Reprinted from *J. Symbolic Computation* (1987) **3**, 69–116

Termination of Rewriting

NACHUM DERSHOWITZ

J. Symbolic Computation (1987) 3, 69–116

Termination of Rewriting†

NACHUM DERSHOWITZ

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

This survey describes methods for proving that systems of rewrite rules are terminating programs. We illustrate the use in termination proofs of various kinds of orderings on terms, including polynomial interpretations and path orderings. The effect of restrictions, such as linearity, on the form of rules is also considered. In general, however, termination is an undecidable property of rewrite systems.

1. Introduction

A *rewrite* (*term-rewriting*) *system* $\mathcal{R}$ over a set of terms $\mathcal{T}$ is a (finite) set of *rewrite rules*, each of the form $l \rightarrow r$, where l and r are terms containing *variables* ranging over $\mathcal{T}$, and such that r only contains variables also in l. A rule $l \rightarrow r$ applies to a term t in $\mathcal{T}$ if a subterm s of t matches the left-hand side l with some substitution σ 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 σ of terms for variables has been made. We write $t \Rightarrow_{\mathcal{R}} u$, or just $t \Rightarrow u$, to indicate that the term t in $\mathcal{T}$ *rewrites* in this way to the term u in $\mathcal{T}$ by a single application of some rule in $\mathcal{R}$. A *derivation* is a sequence of rewrites; if $t \Rightarrow \cdots \Rightarrow u$ in zero or more steps, abbreviated $t \Rightarrow^{*} u$, then we say that u is *derivable* from t; if no rule can be applied to t, we say that t is *irreducible*; when an irreducible term u is derivable from t we say that u is a *normal form* of t. See Huet & Oppen (1980), Dershowitz (1982*b*), and Dershowitz & Jouannaud (1987) for surveys of term-rewriting and its applications.

There are five basic properties of rewrite systems that are of interest:

(1) *termination*—no infinite derivations are possible;
(2) *determinism*—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 desiderata.

Depending on the purpose, various combinations of these properties are needed. 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 "normalisation". A quasi-terminating rewrite

† The preparation of this paper was supported in part by the National Science Foundation under Grant DCR 85–13417. This is a revised version of an invited paper, presented at the First International Conference on Rewriting Techniques and Applications (Dijon, France, May 1985).

0747–7171/87/010069+48 $03.00/0

system is one for which only a *finite* number of different terms are derivable from any given term. A normalising system is one for which every term has *at least* one normal form.

Consider, for example, a simple system, consisting of three rules:

$$\begin{aligned} \textit{white}, \textit{red} &\rightarrow \textit{red}, \textit{white} \\ \textit{blue}, \textit{red} &\rightarrow \textit{red}, \textit{blue} \\ \textit{blue}, \textit{white} &\rightarrow \textit{white}, \textit{blue}. \end{aligned} \tag{0}$$

This program plays the "Dutch National Flag" game: given a sequence of marbles, coloured *red*, *white*, or *blue* and placed side by side in no particular order, they are rearranged 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 may 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" colour is exchanged with the "smaller" one to its right.

The following system (for disjunctive normal form) illustrates some of the difficulties that may be encountered when attempting to determine if, and why, a rewrite system terminates:

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

The first rule eliminates double negations; the second and third rules apply DeMorgan's laws to push negations into sums and products; the last two apply the distributivity of *times* over *plus*. It is not obvious that this system terminates, since some derivations decrease the length of a term, e.g.

$$-(0\times(-1+-1))\Rightarrow\cdots\Rightarrow-0+(1\times1),$$

while others, e.g.

$$-(0\times(1+1))\Rightarrow\cdots\Rightarrow(((-0\times-0)+(-0\times-1))+((-1\times-0)+(-1\times-1)))$$

increase it. Furthermore, applying a rule at a subterm not only affects the structure of that subterm (perhaps duplicating parts of it), but that of its superterms as well. Any proof of termination must take into consideration the many different possible rewrite sequences generated by the *non-deterministic* choice of rules and subterms. For a lively discussion of simple tasks that are difficult to show terminating, see Gardner (1983).

Various methods for proving termination of rewrite systems have been suggested, including Gorn (1967), Iturriaga (1967), Knuth & Bendix (1970), Manna & Ness (1970), Gorn (1973), Lankford (1975*a*, *b*, 1977), Lipton & Snyder (1977), Plaisted (1978*a*, *b*), Dershowitz & Manna (1979), Lankford (1979), Kamin & Lévy (1980), Pettorossi (1981), Dershowitz (1982*a*), Jouannaud *et al.* (1982), Dershowitz *et al.* (1983), Lescanne (1984), Jouannaud & Muñoz (1984), Kapur *et al.* (1985), Bachmair & Plaisted (1985), Bachmair

& Dershowitz (1986), and Rusinowitch (1987). Termination is, in general, an undecidable property of rewrite systems (see, for example, Huet & Lankford (1978)), as it is known to be for non-deterministic Markov systems on strings (see, for example, Tourlakis (1984)).

In the next section we present a proof of the undecidability of termination. In section 3 we show how *well-founded orderings*, in general, and *polynomial interpretations*, in particular, are used in termination proofs. In section 4 *simplification orderings* are defined and their use illustrated; similar methods are described in section 5 for using *quasi-orderings* to prove termination or quasi-termination. Section 6 discusses *multiset orderings*. Then, in section 7, we present various *path orderings* based on an underlying operator "precedence". This is followed in the last two sections with methods for determining when rewriting "modulo" a congruence is terminating, when a rewrite system is normalising, and when systems of restricted form are terminating. Examples are provided throughout; proofs are generally omitted.

2. Non-termination

Given a (countable) set of function symbols $\mathcal{F}$, we consider the set $\mathcal{T}(\mathcal{F})$ of all terms constructed from symbols in $\mathcal{F}$. Function symbols in $\mathcal{F}$ may be *varyadic*, i.e. have variable arity, in which case, whenever f is a function symbol and $t_1, \ldots, t_n$ $(n \geq 0)$ are terms in $\mathcal{T}(\mathcal{F})$, the term $f(t_1, \ldots, t_n)$ is also in $\mathcal{T}(\mathcal{F})$. Or a function symbol f may be restricted to a fixed arity, in which case $f(t_1, \ldots, t_n) \in \mathcal{T}(\mathcal{F})$ only if f is of arity n. Function symbols with arity zero are referred to as *constants*. We use $\mathcal{T}$ for $\mathcal{T}(\mathcal{F})$ when $\mathcal{F}$ is arbitrary. A *rewrite rule* is an ordered pair $l \rightarrow r$ of *free* (*first-order*) terms, i.e. the terms l and r are constructed from function symbols in $\mathcal{F}$ and rule variables from some (countable) set $\mathcal{V}$. A *rewrite system* is a (finite or infinite) set of rewrite rules. (Terms in $\mathcal{T}$ might also contain "term" variables, but for the purposes of this paper, these are usually treated as constants.)

DEFINITION 1. A rewrite system $\mathcal{R}$ is *terminating* for a set of terms $\mathcal{T}$ if there exists no infinite (endless) sequence of terms $t_i \in \mathcal{T}$ such that $t_1 \Rightarrow t_2 \Rightarrow t_3 \Rightarrow \cdots$. A system is *non-terminating* if there exists any such infinite derivation.

This is the same—for finite $\mathcal{R}$—as saying that there are only a finite number of derivations issuing from any given initial term t_1. 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 arity) function symbols $\mathcal{F}$. Rules of a terminating system are called *reductions*.

EXAMPLE. A trivial example of a terminating system is

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

□

EXAMPLE. An equally trivial example of a non-terminating system is

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

□

EXAMPLE. A less trivial example (of what?) is

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

□

THEOREM 1. *It is undecidable whether a rewrite system is terminating, even if it has only two rules.*

PROOF. Turing machines can be simulated by rewrite systems: given any Turing machine $\mathcal{M}$, there exists a two-rule system $R_{\mathcal{M}}$ such that $R_{\mathcal{M}}$ terminates for all initial terms if, and only if, $\mathcal{M}$ 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 function symbols: a binary operator (which we will denote by juxtaposition and which associates to the right), a unary operator ∂ (an *erase* operator), and a ternary operator C.† The binary operator is used to construct a finite non-empty tape segment from individual constants representing tape symbols, with an additional constant $\square$ denoting the end of the segment. Corresponding to a machine in state q with non-blank left portion of the tape $a_1 a_2 \cdots a_m$ (from the left end until the symbol preceding the read head) and right portion $b_1 b_2 \cdots b_n$ (from the symbol being scanned to the end), is the term

$$C(a_m \cdots a_2 a_1 \square,\ q b_1 b_2 \cdots b_n \square,\ machine),$$

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

$$sqa\ s'a'q's''a''$$

signifying "if the machine is in state q reading the symbol a and the symbol immediately to the left of a is s, then replace the tape segment sa with $s'a's''a''$, position the head on s'', and go into state q'". One or two of the tape symbols in $s'a's''a''$ will be extra, and will be represented by the term $\partial(\#)$; as we will see, these terms can be eliminated from the tape by one of the rewrite rules. 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

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

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

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

(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

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

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

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

when a is the blank symbol $\#$ (to handle the right end of the tape). Each such transition term t_i is embedded in an erase operator ∂, so that the machine can (non-deterministically) skip over it, and the term *machine* is just the concatenation $\partial(t_1)\partial(t_2)\cdots\partial(t_k)$ of all transition terms.

† Cf. Bergstra and Tucker (1980), where it is shown that *six* "hidden" functions suffice for the specification of computable data types.

The rewrite system $R_{\mathcal{M}}$ consists of exactly two rules:

$$\partial(\xi)\tau \rightarrow \tau$$
$$C(\alpha\lambda,\ \sigma\beta\rho,\ \partial(\alpha\sigma\beta\alpha'\beta'\sigma'\alpha''\beta'')\tau) \rightarrow C(\beta'\alpha'\lambda,\ \sigma'\alpha''\beta''\rho,\ machine)$$

where the primed and unprimed Greek letters are all variables. The first rule erases transitions from the machine description until an applicable one is at the head, at which time the second rule can be applied to simulate a move. The first rule also erases extra tape symbols introduced by the fixed-format transitions. (Though there are derivations that erase all applicable transitions and therefore do not correspond to a machine computation, they all terminate.) Clearly, if the machine $\mathcal{M}$ does not terminate for some input tape, then the system $R_{\mathcal{M}}$ 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 $R_{\mathcal{M}}$ not to terminate is for one of the occurrences of C to be rewritten infinitely often by the second rule—in a manner corresponding to an infinite computation of $\mathcal{M}$. □

An alternative proof of undecidability of termination is given in Huet & Lankford (1978); see section 9. The number of rules in that proof depends on the number of machine transitions.†

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 l. The point is that there must be an infinite derivation with some rule application at the root (outermost) symbol, if there is an infinite derivation at all. (This lemma is implicit in Dershowitz (1982*a*) and elsewhere.)

Certainly, if a derivation repeats a term, the system is non-terminating; we call this "cycling":

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.

(The previous lemma is given in Guttag *et al.* (1983) for cycling systems; a stronger version appears in Klop (1980).) 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 non-terminating.

† Cf. the above theorem with Lipton & Snyder (1977), which asserts, *sans* proof, that *three* rules suffice for undecidability.

EXAMPLE. System (4) is non-looping, but has the following infinite derivation, beginning with an instance of its left-hand side $-(\alpha+\beta)$:

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

□

To characterise non-termination, therefore, a notion weaker than looping is needed. Viewing terms as ordered trees suggests the following definition:

DEFINITION 4. The *homeomorphic embedding relation* $\trianglerighteq$ on a set $\mathcal{T}$ of terms is defined recursively as follows:

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

if either

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

or

$$f = g \text{ and } s_{i_j} \trianglerighteq t_j \quad \text{for } all\ j = 1, \ldots, n,$$

where $1 \leqslant i_1 < i_2 < \cdots < i_n \leq m$.

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

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

THEOREM 2 (Kruskal, 1954, 1960). *If $\mathcal{F}$ is a finite set of function symbols, then any infinite sequence $t_1, t_2, \cdots$ of terms in the set $\mathcal{T}(\mathcal{F})$ of terms over $\mathcal{F}$ contains two terms t_j and t_k $(j < k)$* such that $t_j \trianglelefteq t_k$.

For a finite set of fixed-arity function symbols, this result is due to Higman (1952); a more general result will be proved in section 7.

This notion of embedding provides a necessary condition for non-termination:

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 3 (Dershowitz, 1982*a*). *If a finite rewrite system is non-terminating, then it is self-embedding.*

PROOF. If a system R does not terminate, then, by definition, there exists at least one infinite derivation $t_1 \Rightarrow t_2 \Rightarrow \cdots$. Since there can be only a finite number of function symbols appearing in the derivation (those in t_1 and in R), by the previous theorem, $t_j \trianglelefteq t_k$ for some $j < k$. □

To show termination, it follows from Theorem 3 that one need only prove the system to be non-self-embedding. The converse, however, does not hold: self-embedding does not imply non-termination.

EXAMPLE. The rewrite system

$$f(f(x)) \rightarrow f(g(f(x))) \quad (5)$$

is both self-embedding and terminating. □

Unfortunately, even this sufficient condition for termination is undecidable:

THEOREM 4 (Plaisted, 1985). *It is undecidable whether a (finite) rewrite system is self-embedding.*

Of course, self-embedding is semi-decidable: just search through all derivations until an embedding is discovered. (This fact is exploited in Plaisted, 1986.) It is similarly undecidable if a system cycles or loops. (For details, see Plaisted, 1985.)

Termination, which is what we have considered up to now, demands that *all* derivations be finite. For non-deterministic programs—which most rewrite systems are—there is a weaker notion that is also of interest:

DEFINITION 6. A rewrite system $\mathscr{R}$ is *normalising* for a set of terms $\mathscr{T}$, if every term $t \in \mathscr{T}$ has a normal form.

A normalising system is also called *weakly-terminating*. Like termination, normalisation is an undecidable property (see section 9).

EXAMPLE. Let f and g be unary function symbols and b a constant. The one-rule system

$$f(g(x)) \rightarrow g(g(f(f(x)))) \quad (6)$$

is not even normalising; witness the term $f(f(g(b)))$ which has no normal form. □

3. Termination

To express proofs of termination, we need the following concepts: a *partially-ordered* set $(\mathscr{S}, \succ)$ consists of a set $\mathscr{S}$ and a transitive and irreflexive binary relation $\succ$ defined on elements of $\mathscr{S}$.† 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 t of $\mathscr{S}$, either $s \succ t$ or $t \succ s$. 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 $\subseteq$. An *extension* of a partial ordering $\succ$ on $\mathscr{S}$ is a partial ordering $\succ'$ also on $\mathscr{S}$ such that $s \succ t$ implies $s \succ' t$ for all $s, t \in \mathscr{S}$; a *restriction* of $\succ$ is a partial ordering $\succ'$ on $\mathscr{S}$ such that $s \succ' t$ implies $s \succ t$ for all $s, t \in \mathscr{S}$. Partial orderings of sets of elements can be used to induce a partial ordering of tuples of component elements: an n-tuple $(s_1, s_2, \ldots, s_n)$ in $(\mathscr{S}_1, \succ_1) \times (\mathscr{S}_2, \succ_2) \times \cdots \times (\mathscr{S}_n, \succ_n)$ is *lexicographically* greater than another such tuple $(t_1, t_2, \ldots, t_n)$ if $s_i \succ_i t_i$ for some i $(1 \leq i \leq n)$, while $s_j = t_j$ for all $j < i$. In the same manner, a partial ordering $\succ$ on a set $\mathscr{S}$ induces a lexicographic ordering $\succ^{*}$ on the set $\mathscr{S}^{*}$ of *finite sequences* (*words*) over $\mathscr{S}$; in this case, a sequence is greater than all of its prefixes.

A partially ordered set $(\mathscr{S}, \succ)$ is said to be *well-founded* if there are no infinite (strictly)

† Asymmetry of such a *strict* partial ordering follows from transitivity and irreflexivity.

descending sequences $s_1 \succ s_2 \succ s_3 \succ \cdots$ of elements of $\mathscr{S}$. Thus, the natural numbers $\mathbf{N}$ under there "natural" ordering $>$ are 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 (positive) rationals or reals. Clearly, any restriction of a well-founded ordering is also well-founded. If $(\mathscr{S}_1, \succ_1)$ and $(\mathscr{S}_2, \succ_2)$ are two well-founded sets, then their lexicographically ordered cross-product $(\mathscr{S}_1 \times \mathscr{S}_2, \succ_{1,2})$ is also well-founded. 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 $>^4$ of quadruples of naturally ordered natural numbers; the lexicographic ordering $>^*$ of unbounded-length sequences of natural numbers is not well-founded. (See, e.g., Manna, 1974.)

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

THEOREM 5 (Manna & Ness, 1969). *A rewrite system $\mathscr{R}$ over a set of terms $\mathscr{T}$ is terminating if, and only if, there exists a well-founded ordering $\succ$ over $\mathscr{T}$ such that*

$$t \Rightarrow_{\mathscr{R}} u \text{ implies } t \succ u$$

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

That is, $\mathscr{R}$ terminates if its rewrite relation $\Rightarrow$ is contained in a well-founded ordering $\succ$. (This theorem holds equally well for finite and infinite systems; the proofs in Manna & Ness, 1969 and Lankford, 1975*a* presuppose finite $\mathscr{R}$.)

EXAMPLE. System (0) terminates, since the lexicographic ordering of tuples of colours (with *blue* > *white* > *red*) is well-founded and the tuple of colours corresponding to a sequence of marbles is reduced with each rule application. By the nature of the lexicographic ordering, one 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 reformulation of Theorem 5 (see Kamin & Lévy, 1980) takes advantage of the structure of terms:

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

$$l \succ r$$

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

$$t \Rightarrow_{\mathscr{R}} u \text{ and } t \succ u \text{ imply } f(\cdots t \cdots) \succ f(\cdots u \cdots)$$

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

EXAMPLE. The system

$$f(f(\alpha)) \to f(g(f(\alpha))) \tag{5}$$

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 $f(g(f(f(a)))) \not> f(f(a))$. But, since $g(f(f(a))) \not\Rightarrow^* f(a)$, this corollary can apply. □

The problem with using the above results lies in the need to consider an infinite number of possible rewrites $t \Rightarrow u$ in termination proofs. To avoid that, we can make use of a definition of monotonicity:

DEFINITION 7. A partial ordering $\succ$ over a set of terms $\mathcal{T}$ is *monotonic* (with respect to term structure) if it possesses 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. This suggests the following means of proving termination:

THEOREM 6 (Lankford, 1977). *A rewrite system $\mathcal{R}$ 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 $l \to r$ in $\mathcal{R}$ and for any substitution of terms in $\mathcal{T}$ for the variables of the rule.

Note that the ordering $\succ$ is defined on the set $\mathcal{T}$ of *ground* (i.e. *closed*) terms, without variables; the theorem requires that $l \succ r$ for *all* (ground) substitutions that yield terms in $\mathcal{T}$. Together with monotonicity, this "local" condition on rules ensures that $t \succ u$ whenever t rewrites to u for terms t and u in $\mathcal{T}$, but requires some means of testing inequality for all substitutions. An alternative is to speak of an ordering of *free* terms, containing variables, while insisting that the ordering be *stable* with respect to substitutions, i.e. that if $t \succ u$, then $t\sigma \succ u\sigma$ for all substitutions σ for variables in t and u. Then one need only require that $l \succ r$ for each rule in some monotonic, stable, and well-founded ordering $\succ$ on free terms. As we will see, it is sometimes possible to "lift" a ground ordering on $\mathcal{T}$ to an ordering of free terms, so that $l \succ r$ in the lifted ordering guarantees that in fact $l\sigma \succ r\sigma$ in the base ordering for all ground substitutions σ.

EXAMPLE. The system

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

terminates. To see this, consider the following stable, well-founded, monotonic ordering on free monadic terms (constructed from the unary symbols f and g, constants, and variables): terms are incomparable if one has a variable not in the other. Otherwise, a term s is greater than a term t if s is longer than t, or if they have the same number of symbols, but the *root* (outermost) symbol of s is f while that of t is g, or if they are of the same length and their root symbols are identical, but the operand in s is (recursively) greater than the operand in t. The above rule is clearly a reduction *vis-à-vis* this ordering. This is an example of the ordering used in Knuth & Bendix (1970); see section 5. □

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

DEFINITION 8. A *termination function* τ from a set of terms $\mathcal{T}(\mathcal{F})$ to a partially-ordered set $(\mathcal{W}, \succ)$ is composed of a set of functions $f_\tau: \mathcal{W}^n \to \mathcal{W}$, one for each function symbol f

and arity n, 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

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

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

In other words, a termination function is a monotonic morphism from the free $\mathcal{F}$-algebra $\mathcal{T}(\mathcal{F})$ to an $\mathcal{F}$-algebra $\mathcal{W}$ that is well-founded under $\succ$. With this definition, we have the following refinement of the previous theorem:

THEOREM 7 (Lankford, 1975*a*). *A rewrite system $\mathcal{R}$ over a set of terms $\mathcal{T}(\mathcal{F})$ is terminating if, and only if, there exists a well-founded set $(\mathcal{W}, \succ)$ and termination function $\tau: \mathcal{T}(\mathcal{F}) \to \mathcal{W}$, such that*

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

for each rule $l \to r$ in $\mathcal{R}$ and for any substitution σ of ground terms in $\mathcal{T}(\mathcal{F})$ for the variables of the rule.

The "if" direction of this theorem, and the preceding two, underlies most of the early termination proofs (e.g. Gorn, 1967; Iturriaga, 1967; Knuth & Bendix, 1970; Manna & Ness, 1970). The "only if" direction is also straightforward (let $\succ$ be the derivation relation itself); a proof for finite $\mathcal{R}$ appears in Lankford (1975*a*).

The use of monotonic *polynomial interpretations* was developed in Lankford (1975*a*, 1979). Using this method, an integer polynomial $F(x_1, \ldots, x_n)$ of degree n is associated with each n-ary function symbol f. The choice of coefficients must ensure monotonicity and that terms are mapped into non-negative integers only, as is, for example, the case when all coefficients are positive. Then each rule must be shown to be reducing; that is, for each rule $l \to r$, the polynomial $\tau(l) - \tau(r)$ must be positive for non-negative interpretations of rule variables. Linear interpretations were used in Knuth & Bendix (1970); linear and quadratic ones were used in Manna & Ness (1970), who also illustrate how the coefficients of linear interpretations can be chosen by solving the desired inequalities; a number of other examples of polynomial interpretations may be found in Dershowitz & Manna (1979); the method in Iturriaga (1967) is based on exponential interpretations. An implementation of the polynomial method was incorporated in the theorem-prover described in Ballantyne & Lankford (1975); recent work on automating polynomial proofs is reported in Ben-Cherifa & Lescanne (1986).

EXAMPLE. Let $\mathcal{T}$ consist of all terms constructed from the constants 0 and 1 and the binary function symbols $+$ and $\times$. To show that the system

$$\begin{aligned} \alpha \times (\beta + \gamma) &\to (\alpha \times \beta) + (\alpha \times \gamma) \\ (\beta + \gamma) \times \alpha &\to (\beta \times \alpha) + (\gamma \times \alpha) \\ (\alpha + \beta) + \gamma &\to \alpha + (\beta + \gamma) \end{aligned} \tag{8}$$

over $\mathcal{T}$ terminates, we use the following polynomial interpretation:

$$\begin{aligned} \tau(\alpha \times \beta) &= \tau(\alpha) \cdot \tau(\beta) & \tau(0) &= 2 \\ \tau(\alpha + \beta) &= 2\tau(\alpha) + \tau(\beta) & \tau(1) &= 2. \end{aligned}$$

Under this interpretation, each rule is a reduction. For each of the first two rules, we have

$$\tau(l) = \tau(\alpha)\cdot\tau(\beta+\gamma) = 2\tau(\alpha)\cdot\tau(\beta)+\tau(\alpha)\cdot\tau(\gamma)+\tau(\alpha)$$
$$\tau(r) = 2\tau(\alpha)\cdot\tau(\beta)+\tau(\alpha)\cdot\tau(\gamma)+1.$$

Since constants are given the interpretation 2, we must have $\tau(\alpha) > 1$ for all terms α. For the third rule, we have

$$\tau(l) = 2\tau(\alpha+\beta)+\tau(\gamma)+1 = 4\tau(\alpha)+2\tau(\beta)+\tau(\gamma)+3$$
$$\tau(r) = 2\tau(\alpha)+2\tau(\beta)+\tau(\gamma)+2.$$

Since $\tau(\alpha)$ is non-negative, this is a reduction. □

Note that, for termination proofs, constants (and hence terms) can be assigned arbitrarily large values; thus, it suffices to show that $\tau(l)-\tau(r)$ is *eventually* positive. This suggests the following recursive test, due to Lankford (1976): let p be a polynomial in variables $x_1, x_2, \ldots, x_n$. It is eventually positive, if all its coefficients are positive, or if $n \geq 1$ and its n first partial derivatives $\partial p/\partial x_1, \partial p/\partial x_2, \ldots, \partial p/\partial x_n$ are eventually positive.

EXAMPLE. Consider the following system (for symbolic differentiation with respect to x):†

$$\begin{aligned}
D_x x &\rightarrow 1\\
D_x a &\rightarrow 0\\
D_x(\alpha+\beta) &\rightarrow D_x\alpha + D_x\beta\\
D_x(\alpha\times\beta) &\rightarrow \beta\times D_x\alpha + \alpha\times D_x\beta\\
D_x(\alpha-\beta) &\rightarrow D_x\alpha - D_x\beta\\
D_x(-\alpha) &\rightarrow -D_x\alpha\\
D_x\left(\frac{\alpha}{\beta}\right) &\rightarrow \frac{D_x\alpha}{\beta} - \alpha\times\frac{D_x\beta}{\beta^2}\\
D_x(\ln\alpha) &\rightarrow \frac{D_x\alpha}{\alpha}\\
D_x(\alpha^\beta) &\rightarrow \beta\times\alpha^{\beta-1}\times D_x\alpha + \alpha^\beta\times(\ln\alpha)\times D_x\beta,
\end{aligned} \tag{9}$$

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

$$\begin{aligned}
\tau(\alpha+\beta) &= \tau(\alpha)+\tau(\beta) & \tau(\alpha\times\beta) &= \tau(\alpha)+\tau(\beta)\\
\tau(\alpha-\beta) &= \tau(\alpha)+\tau(\beta) & \tau(\alpha/\beta) &= \tau(\alpha)+\tau(\beta)\\
\tau(\alpha^\beta) &= \tau(\alpha)+\tau(\beta) & \tau(D_x\alpha) &= \tau(\alpha)^2\\
\tau(-\alpha) &= \tau(\alpha)+1 & \tau(\ln\alpha) &= \tau(\alpha)+1.
\end{aligned}$$

For each of the nine rules $l\rightarrow r$, the value of $\tau(l)$ is greater than that of $\tau(r)$ when the interpretation of the variables is sufficiently large. For example,

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

† This system is taken from Knuth (1973), p. 337. Proving termination of the first five of these rules was one of the problems on a qualifying exam given at Carnegie-Mellon University in 1967.

while

$$\tau\left(\frac{D_x\alpha}{\beta} - \alpha \times \frac{D_x\beta}{\beta^2}\right) = \tau(D_x\alpha) + \tau(\beta) + \tau(\alpha) + \tau(D_x\beta) + \tau(\beta^2)$$
$$= \tau(\alpha)^2 + \tau(\beta)^2 + \tau(\alpha) + 2\tau(\beta) + \tau(2).$$

The polynomial

$$x^2 + y^2 + 2xy - x^2 - y^2 - x - 2y - c$$

(with x for $\tau(\alpha)$, y for $\tau(\beta)$, and c for $\tau(2)$) is eventually positive, since its two derivatives, $2y-1$ and $2x-2$, are. □

Integer polynomials cannot, however, suffice for termination proofs in general, since that would place a super exponential bound on the length of computations; by the same token, primitive recursive interpretations cannot suffice (as pointed out in Stickel, 1976).

EXAMPLE. It seems that System (1) cannot be proved to terminate with any monotonic polynomial interpretation (Dershowitz, 1983).† But termination can be proved using exponentials (Filman, 1978), defining $\tau: \mathcal{T} \to \mathbf{N}$ as follows:

$$\tau(\alpha+\beta) = \tau(\alpha) + \tau(\beta) + 1 \qquad \tau(-\alpha) = 2^{\tau(\alpha)}$$
$$\tau(\alpha \times \beta) = \tau(\alpha) \cdot \tau(\beta) \qquad \tau(u) = 2,$$

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 in all derivations. We call an ordering $\succ$ for which $\succ \cap \Rightarrow^*$ is well-founded for any finite $\mathcal{R}$, *well-founded for derivations*, the advantage being that a derivation (for finite $\mathcal{R}$) can only involve a finite number of function symbols. Thus, to apply Theorem 5, we need only that $\succ$ be a well-founded ordering for derivations. In particular, Theorem 3 implies the following:

LEMMA 2. *A partial ordering $\succ$ is well-founded for derivations if it (has any extension that) extends the embedding relation $\rhd$.*

To apply the "local" method of Theorem 6, we also need $\succ$ to be monotonic. The following definition describes monotonic extensions of $\rhd$:

DEFINITION 9 (Dershowitz, 1982*a*). 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

† This system was presented in Iturriaga (1967) without a proof of termination.

deleting subterms of a (variable arity) function symbol reduces the term in the ordering: if the function symbols f have fixed arity, the deletion condition is superfluous. (Simplification orderings for fixed-arity function symbols were investigated in Higman, 1952.) Together these conditions imply that "syntactically simpler" terms are smaller in the ordering. Hence:

THEOREM 8 (Dershowitz, 1982*a*). *Any simplification ordering is a monotonic well-founded ordering for derivations.*

In the previous section, we observed the use of polynomial interpretations for termination proofs. That method requires that terms be mapped onto the well-founded non-negative integers. Using simplification orderings, on the other hand, allows those methods to be extended to domains that are not themselves well-founded. For example, one can associate a monotonic multivariate polynomial $F(x_1, \ldots, x_n)$ over the *reals* with each n-ary function symbol f (see Dershowitz, 1979). For any given choice of polynomials F to provide a simplification ordering, we must have that

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

and

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

for all positions i and for all real-valued x_i's.† For termination, we need

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

for all rules $l \to r$ and for all assignments σ to the variables in l. Allowing the x_i's to take on any real value is usually too strong a requirement; instead one may show that terms always map into some subset S of the reals, i.e. $x_1, \ldots, x_n$ in S implies $F(x_1, \ldots, x_n)$ in S. Then one need only show that the conditions hold for all x in S. In practice, S is usually the subrange of x greater than some c. The above conditions are all decidable (albeit in superexponential time), since they are logical combinations of multivariate polynomial inequalities over the reals (Tarski, 1951; see Cohen (1969) for a much briefer decision procedure and Collins (1975) for a more efficient one). Thus, the polynomial ordering can be effectively "lifted" to terms containing rule variables (as first suggested for integer polynomials in Lankford (1975*a*) for those cases where the interpretation is reducing for all real values of the variables). It is similarly decidable if there exist polynomials (and a suitable definition of S) 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, 1979).

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

$$(\alpha \times \beta) \times \gamma \to \alpha \times (\beta \times \gamma). \tag{10}$$

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

† The methods of the next section allow the strict inequalities $>$ in these two conditions to be replaced by $\geq$; see Dershowitz (1982*a*).

One example of real polynomials that serve the purpose are:

$$\tau(\alpha \times \beta) = \sqrt{2} \cdot \tau(\alpha) + \tau(\beta)$$
$$\tau(u) = 10^{-6},$$

for all constants u. This termination function τ maps terms to positive reals and satisfies the conditions on simplification orderings. It decreases for the subterm that the rule is applied to: for any terms α, β, and γ,

$$\tau((\alpha \times \beta) \times \gamma) = 2\tau(\alpha) + \sqrt{2}\tau(\beta) + \tau(\gamma) > \sqrt{2}\tau(\alpha) + \sqrt{2}\tau(\beta) + \tau(\gamma) = \tau(\alpha \times (\beta \times \gamma)),$$

since $\tau(\alpha) > 0$. □

Most orderings used in conjunction with Theorem 6 to prove termination of rewrite systems are simplification orderings. In fact:

THEOREM 9 (Dershowitz, 1982*a*). *Any total monotonic ordering $\succ$ is well-founded for derivations if, and only if, it is a simplification ordering.*

In particular, polynomial interpretations must satisfy the subterm property. In general, however, total monotonic orderings, and hence simplification orderings, cannot suffice for termination proofs.†

EXAMPLE. Consider the terminating system‡

$$f(a) \to f(b)$$
$$g(b) \to g(a). \tag{11}$$

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 (Theorem 1) that termination is undecidable for two-rule systems; for one-rule systems, the question of decidability is open. The following is known:

THEOREM 10 (Jouannaud & Kirchner, 1984). *It is decidable whether 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* $(\mathscr{S}, \succsim)$ consists of a set $\mathscr{S}$ and a transitive and reflexive binary relation $\succsim$ defined on elements of $\mathscr{S}$. For example, the set of integers is quasi-ordered under the relation "greater or congruent modulo 10". For any rewrite system $\mathscr{R}$, the derivability relation $\Rightarrow^*_{\mathscr{R}}$ is a quasi-ordering on $\mathscr{T}$. Given a quasi-ordering $\succsim$ on a set $\mathscr{S}$, we define the associated equivalence relation $\approx$ as both $\succsim$ and $\precsim$ and (strict) partial ordering $\succ$ as $\succsim$ but not $\precsim$. An *extension* of a quasi-ordering $\succsim$ *on* $\mathscr{S}$ is a quasi-ordering $\succsim'$ also on $\mathscr{S}$ such that $s \succsim t$ implies $s \succsim' t$ and $s \succ t$ implies $s \succ' t$ for all $s, t \in \mathscr{S}$; the relation $\succsim$ is, in that

† Thus, the requirement that a total monotonic well-founded ordering also have the subterm property (e.g. in Brown (1975)) turns out to be redundant.

‡ Given, for example, in Huet & Oppen (1980).

case, a *restriction* of $\gtrsim'$. A quasi-ordering $\gtrsim$ *on* $\mathscr{S}$ is *total* if, for any two elements s and t in $\mathscr{S}$, either $s \gtrsim t$ or else $s \precsim t$.

Note that the strict part $\succ$ of a quasi-ordering $\gtrsim$ is well-founded if, and only if, all infinite *quasi-descending* sequences $s_1 \gtrsim s_2 \gtrsim s_3 \gtrsim \cdots$ of elements of $\mathscr{S}$ contain a pair $s_j \lesssim s_k$ for some $j < k$. We will refer to a quasi-ordering $\gtrsim$ as well-founded whenever its strict part $\succ$ is. If $\gtrsim$ is well-founded, then from some point on, in any infinite quasi-descending sequence, all elements are equivalent.

Suppose $\gtrsim$ *and* $\gtrsim'$ are two well-founded quasi-orderings on a set $\mathscr{T}$ of terms, and we wish to combine them (lexicographically) to obtain a single well-founded quasi-ordering $\gtrsim''$ on $\mathscr{T}$ for use in termination proofs. That is, we define $t \gtrsim'' u$ if either $t \succ u$, or else $t \approx u$ and $t \gtrsim' u$. In order for $\succ''$ to be a monotonic ordering, we not only need $\succ$ and $\succ'$ to be monotonic, but also need $\approx$ to be a congruence, i.e. $t \approx u$ should imply $f(\cdots t \cdots) \approx f(\cdots u \cdots)$. We have the following definition:

DEFINITION 10. A quasi-ordering $\gtrsim$ over a set of terms $\mathscr{T}$ is *monotonic* if

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

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

Clearly, if $\gtrsim$ is monotonic, then the associated equivalence relation $\approx$ is a congruence; hence a monotonic quasi-ordering is sometimes termed a "pre-congruence". The use of pairs of monotonic polynomial interpretations in termination proofs is illustrated in Manna & Ness (1970) and Lankford (1979); its implementation is described in Ben-Cherifa & Lescanne (1986).

EXAMPLE. To prove termination of

$$\begin{aligned} \alpha \times (\beta + \gamma) &\rightarrow (\alpha \times \beta) + (\alpha \times \gamma) \\ (\beta + \gamma) \times \alpha &\rightarrow (\alpha \times \beta) + (\alpha \times \gamma) \\ (\alpha \times \beta) \times \gamma &\rightarrow \alpha \times (\beta \times \gamma) \end{aligned} \tag{12}$$

over $\mathscr{T}$ we can use the following pair, τ and τ', of monotonic polynomial interpretations:

$$\begin{aligned} \tau(\alpha \times \beta) &= \tau(\alpha) \cdot \tau(\beta) & \tau'(\alpha \times \beta) &= 2\tau'(\alpha) + \tau'(\beta) \\ \tau(\alpha + \beta) &= \tau(\alpha) + \tau(\beta) + 1 & \tau'(\alpha + \beta) &= \tau'(\alpha) + \tau'(\beta) \\ \tau(u) &= 2 & \tau'(u) &= 2, \end{aligned}$$

where u is any constant. The first two rules reduce under τ; while the last reduces under τ'. Since the last rule preserves value under τ, we can use the lexicographic combination of τ and τ' to prove termination of the whole system. □

Well-founded quasi-orderings can be used to prove termination in the following way:

THEOREM 11. *A rewrite system $\mathscr{R}$ over a set of terms $\mathscr{T}$ is terminating if there exists a well-founded quasi-ordering $\gtrsim$, which enjoys the subterm property,*

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

such that

$$l \succ r$$

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

and such that

$$t \Rightarrow_{\mathcal{R}} u \text{ and } t \succsim u \text{ imply } f(\cdots t \cdots) \succsim f(\cdots u \cdots)$$

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

EXAMPLE. To prove that System (10) terminates, the following well-founded quasi-ordering can be used: $t \succsim u$ if the *size* $|t|$ of t (i.e. the number of function symbols in t) is greater, or if t and u are products of equal size, but the size of the first multiplicand of t is at least as big as that of u. The subterm property certainly holds for this quasi-ordering. The two sides of the rule have the same size, but the size $|\alpha \times \beta|$ of the first multiplicand of the left-hand side, $l = (\alpha \times \beta) \times \gamma$, is of necessity greater than the size $|\alpha|$ of the first multiplicand in the right-hand side, $r = \alpha \times (\beta \times \gamma)$; hence $l \succ r$. Since $|t| = |u|$ whenever $t \Rightarrow u$, we have $\alpha \times t \approx \alpha \times u$, as well as $t \times \alpha \approx u \times \alpha$, whenever $t \Rightarrow u$. □

Recall that a derivation cycles if it repeats a term. That suggests a weaker notion than termination:

DEFINITION 11. A rewrite system $\mathcal{R}$ is *quasi-terminating* for a set of terms $\mathcal{T}$ if every infinite derivation $t_1 \Rightarrow t_2 \Rightarrow t_3 \Rightarrow \cdots$ of terms in $\mathcal{T}$ cycles.

EXAMPLE. The following system quasi-terminates, as does any (finite) system that never increases the size of terms:

$$\begin{aligned} (\alpha \times \beta) \times \gamma &\to \alpha \times (\beta \times \gamma) \\ \alpha \times \beta &\to \beta \times \alpha. \end{aligned} \tag{13}$$

□

EXAMPLE. The following non-terminating system† is, nonetheless, quasi-terminating:

$$\begin{aligned} f(a, b, \alpha) &\to f(\alpha, \alpha, \alpha) \\ g(\alpha, \beta) &\to \alpha \\ g(\alpha, \beta) &\to \beta. \end{aligned} \tag{14}$$

To see this, notice that the depth of a term (i.e. the maximum nesting of function symbols) in a derivation is bounded by the depth of the initial term. □

Note that for finite systems $\mathcal{R}$, a term can rewrite in a single step to only a finite number of distinct terms. Thus:

LEMMA 3. *A finite rewrite system $\mathcal{R}$ is quasi-terminating for a set of terms $\mathcal{T}$ if, and only if, all its derivations contain only a finite number of distinct terms.*

(An infinite system can have cycling derivations with an infinite number of distinct terms.) Finite quasi-terminating systems are also *globally finite* in the sense of Huet (1980), i.e. only a finite number of distinct terms are derivable from any given term. As might be expected:

THEOREM 12 (Guttag *et al.*, 1983). *It is undecidable whether a (finite) rewrite system is quasi-terminating.*

† Borrowed from Toyama (1987).

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

We call an equivalence relation $\approx$ that admits only finite equivalence classes *thin*. To prove that a system is quasi-terminating, one can use quasi-orderings and thinness in the following natural way:

THEOREM 13. *A rewrite system $\mathcal{R}$ over a set of terms $\mathcal{T}$ is quasi-terminating if there exists a well-founded quasi-ordering $\gtrsim$, whose equivalence relation $\approx$ is thin, such that*

$$t \Rightarrow_{\mathcal{R}} u \text{ implies } t \gtrsim u$$

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

These conditions on the quasi-ordering (viz. well-foundedness and thinness) are satisfied if for every term t there is only a finite number of terms s such that $t \gtrsim s$ (see Göbel, 1983).

A stronger notion than well-foundedness plays an important role in what follows:

DEFINITION 12 (Kruskal, 1960). A set $\mathcal{S}$ is *well-quasi-ordered* under a quasi-ordering $\gtrsim$ if every infinite sequence $s_1, s_2, \cdots$ of elements of $\mathcal{S}$ contains a pair of elements s_j and s_k, $j < k$, such that $s_j \lesssim s_k$.

Well-quasi-ordered sets are said to have the *finite basis property* in Higman (1952) and to be *partially well-ordered* in Rado (1954). For a survey of the history and applications of well-quasi-orderings, see Kruskal (1972). Note that any finite set is well-quasi-ordered under any quasi-ordering (including equality), and that a well-founded set is well-quasi-ordered when it has only a finite number of pairwise incomparable elements. We have seen already (Theorem 2) that the embedding relation $\trianglerighteq$ is a well-quasi-ordering of the set of terms $\mathcal{T}(\mathcal{F})$ for finite $\mathcal{F}$.

Clearly, any extension of a well-quasi-ordering is also a well-quasi-ordering. Moreover, if a quasi-ordering has only well-founded extensions, then it is a well-quasi-ordering; in other words, a set $\mathcal{S}$ is well-quasi-ordered under $\gtrsim$ if, and only if, all its extensions (and all of their restrictions) are well-founded. In particular, if $\succ$ is a well-ordering (i.e. a total well-founded ordering) of $\mathcal{S}$, then $\mathcal{S}$ is well-quasi-ordered under $\succeq$ (the reflexive closure of $\succ$).

In general, whenever $\succeq$ is a well-quasi-ordering, the equivalence relation $\approx$ must be thin, because any infinite sequence of equivalent terms would have to include repetitions. Furthermore, if $\succeq$ is a well-quasi-ordering, then $\gtrsim$ is well-founded. Hence, we have:

COROLLARY. *A rewrite system $\mathcal{R}$ over a set of terms $\mathcal{T}$ is quasi-terminating if there exists a quasi-ordering $\gtrsim$, such that its restriction $\succeq$ is a well-quasi-ordering, and such that*

$$t \Rightarrow_{\mathcal{R}} u \text{ implies } t \gtrsim u$$

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

In particular, we can—for finite $\mathcal{R}$—use the well-quasi-ordered embedding relation $\trianglerighteq$. (Strictly speaking, $\approx$ is thin, in this case, only when it is restricted to terms appearing in any single derivation.)

EXAMPLE. Consider the one-rule system (for normalising conditionals),†

$$if(if(\alpha, \beta, \gamma), \delta, \varepsilon) \rightarrow if(\alpha, if(\beta, \delta, \varepsilon), if(\gamma, \delta, \varepsilon)) \tag{15}$$

and the monotonic polynomial interpretation,

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

with constants assigned the value 2. The quasi-ordering $\succeq$, where $t \succeq u$ if, and only if, $\tau(t) \geq \tau(u)$, contains the embedding relation $\trianglerighteq$ and is thus a well-quasi-ordering. Since $\tau(l) = \tau(r)$ for the rule, the above corollary establishes quasi-termination. □

Another way to establish thinness is the following:

THEOREM 14. *If the strict part $\succ$ of a quasi-ordering $\succsim$ on a set $\mathcal{T}(\mathcal{F})$ of terms over a finite set $\mathcal{F}$ of function symbols enjoys the strict subterm property,*

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

enjoys the strict deletion property,

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

and has the property that for any term t in $\mathcal{T}(\mathcal{F})$ the length of a strictly descending sequence beginning with t is bounded, then the equivalence relation $\approx$ is thin.

Note that the partial ordering $\succ$ is well-founded, but not necessarily monotonic. This is the essence of the method in Lipton & Snyder (1977), extended to allow varyadic function symbols f.

EXAMPLE. Consider the following system (for multiplication):

$$\begin{aligned} \alpha \times (\beta+\gamma) &\rightarrow (\alpha \times \beta)+(\alpha \times \gamma) \\ (\beta+\gamma)\times\alpha &\rightarrow (\beta \times \alpha)+(\gamma \times \alpha) \\ \alpha \times 1 &\rightarrow \alpha \\ 1 \times \alpha &\rightarrow \alpha \\ \alpha \times 0 &\rightarrow 0 \\ 0 \times \alpha &\rightarrow 0. \end{aligned} \tag{16}$$

Under the "natural" interpretation (+ as addition and $\times$ as multiplication, but all constants as 2), terms map onto natural numbers (and hence the term ordering is of order-type ω), while satisfying the subterm property. Since, under this interpretation, $t \succsim u$ whenever $t \Rightarrow u$, the system quasi-terminates. □

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 (1985).

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

THEOREM 15. *A quasi-terminating rewrite system $\mathcal{R}$ over a set of terms $\mathcal{T}$ is terminating if, and only if, there exists a monotonic quasi-ordering $\succsim$ such that*

$$l \succ r$$

† Circulated by Boyer (1977).

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

The "if" direction appears in Dershowitz (1982*a*); the "only-if" direction is trivial (let $\gtrsim$ be the derivability relation $\Rightarrow^*_{\mathscr{R}}$). Thus, to prove termination one can first find an appropriate quasi-ordering $\gtrsim$ guaranteeing quasi-termination, and then find any monotonic quasi-ordering $\gtrsim'$ under which each rule is a reduction.†

EXAMPLE. A full proof of termination for quasi-terminating System (15) may be obtained via the monotonic quasi-ordering $\gtrsim'$, where $t \gtrsim' u$ if, and only if, $|t| \leq |u|$. A term "decreases" under this quasi-ordering with each application of the size-increasing rule. □

Using monotonicity, we can apply the corollary to Theorem 13 and also give a local condition for quasi-termination:

THEOREM 16. *A rewrite system $\mathscr{R}$ over a set of terms $\mathscr{T}$ is quasi-terminating if there exists a monotonic quasi-ordering $\gtrsim$, such that the relation $\succeq$ is a well-quasi-ordering, and such that*

$$l \gtrsim r$$

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

EXAMPLE. System (16) can be shown to be quasi-terminating using the "natural" interpretation of *plus* and *times*, which preserves the value of a term under rewriting, i.e. $\tau(l) = \tau(r)$, for the first two rules. By letting constants (including 0 and 1) have a value no less than one, the quasi-ordering $\geq$ becomes a monotonic extension of the well-quasi-ordered embedding relation $\trianglerighteq$. □

By combining monotonicity with additional properties, we can extend the results on simplification orderings of the previous section:

DEFINITION 13 (Dershowitz, 1982*a*). A monotonic quasi-ordering $\gtrsim$ is a *quasi-simplification ordering* for a set of terms $\mathscr{T}$ if it possesses the *subterm property*,

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

and *deletion property*,

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

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

A quasi-simplification ordering for fixed-arity function symbols (without the deletion property) is called a *divisibility order* in Higman (1952). This definition means that any quasi-ordering $\gtrsim$ which is a monotonic extension the embedding relation $\trianglerighteq$, is a quasi-simplification ordering. By Theorem 3, its strict part $\succ$ is well-founded for derivations. Thus, as a corollary to Theorem 11, we get:

THEOREM 17 (Dershowitz, 1982*a*). *A finite rewrite system $\mathscr{R}$ over a set of terms $\mathscr{T}$ is terminating if there exists a quasi-simplification ordering $\gtrsim$ such that*

$$l \succ r$$

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

† Lipton & Snyder (1977) and Guttag *et al.* (1983) use "increasing length" where any monotonic quasi-ordering would do.

Suppose we are given two quasi-orderings, one on a set of terms and the other on its set of function symbols. They can be combined to form another ordering on terms:

DEFINITION 14 (Knuth & Bendix, 1970; Dershowitz, 1982*a*). Let $\gtrsim_F$ be a quasi-ordering on a set $\mathcal{F}$ of fixed-arity function symbols and $\gtrsim_T$ a quasi-ordering of the set $\mathcal{T}(\mathcal{F})$ of terms over $\mathcal{F}$. The *Knuth–Bendix ordering* $\gtrsim_{kbo}$ on $\mathcal{T}(\mathcal{F})$ is defined recursively as follows:

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

if

$$s \succ_T t,$$

or else

$$s \approx_T t \text{ and } f \succ_F g,$$

or else

$$s \approx_T t,\ f \approx_F g \text{ and } (s_1, \ldots, s_m) \gtrsim^{*}_{kbo} (t_1, \ldots, t_n),$$

where $\gtrsim^{*}_{kbo}$ is the lexicographic ordering induced by $\gtrsim_{kbo}$.

This generalises the ordering defined in Knuth & Bendix (1970) to any quasi-ordering $\gtrsim_T$.

THEOREM 18 (Dershowitz, 1982*a*). *If $\gtrsim_T$ is a quasi-simplification ordering on a set $\mathcal{T}(\mathcal{F})$ of terms over a set $\mathcal{F}$ of fixed-arity function symbols, such that $f(\cdots t \cdots) \approx_T t$ can hold only when f is unary and maximal under the quasi-ordering $\gtrsim_F$ of $\mathcal{F}$ (i.e. $f \gtrsim_F g$ for all function symbols $g \in \mathcal{F}$), then $\succ_{kbo}$ is a simplification ordering on $\mathcal{T}(\mathcal{F})$.*

The condition on the function symbol ordering $\gtrsim_F$ ensures that $\succ_{kbo}$ possesses the subterm property.

To prove termination via this method, one must find appropriate quasi-orderings $\gtrsim_F$ and $\gtrsim_T$ for which $l \succ_{kbo} r$ for all rules $l \rightarrow r$ in the given system. For example, the method of Knuth & Bendix (1970) totally orders function symbols under an ordering $\succ_F$, and also assigns a positive integer weight to each constant and a non-negative integer weight to each other function symbol, with $\gtrsim_T$ comparing terms according to the sum of the weights of their respective function symbols. Thus, the condition on $\gtrsim_F$ requires that a unary function symbol have zero weight only if it is the largest function symbol under $\succ_F$. Lankford (1979) replaces the linear weight function with monotonic polynomials having non-negative integer coefficients. Since both these methods use total monotonic orderings, by Theorem 9, 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 $\gtrsim_{kbo}$, taking $t \gtrsim_T u$ to be $|t| \geq |u|$ and $\gtrsim_F$ to be equality. □

EXAMPLE. This method applies also to the following system:

$$\begin{aligned} --\alpha &\rightarrow \alpha \\ -(\alpha+\beta) &\rightarrow ---\alpha \times ---\beta \\ -(\alpha \times \beta) &\rightarrow ---\alpha + ---\beta \end{aligned} \tag{17}$$

with $t \gtrsim_T u$ if, and only if, the number of occurrences of function symbols other than *minus* in t is no less than in u, and *minus* is the largest function symbol under $\succ_F$. □

6. Sequence Orderings

A quasi-ordering $\gtrsim$ on a set $\mathscr{S}$ induces a quasi-ordering $\trianglerighteq_{\gtrsim}$ on the set $\mathscr{S}^*$ of finite sequences over $\mathscr{S}$ in the following manner:

DEFINITION 15. The *embedding relation* $\trianglerighteq_{\gtrsim}$ on a set $\mathscr{S}^*$ of finite sequences over a set $\mathscr{S}$, quasi-ordered by $\gtrsim$, is defined as follows:

$$(s_1, s_2, \ldots, s_m) \trianglerighteq_{\gtrsim} (t_1, t_2, \ldots, t_n)$$

if

$$s_{i_j} \gtrsim t_j \quad \text{for } all\ j = 1, \ldots, n,$$

where $1 \leq i_1 < i_2 < \cdots < i_n \leq m$.

That this relation preserves well-quasi-orderedness is known as Higman's lemma:

LEMMA 4 (Higman, 1952). *A set $\mathscr{S}^*$ of finite sequences over a set $\mathscr{S}$ is well-quasi-ordered under the embedding relation $\trianglerighteq_{\gtrsim}$ if, and only if, the set $\mathscr{S}$ is well-quasi-ordered under the quasi-ordering $\gtrsim$.*

PROOF (Nash-Williams, 1963). Suppose the theorem were false. Let the infinite sequence

$$\bar{t} = t_1, t_2, \cdots$$

of words (finite sequences) be a "minimal counterexample". That is, no element of this counterexample can be embedded in a subsequent one, and for every $i = 1, 2, \cdots$ no other counterexample begins with $t_1, t_2, \ldots, t_{i-1}$ followed by a shorter word than t_i. (The Axiom of Choice is needed for such a construction.)

A counterexample cannot contain an infinite subsequence of elements, each of which is a word of length one, since the set $\mathscr{S}$ is itself well-quasi-ordered. So, $\bar{t}$ must contain an infinite subsequence $\bar{r}$ of words of length greater than one. Each of its elements r_i can be split into two strictly shorter, non-empty words r_i' and r_i''. By minimality, the set of left parts must be well-quasi-ordered by the embedding relation (or else $t_1, t_2, \ldots, t_{l-1}, r_1', r_2', \ldots$, where r_1' is the left part of t_l, would be a smaller counterexample than $\bar{t}$). Similarly, the right parts must be well-quasi-ordered.

Now note that (by the infinite versión of Ramsey's theorem) any infinite sequence $q_1, q_2, \cdots$ of elements of a well-quasi-ordered set $(\mathscr{Q}, \gtrsim)$ must contain an *infinite* chain of quasi-ascending elements $q_{i_1} \lesssim q_{i_2} \lesssim \cdots$ (with $1 \leq i_1 < i_2 < \cdots$). For suppose that a chain $q_{i_1} \lesssim q_{i_2} \lesssim \cdots \lesssim q_{i_n}$ could not be extended any further. Then the infinite remainder $q_{i_n+1}, q_{i_n+2}, \cdots$ would either contain an infinite chain or would also contain such an unextendible finite chain. Thus, were there no infinite chain, there would be an infinite number of unextendible finite chains. But the infinite sequence consisting of the final elements of those chains must itself have a quasi-ordered pair, meaning that one of the unextendible chains could, in fact, have been extended.

Thus, the fact that the left parts are well-quasi-ordered by $\trianglerighteq_{\gtrsim}$ means that there is an infinite chain of embeddings $r_{i_1}' \trianglelefteq_{\lesssim} r_{i_2}' \trianglelefteq_{\lesssim} \cdots$. Since the right parts are also well-quasi-ordered, there must also be an embedding among $r_{i_1}'', r_{i_2}'', \cdots$. But then $\bar{t}$ would also contain an embedding.

Since we have shown that there can be no counterexample, the theorem must hold. □

Multisets, or *bags*, are unordered sequences; they are like sets, but allow multiple occurrences of identical elements. 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 $\{3, 4, 0\}$. A quasi-ordering $\gtrsim$ on any given set $\mathscr{S}$ induces a quasi-ordering $\ggtrsim$ on the set $\mathscr{M}(\mathscr{S})$ of finite multisets over $\mathscr{S}$:

DEFINITION 16. For a set $\mathscr{S}$ quasi-ordered by $\gtrsim$, the *multiset ordering* $\ggtrsim$ on the set $\mathscr{M}(\mathscr{S})$ of finite multisets over $\mathscr{S}$ is defined recursively as follows:

$$X = \{x_1, \ldots, x_m\} \ggtrsim \{y_1, \ldots, y_n\} = Y$$

if $X = Y$ or if

$$x_i \approx y_j \text{ and } X - \{x_i\} \ggtrsim Y - \{y_j\},$$

for some $i = 1, \ldots, m$ and $j = 1, \ldots, n$, or

$$x_i \succ y_{j_1}, y_{j_2}, \ldots, y_{j_k} \text{ and } X - \{x_i\} \ggtrsim Y - \{y_{j_1}, y_{j_2}, \ldots, y_{j_k}\},$$

for some $i = 1, \ldots, m$ and $1 \leq j_1 < j_2 < \cdots < j_k \leq n$ $(k \geqslant 0)$.

(A multiset difference $X - Z$ decreases the number of occurrences of each element in X by its number of occurrences in Z.) Two multisets are equivalent under this quasi-ordering if they are the same up to (permutation and) replacement of individual elements with equivalent ones. In the induced strict partial ordering, $M \gg M'$, for two finite multisets M and M' over $\mathscr{S}$, if M' can be obtained from M by replacing one or more elements in M by any (finite) number of elements taken from $\mathscr{S}$, each of which is smaller than one of the replaced elements.

In Dershowitz & Manna (1979) a strict multiset ordering $\gg$ is induced from a given *partial* ordering $\succ$. In that case, two multisets are equivalent only if they are equal as multisets, i.e. have the same elements with the same multiplicities, but perhaps in a different order. If $M \ggeq M'$ in the multiset extension of $\succeq$, but $M \neq M'$, then $M \gg M'$ in the strict multiset ordering. If $\succ$ is the empty relation, then $\gg$ is proper multiset containment. 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 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, 1979.) Alternate definitions of induced orderings on multisets are explored in Jouannaud & Lescanne (1982) and Martin (1986).

EXAMPLE. To prove termination of System (17), we can use Theorem 17 and the quasi-simplification ordering $\gtrsim$, where $t \gtrsim u$ if, and only if,

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

The multisets used here contain the value $|\alpha|_{+\times}$, by which we denote the number of occurrences of symbols other than *minus*, for each operand α of a *minus* sign; these multisets are compared using the multiset ordering induced by $\geq$ for integers. It is easy to see that this monotonic quasi-ordering satisfies the subterm property 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 symbols other than *minus* 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 symbols 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 symbols other than *minus* in $\alpha+\beta$ (and $\alpha \times \beta$) is greater than for each of $--\alpha$, $-\alpha$, α, $--\beta$, $-\beta$, and β. □

Multiset orderings are used in termination proofs (e.g. in Dershowitz & Manna, 1979; Jefferson, 1980; Gardner, 1983) on account of the following:

THEOREM 19 (Dershowitz & Manna, 1979). *A quasi-ordering $\gtrsim$ on a set $\mathcal{S}$ is well-founded if, and only if, the induced multiset ordering $\ggtrsim$ on the set $\mathcal{M}(\mathcal{S})$ of finite multisets over $\mathcal{S}$ is well-founded.*

This result follows from König's lemma (see Dershowitz & Manna, 1979). Well-founded multiset orderings have also been used for inductive proofs in Jouannaud & Kirchner (1986) and Bachmair (1987). Note that, as a result of Higman's lemma, we know that $\ggtrsim$ is a well-quasi-ordering if, and only if, $\gtrsim$ is.

EXAMPLE. To prove termination of System (9), we use the *simple path ordering* of Plaisted (1978*a*). Terms are mapped into multisets of sequences of function symbols; sequences are compared in the *monadic path ordering* $>_{mpo}$. In this ordering, sequences are compared left-to-right. At each step, any function symbol (or constant) less than or equal to the current one in the other sequence is discarded. Whichever sequence becomes a proper subsequence of the other (or is finished first) is smaller. The monotonic termination function used for the simple path ordering is

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

where a *path* is a sequence of function symbols, starting with the root symbol f_1, and taking subterms until an innermost, constant symbol f_k is reached. For the function symbol ordering, we take D to be greater than all else.† For example, consider the term

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

or with the D's numbered for expository purposes,

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

It has three paths:

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

Applying the rule

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

to t yields

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

† Gorn (1973) uses a "stepped" lexicographic ordering (under which longer sequences are always larger) to prove termination of differentiation, but without using multisets, that proof applies only when D's are *not* nested.

(with the labelling 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

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

□

This multiset ordering $\ggtrsim$ is "incremental", in the sense that enlarging the quasi-ordering $\gtrsim$ always enlarges the induced ordering $\ggtrsim$. Furthermore, as has been shown in Jouannaud & Lescanne (1982), no other incremental, induced ordering of multisets contains the multiset ordering $\gg$ for all $\succ$. For other such orderings on multisets, see Martin (1986). When $\succ$ is a total ordering, one may determine whether $M \gg M'$ by first sorting the elements of both M and M' in non-ascending order (with respect to the relation $\succ$) and then comparing the two sorted sequences lexicographically.† 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 $\{3, 3, 4, 0\} \gg \{3, 2, 1, 2, 0, 4\}$. Jouannaud & Lescanne (1982) describe one implementation of multiset orderings for the non-total case.

7. Term Orderings

In this section, we describe well-founded orderings on terms, mostly induced by a given *precedence* ordering $\succ$ (or quasi-ordering $\gtrsim$) on function symbols. These are called *syntactic orderings*. We also describe *semantic orderings*, which are induced by a given ordering $\succ$ (or quasi-ordering $\gtrsim$) on terms. In general, we give definitions of quasi-orderings on terms, with the intention of also defining the partial orderings obtained by excluding equivalent terms.

A quasi-ordering $\gtrsim$ on a set $\mathscr{F}$ of function symbols induces an embedding relation $\trianglerighteq_{\gtrsim}$ on the set $\mathscr{T}(\mathscr{F})$ of terms in the following manner:

DEFINITION 17. For any quasi-ordering $\gtrsim$ on a set $\mathscr{F}$, the *homeomorphic embedding relation* $\trianglerighteq_{\gtrsim}$ on the set $\mathscr{T}(\mathscr{F})$ of terms over $\mathscr{F}$ is defined recursively as follows: for two terms, s and t,

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

if either

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

or

$$f \gtrsim g \text{ and } s_{i_j} \trianglerighteq_{\gtrsim} t_j \quad \text{for } all\ j = 1, \ldots, n,$$

where $1 \le i_1 < i_2 < \cdots < i_n \le m$.

† This is the ordering $I_\times^+$ in Manna (1968).

Note that in the second case, the sequence of immediate subterms of t is embedded in the immediate subterms of s.

The following is known as the "Tree theorem":

THEOREM 20 (Kruskal, 1954, 1960). *A set $\mathscr{F}$ of function symbols is well-quasi-ordered under a quasi-ordering $\gtrsim$ if, and only if, the set $\mathscr{T}(\mathscr{F})$ of terms over $\mathscr{F}$ is well-quasi-ordered under the embedding relation $\trianglerighteq_{\gtrsim}$.*

The same result (for partial orders) was announced in Tarkowski (1960); it extends the result in Higman (1952) for terms over function symbols of bounded arity. Many of the results we have already cited are based on Theorem 2, which is a special case of this result (the quasi-ordering being equality). The following non-constructive proof, due to Nash-Williams (1963), follows the same pattern as the proof of Higman's lemma (Lemma 4):

PROOF (Nash-Williams, 1963). Suppose the theorem were false. Let the infinite sequence

$$\tilde{t} = t_1, t_2, \cdots$$

of terms be a "minimal counterexample", measured by the size of the t_i. By the minimality hypothesis, the set of proper subterms of the t_i must be well-quasi-ordered, or else there would be a smaller counterexample

$$t_1, t_2, \ldots, t_{l-1}, s_1, s_2, \cdots,$$

where $s_1, s_2, \cdots$ is a counterexample of proper subterms, such that s_1 is a subterm of some t_l and all s_i in the counterexample are subterms of one of $t_l, t_{l+1}, \cdots$. (None of $t_1, t_2, \cdots, t_{l-1}$, can embed in any of $s_1, s_2, \cdots$, since that would mean that t_i also embeds in some t_j, $i < l \le j$.)

Since the set $\mathscr{F}$ of function symbols is well-quasi-ordered by $\gtrsim$, there must exist an infinite subsequence $\tilde{r}$ of $\tilde{t}$, the root (outermost) symbols of which constitute a quasi-ascending chain under $\lesssim$. (Recall that any infinite sequence of elements of a well-quasi-ordered set must contain an infinite chain of quasi-ascending elements.) Since the set of proper subterms is well-quasi-ordered, it follows by Higman's lemma that the set of finite sequences consisting of the immediate subterms of the elements in r is also well-quasi-ordered. But then there would have to be an embedding in $\tilde{t}$ itself, in which case it would not be a counterexample. □

A stronger notion than well-quasi-ordering, namely *better-quasi-ordering* (Nash-Williams, 1965), is exploited in Nash-Williams (1965) and Laver (1978) for classes of infinite trees. (A quasi-ordered set Q is said to be better-quasi-ordered if—in some sense—the transfinite closure of Q under the power-set construction is well-quasi-ordered.) Stronger results on trees can be obtained by limiting the contexts in which an embedding may occur; see Ehrenfeucht *et al.* (1983) and Haussler (1985) for finite sequences, Puel (1985) for fixed-arity terms, and Friedman (1982), Simpson (1985), Okada & Takeuti (1986), and Okada (1986*a*) for finite terms (trees).†

The Tree theorem plays an important role in the proof of well-foundedness of the following ordering:

† A weaker form of "embedding" (allowing edges to map into non-disjoint paths), and correspondingly weaker results, appear as an exercise in Knuth (1973, p. 385), where it was suggested that embedding "may be used to prove that certain algorithms must terminate". The analogue of the Tree Theorem for finite graphs with this weaker embedding has been conjectured to hold in Nash-Williams (1965); it does not hold for homeomorphic embedding (Kruskal, 1954).

DEFINITION 18 (Dershowitz, 1982*a*). Let $\gtrsim$ be a quasi-ordering on a set $\mathscr{F}$ of function symbols. The *recursive path ordering* $\gtrsim_{rpo}$ on the set $\mathscr{T}(\mathscr{F})$ of terms over $\mathscr{F}$ is defined recursively as follows:

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

if

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

or

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

or

$$f \approx g \text{ and } \{s_1, \ldots, s_m\} \gtrsim\!\!\gtrsim_{rpo} \{t_1, \ldots, t_n\},$$

where $\gtrsim\!\!\gtrsim_{rpo}$ is the multiset ordering induced by $\gtrsim_{rpo}$.

Two terms are equivalent under $\approx_{rpo}$ if they are the same up to equivalent function symbols and *permutative congruence* (permutations of subterms), in which case they fall completely under the last case of the definition. The above definition is similar to a characterisation of the *path of subterms ordering* given in Plaisted (1978*b*). The idea is that a term is decreased by replacing a subterm with any number of smaller (recursively) subterms connected by any structure of function symbols smaller (in the precedence ordering) than the root symbol of the replaced subterm.†

To determine, then, if a term s is strictly greater under $\succ_{rpo}$ than a term t, the root symbols of the two terms are compared first. If they are equivalent, then those (immediate) subterms of t that do not have an equivalent counterpart in s must each be smaller (recursively in the term ordering) than some unaccounted for subterm of s. Furthermore, there must be at least one subterm of s for which there is no equivalent subterm of t. If the root symbol of s is greater than that of t, then s must be greater than each subterm of t. In any case, if a subterm of s is greater than or equivalent to t, then s is greater than t. For example, suppose $\times > +$ and $a \approx d$. We have

$$s = a \times (b+c) >_{rpo} (a \times b) + (c \times d) = t$$

under the corresponding recursive path ordering $\gtrsim_{rpo}$ by the following line of reasoning:

$$s >_{rpo} t \text{ since } \times > + \text{ and } s >_{rpo} a \times b,\ c \times d$$

$$s >_{rpo} a \times b \text{ since } a = a \text{ and } b+c >_{rpo} b$$

$$b+c >_{rpo} b \text{ since } b \gtrsim_{rpo} b$$

$$s >_{rpo} c \times d \text{ since } a \approx d \text{ and } b+c >_{rpo} c$$

$$b+c >_{rpo} c \text{ since } c \gtrsim_{rpo} c.$$

It is easy to see that the recursive path ordering is monotonic and satisfies the subterm and deletion properties of simplification orderings. (It is harder to show that it is a transitive relation.) Hence:

THEOREM 21 (Dershowitz, 1982*a*). *For any quasi-ordering $\gtrsim$ on a set $\mathscr{F}$ of function symbols, the recursive path ordering $\succ_{rpo}$ on the set $\mathscr{T}(\mathscr{F})$ of terms is a simplification ordering.*

† Thus, this ordering addresses the problem posed in Lévy (1980).

Using the recursive path ordering to prove the termination of rewrite systems generalises the (exponential interpretation) method in Iturriaga (1967).†

EXAMPLE. We can use a recursive path ordering to prove termination of System (1). Let the function symbols be ordered by $- > \times > +$. Since this is a simplification ordering on terms, by Theorem 8, we need only show that

$$--\alpha >_{rpo} \alpha$$
$$-(\alpha+\beta) >_{rpo} -\alpha \times -\beta$$
$$-(\alpha \times \beta) >_{rpo} -\alpha + -\beta$$
$$\alpha \times (\beta+\gamma) >_{rpo} (\alpha \times \beta)+(\alpha \times \gamma)$$
$$(\beta+\gamma) \times \alpha >_{rpo} (\beta \times \alpha)+(\gamma \times \alpha)$$

for any terms α, β, and γ. The first inequality follows, for any α, from the subterm property of simplification orderings. By the definition of the recursive path ordering, to show that $-(\alpha+\beta) >_{rpo} (-\alpha) \times (-\beta)$ when $- > \times$, we must show that $-(\alpha+\beta) >_{rpo} -\alpha$, and $-(\alpha+\beta) >_{rpo} -\beta$. Now, since the root symbols 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 property. Thus, the second inequality holds. By an analogous argument, the third inequality also holds. For the fourth inequality, since $\times > +$, 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 element $\beta+\gamma$ is greater than both β and γ with which it is replaced. The same argument holds for the last inequality □

The *recursive decomposition ordering* $\gtrsim_{rdo}$ (defined in Lescanne (1982) and Plaisted (1979) for the case when the precedence $\gtrsim$ is total) "preprocesses" terms in an attempt to improve the efficiency of computing precedence-based orderings. Suppose $\gtrsim$ is total, and let $\hat{t}$ denote the term $t = g(\hat{t}_{j_1}, \ldots, t_n)$ with all subterms preprocessed and sorted according to $\gtrsim_{rdo}$, i.e. $\hat{t} = g(\hat{t}_{j_1}, \ldots, \hat{t}_{j_n})$, where $\hat{t}_{j_1} \gtrsim_{rdo} \cdots \gtrsim_{rdo} \hat{t}_{j_n}$ and $(j_1, \ldots, j_n)$ is a permutation of $(1, \ldots, n)$. Consider two preprocessed terms $\hat{s} = u[f(s_1, \ldots, s_m)]$ and $\hat{t} = v[g(t_1, \ldots, t_n)]$, where f and g are the greatest function symbols 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 $\hat{s}$,

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

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

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

where the three components are compared lexicographically, the symbols 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 symbol $\circ$ is considered to be greater than any term not containing $\circ$; in choosing greatest f and g, circles are ignored. With this definition, the comparison of preprocessed terms is essentially lexicographic. Sorting a list of sorted terms and building the decomposition are believed to be relatively inexpensive (Dershowitz & Zaks, 1981; Lescanne & Steyaert, 1983). The definition of

† The cases where Iturriaga's method works are those for which the function symbols are partially ordered so that the root ("virtual") symbols of the left-hand side of the rules are greater than any other function symbol.

"decomposition" can also be extended to the non-total case (Jouannaud *et al.*, 1982; Rusinowitch, 1987).

For example, suppose $0 > - > \times > + > 1$, $s = -(1 \times (1+0))$, and $t = -1 + -(0 \times 1)$. Their sorted terms are $\hat{s} = -((0+1)$ and $\hat{t} = -(0 \times 1) + -1$. The full decomposition of $\hat{s}$ is

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

that of $\hat{t}$ is

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

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

The recursive decomposition ordering, as well as the *path of subterms ordering* of Plaisted (1978*b*) and *path ordering* of Kapur & Sivakumar (1983), extend the recursive path ordering somewhat when the ordering $\succ$ on function symbols is partial (see Rusinowitch, 1987), but are all equivalent in the total case.† For example, the path of subterms ordering $\succ_{pso}$ makes

$$h(f(a), f(b)) \succ_{pso} h(g(a, b), g(a, b))$$

if $f \succ g$, but the two are incomparable under $\succ_{rpo}$. These orderings are also equivalent for monadic terms; an efficient implementation of the monadic case is given in Lescanne (1981).

We have seen examples of the use of the subterm property to establish inequalities between free terms containing rule variables. In effect, these precedence orderings are lifted to apply to free terms by considering variables as constants, unrelated to any other symbol. This means that a non-variable term t is greater than a variable x if, and only if, t contains an occurrence of x. For the recursive path ordering this idea was illustrated in Dershowitz (1982*a*) and formalised in Huet & Oppen (1980); for the recursive decomposition ordering this is done in Jouannaud *et al.* (1982); for the path of subterms ordering, see Plaisted (1978*b*). As an example, we have $-(\alpha+\beta) >_{rpo} -\alpha \times -\beta$, where α and β 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 $\alpha \geq_{rpo} \alpha$.

These orderings are also incremental. That is, one can start with an empty precedence, and add to it (and hence to the term ordering) only as necessary to satisfy given inequalities between terms. How this may be done with the recursive decomposition ordering is described in Jouannaud *et al.* (1982); for the recursive path ordering, this is done in Ait-Kaci (1985). 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 within the REVE system of Lescanne (1983); details may be found in Choque (1984) and Detlefs & Forgaard (1985).) 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 $- > \times$. Determining if a precedence exists that makes two terms comparable under the recursive path ordering is, however, NP-complete (Krishnamoorthy & Narendran, 1984).‡

EXAMPLE. Consider the system§

$$\begin{aligned} h(f(\alpha), \beta) &\to f(g(\alpha, \beta)) \\ g(\alpha, \beta) &\to h(\alpha, \beta). \end{aligned} \tag{18}$$

† These total orderings are known to suffice for termination proofs of all primitive recursive functions, in the sense that every primitive recursive function can be computed by some system that reduces terms, for some total precedence (Plaisted, 1978*b*).

‡ Cf. the conjecture in Ait-Kaci (1985) that the given choice procedure does not require backtracking.

§ Based on an example in Bergstra & Klop (1983).

The first rule suggests the precedence $h > f$ and $h \gtrsim g$; the second rule, on the other hand, requires g to be greater than h. We can, nevertheless, prove termination by letting $g \approx h > f$ and $g >' h$ and using a lexicographic combination of $\gtrsim_{rpo}$ and $>'_{rpo}$, since $g(\alpha, \beta) \approx_{rpo} h(\alpha, \beta)$. □

Note that, for all these precedence orderings, terms are totally ordered when function symbols are. In that case, one can establish that $h(f(\alpha), f(\beta)) \succ_{rpo} h(g(\alpha, \beta), g(\alpha, \beta))$ (with $f \succ g$) by considering all three possible cases: $\alpha \succ \beta$, $\beta \succ \alpha$, and $\alpha \approx \beta$. Furthermore, given a partial quasi-ordering $\gtrsim$ of function symbols $\mathcal{F}$, the following ordering could be used:

$$\bigcap_{\substack{\gtrsim' \text{ extends } \gtrsim \\ \text{total } \gtrsim'}} \succ'_{rpo},$$

where all possible total extensions $\gtrsim'$ of the given precedence $\gtrsim$ are considered. Only if $t \succ'_{rpo} u$ in the induced recursive path ordering for all total extensions, is t greater than u in this ordering. (See Forgaard, 1984.) For example, $h(f(a), f(b))$ is greater than $h(g(a, b), g(a, b))$ in this ordering as long as f is greater than g in the given precedence. Moreover, $h(f(\alpha), f(\beta))$ is greater than $h(g(\alpha, \beta), g(\alpha, \beta))$ in the lifted version of this ordering.

Another useful ordering is the following lexicographic variant:

DEFINITION 19 (Kamin & Lévy, 1980). Let $\gtrsim$ be a quasi-ordering on a set $\mathcal{F}$ of fixed-arity function symbols. The *lexicographic path ordering* $\gtrsim_{lpo}$ on the set $\mathcal{T}(\mathcal{F})$ of terms over $\mathcal{F}$ is defined recursively as follows:

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

if

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

or

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

or

$$f \approx g \ and\ (s_1, \ldots, s_m) \gtrsim^{*}_{lpo} (t_1, \ldots, t_n) \ and\ s \succ_{lpo} t_j \quad \text{for } all\ j = 2, \ldots, n,$$

where $\gtrsim^{*}_{lpo}$ is the lexicographic ordering induced by $\gtrsim_{lpo}$.

Two terms are equivalent in this quasi-ordering if they are the same up to equivalent function symbols. This ordering generalises ideas in Dershowitz (1982*a*) on treating an operand of a term as its function symbol.†

THEOREM 22 (Kamin & Lévy, 1980). *For any quasi-ordering $\gtrsim$ on a set $\mathcal{F}$ of fixed-arity function symbols, the lexicographic path ordering $\succ_{lpo}$ on the set $\mathcal{T}(\mathcal{F})$ of terms is a simplification ordering.*

EXAMPLE. To prove that System (15) terminates we use the lexicographic path ordering $=_{lpo}$, i.e. the precedence quasi-ordering is just equality. The condition $if(\alpha, \beta, \gamma)$ of the left-hand side is greater (by the subterm property) than the condition α of the right-hand side. Thus, we need only show that the left-hand side is greater than the two operands, $if(\beta, \delta, \varepsilon)$ and $if(\gamma, \delta, \varepsilon)$. Again, $if(\alpha, \beta, \gamma)$ is greater than both β and γ, and is also greater than the remaining operands, δ and ε. □

† The same lexicographic path ordering has been described in Sakai (1984), but note that the recursive and lexicographic path orderings are in fact incomparable. The suggestion in Pettorossi (1981) how one might encode terms so that $t \succ_{lpo} u$ if, and only if, $\tau(t) \succ_{rpo} \tau(u)$ contains errors.

EXAMPLE. The following system (for combinator C) can be shown to terminate with the same lexicographic path ordering $=_{lpo}$:

$$(((C\cdot\alpha)\cdot\beta)\cdot\gamma)\cdot\delta \rightarrow (\alpha\cdot\gamma)\cdot(((\alpha\cdot\beta)\cdot\gamma)\cdot\delta). \tag{19}$$

The left subterm $((C\cdot\alpha)\cdot\beta)\cdot\gamma$ is greater than $\alpha\cdot\gamma$ (because the latter is embedded in the former) and the whole left-hand side is greater than $((\alpha\cdot\beta)\cdot\gamma)\cdot\delta$ (for the same reason).† □

EXAMPLE. The following system (for Ackermann's function) can easily be seen to terminate with a lexicographic path ordering and precedence $a > s$:

$$\begin{aligned} a(0,\beta) &\rightarrow s(\beta)\\ a(s(\alpha),0) &\rightarrow a(\alpha,s(0))\\ a(s(\alpha),s(\beta)) &\rightarrow a(\alpha,a(s(\alpha),\beta)). \end{aligned} \tag{20}$$

The lexicographic aspect of the ordering is needed for the last rule, in particular. □

The above orderings can be combined by allowing some function symbols to have their operands compared lexicographically (left-to-right as above, or in any fixed order), while others are compared using multisets (depending on what is called the "status" of a symbol in Lescanne (1984)). Multiset and lexicographic versions of the path orderings have been implemented in REVE (Lescanne, 1984; Detlefs & Forgaard, 1985), RRL (Kapur & Sivakumar, 1983) and FORMEL (Fages, 1984). In Kamin & Lévy (1980), it is shown that not only is a lexicographic ordering of operands possible, but any mapping $*$ of a partial ordering $\succ$ to a partial ordering $\succ^*$ that satisfies

$$t \succ u \textit{ implies } f(\cdots t \cdots) \succ^* f(\cdots u \cdots)$$

and depends on only a finite number of comparisons under $\succ$ of smaller terms would work as well.

Not only are the recursive and lexicographic path orderings simplification orderings, but as long as the function symbols are well-founded, these orderings are, too:

THEOREM 23 (Dershowitz, 1982*a*). *A quasi-ordering $\gtrsim$ on a set $\mathcal{F}$ of function symbols is well-founded if, and only if, the induced recursive path ordering $\gtrsim_{rpo}$ on the set $\mathcal{T}(\mathcal{F})$ of terms over $\mathcal{F}$ is well-founded.*

THEOREM 24 (Kamin & Lévy, 1980). *A quasi-ordering $\gtrsim$ on a set $\mathcal{F}$ of function symbols is well-founded if, and only if, the induced lexicographic path ordering $\gtrsim_{lpo}$ on the set $\mathcal{T}(\mathcal{F})$ of terms over $\mathcal{F}$ is well-founded, provided that equivalent symbols have the same fixed arity.*

The following proof sketch for both results is typical of proofs of well-foundedness based on well-quasi-orderedness.‡

† This kind of proof is possible whenever a combinator has the *non-ascending property* described in Pettorossi (1978, 1981).

‡ The fact that the path of subterms ordering, recursive path ordering, recursive decomposition ordering, and path ordering are all identical when the precedence is total means that this well-foundedness proof proves them all to be well founded (Dershowitz, 1982*a*; Lescanne, 1982).

PROOF. By Zorn's lemma, any given well-founded precedence $\gtrsim$ may be extended to a total well-founded ordering $\gtrsim$ of function symbols, in which case $\gtrsim$ well-quasi-orders $\mathscr{F}$. By the Tree theorem (Theorem 20), the embedding relation $\trianglerighteq_{\gtrsim}$ well-quasi-orders $\mathscr{T}(\mathscr{F})$. The total ordering $\gtrsim_{rpo}$ ($\gtrsim_{lpo}$) is well-founded, since it can be proved by induction that it is an extension of the well-quasi-ordering $\trianglerighteq_{\gtrsim}$ (with the added requirement of fixed arity for equivalent function symbols in the lexicographic case). Since the recursive (lexicographic) path ordering is incremental, $\gtrsim_{rpo}$ ($\gtrsim_{lpo}$) is an extension of $\gtrsim_{rpo}$ ($\gtrsim_{lpo}$). Hence, the latter is also well-founded. □

In general, if a totally ordered set $(\mathscr{F}, \succ)$ of varyadic function symbols is of order type α, then the recursive path ordering on the set $\mathscr{T}(\mathscr{F})$ of terms is of order type $\phi^{\alpha}(0)$, where $\phi^0(\beta) = \beta$, $\phi^1(\beta) = \varepsilon_\beta$ (the βth epsilon number), and $\phi^{\alpha}(\beta)$ is the βth fixpoint ξ of $\phi^{\mu}(\xi) = \xi$ common to all ϕ^{μ} for ordinals $\mu < \alpha$ (see Feferman, 1968). If one lets arbitrary terms serve as operators (function symbols), then the recursive path ordering on the set of terms constructed from operators built using instances of a single symbol $\bigcirc$ has as its order type the least impredicative ordinal Γ_0 (see Veblen, 1908 and Schutte, 1965). This can be shown with the following order-preserving mapping ψ (of Dershowitz, 1980) from $\mathscr{T}(\{\bigcirc\})$ onto Γ_0 defined by:

$$\psi(\bigcirc) = 0$$
$$\psi(\alpha(\beta_1, \beta_2, \ldots, \beta_n)) = \phi^{\psi(\alpha)}\left(\sum_{i=1}^{n} \omega^{\psi(\beta_i)}\right) + \delta,$$

where $\sum$ is the natural (i.e. commutative) sum of ordinals and δ is 1 if $\psi(\alpha) = 0$, $n = 1$, and $\psi(\beta_1)$ is an epsilon number and is 0 otherwise. (The purpose of δ is to ensure that $\psi(\bigcirc(\beta)) = \omega^{\psi(\beta)} + \delta > \psi(\beta)$ even if $\psi(\beta)$ is an epsilon number.) That this mapping is order-preserving follows from the fact (Feferman, 1968; Weyhrauch, 1978) 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 $\beta > \phi^{\alpha'}(\beta')$. The recursive and lexicographic path orderings are related to Ackermann's ordinal notation, in which $\Gamma_0 = A_2(\{\bigcirc\}, \{\bigcirc\})$ (see Ackermann, 1951). Combining the two orderings into one, as described in Lescanne (1984), yields a more powerful ordering than either alone. (See Okada, 1986*b*.)

The multiset ordering and simple path ordering (described in the previous section) may be thought of as special cases of the recursive path ordering, in which the multiset constructor $\{\cdots\}$ is greater than other function symbols. The *nested multiset ordering*, defined in Dershowitz & Manna (1979), is just a recursive path ordering on all terms constructed from that one varyadic constructor symbol (and a well-founded set of smaller constants). With just that one constructor (and no constants), its order type is ε_0.† Gentzen (1938) used an ordering of this order type to show termination of his "normalisation procedure"; two other interesting examples of ε_0 termination arguments may be found in Kirby & Paris (1982) (see also Kent & Hodgson, 1986).

It is sometimes necessary to transform terms before comparing them in the recursive path ordering. As long as the precedence for the function symbols of transformed terms is well-founded, we know that the recursive path ordering will also be. But the transform τ, which acts as termination function, needs to satisfy the monotonicity condition

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

† That the nested multiset ordering has the properties of simplification orderings was pointed out in Scherlis (1980). For a "constructive" discussion of this ordering, see Paulson (1984).

for Theorem 6 to apply. Depending on the particular τ, this condition may or may not hold. For instance, let $\bar{t}$ denote the *flattened* version of a term t. For example,

$$\overline{f(f(a,b),g(b,g(a,b)))} = f(a,b,g(b,a,b)),$$

with all nested occurrences of the symbols f and g stripped. With precedence $f > g$, we get, on the one hand,

$$\overline{f(a,a)} = f(a,a) >_{rpo} g(a,a) = \overline{g(a,a)},$$

while, on the other, we have

$$\overline{f(f(a,a),a)} = f(a,a,a) <_{rpo} f(g(a,a),a) = \overline{f(g(a,a),a)}.$$

The general use of rewrite systems as termination functions and the formulation of abstract conditions for monotonicity are explored in Bachmair & Dershowitz (1986) and Bellegarde & Lescanne (1987). See section 8.

Rather than transform terms, we can sometimes use the following "semantic" ordering:

Definition 20 (Kamin & Lévy, 1980; Plaisted, 1979). Let $\gtrsim$ be a quasi-ordering on a set $\mathcal{T}$ of terms. The *semantic path ordering* $\gtrsim_{spo}$ 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

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

or

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

or

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

where $\ggtrsim_{spo}$ is the multiset ordering induced by $\gtrsim_{spo}$.

This ordering is not in general monotonic, but we do have the following result, based on Theorem 11:

Theorem 25 (Kamin & Lévy, 1980). *A rewrite system $\mathcal{R}$ over a set $\mathcal{T}$ of terms is terminating if, and only if, there exists a well-founded quasi-ordering $\gtrsim$ on $\mathcal{T}$ such that*

$$t \Rightarrow_{\mathcal{R}} u \text{ implies } f(\cdots t \cdots) \gtrsim f(\cdots u \cdots)$$

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

$$l \succ_{spo} r$$

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

Note the condition linking $\Rightarrow$ to $\gtrsim$; it helps ensure that

$$t \succ_{spo} u \text{ and } t \Rightarrow_{\mathcal{R}} u \text{ imply } f(\cdots t \cdots) \succ_{spo} f(\cdots u \cdots).$$

This condition is trivially satisfied by any quasi-ordering $\gtrsim$ that simply compares root symbols; in that case, the ordering $\succ_{spo}$ specialises to the recursive path ordering $\succ_{rpo}$. One can similarly define a semantic version of the lexicographic path ordering, akin to the Knuth–Bendix ordering, defined in section 5. Other variants of this ordering may be possible; for details, see Kamin & Lévy (1980).

Example. No simplification ordering (and, in particular, no polynomial interpretation) can be used to prove termination of the following system over $\mathcal{T}(\{+, \times, 0, 1\})$:

$$
\begin{aligned}
\alpha \times (\beta+1) &\rightarrow (\alpha \times (\beta+(1 \times 0)))+\alpha \\
\alpha \times 1 &\rightarrow \alpha \\
\alpha+0 &\rightarrow \alpha \\
\alpha \times 0 &\rightarrow 0.
\end{aligned} \tag{21}
$$

We can, however, use a semantic path ordering in which $\gtrsim$ makes products greater than other terms, and compares products based on the "natural" interpretation τ of their right multiplicands, where

$$
\tau(\alpha \times \beta) = \tau(\alpha)\cdot\tau(\beta) \qquad \tau(0)=0
$$
$$
\tau(\alpha+\beta) = \tau(\alpha)+\tau(\beta) \qquad \tau(1)=1.
$$

Since $\tau(\beta+1) > \tau(\beta+(1\times 0))$, the first rule is reducing. Since applying a rule cannot affect the value—under this interpretation—of any multiplicand, the system is terminating. □

EXAMPLE. The lexicographic path ordering cannot directly handle the following system:†

$$
\begin{aligned}
(\alpha\cdot\beta)\cdot\gamma &\rightarrow \alpha\cdot(\beta\cdot\gamma) \\
(\alpha+\beta)\cdot\gamma &\rightarrow (\alpha\cdot\gamma)+(\beta\cdot\gamma) \\
\gamma\cdot(\alpha+f(\beta)) &\rightarrow g(\gamma,\beta)\cdot(\alpha+a).
\end{aligned} \tag{22}
$$

But termination can be proved using a semantic path ordering $\succ_{spo}$, with any term of the form $\gamma\cdot(\alpha+f(\beta))$ greater under $\succ$ than any other product; any product greater than any other term; and products treated lexicographically (left-to-right). Note that no rule application changes the value (under $\succ$) of its superterms. Examining all cases shows that each rule is a reduction, whatever the value of its products. □

8. Combined Systems

In this section we consider the termination of combinations of systems. First we consider "rewriting modulo a congruence". By a *congruence system* $\mathscr{E}$, we mean a (finite or infinite) set of rules, such that if $l \rightarrow r$ is a rule in $\mathscr{E}$, then $r \Rightarrow^*_{\mathscr{E}} l$ must also be a derivation for $\mathscr{E}$. (We shall assume that l and r have the same set of variables occurring within them.) If $\mathscr{R}$ is a terminating system and $\mathscr{E}$ is a (non-terminating) congruence system, we need methods of proving that the combined system $\mathscr{R}\cup\mathscr{E}$ does not allow for a derivation with an infinite number of applications of rules in $\mathscr{R}$. We also consider conditions for termination and normalisation of a system $\mathscr{R}\cup\mathscr{S}$, containing all the rules of two terminating, or normalising, systems, $\mathscr{R}$ and $\mathscr{S}$.

Proving termination of rewriting modulo a congruence is, in practice, considerably more difficult than for plain rewrite systems. Given a congruence system $\mathscr{E}$, we define the *rewrite relation* $\mathscr{R}/\mathscr{E}$ over a set of terms $\mathscr{T}$ as follows: $t \Rightarrow_{\mathscr{R}/\mathscr{E}} u$ if $t \Rightarrow^*_{\mathscr{E}} v \Rightarrow_{\mathscr{R}} w \Rightarrow^*_{\mathscr{E}} u$, for some terms v and w in $\mathscr{T}$. The question then is: for given $\mathscr{R}$ and $\mathscr{E}$, does there exist an infinite sequence of terms $t_i \in \mathscr{T}$ such that $t_1 \Rightarrow_{\mathscr{R}/\mathscr{E}} t_2 \Rightarrow_{\mathscr{R}/\mathscr{E}} \cdots$? When $\mathscr{E}$ is empty, this is simply the question of termination of $\mathscr{R}$. System (0) is an example of rewriting modulo associativity, the infix comma obeying the axiom $(\alpha,(\beta,\gamma)) = ((\alpha,\beta),\gamma)$.

EXAMPLE. Let I denote the following congruence system (idempotence):

$$
\begin{aligned}
\alpha+\alpha &\rightarrow \alpha \\
\alpha &\rightarrow \alpha+\alpha.
\end{aligned}
$$

† Abstracted from a problematic example in Bellegarde (1984).

For any non-empty $\mathcal{R}$, the relation $\mathcal{R}/I$ cannot be terminating, since there would be an infinite derivation $l \Rightarrow_I l+l \Rightarrow_{\mathcal{R}} l+r \Rightarrow_I (l+l)+r \Rightarrow_{\mathcal{R}} \cdots$ for each $l \rightarrow r \in \mathcal{R}$. □

The congruence system AC, consisting of the associative and commutative axioms,

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

for each associative-commutative function symbol f, is of particular importance. Since addition and multiplication are themselves associative and commutative, monotonic polynomial interpretations can be helpful. To provide an ordering for rewriting modulo these axioms, an interpretation should preserve its value under associativity and commutativity. Then if $\tau(l) > \tau(r)$ for each rule, it would follow that $\tau(t) > \tau(u)$ whenever $t \Rightarrow_{\mathcal{R}\,AC} u$. 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 preserve value under associativity. In general, polynomials for associative-commutative symbols must either be of the quadratic form

$$F(x, y) = axy + bx + by + b(b-1)/a \quad (a \neq 0)$$

or of the linear form $F(x, y) = x+y+c$ (Ben-Cherifa & Lescanne, 1986; cf. Lankford, 1979). Since the degree of value-preserving polynomials is bounded (by two), it is decidable (see section 4) whether there exists a polynomial interpretation τ that preserves value under associativity and commutativity, and which (eventually) is monotonic, has the subterm property, and decreases for each rule. Though decidable, this is a very restricted class of simplification orderings.

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

$$\begin{aligned} \alpha \times 1 &\rightarrow \alpha \\ \alpha \times 0 &\rightarrow 0 \\ \alpha \times \alpha &\rightarrow \alpha \\ \alpha + 0 &\rightarrow \alpha \\ \alpha + \alpha &\rightarrow 0 \\ (\alpha+\beta) \times \gamma &\rightarrow (\alpha \times \gamma) + (\beta \times \gamma). \end{aligned} \tag{23}$$

One can use the following polynomial interpretation to prove termination of this system modulo AC:

$$\tau(\alpha+\beta) = \tau(\alpha) + \tau(\beta) + 1$$
$$\tau(\alpha \times \beta) = \tau(\alpha) \cdot \tau(\beta).$$

With constants assigned a sufficiently large value (>1), all the necessary conditions are fulfilled. □

Two terms u and v are equal in AC if, and only if, $\bar{u}$ and $\bar{v}$ are the same (up to permutation of arguments of associative-commutative symbols), where $\bar{t}$ is the term t with all associative-commutative symbols "flattened", and where the order of operands of f is not significant. It is natural, therefore, to consider orderings on the set $\bar{\mathcal{T}} = \{\bar{t} : t \in \mathcal{T}\}$ of

† Used in Hsiang & Dershowitz (1983).

flattened terms. For example, instead of the non-preserving interpretation $F(x, y) = xy + 1$ for an associative-commutative symbol f, one could interpret a varyadic f with $F(x_1, \ldots, x_n) = x_1 \cdot \cdots \cdot x_n + 1$, when flattening results in n operands. To be useful, though, any ordering on flattened terms must satisfy the conditions of the following theorem:

THEOREM 26 (Dershowitz *et al.*, 1983). *Let $\mathscr{R}$ be a rewrite system over some set $\mathscr{T}$ of terms and $\mathscr{F}$ a set of associative-commutative symbols. The rewrite relation $\mathscr{R}/AC$ is terminating if, and only if, there exists a well-founded ordering $\succ$ on flattened terms in $\bar{\mathscr{T}}$ such that*

$$\bar{l} \succ \bar{r}$$

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

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

for each rule $l \rightarrow r$ in $\mathscr{R}$ whose left-hand side l or right-hand side r has associative-commutative root symbol $f \in \mathscr{F}$ or for which r is just a variable and for any substitution of terms in $\mathscr{T}$ for the variables of the rule and for ξ (a new variable, otherwise not occurring in the rule), and such that

$$t \Rightarrow_{\mathscr{R}/AC} u \text{ and } \bar{t} \succ \bar{u} \text{ imply } f(\cdots \bar{t} \cdots) \succ f(\cdots \bar{u} \cdots)$$

for all terms t and u in $\mathscr{T}$ and $f(\cdots \bar{t} \cdots)$ and $f(\cdots \bar{u} \cdots)$ in $\bar{\mathscr{T}}$.

EXAMPLE. The following system terminates modulo associativity and commutativity of $\times$ and $+$:

$$\alpha \times (\beta + \gamma) \rightarrow (\alpha \times \beta) + (\alpha \times \gamma). \tag{24}$$

To prove termination, we use a semantic path ordering on flattened terms that takes the arity of $\times$ into account. That is, the more multiplicands, the greater the product, and all products are greater than all sums. The left-hand side l is greater than the right-hand side r in this ordering. Furthermore, under this ordering, $\overline{l \times \xi}$ and $\overline{l + \xi}$ are greater than $\overline{r \times \xi}$ and $\overline{r + \xi}$, respectively. Since all the conditions of the theorem are satisfied, termination is assured. □

As we saw in the previous section, flattening alone makes the recursive path ordering non-monotonic. To overcome this, the ordering has been adapted (in a series of papers: Dershowitz *et al.* (1983), Plaisted (1983), Bachmair (1984), Bachmair & Plaisted (1985), Bachmair & Dershowitz (1986), Gnaedig (1986), and Gnaedig & Lescanne (1986)) to handle associative-commutative symbols by flattening and transforming terms (distributing large function symbols over small ones) before comparing them.

EXAMPLE. Consider System (23) modulo associativity and commutativity of $\times$ and $+$ and a recursive path ordering with precedence $\times > +$. Flattening does not satisfy the conditions in the above theorem, since $(\alpha + \beta) \times \gamma \times \xi$ is not greater under $>_{rpo}$ than $((\alpha \times \gamma) + (\beta \times \gamma)) \times \xi$. The system can be shown terminating, however, by distributing *times* over *plus* before comparing; see Bachmair & Dershowitz (1986). □

We turn now to consider combinations of two terminating rewrite relations:

DEFINITION 21. A rewrite relation $\mathscr{R}$ *commutes over* another relation $\mathscr{S}$ if whenever $t \Rightarrow_{\mathscr{S}} u \Rightarrow_{\mathscr{R}} v$, there is an alternative derivation of the form $t \Rightarrow_{\mathscr{R}} w \Rightarrow_{\mathscr{S}} v$, for some w. A rewrite relation $\mathscr{R}$ *quasi-commutes* over another relation $\mathscr{S}$ if whenever $t \Rightarrow_{\mathscr{S}} u \Rightarrow_{\mathscr{R}} v$, there is an alternative derivation of the form $t \Rightarrow_{\mathscr{R}} w \Rightarrow^{*}_{\mathscr{R} \cup \mathscr{S}} v$, for some w.

Other investigations of properties of commuting relations include Newman (1942), Rosen (1973), O'Donnell (1977), Huet & Lévy (1979), Huet (1980), and Raoult & Vuillemin (1980). With this notion, we can reduce termination of the union of $\mathscr{R}$ and $\mathscr{S}$ to termination of each:

THEOREM 27 (Bachmair & Dershowitz, 1986). *Let $\mathscr{R}$ and $\mathscr{S}$ be two rewrite relations over some set $\mathscr{T}$ of terms such that $\mathscr{R}$ quasi-commutes over $\mathscr{S}$. Then, the combined rewrite relation $\mathscr{R} \cup \mathscr{S}$ is terminating if, and only if, $\mathscr{R}$ and $\mathscr{S}$ both are.*

This theorem applies equally well to rewriting modulo a congruence and to ordinary rewriting.

EXAMPLE. Let AC contain associative and commutative axioms for both $\times$ and $+$. Let R be

$$(\alpha+\beta)\times\gamma \rightarrow (\alpha\times\gamma)+(\beta\times\gamma) \tag{23a}$$

and S be

$$\begin{aligned} \alpha\times 1 &\rightarrow \alpha \\ \alpha\times 0 &\rightarrow 0 \\ \alpha\times\alpha &\rightarrow \alpha \\ \alpha+0 &\rightarrow \alpha \\ \alpha+\alpha &\rightarrow 0. \end{aligned} \tag{23b}$$

The rewrite relation R/AC quasi-commutes over S (and, therefore, over S/AC, too). If, say,

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

then by the same token

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

The rewrite relation S/AC terminates, since it shortens terms, and we saw above that R/AC (System (24)) terminates. Hence, System (23) modulo AC also terminates. □

For rewriting modulo a congruence, we also have the following related result:

THEOREM 28 (Jouannaud & Muñoz, 1984). *If a rewrite relation $\mathscr{R}$ quasi-commutes over a congruence relation $\mathscr{E}$, then the rewrite relation $\mathscr{R}/\mathscr{E}$ is terminating if, and only if, $\mathscr{R}$ is terminating.*

Some suggestions of how non-commuting $\mathscr{R}$ and $\mathscr{E}$ might be handled are given in Jouannaud & Muñoz (1984).

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

DEFINITION 22. A rewrite 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. A system is *linear* if it is both left- and right-linear.

DEFINITION 23. A term u *overlaps* a term t if u can be unified with (i.e. made the same as) some non-variable subterm s of t by substituting terms for the variables in each (with the variables of t and u considered disjoint). 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 $\mathscr{R}$ is said to be *non-overlapping* if there is no overlap among the left-hand sides of $\mathscr{R}$, that is, no left-hand side l_i overlaps a different left-hand side l_j and no left-hand side l_i overlaps a non-variable proper subterm of itself.

Non-overlapping systems are called "non-ambiguous" in Huet & Lévy (1979); they have no "critical pairs" in the sense of Knuth & Bendix (1970).

EXAMPLE. The linear system

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

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

$$\alpha\times(\beta+\gamma)\to(\alpha\times\beta)+(\alpha\times\gamma) \tag{24}$$

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

$$(\alpha\times\beta)+(\alpha\times\gamma)\to\alpha\times(\beta+\gamma) \tag{25}$$

is right-linear, but not left-linear; both are non-overlapping. All three are terminating. □

In Raoult & Vuillemin (1980) (and in Rosen, 1973, for the ground case), it has been shown that if $\mathscr{R}$ and $\mathscr{S}$ are two left-linear systems and there is no overlap between their left-hand sides, then whenever $t\Leftarrow_{\mathscr{S}}u\Rightarrow_{\mathscr{R}}v$, it is also the case that $t\Rightarrow^{*}_{\mathscr{R}}w\Leftarrow^{*}_{\mathscr{S}}v$, for some w. Using that idea, we obtain the following corollary to Theorem 27:

THEOREM 29 (Bachmair & Dershowitz, 1986). *Let $\mathscr{R}$ and $\mathscr{S}$ be two rewrite systems over some set $\mathscr{T}$ of terms. Suppose that $\mathscr{R}$ is left-linear, $\mathscr{S}$ is right-linear, and there is no overlap between left-hand sides of $\mathscr{R}$ and right-hand sides of $\mathscr{S}$. Then, the combined system $\mathscr{R}\cup\mathscr{S}$ is terminating if, and only if, $\mathscr{R}$ and $\mathscr{S}$ both are.*

This generalises the case exploited in Bidoit (1981).

EXAMPLE. The systems

$$\left.\begin{aligned}\alpha\times(\beta+\gamma)&\to(\alpha\times\beta)+(\alpha\times\gamma)\\(\beta+\gamma)\times\alpha&\to(\beta\times\alpha)+(\gamma\times\alpha)\\\alpha\times1&\to\alpha\\1\times\alpha&\to\alpha\end{aligned}\right. \tag{16a}$$

and

$$\left.\begin{aligned}\alpha\times0&\to0\\0\times\alpha&\to0\end{aligned}\right. \tag{16b}$$

each terminate. The first is left-linear; the second has a constant on the right that does not appear in the first. Therefore, their union terminates. □

Each of the three requirements of the above theorem is necessary, as evidenced by the following examples of non-terminating systems:

Example. The system

$$\begin{aligned} b &\to a \\ f(\alpha, \alpha) &\to f(a, b) \end{aligned} \tag{26}$$

has an infinite derivation $f(a, a) \Rightarrow f(a, b) \Rightarrow f(a, a) \Rightarrow \cdots$. Though each rule terminates, the first rule is linear, and there is no forbidden overlap between the right-hand side of the first and the left-hand side of the second, termination does not follow, since the second is not left-linear (although it is right-linear). □

Example. The system

$$\begin{aligned} f(a, b, \alpha) &\to f(\alpha, \alpha, \alpha) \\ b &\to a \end{aligned} \tag{27}$$

has an infinite derivation $f(a, b, b) \Rightarrow f(b, b, b) \Rightarrow f(a, b, b) \Rightarrow \cdots$. Each rule terminates, the second is left-linear, and there is no forbidden overlap, but the first is not right-linear (although it is left-linear). □

Example. The system

$$\begin{aligned} b &\to g(a) \\ a &\to g(b) \end{aligned} \tag{28}$$

has an infinite derivation $b \Rightarrow g(a) \Rightarrow g(g(b)) \Rightarrow \cdots$. Each rule terminates and both are linear, but (in either order) there is a forbidden overlap. □

Note that, by definition, a variable right-hand side overlaps any left-hand side.†

Example. Consider System (14), divided into two parts:

$$f(a, b, \alpha) \to f(\alpha, \alpha, \alpha) \tag{14a}$$

$$\begin{aligned} g(\alpha, \beta) &\to \alpha \\ g(\alpha, \beta) &\to \beta. \end{aligned} \tag{14b}$$

Though (14a) is left-linear, (14b) is right-linear, and each part alone is terminating, the combined system is non-terminating. This is because the right-hand sides of (14b) overlap the left-hand side of (14a). □

For related results on rewriting modulo a congruence, see Bachmair & Dershowitz (1986). The use of commutation properties in establishing "fair termination" is investigated in Porat & Francez (1986).

Recall that a system is normalising if every term rewrites to an irreducible term. To prove that a system is normalising, one can choose a particular evaluation strategy and only show that any term to which rules are applied is (eventually) reduced in some well-founded ordering for those rewrites allowed by the chosen strategy. (The ordering need not be monotonic.)

† The definition of "overlap" in Dershowitz (1981) and the examples in Bachmair & Dershowitz (1986) are wrong on this account.

EXAMPLE. The following system† (cf. System (1)) over $\mathscr{T}(\{\times, +, -, 0, 1\})$ does not always terminate, but is normalising and its irreducible terms are in disjunctive normal form:

$$\begin{aligned} --\alpha &\to \alpha \\ -(\alpha+\beta) &\to ---\alpha \times ---\beta \\ -(\alpha \times \beta) &\to ---\alpha + ---\beta \\ \alpha \times (\beta+\gamma) &\to (\alpha \times \beta)+(\alpha \times \gamma) \\ (\beta+\gamma) \times \alpha &\to (\beta \times \alpha)+(\gamma \times \alpha). \end{aligned} \tag{29}$$

To see that it does not terminate, consider the (embedding) derivation

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

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), which is terminating. Hence a normal form always exists. □

To prove that the union of two normalising systems $\mathscr{R}$ and $\mathscr{S}$ is also normalising, one can choose to first compute an $\mathscr{R}$-normal form and only then apply $\mathscr{S}$. Then, if one can show that applying $\mathscr{S}$ to an $\mathscr{R}$-normal form results in an $\mathscr{R}$-normal form, it would follow that $\mathscr{R} \cup \mathscr{S}$ is normalising.

EXAMPLE. The non-terminating System (29) is normalising by the following line of reasoning: the first three rules alone are terminating (they are System (17)), as are the last two (they are part of System (1)). Since the first three rules eliminate all negations of non-constant terms and the two distributivity rules cannot introduce other negations, the whole system is normalising. □

9. Restricted Systems

In this section, we consider how linearity and non-overlapping of rules make it possible to restrict the derivations that must be considered when proving termination or non-termination of a rewrite system. *Non-deterministic Markov systems* (i.e. *semi-Thue systems*) are rewrite systems over words, consisting of monadic (hence, linear) terms. Nevertheless:

THEOREM 30 (Huet & Lankford, 1978). *It is undecidable whether a (finite) rewrite system is terminating, even if it is non-overlapping and contains only monadic function symbols and constants.*

† From Dershowitz (1982*a*).

By the same token, we have:

THEOREM 31. *It is undecidable whether a (finite) rewrite system is normalising, even if it is non-overlapping and contains only monadic function symbols and constants.*

And:

THEOREM 32 (Guttag *et al.*, 1983). *It is undecidable whether a (finite) rewrite system is quasi-terminating, even if it is non-overlapping and contains only monadic function symbols and constants.*

For the purposes of this section, we need to distinguish between constants and term variables appearing in derivations. If a variable appears, then the same rule can be applied when the variable is replaced by any term. That is, if $u[\alpha, \ldots, \alpha] \Rightarrow v[\alpha, \ldots, \alpha]$, for some contexts u and v occurrences within them of a variable α, then $u[t, \ldots, t] \Rightarrow v[t, \ldots, t]$ for any $t \in \mathcal{T}$. Two derivations are considered equal if they can be obtained one from the other by only renaming variables.

We are interested in the following restricted class of derivations:

DEFINITION 24 (Lankford & Musser, 1978). The set of *forward closures* for a given rewrite system $\mathcal{R}$ may be inductively defined as follows: every rule $l \rightarrow r$ in $\mathcal{R}$ is a forward closure $l \Rightarrow r$. Let the derivations

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

and

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

be two forward closures. If c_m has a non-variable subterm s within some context u such that s unifies with d_1 via most general unifier σ, 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.

This definition is related to the *narrowing* process, as defined in Slagle (1974) and Hullot (1980). (Forward closures are referred to as "chains" in Lankford & Musser (1978) and Dershowitz (1981).)

EXAMPLE. Consider the terminating system

$$\begin{aligned} --\alpha &\rightarrow \alpha \\ -(\alpha+\beta) &\rightarrow -\alpha + -\beta. \end{aligned} \tag{30}$$

The derivation

$$\begin{aligned} -((\alpha + -\beta) + -\gamma) &\Rightarrow -(\alpha + -\beta) + --\gamma \Rightarrow (-\alpha + --\beta) + --\gamma \\ &\Rightarrow (-\alpha+\beta) + --\gamma \Rightarrow (-\alpha+\beta)+\gamma \end{aligned}$$

is a forward closure for that system. All of its forward closures are of similar form; they begin with the negation of a sum of either negated or unnegated variables and end with such a sum. □

To determine if a right-linear system terminates, one need only consider its forward closures:

THEOREM 33 (Dershowitz, 1981). *A right-linear rewrite system is terminating if, and only if, it has no infinite forward closures.*

This is a stronger result than the related one—for quasi-terminating systems—given in Guttag *et al.* (1983).

EXAMPLE. The self-embedding rewrite system

$$f(g(\alpha)) \rightarrow f(h(g(\alpha))) \tag{31}$$

is right-linear and has only one forward closure:

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

Since this forward closure is finite, the system must terminate. Recall that, by Theorem 9, no total monotonic ordering can prove termination of this system. □

EXAMPLE. The forward closures of

$$f(g(\alpha)) \rightarrow g(g(f(\alpha))) \tag{32}$$

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 above theorem, it must terminate for all inputs. □

EXAMPLE. The forward closures of

$$f(g(\alpha)) \rightarrow g(g(f(f(\alpha)))) \tag{33}$$

include

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

Thus, the system does not terminate. □

EXAMPLE. Consider the linear System (30). Termination of its closures, and hence of the system, can easily be proved using a multiset ordering on the sizes of all negated subterms. □

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

EXAMPLE. The non-right-linear and overlapping system

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

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

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

□

For left-linear systems, we know the following:

THEOREM 34 (Dershowitz, 1981). *A non-overlapping left-linear rewrite system is terminating if, and only if, it has no infinite forward closures.*

EXAMPLE. None of the forward closures of the non-overlapping left-linear System (9) have nested D symbols. (This can be shown by induction.) Thus, the finiteness of those forward closures—and consequently the termination of the system—can be proved by considering the multiset of the sizes of the arguments of the D's in a term. Any rule application is a reduction under the multiset ordering $\gg$ induced by the natural ordering $>$ of positive integers. □

In Guttag *et al.* (1983), the notion of closure is expanded so that derivations are also extended if the last term c_m of a closure $c_1 \Rightarrow \cdots \Rightarrow c_m$ unifies with a non-variable subterm of the first term d_1 of a closure $d_1 \Rightarrow \cdots \Rightarrow d_n$, as well as if a non-variable subterm of c_m unifies with d_1. These restricted derivations are called *overlap closures*. It is unknown if there are non-terminating systems that do not have an infinite overlap closure.†

EXAMPLE. Left-linear System (34), though it has no infinite forward closure, does have the following cycling overlap closure:

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

THEOREM 35 (Guttag *et al.*, 1983). *A quasi-terminating left-linear rewrite system is terminating if, and only if, it has no cycling overlap closures.*

An advantage of using closures is that non-termination may be more easily detectable, as the next theorem will demonstrate. First, we extend the definition of "looping":

DEFINITION 25. A derivation $t_1 \Rightarrow t_2 \Rightarrow \cdots \Rightarrow t_j \Rightarrow \cdots \Rightarrow t_k \Rightarrow \cdots$ *loops* if some t_k has a subterm that is an *instance* of t_j (with variables of the two terms considered distinct) for some $j < k$.

Looping closures are indicative of a non-terminating system. Moreover:

THEOREM 36 (Dershowitz, 1981). *A right-linear rewrite system with only a finite number of forward closures (beginning with different terms) is terminating if, and only if, it has no looping forward closures.*

(Recall that two closures are considered equal if they differ only in the names of their variables.) A similar result is given in Dershowitz (1981) for non-overlapping left-linear systems.

EXAMPLE. The non-terminating right-linear System (33) has a looping forward closure

$$f(g(g(\alpha))) \Rightarrow g(g(f(f(g(\alpha))))) \Rightarrow g(g(f(g(g(f(f(\alpha))))))) \Rightarrow \cdots .$$ □

† The idea of decomposing proofs of termination, by looking at overlappings between rules (but ignoring the difficulties caused by non-left-linear rules), appears in Pettorossi (1981).

Bergstra, J. A., Klop, J. W. (1983). *A process algebra for the operational semantics of static data flow networks.* Preprint IW 222/83, Mathematisch Centrum, Amsterdam, The Netherlands.

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

Bidoit, M. (1981). *Une méthode de présentation de types abstraits: Applications.* Thèse de Troisième Cycle, Université de Paris-Sud, Orsay, France.

Brown, T. C., Jr. (1975). *A structured design-method for specialized proof procedures.* Ph.D. thesis, California Institute of Technology, Pasadena, CA.

Choque, G. (1984). *Calcul d'un ensemble complet d'incrémentations minimales pour l'ordre récursif de décomposition.* Technical report, Centre de Recherche en Informatique de Nancy, Nancy, France.

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

Collins, G. (1975). Quantifier elimination for real closed fields by cylindrical algebraic decomposition. *Proceedings of the Second GI Conference on Automata Theory and Formal Languages,* Kaiserslautern, West Germany. *Springer Lec. Notes Comp. Sci.* **33,** 134–183.

Dauchet, M., Tison, S. (1985). Tree automata and decidability in ground term rewriting systems. Proceedings of the Fifth International Conference on Fundamentals of Computation Theory, Cottbus, East Germany. *Springer Lec. Notes Comp. Sci.* **199,** 80–89.

Dershowitz, N. (1979). A note on simplification orderings. *Inf. Proc. Lett.* **9,** 212–215.

Dershowitz, N. (1980). *On representing ordinals up to* Γ_0. Unpublished note, Department Computer Science, University of Illinois, Urbana, IL.

Dershowitz, N. (1981). Termination of linear rewriting systems (Preliminary version). *Proceedings of the Eighth EATCS International Colloquium on Automata, Languages and Programming,* Acre, Israel. *Springer Lec. Notes Comp. Sci.* **115,** 448–458.

Dershowitz, N. (1982*a*). Orderings for term-rewriting systems. *Theor. Comp. Sci.* **17,** 279–301. (Previous version appeared in *Proceedings of the Symposium on Foundations of Computer Science,* San Juan, PR, pp. 123–131 [October 1979].)

Dershowitz, N. (1982*b*). Applications of the Knuth–Bendix completion procedure. *Proceedings of the Seminaire d'Informatique Theorique,* Paris, France, pp. 95–111.

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

Dershowitz, N., Hsiang, J., Josephson, N. A., Plaisted, D. A. (1983). Associative-commutative rewriting. *Proceedings of the Eighth International Joint Conference on Artificial Intelligence,* Karlsruhe, West Germany, pp. 940–944.

Dershowitz, N., Jouannaud, J.-P. (1987). Rewrite systems. In: (Meyer, A., Nivat, M., Paterson, M., Perrin, D., eds.) *Handbook of Theoretical Computer Science,* North-Holland, Amsterdam. (In preparation.)

Dershowitz, N., Manna, Z. (1979). Proving termination with multiset orderings. *Commun. ACM* **22,** 465–476. (Also in *Proceedings of the International Colloquium on Automata, Languages and Programming,* Graz, pp. 188–202 [July 1979].)

Dershowitz, N., Zaks, S. (1981). Applied tree enumerations. Proceedings of the Sixth Colloquium on Trees in Algebra and Programming, Genoa, Italy. *Springer Lec. Notes Comp. Sci.* **112,** 180–193.

Detlefs, D., Forgaard, R. (1985). 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. *Springer Lec. Notes Comp. Sci.* **202,** 255–270.

Ehrenfeucht, A., Haussler, D., Rozenberg, G. (1983). On regularity of context-free languages. *Theor. Comp. Sci.* **27,** 311–332.

Fages, F. (1984). *Le système KB: manuel de référence: présentation et bibliographie, mise en œuvre.* Report R.G.10.84, Greco de Programmation, C.N.R.S., Bordeaux, France.

Feferman, S. (1968). Systems of predicative analysis II: Representation of ordinals. *J. Symb. Logic* **33,** 193–220.

Filman, R. (1978). Personal communication.

Forgaard, R. (1984). *A program for generating and analyzing term rewriting systems.* Report 343, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA. (Master's thesis.)

Friedman, H. (1982). *Beyond Kruskal's theorem I–III.* Unpublished reports, Ohio State University, Columbus, OH.

Gardner, M. (1983). Mathematical games: Tasks you cannot help finishing no matter how hard you try to block finishing them. *Sci. Am.* **24,** 12–21.

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

Gnaedig, I. (1986). *Preuves de terminaison des systèmes de réécriture associatifs commutatifs une méthode fondée sur la réécriture elle-même.* Thèse de Troisième Cycle, Université de Nancy I, Nancy, France.

Gnaedig, I., Lescanne, P. (1986). Proving termination of associative rewriting systems by rewriting. Proceedings of the Eighth Conference on Automated Deduction, Oxford, England. *Springer Lec. Notes Comp. Sci.* **230,** 52–61.

Göbel, R. (1983). A completion procedure for globally finite term rewriting systems. *Proceedings of an NSF Workshop on the Rewrite Rule Laboratory*, Schenectady, NY, pp. 155–203. (Available as Report 84GEN008, General Electric Research and Development [April 1984].)
Gorn, S. (1967). Handling the growth by definition of mechanical languages. *Proceedings of the Spring Joint Computer Conference*, pp. 213–224.
Gorn, S. (1973). On the conclusive validation of symbol manipulation processes (How do you know it has to work?). *J. Franklin Inst.* **296,** 499–518.
Guttag, J. V., Kapur, D., Musser, D. R. (1983). On proving uniform termination and restricted termination of rewriting systems. *SIAM J. Comput.* **12,** 189–214.
Haussler, D. (1985). Another generalization of Higman's well quasi order result on a finitely generated free monoid. *Discrete Math.* **57,** 237–243.
Higman, G. (1952). Ordering by divisibility in abstract algebras. *Proc. London Math. Soc. (3)* **2,** 326–336.
Hsiang, J., Dershowitz, N. (1983). Rewrite methods for clausal and non-clausal theorem proving. Proceedings of the Tenth EATCS International Colloquium on Automata, Languages and Programming, Barcelona, Spain. *Springer Lec. Notes Comp. Sci.* **154,** 331–346.
Huet, G. (1980). Confluent reductions: Abstract properties and applications to term rewriting systems. *J. Assoc. Comp. Mach.* **27,** 797–821. (Previous version in *Proceedings of the Symposium on Foundations of Computer Science*, Providence, RI, pp. 30–45 [October 1977].)
Huet, G., Lankford, D. S. (1978). *On the uniform halting problem for term rewriting systems.* Rapport Laboria 283, Institut de Recherche en Informatique et en Automatique, Le Chesnay, France.
Huet, G., Lévy, J.-J. (1979). *Call by need computations in non-ambiguous linear term rewriting systems.* Rapport Laboria 359, Institut National de Recherche en Informatique et en Automatique, Le Chesnay, France.
Huet, G., Oppen, D. C. (1980). Equations and rewrite rules: A survey. In (Book, R., ed.) *Formal Language Theory: Perspectives and Open Problems*, pp. 349–405. New York: Academic Press.
Hullot, J.-M. (1980). Canonical forms and unification. *Proceedings of the Fifth Conference on Automated Deduction*, Les Arcs, France, pp. 318–334.
Iturriaga, R. (1967). *Contributions to mechanical mathematics.* Ph.D. thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA.
Jefferson, D. R. (1980). *Type reduction and program verification.* Ph.D. thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA.
Jouannaud, J.-P., Kirchner, H. (1984). Construction d'un plus petit ordre de simplification. *RAIRO Theor. Inform.* **18,** 191–207.
Jouannaud, J.-P., Kirchner, H. (1986). Completion of a set of rules modulo a set of equations. *SIAM J. Comput.* **15,** 1155–1194.
Jouannaud, J.-P., Lescanne, P. (1982). On multiset orderings. *Inf. Proc. Lett.* **15,** 57–63.
Jouannaud, J.-P., Lescanne, P., Reinig, F. (1982). Recursive decomposition ordering. *Proceedings of the Second IFIP Workshop on Formal Description of Programming Concepts*, Garmisch-Partenkirchen, West Germany, pp. 331–348.
Jouannaud, J.-P., Muñoz, M. (1984). Termination of a set of rules modulo a set of equations. Proceedings of the Seventh International Conference on Automated Deduction, Napa, CA. *Springer Lec. Notes Comp. Sci.* **170,** 175–193.
Kamin, S., Lévy, J.-J. (1980). *Two generalizations of the recursive path ordering.* Unpublished note, Department of Computer Science, University of Illinois, Urbana, IL.
Kapur, D., Sivakumar, G. (1983). Experiments with and architecture of RRL, a rewrite rule laboratory. *Proceedings of an NSF Workshop on the Rewrite Rule Laboratory*, Schenectady, NY, pp. 33–56. (Available as Report 84GEN008, General Electric Research and Development [April 1984].)
Kapur, D., Narendran, P., Sivakumar, G. (1985). A path ordering for proving termination of term rewriting systems. Proceedings of the Tenth Colloquium on Trees in Algebra and Programming, Berlin, West Germany. *Springer Lec. Notes Comp. Sci.* **185,** 173–185.
Kent, C. F., Hodgson, B. R. (1986). *Extensions of arithmetic for proving termination of computations.* Report 86–10, Département de Mathématiques, Université Laval, Québec, Québec.
Kirby, L., Paris, J. (1982). Accessible independence results for Peano arithmetic. *Bull. London. Math. Soc.* **14,** 285–293.
Klop, J. W. (1980). Reduction cycles in combinatory logic. In: (Seldin, J. P., Hendley, R., eds.) *To H. B. Curry, Essays on Combinatory Logic, Lambda Calculus and Formalism*, pp. 193–214. New York: Academic Press.
Knuth, D. E. (1973). Fundamental algorithms. In: *The Art of Computer Programming*, vol. 1, 2nd edn. Reading, MA: Addison-Wesley.
Knuth, D. E., Bendix, P. B. (1970). Simple word problems in universal algebras. In: (Leech, J., ed.) *Computational Problems in Abstract Algebra*, pp. 263–297. Oxford: Pergamon Press.
Krishnamoorthy, M. S., Narendran, P. (1984). *A note on recursive path ordering.* Unpublished note, General Electric Corporate Research and Development, Schenectady, NY.
Kruskal, J. B. (1954). *The theory of well-partially-ordered sets.* Ph.D. thesis, Princeton University, Princeton, NJ.

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

Kruskal, J. B. (1972). The theory of well-quasi-ordering: A frequently discovered concept. *J. Comb. Theor. Ser. A* **13**, 297–305.

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

Lankford, D. S. (1975*b*). Canonical inference. Memo ATP-32, Automatic Theorem Proving Project, University of Texas, Austin, TX.

Lankford, D. S. (1976). *A finite termination algorithm.* Internal memo, Southwestern University, Georgetown, TX.

Lankford, D. S. (1977). *Some approaches to equality for computational logic: A survey and assessment.* Memo ATP-36, Automatic Theorem Proving Project, University of Texas, Austin, TX.

Lankford, D. S. (1979). *On proving term rewriting systems are Noetherian.* Memo MTP-3, Mathematics Department, Louisiana Tech. University, Ruston, LA. (Revised October 1979.)

Lankford, D. S., Musser, D. R. (1978). *A finite termination criterion.* Unpublished draft, Information Sciences Institute, University of Southern California, Marina-del-Rey, CA.

Laver, R. (1978). Better-quasi-orderings and a class of trees. In: (Rota, G.-C., ed.) *Studies in Foundations and Combinatorics*, pp. 31–48. New York: Academic Press.

Lévy, J.-J. (1980). Problem 80-5. *J. Algorithms* **1**, 108–109.

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

Lescanne, P. (1982). Some properties of decomposition ordering, A simplification ordering to prove termination of rewriting systems. *RAIRO Theor. Inform.* **16**, 331–347.

Lescanne, P. (1983). Computer experiments with the REVE term rewriting system generator. *Proceedings of the Tenth ACM Symposium on Principles of Programming Languages*, Austin, TX, pp. 99–108.

Lescanne, P. (1984). Uniform termination of term-rewriting systems: Recursive decomposition ordering with status. *Proceedings of the Ninth Colloquium on Trees in Algebra and Programming*, pp. 181–194, Bordeaux, France. Cambridge: Cambridge University Press.

Lescanne, P., Steyaert, J.-M. (1983). 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. *Springer Lec. Notes Comp. Sci.* **154**, 466–475.

Lipton, R., Snyder, L. (1977). On the halting of tree replacement systems. *Proceedings of the Conference on Theoretical Computer Science*, University of Waterloo, Waterloo, Canada, pp. 43–46.

Manna, Z. (1968). *Termination of algorithms.* Ph.D. thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA.

Manna, Z. (1974). *Mathematical Theory of Computation.* New York: McGraw-Hill.

Manna, Z., Ness, S. (1969). *Termination of Markov algorithms.* Unpublished manuscript, Department of Computer Science, Stanford University, Stanford, CA.

Manna, Z., Ness, S. (1970). On the termination of Markov algorithms. *Proceedings of the Third Hawaii International Conference on System Science*, Honolulu, HI, pp. 789–792.

Martin, U. (1986). *Multiset Orderings.* Technical Report UMCS-86-5-1, Department of Computer Science, University of Manchester, Manchester, England.

Metivier, Y. (1983). About the rewriting systems produced by the Knuth–Bendix completion algorithm. *Inf. Proc. Lett.* **16**, 31–34.

Metivier, Y. (1985). Calcul de longueurs de chaînes de réécriture dans le monoïde libre. *Theor. Comp. Sci.* **35**, 71–87.

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

Nash-Williams, C. St. J. A. (1965). On well-quasi-ordering infinite trees. *Proc. Cambridge Phil. Soc.* **61**, 697–720.

Newman, M. H. A. (1942). On theories with a combinatorial definition of 'equivalence'. *Ann. Math.* **43**, 223–243.

O'Donnell, M. J. (1977). Computing in systems described by equations. *Springer Lec. Notes Comp. Sci.* **58**.

Okada, M. (1986*a*). *An extended Kruskal theorem with a restricted gap condition for the transfinitely labeled finite trees.* Unpublished report, Department of Mathematics, University of Illinois, Urbana, IL.

Okada, M. (1986*b*). Ackermann's ordering and its relationship with ordering systems of term rewriting theory. *Proceedings of the Twenty-fourth Allerton Conference on Communication, Control, and Computing*, Monticello, IL.

Okada, M., Takeuti, G. (1986). On the theory of quasi ordinal diagrams. In: (Simpson, S. G., ed.) *Logic and Combinatorics.* Providence, RI: American Mathematical Society.

Paulson, L. C. (1984). *Constructing recursion operations in intuitionistic type theory.* Technical Report 57, Computer Laboratory, University of Cambridge, Cambridge, UK.

Pettorossi, A. (1978). *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.

Pettorossi, A. (1981). Comparing and putting together recursive path orderings, simplification orderings and non-ascending property for termination proofs of term rewriting systems. Proceedings of the Eighth

J. Symbolic Computation (1987) 4, 409–410

Corrigendum

Termination of Rewriting

NACHUM DERSHOWITZ

J. Symbolic Computation (1987) 3, 69-116.

(1) The last line on page 78 should read:

$$\tau(\alpha+\beta) = 2\tau(\alpha) + \tau(\beta) + 1 \qquad \tau(1) = 2.$$

(2) The second sentence on page 80 should read:

The polynomial

$$x^2 + y^2 + 2xy - x^2 - y^2 - x - 2y - c$$

(with x for $\tau(\alpha)$, y for $\tau(\beta)$, and c for $\tau(2)$) is no less than

$$2xy - 2x - 2y$$

(assuming that $x \geq c$). The latter is eventually positive, since its two derivatives, $2y-2$ and $2x-2$, are.

In general, the test for eventually positive polynomials (Lankford, 1976) only helps when there are no constants in the polynomial (Lescanne, 1987).

(3) After the second sentence of section 4 (page 80), the following should have been stated explicitly:

(Actually, an ordering over fixed-arity terms is "well-founded for derivations" if, and only if, it is well-founded over terms constructed from a finite number of function symbols.)

(4) Theorem 9 (page 82) should read:

THEOREM 9 (Dershowitz, 1982*a*). *Any total monotonic ordering on fixed-arity terms is well-founded for derivations if, and only if, it is a simplification ordering.*

The essence of the "only-if" direction appears in Plaisted (1978*a*).

(5) The fifth sentence of the example on page 91 should read:

Whichever sequence becomes a proper subsequence of the other (or becomes smaller than a subsequence of the other) is smaller.

(6) Definition 25 (page 110) as stated (adapted from Dershowitz, 1981) does not work (Purdom, 1987). Instead, it should read:

DEFINITION 25 (Purdom, 1987). A derivation $t_1 \Rightarrow t_2 \Rightarrow \cdots \Rightarrow t_j \Rightarrow \cdots \Rightarrow t_k \Rightarrow \cdots$ *loops* if some instance of t_k has a subterm that is the same as a less or equally general instance of t_j for some $j < k$.

References

Dershowitz, N. (1981). Termination of linear rewriting systems (Preliminary version). *Proceedings of the Eighth EATCS International Colloquium on Automata, Languages and Programming*, Acre, Israel. *Springer Lec. Notes Comp. Sci.* 115, 448-458.

Dershowitz, N. (1982*a*). Orderings for term-rewriting systems. *Theor. Comp. Sci.* 17, 279-301. (Previous version appeared in *Proceedings of the Symposium on Foundations of Computer Science*, San Juan, PR, pp. 123-131 [October 1979].)

Lankford, D. S. (1976). *A finite termination algorithm.* Internal Memo, Southwestern University, Georgetown, TX.

Lescanne, P. (1987). Personal communication.

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

Purdom, P. W., Jr. (1987). Detecting Looping Simplifications. *Proceedings of the Second International Conference on Rewriting Techniques and Applications*, Bordeaux, France. *Springer Lec. Notes Comp. Sci.* 256, 54-61.