85] Puel, L. "personal communication", May 1985.

[Raoult-Vuillemin-80] Raoult, J. C., and Vuillemin, J. "Operational and semantic equivalence between recursive programs". *J. of the Association of Computing Machinery*, Vol. 27, No. 4 (October 1980), pp. 772-796.

[Rosen-73] Rosen, B. "Tree-manipulating systems and Church-Rosser theorems". *J. of the Association for Computing Machinery*, Vol. 20 (1973), pp. 160-187.

[Rusinowitch-85] Rusinowitch, M. "Plaisted ordering and recursive decomposition ordering revisited". *Proceedings of the First International Conference on Rewriting Techniques and Applications*, Dijon, France (May 1985).

[Sakai-84] Sakai, K. "An ordering method for term rewriting systems". *Proceedings of the First International Conference on Fifth Generation Computer Systems*, Tokyo, Japan (November 1984).

[Scherlis-80] Scherlis, W. L. "Expression procedures and program derivation", Ph.D. Dissertation, Department Computer Science, Stanford University, Stanford, CA, August 1980.

[Slagle-74] Slagle, J. R. "Automated theorem-proving for theories with simplifiers, commutativity, and associativity". *J. of the Association for Computing Machinery*, Vol. 21, No. 4 (1974), pp. 622-642.

[Smullyan-79] Smullyan, R. M. "Trees and ball games". *Annals of the New York Academy of Science*, Vol. 321 (1979), pp. 86-90.

[Tarski-51] Tarski, A. *A Decision Method for Elementary Algebra and Geometry*. University of California Press, Berkeley, CA, 1951.

[Veblen-08] Veblen, O. "Continuous increasing functions of finite and transfinite ordinals". *Transactions of the American Mathematical Society*, Vol. 9 (1908), pp. 280-292.

[Weyhrauch-78] Weyhrauch, R. W. "Some notes on ordinals up to $\Gamma_0$", Informal note 13, Stanford University, Stanford, CA, March 1978.