PROVING TERMINATION WITH MULTISET ORDERINGS

by

Nachum Dershowitz[1] and Zohar Manna[2]

Stanford University and Weizmann Institute

ABSTRACT

A common tool for proving the termination of programs is the *well-founded set*, a set ordered in such a way as to admit no infinite descending sequences. The basic approach is to find a *termination function* that maps the values of the program variables into some well-founded set, such that the value of the termination function is continually reduced throughout the computation. All too often, the termination functions required are difficult to find and are of a complexity out of proportion to the program under consideration. However, by providing more sophisticated well-founded sets, the corresponding termination functions can be simplified.

Given a well-founded set S, we consider *multisets* over S, "sets" that admit multiple occurrences of elements taken from S. We define an ordering on all finite multisets over S that is induced by the given ordering on S. This *multiset ordering* is shown to be well-founded. The value of the multiset ordering is that it permits the use of relatively simple and intuitive termination functions in otherwise difficult termination proofs. In particular, we apply the multiset ordering to prove the termination of *production systems*, programs defined in terms of sets of rewriting rules.

An extended version of this paper appeared as Memo AIM-310, Stanford Artificial Intelligence Laboratory, Stanford, California.

This research was supported in part by the United States Air Force Office of Scientific Research under Grant AFOSR-76-2909 (sponsored by the Rome Air Development Center, Griffiss AFB, NY), by the National Science Foundation under Grant MCS 76-83655, and by the Advanced Research Projects Agency of the Department of Defense under Contract MDA 903-76-C-0206.

[1] Current Address:
Department of Computer Science
University of Illinois
Urbana, Illinois 61801

[2] Current Address:
Department of Computer Science
Stanford University
Stanford, California 94305

I. INTRODUCTION

The use of well-founded sets for proving that programs terminate has been suggested by Floyd [1967]. A *well-founded set* consists of a set of elements S and a transitive and irreflexive ordering $\succ$ defined on the elements such that there can be no infinite descending sequences of elements. The idea is to find a well-founded set and a *termination function* that maps the values of the program variables into that set such that the value of the termination function is continually reduced throughout the computation. Since, by the nature of the set, the value cannot decrease indefinitely, the program must terminate.

The well-founded sets most frequently used for this purpose are the natural numbers under the "greater-than" ordering and n-tuples of natural numbers under the lexicographic ordering. In practice using these conventional orderings often leads to complex termination functions that are difficult to discover. For example, the termination proofs of programs involving stacks and production systems are often quite complicated and require much more subtle orderings and termination functions. Finding an appropriate ordering and termination function for such programs is a well-known challenge among researchers in the field of program verification. In this paper, we introduce a powerful ordering that can sometimes make the task of proving termination easier.

II. THE MULTISET ORDERING

For a given partially-ordered set $(S,\succ)$, we consider *multisets* (sometimes called "bags") over S, i.e. unordered collections of elements that may have multiple occurrences of identical elements. For example, $\{3,3,3,4,0,0\}$ is a multiset of natural numbers; it is identical to the multiset $\{0,3,3,0,4,3\}$, but distinct from $\{3,4,0\}$. We denote by $\mathcal{M}(S)$ the set of all finite multisets with elements taken from the set S.

For a partially-ordered set $(S,\succ)$, the *multiset ordering* $\succ\!\succ$ on $\mathcal{M}(S)$ is defined as follows:

$$M \succ\!\succ M'$$

if for some multisets $X,Y\varepsilon\mathcal{M}(S)$, where $\{\}\neq X\subseteq M$,

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

and

$$(\forall y\varepsilon Y)(\exists x\varepsilon X)\ x\succ y.$$

In words, a multiset is reduced by the removal of at least one element (those in X) and their replacement with any finite number - possibly zero - of elements (those in Y), each of which is smaller than one of the elements that have been removed. Thus, if S is the set N of natural numbers 0,1,2,... with the > ordering, then under the corresponding multiset ordering $\gg$ over N, the multiset $\{3,3,4,0\}$ is greater than each of the three multisets $\{3,4\}$, $\{3,2,2,1,1,1,4,0\}$, and $\{3,3,3,3,2,2\}$. In the first case, two elements have been removed; 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. The empty multiset {} is clearly smaller than any other multiset.

The multiset ordering is in fact a partial ordering, i.e. if $\succ$ is irreflexive and transitive, then $\gg$ also is. We have the

THEOREM: *The multiset ordering* $(\mathcal{M}(S),\gg)$ *over* $(S,\succ)$ *is well-founded, if and only if* $(S,\succ)$ *is.*

Proof: The "only if" part is trivial. For the "if" part, assume that $(S,\succ)$ is well-founded. Let $S' = S \cup \{\perp\}$ be S extended with a least element $\perp$, i.e. for every element $s \varepsilon S$, $s \succ \perp$ in the ordering on S'. Clearly S' is well-founded if S is. Now, suppose that $(\mathcal{M}(S),\gg)$ is not well-founded; therefore, there exists an infinite descending sequence $M_1 \gg M_2 \gg M_3 \gg \ldots$ of multisets of $\mathcal{M}(S)$. We derive a contradiction by constructing the following tree. Each node in the tree is labelled with some element of S'; at each stage of the construction, the set of all terminal nodes in the tree forms a multiset in $\mathcal{M}(S')$.

> Begin with a root node with children corresponding to each element of M_1. Then since $M_1 \gg M_2$, there must exist multisets X and Y, such that $\{\} \neq X \subseteq M_1$, $M_2 = (M_1 \setminus X) \cup Y$, and $(\forall y \varepsilon Y)(\exists x \varepsilon X) x \succ y$. Then for each $y \varepsilon Y$, add a child labelled y to the corresponding x. In addition, grow a child $\perp$ from each of the elements of X. (Since X is nonempty, growing $\perp$ ensures that even if Y is empty, at least one node is added to the tree. Since Y is finite, the nodes corresponding to X each have a finite number of children.) Repeat the process for $M_2 \gg M_3$, $M_3 \gg M_4$, and so on.

Since at least one node is added to the tree for each multiset M_i in the sequence, were the sequence infinite, the tree corresponding to the sequence would also be. But by Konig's Infinity Lemma, an infinite tree (with a finite number of children for each node) must have an infinite path. On the other hand, by our construction, all paths in the tree are descending in the well-founded ordering $\succ$ on S', and must be finite. Thus, we have derived a contradiction, implying that the sequence $M_1, M_2, M_3, \ldots$ cannot be infinite.

Remarks:

- If $(S,\succ)$ is totally ordered, then for any two multisets $M,M' \varepsilon \mathcal{M}(S)$, one may decide whether $M \gg M'$ by first sorting the elements of both M and M' in descending order (with respect to the relation $\succ$) and then comparing the two sorted sequences lexicographically.
- *If* $(S,\succ)$ *is of order type* α, *then the multiset ordering* $(\mathcal{M}(S),\gg)$ *over* $(S,\succ)$ *is of order type* ω^{α}.

- Consider the special case where there is a bound k on the number of replacement elements, i.e. take the (irreflexive) transitive closure of the relation $M \gg' M'$ which holds if $M'=(M \setminus X) \cup Y$ and $|Y| < k$. *Any termination proof using this bounded multiset ordering over* N *may be translated into a proof using* (N,>). This may be done using, for example, the order-preserving function

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

which maps multisets over the natural numbers into the natural numbers by summing the number k^n for every number n in a multiset M. □

We turn now to consider *nested multisets*, by which we mean that the elements of the multisets may belong to some base set S, or may be multisets containing both elements of S and multisets of elements of S, and so on. For example, {{1,1}, {{0},1,2},0} is a nested multiset. More formally, given a partially-ordered set $(S,\succ)$, a *nested multiset over* S is either an element of S, or else it is a finite multiset of nested multisets over S. We denote by $\mathcal{M}^*(S)$ the set of nested multisets over S.

We define now a *nested multiset ordering* $\gg^*$ on $\mathcal{M}^*(S)$; it is a recursive version of the standard multiset ordering. For two elements $M,M' \in \mathcal{M}^*(S)$, we say that

$$M \gg^* M'$$

if

- $M,M' \in S$ and $M \succ M'$

(two elements of the base set are compared using $\succ$), or else

- $M \notin S$ and $M' \in S$

(any multiset is greater than any element of the base set), or else

- $M,M' \in S$, and for some $X,Y \in \mathcal{M}^*(S)$, where $\{\} \neq X \subseteq M$,

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

and

$$(\forall y \in Y)(\exists x \in X)\ x \gg^* y.$$

For example, the nested multiset {{1,1},{{0},1,2},0} is greater than {{1,0,0},5, {{0},1,2},0}, since {1,1} is greater than both {1,0,0} and 5. The same nested multiset {{1,1}, {{0},1,2},0} is also greater than {{{},1,2},{5,5,2},5}, since {{0},1,2} is greater than each of the three elements {{},1,2}, {5,5,2}, and 5.

Let $\mathcal{M}^i(S)$ denote the set of all nested multisets of *depth* i. In other words $\mathcal{M}^0(S)=S$, and $\mathcal{M}^{i+1}(S)$ contains the multisets whose elements are taken from $\mathcal{M}^0(S)$, $\mathcal{M}^1(S),\ldots,\mathcal{M}^i(S)$, with at least one element taken from $\mathcal{M}^i(S)$. Thus, the set $\mathcal{M}^*(S)$ is the infinite union of the disjoint sets $\mathcal{M}^0(S)$, $\mathcal{M}^1(S)$, $\mathcal{M}^2(S)$, ... The following property holds:

> *For two nested multisets, M and M', if the depth of M is greater than the depth of M', then $M \gg^* M'$.*

In other words, the multisets of $\mathcal{M}^i(S)$ are all greater than the multisets of $\mathcal{M}^j(S)$, under the ordering $\gg^*$, for any $j<i$.

The relation $\gg^*$ is a partial ordering; it can be shown to be both irreflexive and transitive. The following theorem gives the condition under which it is well-founded:

THEOREM: *The nested multiset ordering* $(\mathcal{M}^*(S), \gg^*)$ *over* $(S,\succ)$ *is well-founded, if and only if* $(S,\succ)$ *is well-founded.*

In order to show that $(\mathcal{M}^*(S), \gg^*)$ is well-founded, it suffices to show that each $\mathcal{M}^i(S)$ is itself well-founded under $\gg^*$. This may be proved by induction on i.

Remark: It can be shown that *if* $(S,\succ)$ *is of order type less than* ε_0, *then* $(\mathcal{M}^*(S), \gg^*)$ *is of order type* ε_0. (Gentzen [1938] used in ε_0 ordering to prove the termination of his normalization procedure for proofs in arithmetic.) □

In the following two sections, we shall apply the multiset ordering to problems of termination, first proving the termination of conventional programs, and then proving the termination of production systems.

III. TERMINATION OF PROGRAMS

In the following examples, we shall prove the termination of programs using multiset orderings as the well-founded set.

EXAMPLE 1: *Counting tips of binary trees.*

Consider a simple program to count the number of tips - terminal nodes (without descendents) - in a full binary tree. Each tree y that is not a tip has two subtrees, $left(y)$ and $right(y)$. The program is

```
S := (t)
c := 0
loop until S=()
     y := head(S)
     if tip(y) then S := tail(S)
                    c := c+1
               else S := left(y)·right(y)·tail(S)
               fi
     repeat.
```

It employes a stack S and terminates when S is empty. At that point, the variable c is to contain the total number of tip nodes in the given tree t. The termination of this program may be proved using the well-founded set (N,>). The appropriate termination function is

$$\tau(S) = \sum_{s\in S} nodes(s),$$

where $nodes(s)$ is the total number of nodes in the subtree s - not just the tip nodes.

Using the multiset ordering over trees, we can prove termination with the simple termination function

$$\tau(S) = \{s : s \varepsilon S\},$$

giving the multiset of trees appearing in the stack. The trees themselves are ordered by the natural well-founded subtree ordering, i.e. any tree is greater than its subtrees. Thus, removal of a tree from the stack decreases τ in the multiset ordering by removing an element, and the replacement of a tree with two smaller subtrees decreases τ.

In general, any program in which elements are repeatedly removed from a stack, queue, bag, etc. and replaced with any number of smaller elements (in some well-founded ordering) can be shown to terminate with the corresponding multiset ordering.

EXAMPLE 2: *McCarthy's 91-function*

The following is a contrived program to compute the simple function

$$f(x) = \textit{if } x>100 \textit{ then } x-10 \textit{ else } 91$$

over the set of integers Z, in a round-about manner. Though this program is short, the proof of its correctness and termination are nontrivial, and for this reason it is often used to illustrate proof methods.

The program is:

```
$n := 1$
$z := x$
loop $L$: assert $f(x)=f^n(z)$, $n \geq 1$
     if $z>100$ then $n := n-1$
                     $z := z-10$
               else $n := n+1$
                     $z := z+11$
               fi
     until $n=0$
     repeat
assert $z=f(x)$.
```

The predicates $f(x)=f^n(z)$ and $n \geq 1$ are loop invariants. The loop is exited if control reaches the until clause with $n=0$; at that point $f(x)=f^0(z)=z$.

Consider the following well-founded partial-ordering $\succ$ on the integers:

$$a \succ b \text{ if and only if } a<b<111.$$

(This is the same ordering on integers as in the familiar structural-induction proof, due to Rod Burstall, of the recursive version of this program.) As the well-founded set, we use the set $(\mathcal{M}(Z), \ggcurly)$ of all multisets of integers, under the corresponding multiset ordering. The appropriate termination function τ at L yields a multiset in $\mathcal{M}(Z)$, and is defined as

$$\tau(n,z) = \{z,f(z),\ldots,f^{n-1}(z)\}.$$

We must show that for each loop iteration this function decreases. There are three cases to consider:

1. $z>100$ at L: In this case, the then branch of the conditional is executed and both n and z are decremented. When control returns to L (assuming that the loop has not been exited), we have, in terms of the old values of n and z,

$$\begin{aligned}\tau(n-1,z-10) &= \{z-10,f(z-10),\ldots,f^{n-2}(z-10)\}\\ &= \{f(z),f^2(z),\ldots,f^{n-1}(z)\}.\end{aligned}$$

Thus, the value of the termination function τ has been decreased by removing the element z from the original multiset $\{z,f(z),\ldots,f^{n-1}(z)\}$.

2. $90\leq z\leq 100$ at L: In this case, the else branch is taken and both n and z are incremented, yielding

$$\tau(n+1,z+11) = \{z+11,f(z+11),f^2(z+11),\ldots,f^n(z+11)\}.$$

Either $z+1=101$ or else $z+1\leq 100$; in both cases $f^2(z+11)=f(z+1)=91=f(z)$.
Thus, we get

$$\tau(n+1,z+11) = \{z+11,z+1,f(z),\ldots,f^{n-1}(z)\}$$

Since $z<z+1<z+11\leq 111$, we have $z\succ z+11$ and $z\succ z+1$. Accordingly, the multiset has been reduced by replacing the element z with the two smaller elements, $z+11$ and $z+1$.

3. $z<90$ at L: The else branch is taken and we have

$$\begin{aligned}\tau(n+1,z+11) &= \{z+11,f(z+11),f^2(z+11),\ldots,f^n(z+11)\}.\\ &= \{z+11,91,f(z),\ldots,f^{n-1}(z)\}.\end{aligned}$$

Again z has been replaced by two smaller elements (under the $\succ$ relation), $z+11$ and 91. □

EXAMPLE 3: *Ackermann's function.*

The following iterative program computes Ackermann's function $a(m,n)$ over pairs of natural numbers:

```
S := (m)
z := n
loop L: assert $a(m,n) = a(s_k,a(s_{k-1},\dots,a(s_2,a(s_1,z))\dots))$
     y := head(S)
     S := tail(S)
     if y=0 then z := z+1
     else
     if z=0 then S := (y-1)·S
                 z := 1
            else S := y·(y-1)·S
                 z := z-1
            fi fi
     until S=()
     repeat
assert z = a(m,n),
```

where the stack S has k elements $s_1,s_2,\dots,s_k$.

To prove termination, consider the set N × N of lexicographically-ordered pairs of natural numbers and use the corresponding multiset ordering over N × N. Let $y=head(S)=s_1$ The termination function at L is

$$\tau(S,z) = \{(s_k+1,0),(s_{k-1}+1,0),\dots,(s_2+1,0),(y,z)\}.$$

Thus, $\tau(S,z)$ yields a multiset containing one pair per element in the stack S. Note that at L, the stack S is nonempty, and all the elements of S as well as z are non-negative.

The proof considers three cases, corresponding to the three branches of the conditional in the loop:

1. $y=0$: If the loop is not exited, then the new value of τ at L is

$$\tau((s_2,\dots,s_k),z+1) = \{(s_k+1,0),\dots,(s_2+1,0),(s_2,z+1)\}.$$

This represents a decrease in τ under the multiset ordering, since the element (y,z) has been removed and the element $(s_2+1,0)$ has been replaced by the smaller $(s_2,z+1)$.

2. $y\neq 0$ and $z=0$: In this case we obtain

$$\tau((y-1,s_2,\dots,s_k),1) = \{(s_k+1,0),\dots,(s_2+1,0),(y-1,1)\}.$$

Thus, the element (y,z) has been replaced by the smaller element $(y-1,1)$.

3. $y\neq 0$ and $z\neq 0$: Here we have

$$\tau((y,y-1,s_2,\dots,s_k),z-1) = \{(s_k+1,0),\dots,(s_2+1,0),(y,0),(y,z-1)\}.$$

The element (y,z) has been replaced by the two smaller elements $(y,0)$ and $(y,z-1)$.

Remark: The previous examples suggest the following heuristic for proving termination: given a program over a domain $(D,\succ)$ that computes some function $f(x)$, if the prógram has a loop invariant of the form

$$f(x) = h(f(g_1(y)), f(g_2(y)), \ldots, f(g_n(y))),$$

where the g_i are the arguments of occurrences of f in the right-hand side, then try the multiset ordering $(\mathcal{M}(D), \gg)$ and use the termination function

$$\tau(y) = \{g_1(y), g_2(y), \ldots, g_n(y)\}.$$

The idea underlying this heuristic is that τ represents the set of unevaluated arguments of some recursive expansion of the function f.

IV. TERMINATION OF PRODUCTION SYSTEMS

A production system Π (also called a *term-writing system*) over a set of expressions E is a (finite or infinite) set of rewriting rules, called *productions*, each of the form

$$\pi(\alpha, \beta, \ldots) \rightarrow \pi'(\alpha, \beta, \ldots),$$

where π and π' are expressions containing variables $\alpha, \beta, \ldots$ ranging over E. (The variables appearing in π' must be a subset of those in π.) Such a rule is applied in the following manner: given an expression $e \varepsilon E$ that contains a subexpression

$$\pi(a, b, \ldots),$$

(i.e. the variables $\alpha, \beta, \ldots$ are instantiated with the expressions $a, b, \ldots$, respectively), replace that subexpression with the corresponding expression

$$\pi'(a, b, \ldots).$$

We write $e \Rightarrow e'$, if the expression e' can be derived from e by a single application of some rule in Π to one of the subexpressions of e.

For example, the following is a production system that differentiates an expression, containing + and $\cdot$, with respect to x:

$$\boxed{\begin{array}{lll} Dx & \rightarrow & 1 \\ Dy & \rightarrow & 0 \\ D(\alpha+\beta) & \rightarrow & (D\alpha + D\beta) \\ D(\alpha\cdot\beta) & \rightarrow & ((\beta\cdot D\alpha) + (\alpha\cdot D\beta)), \end{array}}$$

where y can be any constant or any variable other than x. Consider the expression

$$D(D(x\cdot x)+y).$$

We could either apply the third production to the outer D, or else we could apply the fourth production in the inner D. In the latter case, we obtain

$$D(((x\cdot Dx)+(x\cdot Dx))+y)$$

which now contains three occurrences of D. At this point, we can still apply the third production to the outer D, or we could apply the first production to either one of the inner D's. Applying the third production yields

$$(D((x\cdot Dx)+(x\cdot Dx)+Dy).$$

Thus,

$$D(D(x\cdot x)+y) \Rightarrow D(((x\cdot Dx)+(x\cdot Dx))+y) \Rightarrow (D((x\cdot Dx)+(x\cdot Dx))+Dy).$$

In general, at each stage in the computation there are many ways to proceed, and the choice is made nondeterministically. In our case, all choices eventually lead to the expression

$$((((1\cdot 1)+(x\cdot 0))+((1\cdot 1)+(x\cdot 0)))+0),$$

for which no further application of a production is possible.

A production system Π *terminates* over E, if there exist no infinite sequences of expressions $e_1,e_2,e_3,\ldots$ such that $e_1 \Rightarrow e_2 \Rightarrow e_3 \Rightarrow \ldots$ and $e_1 \varepsilon E$. In other words, given any initial expression, execution always reaches a state for which there is no way to continue applying productions. The difficulty in proving the termination of a production system, such as the one for differentiation above, stems from the fact that while some productions (the first two) may decrease the size of an expression, other productions (the last two) may increase its size. Also, a production (the fourth) may actually duplicate occurrences of subexpressions. Furthermore, applying a production to a subexpression, not only affects the structure of that subexpression, but also changes the corresponding superexpressions, including the top-level expression. And a proof of termination must hold for the many different possible sequences generated by the nondeterministic choice of productions and subexpressions.

The following theorem has provided the basis for most of the techniques used for proving the termination of production systems:

THEOREM: *A production system over E terminates, if and only if there exists a well-founded set* $(W,\succ)$ *and a termination function* $\tau : E \rightarrow W$, *such that for any* $e,e' \varepsilon E$

$$e \Rightarrow e' \textit{ implies } \tau(e) \succ \tau(e').$$

Several researchers have considered the problem of proving the termination of production systems. Among them: Gorn [1965] in an early work addresses this issue; Iturriaga [1967] gives sufficient conditions under which a class of production systems terminates; Knuth and Bendix [1969] define a well-founded ordering based on a weighted size for expressions; Manna and Ness [1970] and Lankford [1975] use a "monotonic interpretation" that decreases with each application of a production; Lipton and Snyder [1977] make use of a "value-preserving" property as the basis for a method of proving termination. Recently, Plaisted [July 1978, Oct. 1978] has applied two classes of well-founded orderings on terms to the termination of production systems.

In the following examples, we illustrate the use of multisets in proving termination. We begin with a very simple example.

EXAMPLE 1: *Associativity.*

Consider the set of arithmetic expressions E constructed from some set of atoms (symbols) and the single operator +. The production system

$$\boxed{(\alpha+\beta)+\gamma \rightarrow \alpha+(\beta+\gamma)}$$

over E contains just one production which reparenthesizes a sum by associating to the right. For example, the expression $(a+b)+((c+d)+g)$ becomes either $a+(b+((c+d)+g))$ or $(a+b)+(c+(d+g))$, both of which become $a+(b+(c+(d+g)))$. Since the length of the expression remains constant when the production is applied, some other measure is needed to prove termination.

To prove termination, we use the multiset ordering over the natural numbers, $(\mathcal{M}(N),\gg)$, and let $\tau:E\rightarrow\mathcal{M}(N)$ return the multiset of the lengths of all the subexpressions in e to which the production is applicable, i.e.

$$\tau(e) = \{|(\alpha+\beta)+\gamma|:(\alpha+\beta)+\gamma \text{ in } e\}.$$

For example,

$$\tau((a+b)+((c+d)+g)) = \{|(a+b)+((c+d)+g)|,|(c+d)+g|\} = \{9,5\}.$$

1. The value of the termination function τ *decreases* with each application of a production, i.e. for any possible values of α, β, and γ,

$$\tau((\alpha+\beta) \gg \tau(\alpha+(\beta+\gamma)).$$

Before an application of the production, the multiset $\tau((\alpha+\beta)+\gamma)$ includes an occurrence of $|(\alpha+\beta)+\gamma|$, along with elements corresponding to the subexpressions of α, β, and γ. With application of the production, that element is removed; the only element that may be added is $|\beta+\gamma|$ (if β is of the form $(\beta_1+\beta_2)$), which is smaller. The multiset has accordingly been decreased.

2. Since the production does not change the length of the expression it is applied to, i.e.

$$|\pi| = |\pi'|,$$

the length of superexpressions containing $(\alpha+\beta)+\gamma$ is also unchanged.

The multiset $\tau(e)$ consists of all the elements in $\tau((\alpha+\beta)+\gamma)$ plus the lengths of some of their superexpressions and other subexpressions. The only elements in $\tau(e)$ that are changed by the production are those in $\tau((\alpha+\beta)+\gamma)$ and they have been decreased by the production. Thus, $e\Rightarrow e'$ implies that $\tau(e)\gg\tau(e')$. □

EXAMPLE 2: *Differentiation.*

The following system symbolically differentiates an expression with respect to x:

$$\boxed{\begin{array}{l}
Dx \rightarrow 1 \\
Dy \rightarrow 0 \\
D(\alpha+\beta) \rightarrow (D\alpha+D\beta) \\
D(\alpha\cdot\beta) \rightarrow ((\beta\cdot D\alpha) + (\alpha\cdot D\beta)) \\
D(-\alpha) \rightarrow (-D\alpha) \\
D(\alpha-\beta) \rightarrow (D\alpha-D\beta) \\
D(\alpha/\beta) \rightarrow ((D\alpha/\beta) - ((\alpha\cdot D\beta)/(\beta\uparrow 2))) \\
D(\ln\alpha) \rightarrow (D\alpha/\alpha) \\
D(\alpha\uparrow\beta) \rightarrow ((D\alpha\cdot(\beta\cdot(\alpha\uparrow(\beta-1)))) + (((\ln\alpha)\cdot D\beta)\cdot(\alpha\uparrow\beta))) .
\end{array}}$$

We present two solutions. The first uses a multiset ordering; the second uses nested multisets.

•Solution 1.

We use the multiset ordering over sequences of natural numbers. The sequences are compared under the well-founded *stepped lexicographic* ordering $\succ$, i.e. longer sequences are greater than shorter ones (regardless of the values of the individual elements), and equal length sequences are compared lexicographically. The termination function is

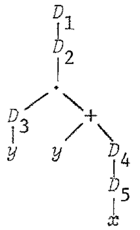
$$\tau(e) = \{(d_1(x), d_2(x), \ldots): \ x \text{ is an occurrence of an atom in } e\},$$

where $d_i(x)$ is the distance (number of operators) between x and the ith enclosing D.

For example, consider the expression

$$e = DD(Dy\cdot(y+DDx)),$$

or in the tree form (with the D's enumerated for expository purposes),



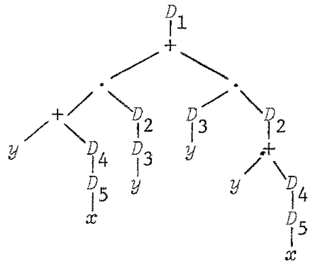
There are three atoms: y, y, and x. The left atom y contributes the element $(0,2,3)$ to the multiset, since there are no operators between D_3 and y, there are two operators ($\cdot$ and D_3) between D_2 and y, and there are three operators (D_2, $\cdot$, and D_3) between D_1 and y. Similarly the other two atoms contribute $(2,3)$ and $(0,1,4,5)$. Thus,

$$\tau(e) = \{(0,2,3),\ (2,3),\ (0,1,4,5)\}.$$

Applying the production

$$D(\alpha\cdot\beta) \rightarrow ((\beta\cdot D\alpha) + (\alpha\cdot D\beta)),$$

to e, yields $e' = D(((y+DDx)\cdot DDy) + (Dy\cdot D(y+DDx)))$. In the tree form (with the labelling of the D's retained), we have



and accordingly

$$\tau(e') = \{(3),(0,1,5),(0,1,4),(0,3),(1,4),(0,1,3,6)\},$$

Thus, $\tau(e) \gg \tau(e')$, since the element (0,1,4,5) has been replaced by five shorter sequences and by the lexicographically smaller (0,1,3,6).

In general, applying any of the productions decreases τ, and the productions only affect the sequences in $\tau(e)$ corresponding to the atoms of the subexpression that they are applied to. Therefore, for any application of a production, $e \Rightarrow e'$ implies $\tau(e) \gg \tau(e')$.

• Solution 2.

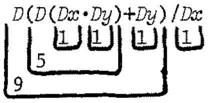
For the alternative solution, we use nested multisets. Note that the arguments to D are reduced in length by each production. One would therefore like to prove termination using the well-founded set $(\mathcal{M}(N), \gg)$ and a termination function that yields the multiset containing the length of the arguments to each occurrence of D, i.e.

$$\tau(e) = \{|\alpha|: \ D\alpha \text{ in } e\}.$$

The value of this function is decreased by the application of a production, i.e. $\tau(\pi) \gg \tau(\pi')$ for each of the productions $\pi \rightarrow \pi'$. The problem is that the size of superexpressions increases, since $|\pi'| > |\pi|$; applying a production to a subexpression of e will therefore increase $\tau(e)$.

To overcome this problem, we need a termination function that takes the nested structure of the expression into consideration and gives more significance to more deeply nested subexpressions. Fortunately, this is exactly what nested multisets can do for us.

Let the well-founded set be the nested multisets over the natural numbers, $(\mathcal{M}^*(N), \gg^*)$, and let the termination function $\tau: E \rightarrow \mathcal{M}^*(N)$ yield $|\alpha|$ for each occurrence of $D\alpha$, while preserving the nested structure of the expression. For example, the arguments of the six occurrences of D in the expression $D(D(Dx \cdot Dy)+Dy)/Dx$ are $D(Dx \cdot Dy)+Dy$, $Dx \cdot Dy$, x, y, y, and x. They are of lengths 9, 5, 1, 1, 1, and 1, respectively. Considering the nested depths of the D's, the structure of the expression is



Thus, for

$$e = D\,(D\,(Dx \cdot Dy) + Dy)\,/\,Dx$$

we have

$$\tau(e) = \{\{9,\{5,\{1\},\{1\}\},\{1\}\},\{1\}\}.$$

For each production $\pi \rightarrow \pi'$, we have $\tau(\pi) \gg^{*} \tau(\pi')$ under the nested multiset ordering. It remains to ascertain what happens to the value of τ for superexpressions. The crucial point here is that the termination function gives greater weight to the more deeply nested D's by placing their length at a greater depth in the nested multiset. The effect of the productions on lower-level expressions is therefore more significant than their effect on higher-level expressions, and the decrease in τ for the subexpression to which the production is applied overshadows any increase in the length of a superexpression.

Consider, for example,

$$D(D(x \cdot x)+y) \Rightarrow D(((x \cdot Dx)+(x \cdot Dx))+y).$$

The value of τ for the expression on the left is $\{\{6,\{3\}\}\}$, while for the right-hand side expression it is $\{\{11,\{1\},\{1\}\}\}$. Note that this represents a decrease in the nested multiset ordering over N, despite the fact that the element 6, corresponding to the length of the top-level argument, has been increased to 11. This is the case since the production has replaced the element $\{3\}$ in the multiset $\{6,\{3\}\}$ by two occurrences of the smaller $\{1\}$, and $\{3\}$ is also greater than 11 - or any number for that matter - on account of its greater depth.

Thus, $e \Rightarrow e'$ implies $\tau(e) \gg^{*} \tau(e')$. □

In this section, we have illustrated the use of multiset and nested multiset ordering in proofs of termination of production systems, by means of examples. Along similar lines, using these orderings, one can give general theorems which express sufficient conditions for the termination of broad classes of production systems.

ACKNOWLEDGMENT

We thank Bob Boyer, John Doner, Chris Goad, John McCarthy, Steve Ness, Amir Pnueli, Adir Pridor, and Richard Weyhrauch for stimulating discussions.

REFERENCES

Floyd, R. W. [1967], *Assigning meanings to programs*, Proc. Symp. in Applied Mathematics, vol. 19 (J. T. Schwartz, ed.), American Mathematical Society, Providence, RI, pp. 19-32.

Gentzen, G. [1938], *New version of the consistency proof for elementary number theory*, The collected papers of Gerhart Gentzen (M. E. Szabo, ed.), North Holland, Amsterdam (1969), pp. 252-286.

Gorn, S. [Sept. 1965], *Explicit definitions and linguistic dominoes*, Proc. Conf. on Systems and Computer Science, London, Ontario, pp. 77-115.

Iturriaga, R. [May 1967], *Contributions to mechanical mathematics*, Ph.D. thesis, Carnegie-Mellon Univ., Pittsburgh, PA.

Knuth, D. E. and P. B. Bendix [1969], *Simple word problems in universal algebras*, Computational Problems in Universal Algebras (J. Leech, ed.), Pergamon Press, Oxford, pp. 263-297.

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

Lipton, R. J. and L. Snyder [Aug 1977], *On the halting of tree replacement systems*, Proc. Conf. on Theoretical Computer Science, Waterloo, Ontario, pp. 43-46.

Manna, Z. and S. Ness [Jan 1970], *On the termination of Markov algorithms*, Proc. Third Hawaii Intl. Conf. on Systems Sciences, Honolulu, HI, pp. 789-792.

Plaisted, D. [July 1978], *Well-founded orderings for proving the termination of rewrite rules*, Memo R-78-932, Dept. of Computer Science, Univ. of Illinois, Urbana, IL.

Plaisted, D. [Oct. 1978], *A recursively defined ordering for proving termination of term rewriting systems*, Memo R-78-943, Dept. of Computer Science, Univ. of Illinois, Urbana, IL.