UIUCDCS-R-78-943

A Recursively Defined Ordering for Proving Termination of Term Rewriting Systems

by

David A. Plaisted

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

September 1978

This research was supported in part by the National Science Foundation under Grant MCS 77-22830.

1. Introduction

Many programs can easily be expressed as sets of term rewriting rules of the form $s_1 \to t_1, \ldots, s_n \to t_n$ where $s_i \to t_i$ indicates that a term of the form s_i can be replaced by a term of the form t_i. For example, the factorial function can be expressed as

$$fact(s(x)) \to s(x)*fact(x)$$

$$fact(0) \to 1.$$

Algebraic manipulation routines can often be expressed in this way also. For example, the following rules will transform a Boolean expression involving $\wedge$, $\vee$, and $\neg$ into disjunctive normal form:

$$\neg(x \wedge y) \to (\neg x) \vee (\neg y)$$

$$\neg(x \vee y) \to (\neg x) \wedge (\neg y)$$

$$\neg\neg x \to x$$

$$x \wedge (y \vee z) \to (x \wedge y) \vee (x \wedge z)$$

To show such programs are correct, one only needs to show that the equations are valid in the intended interpretations, and that the final expression will have some desired form. We are interested in proving <u>termination</u> of such programs. Namely, we want to show that there is no infinite sequence $u_1, u_2, u_3, \ldots$ such that for all i, u_{i+1} can be obtained from u_i by a replacement using a term rewriting rule. It is easy to show [3] that termination is in general an undecidable property of such programs. However, we are interested in methods for proving termination that frequently work in cases of practical interest. These methods are based on well-founded orderings. We presented some work in this area in [9], but at that time we did not have a good way to deal with the replacement

$x*(y + z) \rightarrow x+y + x*z$ and similar replacements. We now present a general proof technique which easily handles such replacements, together with replacements such as those mentioned in [9]. For other work in this area, see [1], [2], [4], [5], [6], [7], [8].

We present an ordering on terms called the *path of subterms ordering*. It is similar to the path ordering defined in [9] but is defined recursively. We show that this is a partial ordering, and then that it is a "partial simplification ordering" as defined in [9]. Proving well-foundedness turns out to be somewhat difficult. We then present efficient methods for computing the ordering on ground terms and on non-ground terms. Finally, we present a general class of replacements $s \rightarrow t$ all of which are simplifications in this ordering. That is, $t < s$ in the ordering. Any term rewriting system all of whose replacements are of this form is guaranteed to terminate. The two examples mentioned above are both of this form.

Given terms s and t, we can in many cases efficiently compute whether $t < s$ in the path of subterms ordering. If $t_i < s_i$ in this ordering for $1 \leq i \leq n$, then the set $s_1 \rightarrow t_1, \ldots, s_n \rightarrow t_n$ of rewrite rules is guaranteed to terminate. This research therefore gives an efficient, simple method for providing proofs of termination of sets of rewrite rules.

2. Simplification Orderings

Definition: A *term* is an expression formed from function symbols, constant symbols, and variables, properly combined. Thus $f(x, g(c, y))$ is a term. We consider constant symbols as function symbols having no arguments. A term without variables in it is a *ground* term.

We use the usual notion of substitution. A substitution is considered as a multiple replacement, simultaneously replacing all variables of a term by terms. Thus $\{x \leftarrow f(x), y \leftarrow g(c)\}$ is a substitution. We assume all function and constant symbols are taken from same set F of symbols.

Definition: We say an ordering "<" is well-founded if there is no infinite sequence $t_1, t_2, t_3, \ldots$ such that $t_i > t_{i+1}$ for all $i \geq 1$. (Such a sequence $t_1, t_2, t_3, \ldots$ is called an infinite descending sequence.)

Definition: A partial ordering "<" on terms is a simplification ordering if it has the following four properties:

1. It is a total ordering on ground terms.
2. It is a well-founded ordering.
3. (Consistency with respect to substitution) If t_1 and t_2 are terms and $t_1 < t_2$ in the ordering, then for all substitutions θ, $t_1\theta < t_2\theta$ in the ordering.
4. (Consistency with respect to subterm replacement) If s_1 is a subterm of t_1, and t_2 is obtained from t_1 by replacing s_1 by s_2, and $s_1 < s_2$ in the ordering, then $t_1 < t_2$ in the ordering.

Intuitively, these are desirable properties for a simplification ordering to have. Property 2 guarantees that the simplification process must terminate. Property 4 guarantees that simplifying a subterm will also simplify the whole term, so simplification can be done "locally".

Definition: A partial ordering on terms is a partial simplification ordering if it has properties 2, 3, and 4 as above. Thus it need not be a total ordering on ground terms.

A replacement is an equation that can only be used in one direction. We write $s_1 \to s_2$ to mean that any instance of s_1 can be replaced by the corresponding instance of s_2.

Definition: A replacement $s_1 \to s_2$ is a simplification with respect to a simplification ordering if s_2 is less than s_1 with respect to the ordering.

Definition: A term t is obtained from term s using the replacement $s_i \to t_i$ if there is some substitution θ with the following properties:

> $s_i\theta$ is a subterm of s, and t is obtained from s by replacing one occurrence of $s_i\theta$ in s by $t_i\theta$.

Note that if $t_i < s_i$ in some partial simplification ordering then $t_i\theta < s_i\theta$ also by consistency with respect to substitution. Hence $t < s$ by consistency with respect to subterm replacement.

We say that a set $s_1 \to t_1$, $s_2 \to t_2$, ..., $s_n \to t_n$ of rewrite rules fails to terminate on input u_1 if there is an infinite sequence u_1, u_2, u_3, ... such that for all $i \geq 1$, u_{i+1} is obtained from u_i using some replacement in the set. If no such infinite sequence exists, we say the set of rewrite rules terminates on input u_1.

Theorem [8] Suppose $s_1 \to t_1$, ..., $s_n \to t_n$ is a set of rewrite rules. Suppose there is a partial simplification ordering "<" such that $t_i < s_i$ in the ordering for $1 \leq i \leq n$. Then the set of rewrite rules terminates on all inputs.

Proof: Assume the rules fail to terminate on input u_1. Let u_1, u_2, u_3, ... be an infinite sequence of terms such that for all $i \geq 1$, u_{i+1} is obtained from u_i by using some replacement in the set of rewrite rules. We showed above that $u_{i+1} < u_i$ in the ordering, for all i.

Hence $u_1, u_2, u_3, \ldots$ is an infinite descending sequence in the ordering. But this is impossible, because a partial simplification ordering is well-founded. Thus the given set of rewrite rules is guaranteed to terminate on all inputs.

3. Multisets

A multiset is a set in which an element can occur more than once. We write multiset union as $\uplus$ or sometimes $\cup$. The number of occurrences of an element in $A \cup B$ is the sum of the number of its occurrences in A and B.

Definition: If S is a multiset of elements ordered by some total order relation, let List(S) be a list $\{s_1, s_2, \ldots, s_m\}$ of the elements of S in non-increasing order. Each element must appear the same number of times in List(S) as in S.

Definition: Suppose "<" is a partial ordering such that there exists an equivalence relation "≡" satisfying the following conditions:

1. $x_1 \equiv x_2$ and $y_1 \equiv y_2$ implies $x_1 < y_1$ iff $x_2 < y_2$.
2. $x_1 \equiv x_2$ iff x_1 and x_2 are unrelated in the ordering (that is, $\neg(x_1 < x_2)$ and $\neg(x_2 < x_1)$).

We then write $x_1 \leq x_2$ to indicate $x_1 < x_2$ or $x_1 \equiv x_2$. Also, we write $[x]$ to refer to the equivalence class of x in this equivalence relation. That is, $[x] = \{y: y \equiv x\}$.

Given such a partial ordering on some set S, we define a partial ordering having the same property, on finite multisets of elements of S as follows:

Suppose U and V are multisets of elements of S. Suppose $U \neq V$. Let $\{u_1, u_2, \ldots, u_k\}$ be List(U) and let $\{v_1, v_2, \ldots, v_\ell\}$ be List(V). If $[u_1], [u_2], \ldots, [u_k]$ is a prefix of $[v_1], [v_2], \ldots, [v_\ell]$ we say $U < V$. If $[v_1], [v_2], \ldots, [v_\ell]$ is a prefix of $[u_1], [u_2], \ldots, [u_k]$ we say $V < U$. Otherwise, let j be $\min\{i: u_i < v_i \text{ or } v_i < u_i\}$.

Then we say $U < V$ if $u_j < v_j$, $U > V$ if $u_j > v_j$. We call this ordering on multisets the ordering induced by the ordering on S.

4. The Path of Subterms Ordering

We now define the path of subterms ordering and show that it is a partial simplification ordering. We assume that there is a total ordering on the function and constant symbols F appearing in terms t that we are dealing with. If $g > f$ in this ordering, we intuitively think of g as being more complicated than f, and we say g is larger than f.

4.1 Definition of the Ordering

Definition: If t is a term of form $f(t_1, \ldots, t_n)$, a path of subterms of t is the sequence consisting of t itself followed by a path of subterms for t_i for some i, $1 \le i \le n$. If t is a variable, then t itself is the only path of subterms for t. Thus a path of subterms for f(g(a, b), c) is the sequence f(g(a, b), c), g(a, b), b. Also, a path of subterms of the term f(x, g(y, c)) is the sequence f(x, g(y, c)), g(y, c), y.

Definition: Let SPaths(t) be the multiset of paths of subterms of t. More precisely, if t is of the form $f(t_1, \ldots, t_n)$ then SPaths(t) is $\{t\} \biguplus_{i=1}^{n}$ SPaths(t_i). If t is a constant symbol c, then SPaths(t) is {c}. If t is a variable, then SPaths(t) is {t}.

Definition: If t is a term of the form $f(t_1, \ldots, t_n)$, let PSPaths(t) be $\biguplus_{i=1}^{n}$ SPaths(t_i).

Definition: If α is a sequence of terms, then Subseq(α) is the multiset of subsequences of α. Thus Subseq($t\beta$) = {t, Λ}Subseq(β) = {t}Subseq(β) $\uplus$ Subseq(β). Also, Subseq(Λ) = Λ. (Here Λ is the sequence of length zero.)

Definition: Suppose t and u are terms. We say t and u are equivalent up to a permutation of arguments if t and u are both variables, or if

a) the top-level function symbols of t and u are identical. (Suppose t is of the form $f(t_1, \ldots, t_n)$ and u is of the form $f(u_1, \ldots, u_n)$), and

b) there is a one-to-one correspondence between the multiset $\{t_1, \ldots, t_n\}$ and $\{u_1, \ldots, u_n\}$ such that if t_i and u_j correspond to each other then t_i and u_j are equivalent up to a permutation of arguments.

Example: f(g(a, b), h(a, c)) and f(h(a, c), g(b, a)) are equivalent up to a permutation of arguments. We write $s \sim t$ if s and t are identical up to a permutation of arguments.

Definition: If t is a term, then Size(t) is the sum of the number of occurrences of all function symbols, constant symbols, and variables in t. For example, Size(f(x, f(x, c))) = 5.

Given ground terms s and t, the following procedure will efficiently determine if s and t are identical up to a permutation of arguments.

Suppose s is of form $f(s_1, \ldots, s_n)$ and t is of form $f(t_1, \ldots, t_n)$. (If the top-level function symbols of s and t are not the same, s and t are clearly not identical up to a permutation of arguments.)

Let s_i' and t_i' be "canonical forms" of s_i and t_i, respectively. We obtain the canonical form recursively in a manner to be described. Let $s_1'' s_2'' \ldots s_n''$ and $t_1'' t_2'' \ldots t_n''$ be a listing of the multisets $\{s_1', \ldots, s_n'\}$ and $\{t_1', \ldots, t_n'\}$, respectively, in non-increasing order. Any total ordering on ground terms will do here; lexicographic ordering is probably the simplest to use.

The canonical form of s is defined to be $f(s_1'', \ldots, s_n'')$ and that of t is defined to be $f(t_1'', \ldots, t_n'')$. If these canonical forms are identical, then s and t are identical up to a permutation of arguments. Otherwise, s and t are not identical up to a permutation of arguments.

Suppose s and t are not ground terms. Let $x_1 \ldots x_m$ be the variables appearing in s and t. Let $c_1 \ldots c_m$ be new constant symbols. Let s* and t* be s and t with x_i replaced by c_i everywhere, for $1 \leq i \leq m$. Then s* and t* are ground terms, and $s^* \sim t^*$ iff $s \sim t$. Hence the above procedure can be applied to s* and t*.

Assuming that the set of function symbols is fixed, the time required to put s in canonical form is $O(\mathrm{Size}(s)^2)$ using this method. The time to test if s and t are identical up to a permutation of arguments is then $O(\mathrm{Size}(s)^2 + \mathrm{Size}(t)^2)$. Later we will give a more complicated method for deciding if $s \sim t$, that has a better asymptotic time bound.

Definition: Suppose $s_1, s_2, \ldots, s_m$ and $t_1, t_2, \ldots, t_n$ are paths of subterms. We write $s_1, s_2, \ldots, s_m \sim t_1, t_2, \ldots, t_n$ iff

a) $m = n$ and

b) for all i, $1 \leq i \leq m$, $s_i \sim t_i$.

In this case, we say that the paths $s_1, s_2, \ldots, s_m$ and $t_1, t_2, \ldots, t_n$ are identical up to a permutation of arguments.

Definition: The intermediate path of subterms ordering on ground terms is defined as follows:

Suppose f is the top-level function symbol of ground term t and g is the top-level function symbol of ground term u. Then $t < u$ in the intermediate path of subterms ordering if $f < g$ in the function symbol ordering.

Suppose both t and u have the same top-level function symbol f. Suppose t is of the form $f(t_1, t_2, \ldots, t_n)$ and u is of the form $f(u_1, u_2, \ldots, u_n)$. In this case, we define the ordering recursively, assuming that we already know how proper subterms of t and u are ordered with respect to each other in the intermediate path of subterms ordering.

Let S be the set of terms r such that r is a proper subterm of t or u. Assume inductively that all elements of S are ordered with respect to each other by the intermediate path of subterms ordering, except those terms which are equivalent up to a permutation of arguments.

Order sequences of elements of S lexicographically. That is, if $\bar{q} = q_1, \ldots, q_\ell$ and $\bar{r} = r_1, \ldots, r_m$ are two sequences of elements of S, then $\bar{q} < \bar{r}$ if

a) $\bar{q}$ is empty and $\bar{r}$ is non-empty

b) $q_1 < r_1$ in the intermediate path of subterms ordering, or

c) q_1 and r_1 are equivalent up to a permutation of arguments, and $(q_2, \ldots, q_\ell) < (r_2, \ldots, r_m)$ in the lexicographic ordering on sequences of elements of S.

Order multisets of sequences of elements of S by the multiset ordering induced by the above ordering on sequences.

If α and β are paths in PSPaths(t) ⊎ PSPaths(u), then order α and β by the above multiset ordering on Subseq(α) and Subseq(β).

We say t < u in the intermediate path of subterms ordering if PSPaths(t) < PSPaths(u) in the multiset ordering induced by the above ordering on paths.

It is not difficult to show that t and u are ordered with respect to each other in this ordering unless they are equivalent up to a permutation of arguments. Thus we obtain by induction the result that ground terms are ordered by this ordering unless they are equivalent up to a permutation of arguments.

Definition: Let t and u be arbitrary ground terms. Then t < u in the path of subterms ordering if SPaths(t) < SPaths(u). That is, we order SPaths(t) and SPaths(u) by the multiset ordering induced by the ordering on paths of subterms. The purpose of this definition is to eliminate the large dependence of the ordering on the top-level function symbols of t and u.

The intuition behind this definition is that we want to weight function symbols more if they occur at the top of a more complex subterm. Thus the * in x*(y + z) would be weighted more highly than either * in x*y + x*z, and so we get that x*(y + z) → x*y + x*z is a simplification in the path of subterms ordering, extended to non-ground terms in the usual way.

Definition: Suppose s and t are terms, possibly with variables in them. Then $s < t$ in the path of subterms ordering if for all substitutions θ such that $s\theta$ and $t\theta$ are ground terms, $s\theta < t\theta$ in the path of subterms ordering on ground terms. Note that this definition guarantees that the path of subterms ordering is consistent with respect to substitution.

4.2 Proof that the Ordering is a Partial Simplification Ordering

We now show that the path of subterms ordering is a partial ordering, is consistent with respect to subterm replacement, and is well-founded. We have just noted that it is consistent with respect to substitution.

Theorem 4.1. The intermediate path of subterms ordering on ground terms is a partial ordering. That is, it is non-reflexive, anti-symmetric, and transitive.

Proof: Suppose t_1 and t_2 and t_3 are ground terms. Let Subterms(t) be the set of proper subterms of t. Let S be Subterms(t_1) $\cup$ Subterms(t_2) $\cup$ Subterms(t_3). Assume inductively that the intermediate path of subterms ordering is a partial ordering on S. Also, note that if $s_1 \in S$ and $s_2 \in S$ and $s_1 \sim s_2$ fails to hold, then either $s_1 < s_2$ or $s_2 < s_1$ in the intermediate path of subterms ordering. It follows that this ordering is a partial ordering on sequences of elements of S, ordered as specified. Hence it is a partial ordering on multisets of sequences of elements of S, ordered by the induced multiset ordering. Hence if t_1, t_2, and t_3 have the same top-level function symbol, the intermediate ordering is a partial ordering on $\{t_1, t_2, t_3\}$. If the top-level function symbols are not the same, it is easy to show that the intermediate ordering is still a partial ordering on $\{t_1, t_2, t_3\}$.

Furthermore, since the intermediate path of subterms ordering is a partial ordering on all sets of three terms, it is a partial ordering on the set of all terms.

Corollary: The path of subterms ordering is a partial ordering.

Proof: The path of subterms ordering on t_1 and t_2 is the same as the intermediate path of subterms ordering on $f(t_1)$ and $f(t_2)$ for some function symbol f.

We now recall some basic properties of the subsequence ordering adapted from [9].

Suppose α and β are sequences of terms. Suppose s and t are terms, and $s \sim t$. Then $\text{Subseq}(\alpha) < \text{Subseq}(\beta)$ iff $\text{Subseq}(s\alpha) < \text{Subseq}(t\beta)$. Also, $\text{Subseq}(\alpha) < \text{Subseq}(\beta)$ iff $\text{Subseq}(\alpha s) < \text{Subseq}(\beta t)$. In addition, if $\alpha \sim \beta$ does not hold, then either $\text{Subseq}(\alpha) < \text{Subseq}(\beta)$ or $\text{Subseq}(\beta) < \text{Subseq}(\alpha)$. It follows that if α is a proper prefix or a proper suffix of β, then $\text{Subseq}(\alpha) < \text{Subseq}(\beta)$.

Suppose α and β are sequences of terms. Suppose $\alpha = \alpha_1 s \alpha_2$ and $\beta = \beta_1 t \beta_2$ where α_1, α_2, β_1, and β_2 are sequences of terms and s and t are terms. Suppose s is a maximal term in α and t is a maximal term in β in the intermediate path of subterms ordering. Suppose s does not occur in α_1 and t does not occur in β_1. Then $\text{Subseq}(\alpha) < \text{Subseq}(\beta)$ iff any of the following conditions are true:

1. $s < t$ in the intermediate path of subterms ordering
2. $s \sim t$ and $\text{Subseq}(\alpha_2) < \text{Subseq}(\beta_2)$
3. $s \sim t$ and $\alpha_2 \sim \beta_2$ and $\text{Subseq}(\alpha_1) < \text{Subseq}(\beta_1)$.

Theorem 4.2 Suppose s and t are ground terms and s is a strict subterm of t. Then $s < t$ in the path of subterms ordering.

Proof: We know that s < t iff SPaths(s) < SPaths(t) in the multiset ordering. For every path α in SPaths(s), there is a path β in SPaths(t) such that α is a proper suffix of β. Hence $\alpha < \beta$ in the ordering on paths of subterms. Hence the maximal element of SPaths(s) is less than the maximal element of SPaths(t) in the ordering on paths of subterms. Hence SPaths(s) < SPaths(t), and s < t in the path of subterms ordering.

Theorem 4.3 Suppose s and t are ground terms and s is a strict subterm of t. Suppose that the top-level function symbols of s and t are identical. Then s < t in the intermediate path of subterms ordering.

Proof: We know that s < t in this ordering if PSPaths(s) < PSPaths(t) in the ordering on multisets of paths of terms. But PSPaths(s) < PSPaths(t), by an argument similar to that used in the above theorem.

Theorem 4.4 Suppose t is a ground term of form $f(t_1, \ldots, t_n)$. Suppose t_j' is a ground term which is simpler than t_j in the path of subterms ordering, for some j, $1 \leq j \leq n$. Let t' be the term $f(t_1, \ldots, t_{j-1}, t_j', t_{j+1}, \ldots, t_n)$. Then t' < t in the path of subterms ordering.

Proof: We know that t' < t iff SPaths(t') < SPaths(t). Now, SPaths(t') = {t'}PSPaths(t') and SPaths(t) = {t}PSPaths(t).

We first show that t' < t in the intermediate path of subterms ordering. Since t' and t have the same top-level function symbol, t' < t in this ordering iff PSPaths(t') < PSPaths(t). But

$$\mathrm{PSPaths}(t') = \left[\biguplus_{\substack{i=1 \\ i \neq j}}^{n} \mathrm{SPaths}(t_i)\right] \uplus \mathrm{SPaths}(t_j') \text{ and}$$

$\mathrm{PSPaths}(t) = \biguplus_{i=1}^{n} \mathrm{SPaths}(t_i)$. Hence PSPaths(t') < PSPaths(t) iff $\mathrm{SPaths}(t_j') < \mathrm{SPaths}(t_j)$. But this is true because $t_j' < t_j$ in the path of subterms ordering.

We now show that t' < t in the path of subterms ordering. We need to show that {t'}PSPaths(t') < {t}PSPaths(t). This is true because t' < t in the intermediate path of subterms ordering and because PSPaths(t') < PSPaths(t) as shown above. This completes the proof.

Corollary 1: Suppose t is a ground term and s is a subterm of t. Suppose s' is a ground term and s' < s in the path of subterms ordering. Suppose t' is t with an occurrence of s replaced by s'. Then t' < t in the path of subterms ordering.

Proof: By induction, using the above theorem.

Corollary 2: Suppose t is an arbitrary term, possibly with variables in it. Suppose s is a subterm of t. Suppose s' is another term and s' < s in the path of subterms ordering. Suppose t' is t with an occurrence of s replaced by s'. Then t' < t in the path of subterms ordering.

Proof: Using Corollary 1 and the definition of the path of subterms ordering on non-ground terms.

4.2.1 Well-Foundedness

We now show that the path of subterms ordering is well-founded.

Theorem 4.5. Suppose α is a maximal path of subterms in SPaths(t) for ground term t. Suppose α is $t_1, t_2, \ldots, t_n$. (Thus each t_i is a top-level subterm of t_{i-1}, for $2 \le i \le n$, and t_1 is t and t_n is a constant.) Then for all i, $2 \le i \le n$, the path $t_i, t_{i+1}, \ldots, t_n$ is a maximal path of subterms in PSPaths(t_{i-1}).

Proof: If not, we could replace $t_i, t_{i+1}, \ldots, t_n$ by a larger path to get a larger element of SPaths(t).

Theorem 4.6. Suppose t_1 and t_2 are ground terms and α_1 and α_2 are maximal paths in SPaths(t_1) and SPaths(t_2), respectively. Suppose $\alpha_1 = \beta_1 v_1 \gamma_1$ and $\alpha_2 = \beta_2 v_2 \gamma_2$ where β_1, β_2, γ_1, γ_2 are sequences of terms and v_1, v_2 are terms. Suppose that v_1 and v_2 are the same to within a permutation of arguments. Then

a) γ_1 and γ_2 are the same to within a permutation of arguments, and

b) $\alpha_1 < \alpha_2$ in the path ordering iff $\beta_1 < \beta_2$ in the path ordering.

Proof: By the above result, γ_1 and γ_2 are maximal elements of SPaths(v_1) and SPaths(v_2), respectively. Since v_1 and v_2 are the same to within a permutation of arguments, so are γ_1 and γ_2. Therefore α_1 and α_2 have essentially the same suffix (namely, $v_1\gamma_1$ or $v_2\gamma_2$). Hence Subseq(α_1) < Subseq(α_2) iff $\beta_1 < \beta_2$. This completes the proof.

Definition: If α is a path of subterms, let mtf(α) be the maximum top-level function symbol of any term in α, in the function symbol ordering.

Definition: If α is a path of subterms, let mt(α) be the first term in α whose top-level function symbol is mtf(α).

Theorem 4.7. The term mt(α) is the largest term in α in the intermediate path of subterms ordering.

Proof: The largest term in α in this ordering must have mtf(α) as its top-level function symbol, by definition of the intermediate path of subterms ordering. Also, if v_1 and v_2 both occur in α and both have f as their top-level function symbol, and if v_1 occurs before

v_2 in α, then v_2 is a proper subterm of v_1 and so $v_2 < v_1$ in the intermediate path of subterms ordering by Theorem 4.3. This completes the proof.

Theorem 4.8. Suppose t_1 and t_2 are ground terms and α_1 and α_2 are maximal paths in SPaths(t_1) and SPaths(t_2), respectively. Suppose $\alpha_1 = \beta_1 v_1 \gamma_1$ and $\alpha_2 = \beta_2 v_2 \gamma_2$ where β_1, β_2, γ_1, γ_2 are sequences of terms and v_1, v_2 are terms. Suppose that v_1 and v_2 are the maximal terms in α_1 and α_2, respectively, in the intermediate path of subterms ordering. Then

a) if $v_1 < v_2$ in the intermediate path of subterms ordering, then $\alpha_1 < \alpha_2$ in the path ordering, and

b) if v_1 and v_2 are identical to within a permutation of arguments, then $\alpha_1 < \alpha_2$ in the path ordering iff $\beta_1 < \beta_2$ in the path ordering.

Proof: The first part follows because Subseq(α_1) < Subseq(α_2) if the maximal element of α_1 is less than the maximal element of α_2. The second part follows by theorem 4.6.

Note that this result, together with the previous result, gives us a reasonably fast way to decide whether $\alpha_1 < \alpha_2$ in the path ordering, assuming we know how all terms in α_1 and α_2 are ordered in the intermediate path of subterms ordering.

Definition: If s is a ground term and α is a path of subterms of s, define the stepping sequence of α to be the sequence $v_1, v_2, \ldots,$ of elements of α defined as follows:

a) v_1 is mt(α)

b) For $1 \leq i < m$, let α_i be the portion of α up to but not including v_i. Then v_{i+1} is mt(α_i) for $1 \leq i < m$.

c) $v_n = s$.

<u>Theorem 4.9</u>. Suppose s and t are ground terms and α and β are paths of subterms from s and t, respectively. Let $s_1, \ldots, s_m$ and $t_1, \ldots, t_n$ be the stepping sequences of α and β, respectively. Let j be min{i: s_i and t_i are not the same to within a permutation of arguments}, if it exists. Then

1. $\alpha < \beta$ in the path ordering iff
   a) the stepping sequence of α is a proper prefix of that of β or
   b) j as defined above exists and $s_j < t_j$ in the intermediate path of subterms ordering.
2. If j as defined above exists, then for all $i \geq j$ such that $i \leq \min(m, n)$, s_i and t_i are <u>not</u> the same to within a permutation of arguments.

<u>Proof</u>: A combination of previous results.

Note that this allows us to compare paths in time proportional to the number of distinct function symbols, assuming that the stepping sequences are known and that we can in constant time compare terms in the paths in the intermediate path of subterms ordering. In fact, we can do it in $O(\log|F|)$ time, where F is the set of function symbols, by doing binary search and making use of observation 2.

<u>Theorem 4.10</u>. Suppose s and t are ground terms. Then $s < t$ in the path of subterms ordering iff

a) s and t have the same top-level function symbol and $s < t$ in the intermediate path of subterms ordering or
b) s and t have different top-level function symbols and $\alpha < \beta$ in the path ordering, where $\alpha = \max(\mathrm{SPaths}(s))$ and $\beta = \max(\mathrm{SPaths}(t))$.

In fact, even if a) is true, $\alpha < \beta$ where $\alpha = \max(\text{SPaths}(s))$ and $\beta = \max(\text{SPaths}(t))$. However, since α begins with s and β with t, we need to know the intermediate path of subterms ordering on s and t in order to compute the ordering on α and β.

For the remainder of this section, we will allow the set of function symbols to be infinite. We require the function symbol ordering to be total and well-founded. The use of infinite sets of function symbols will be useful for other applications (not discussed in this paper).

We write the sequence $t_1, t_2, t_3, \ldots$ as $\{t_i\}_i$. Thus $t_1^i, t_2^i, t_3^i, \ldots$ is written $\{t_j^i\}_j$, and $t_j^1, t_j^2, t_j^3, \ldots$ is written $\{t_j^i\}_i$.

Definition: Suppose $\{a_i\}_i$ is an infinite sequence of arbitrary objects. We say a is the limit of $\{a_i\}_i$ if there exists k such that for all $i \geq k$, $a_i = a$.

Definition: Suppose $\{t_i\}_i$ is an infinite sequence of terms. Suppose $i_1, i_2, i_3, \ldots$ is a monotone increasing sequence of positive integers. Suppose $\{u_j\}_j$ is a sequence of terms such that for all $j \geq 1$, u_j is a proper subterm of t_{i_j}. Then we say $\{u_j\}_j$ is a subsequence of proper subterms of $\{t_i\}_i$.

We now show that every infinite descending sequence of ground terms has an infinite descending subsequence of proper subterms, all with the same top-level function symbol. Let $\{t_i\}_i$ be an infinite descending sequence of ground terms. Then there must exist an infinite descending sequence $\{\alpha_i\}_i$ of paths such that $\alpha_i \in \text{SPaths}(t_i)$ for all $i \geq 1$. By considering the stepping sequences of the α_i, and making use of the fact that the function symbol ordering is well-founded, we can show that there is an infinite descending sequence $\{v_j\}_j$ of terms such that all v_j have the same top-level function symbol and such that for some $n \geq 0$, for all $j \geq 1$, v_j occurs on the path α_{j+n}. By considering elements of PSPaths(v_j) for $j \geq 1$, we can show that $\{v_j\}_j$ has an infinite decreasing subsequence of proper subterms, all with the same top-level function symbol. Hence $\{t_i\}_i$ has an infinite descending subsequence of proper subterms, all with the same top-level function symbol.

Definition: Suppose t is a ground term and f is a function symbol. Then $msub_f(t) = max\{u$: u is a proper subterm of t and the top-level function symbol of u is greater than or equal to f$\}$, if some such subterm u exists. By "max" we mean a maximal term in the path of subterms ordering, chosen in some consistent way.

Definition: Suppose t is a ground term and f is a function symbol. Then $Msub_f(t)$ is $max\{u$: u is a subterm of t and the top-level function symbol of u is f or larger than f$\}$, if some such subterm u exists. Thus $Msub_f(t)$ is the same as $msub_f(t)$ except that we do not require $Msub_f(t)$ to be a proper subterm of t. In fact, if the top-level function symbol of t is f or larger than f, then $Msub_f(t) = t$. Also, if the top-level function symbol of t is smaller than f, then $Msub_f(t)$ exists iff $msub_f(t)$ does, and in that case $Msub_f(t) = msub_f(t)$.

In particular, $Msub_f(s)$ exists iff s contains a function symbol larger than or equal to f, and $msub_f(s)$ exists iff some proper subterm of s contains a function symbol larger than or equal to f.

Definition: Suppose $\{t_i\}_i$ is an infinite sequence of ground terms and f is a function symbol. Suppose $u_i \sim Msub_f(t_i)$ for all $i \geq 1$.

Then we call $\{u_i\}_i$ a <u>main sequence of subterms</u> of $\{t_i\}_i$. Usually we say $\{u_i\}_i$ is a <u>main sequence</u> of $\{t_i\}_i$. Note that all maximal elements of SPaths(t_i) contain a term equivalent to u_i up to a permutation of arguments.

For convenience, we say u is <u>equivalent to</u> v if $u \sim v$.

<u>Definition</u>: Suppose $\{t_i\}_i$ is an infinite sequence of terms. Then f is a <u>top-level function symbol</u> for $\{t_i\}_i$ if f is the top-level function symbol of infinitely many elements of the sequence $\{t_i\}_i$. That is, $\{i$: f is the top-level function symbol of $t_i\}$ is infinite.

<u>Definition</u>: Suppose $\{t_i\}_i$ is an infinite descending sequence of ground terms. Then the <u>coarseness</u> of $\{t_i\}_i$ is the smallest function symbol g having the following properties:

1. For all function symbols $h \geq g$, no subsequence $\{u_j\}_j$ of proper subterms of $\{t_i\}_i$ which has h as a top-level function symbol, is an infinite descending sequence.
2. The set of function symbols $h \geq g$ such that for some subsequence $\{u_j\}_j$ of proper subterms of $\{t_i\}_i$, the following are true, is finite:
   a) $\{u_j\}_j$ is an infinite descending sequence.
   b) $\{u_j\}_j$ has a main sequence with h as a top-level function symbol.

We call a function symbol g having properties 1. and 2. a <u>coarseness bound</u> for $\{t_i\}_i$. We will show that if there is an infinite descending sequence whose coarseness is g, then there is an infinite descending sequence whose coarseness is less than g.

We now show that the coarseness of any infinite descending sequence of ground terms is well-defined. Let $\{t_i\}_i$ be an infinite descending sequence of ground terms, and let g_i be the maximal function symbol in t_i. Then $\{g_i\}_i$ is an infinite non-increasing sequence, which must reach a limiting value since the function symbol ordering is a well-founded ordering. Let this limit be g. Let h be the smallest function symbol larger than g, if it exists. Then h is a coarseness bound for $\{t_i\}_i$. Hence there is a smallest coarseness bound for $\{t_i\}_i$, since the symbol ordering is well-founded. This smallest coarseness bound is then the coarseness of $\{t_i\}_i$. If no such h exists (i.e., g is the maximal symbol in the set of function symbols), we can add such an h to the set of symbols in order to define coarseness uniformly.

Lemma 1: Suppose $\{t_i\}_i$ is an infinite descending sequence of ground terms. Let h be a function symbol. Let W be any subset of the set of terms w satisfying the following properties:

1. For some subsequence $\{u_j\}_j$ of proper subterms of $\{t_i\}_i$, w occurs infinitely often in some main sequence of $\{u_j\}_j$.
2. The top-level function symbol of w is h.

If W has no minimal element, then there is an infinite descending sequence $\{v_j\}_j$ having the following properties:

1. $\{v_j\}_j$ is a subsequence of proper subterms of $\{t_i\}_i$.
2. The top-level function symbol of v_j is h, for all $j \geq 1$.

Proof of lemma: Let v_1 be an arbitrary element of W. For $j \geq 1$, let v_{j+1} be an element of W smaller than v_j. Since W has no smallest element in the path of subterms ordering, such a term v_{j+1} always exists. Then $\{v_j\}_j$ is an infinite descending sequence, all of whose elements have h as the top-level function symbol. Also, since each v_j occurs as a proper subterm of t_i for infinitely many i, $\{v_j\}_j$ is a subsequence of proper subterms of $\{t_i\}_i$. (No v_j can occur in the sequence $\{t_i\}_i$ itself, since $\{t_i\}_i$ is a descending sequence. Hence the v_j must all be proper subterms of t_i for infinitely many i.)

Lemma 2: Suppose s_1 and s_2 are ground terms, and f is a function symbol. Suppose s_1 and s_2 have the same top-level function symbol (not necessarily f). Then

a) If $msub_f(s_1)$ and $msub_f(s_2)$ exist and $s_1 \geq s_2$ in the path of subterms ordering, $msub_f(s_1) \geq msub_f(s_2)$ in the path of subterms ordering.

b) If $msub_f(s_1)$ exists and $msub_f(s_2)$ does not exist, then $s_1 > s_2$ in the path of subterms ordering.

c) If $msub_f(s_1)$ exists, then something equivalent to $msub_f(s_1)$ occurs on a maximal element of $SPaths(s_1)$.

Proof: By considering the stepping sequences of maximal elements of $PSPaths(s_1)$ and $PSPaths(s_2)$, respectively.

Also, if s_1 and s_2 have different top-level function symbols, both smaller than f, and $s_1 \geq s_2$, and $msub_f(s_1)$ and $msub_f(s_2)$ exist, then $msub_f(s_1) \geq msub_f(s_2)$.

Definition: Suppose t is a ground term and f is a function symbol. Let $msub'_f(t)$ be max{u: u is a proper subterm of t and the top-level function symbol of u is larger than f}, if some such subterm u exists.

We can show that $msub'_f(t)$ satisfies all three properties given in lemma 2 for $msub_f(t)$, in a similar way.

<u>Lemma 3</u>: Suppose s_1 and s_2 are ground terms, and f is a function symbol. Then

a) If $Msub_f(s_1)$ and $Msub_f(s_2)$ exist and $s_1 \geq s_2$ in the path of subterms ordering, then $Msub_f(s_1) \geq Msub_f(s_2)$ in the path of subterms ordering.

b) If $Msub_f(s_1)$ exists and $Msub_f(s_2)$ does not exist, then $s_1 > s_2$ in the path of subterms ordering.

c) If $Msub_f(s_1)$ exists, then $Msub_f(s_1)$ or something equivalent to it up to a permutation of arguments, occurs on a maximal element of $SPaths(s_1)$.

<u>Proof</u>: By considering the stepping sequences of maximal paths of s_1 and s_2, respectively.

<u>Definition</u>: Suppose t is a ground term and f is a function symbol. Then $Msub_f'(t)$ is max{u: u is a subterm of t and the top-level function symbol of u is strictly <u>larger</u> than f}, if some such subterm u exists. We can show that $Msub_f'(t)$ has properties similar to $Msub_f(t)$, by similar arguments.

<u>Lemma 4</u>: Suppose f and g are two function symbols, and $f < g$. Suppose t is a ground term, and $Msub_g(t)$ exists. Then $Msub_g(t) \sim Msub_g(Msub_f(t))$.

<u>Proof</u>: $Msub_f(t) \leq t$ hence $Msub_g(Msub_f(t)) \leq Msub_g(t)$. However, $Msub_f(t) \geq Msub_g(t)$ so $Msub_g(Msub_f(t)) \geq Msub_g(Msub_g(t)) = Msub_g(t)$. Thus $Msub_g(t) \sim Msub_g(Msub_f(t))$. Similarly, $Msub_g(t) \sim Msub_g(Msub_f'(t))$.

Lemma 5: Suppose that s_1 and s_2 are ground terms and g is a function symbol. Suppose the top-level function symbols of s_1 and s_2 are both smaller than g. Also, suppose $s_1 \geq s_2$ in the path of subterms ordering and that $msub_g(s_1)$ and $msub_g(s_2)$ exist. Let f1 be the largest function symbol such that $Msub_{f1}(s_1)$ exists and has a proper subterm t_1 such that $t_1 \sim msub_g(s_1)$. Similarly, let f2 be the largest function symbol such that $Msub_{f2}(s_2)$ exists and has a proper subterm t_2 such that $t_2 \sim msub_g(s_2)$. Then either $msub_g(s_1) > msub_g(s_2)$ or $f1 \geq f2$.

Proof: We know by lemma 2 that $msub_g(s_1) \geq msub_g(s_2)$. Suppose $msub_g(s_1) \sim msub_g(s_2)$ and $f_1 < f_2$. Then $Msub_{f2}(s_1) \sim msub_g(s_1)$ and $Msub_{f2}(s_2) > msub_g(s_2)$. Hence $Msub_{f2}(s_2) > Msub_{f2}(s_1)$ and so $s_2 > s_1$, contrary to hypothesis.

Lemma 6: Let T^i for $i \geq 1$ represent the sequence $t_1^i, t_2^i, t_3^i, \ldots$ of ground terms. Suppose that T^i is an infinite descending sequence for all $i \geq 1$, and that the top-level function symbol of t_j^i is f for all $i \geq 1$, $j \geq 1$. Suppose that T^{i+1} is a subsequence of proper subterms of T^i, for all $i \geq 1$. Then for some i, some $t \in T^i$, $msub_f'(t)$ exists and for some $j \geq 1$, $\delta \geq 1$, $msub_f'(t) \geq t_j^{i+\delta}$.

Proof: For all $i \geq 1$, $j \geq 1$ there exists $\ell \geq 0$, $m \geq 1$ such that t_m^{i+1} is a proper subterm of $t_{j+\ell}^i$. Hence $msub_f(t_{j+\ell}^i)$ exists. Therefore $msub_f(t_j^i)$ must also exist. In addition, $msub_f(t_{j+\ell}^i) \geq t_m^{i+1}$. Also, $msub_f(t_j^i) \geq msub_f(t_{j+1}^i) \geq \ldots \geq msub_f(t_{j+\ell}^i)$. Hence $msub_f(t_j^i) \geq t_m^{i+1}$. We have shown that for all $i \geq 1$, for all $j \geq 1$, $msub_f(t_j^i)$ exists and for some $m \geq 1$, $msub_f(t_j^i) \geq t_m^{i+1}$.

Let z_{ij}^o be t_j^i for all $i \geq 1$, $j \geq 1$. Let z_{ij}^{k+1} be $msub_f(z_{ij}^k)$ for all $k \geq 1$ such that $msub_f(z_{ij}^k)$ exists and such that the top-level function symbol of $msub_f(z_{ij}^k)$ is f. Let n_{ij} be the maximum ℓ such that z_{ij}^ℓ is thus defined. Let n be $\min\{n_{ij}: i \geq 1, j \geq 1\}$. Note that if $n \geq 1$, then

$z^1_{ij} \geq z^1_{i,j+1}$; if $n \geq 2$, then $z^2_{ij} \geq z^2_{i,j+1}$. In general, if $n \geq k$, then $z^k_{ij} \geq z^k_{i,j+1}$.

We showed above that for all i, j there exists $m_1 \geq 1$ such that $msub_f(t^i_j) \geq t^{i+1}_{m_1}$. Similarly, there exists $m_2 \geq 1$ such that $msub_f(t^{i+1}_{m_1}) \geq t^{i+2}_{m_2}$. Hence if $n \geq 1$, $msub_f(msub_f(t^i_j)) \geq t^{i+2}_{m_2}$. In general, if $n \geq k$ then for all i, j there exists $m \geq 1$ such that $z^k_{ij} \geq z^o_{i+k,m}$. In particular, choose i and j so that $n_{ij} = n$; then there exists $m \geq 1$ so that $z^n_{ij} \geq z^o_{i+n,m}$. Now, $z^o_{i+n,m} = t^{i+n}_m$ and $msub_f(t^{i+n}_m)$ must exist, as we showed earlier. Hence $msub_f(z^n_{ij})$ exists, and therefore has top-level function symbol strictly <u>larger</u> than f. Also, for some $\ell \geq 1$, $msub_f(t^{i+n}_m) \geq t^{i+n+1}_\ell$ as we showed earlier. Hence $msub'_f(z^n_{ij}) = msub_f(z^n_{ij}) \geq msub_f(z^o_{i+n,m}) \geq t^{i+n+1}_\ell$. Thus $msub'_f(t^i_j) \geq t^{i+n+1}_\ell$

<u>Theorem 4.11</u>: Suppose $\{t_i\}_i$ is an infinite descending sequence of ground terms. Suppose the coarseness of $\{t_i\}_i$ is g. Then $\{t_i\}_i$ has an infinite descending subsequence of proper subterms whose coarseness is less than g.

<u>Proof</u>: Let H be the set of function symbols h such that $h \geq g$ and such that for some infinite descending subsequence $\{u_j\}_j$ of proper subterms of $\{t_i\}_i$, the sequence $\{Msub_g(u_j)\}_j$ has h as a top-level function symbol. We know that H is finite, by definition of coarseness. For $h \in H$, let w_h be a minimal term having h as a top-level function symbol such that w_h occurs infinitely often in some such main sequence $\{Msub_g(u_j)\}_j$. We know that w_h exists for $h \in H$ by lemma 1. Let w be $\min\{w_h: h \in H\}$. We have two cases.

Case 1: Some infinite descending subsequence $\{u_j\}_j$ of proper subterms of $\{t_i\}_i$ has no main sequence with any element of H as a top-level function symbol. Let f be the largest symbol occurring infinitely often in $\{u_j\}_j$. Then $\{u_j\}_j$ has a main sequence with f as a top-level symbol. We know that $f < g$ since $f \notin H$.

Case 1a: Suppose that $\{u_j\}_j$ itself or some subsequence $\{v_j\}_j$ of proper subterms of $\{u_j\}_j$ has the following properties:

a) $\{v_j\}_j$ is an infinite descending sequence.

b) $\{v_j\}_j$ has no infinite descending subsequence of proper subterms with f as a top-level function symbol.

We show that f is a coarseness bound for $\{v_j\}_j$. Since no function symbol larger than f occurs in infinitely many u_j, f satisfies the first part of the definition of a coarseness bound. The second part is true for the same reason.

Case 1b: The sequence $\{u_j\}_j$ and every infinite descending subsequence of proper subterms of $\{u_j\}_j$ has an infinite descending subsequence of proper subterms with f as a top-level function symbol. In this case, we can show that there exist sequences $U^i = \{u^i_j\}_j$ of ground terms with the following properties, for all $i \geq 1$:

a) U^i is an infinite descending sequence.

b) U^{i+1} is a subsequence of proper subterms of U^i.

c) For all $i \geq 1$, $j \geq 1$, the top-level function symbol of u^i_j is f, and no function symbol larger than f appears in u^i_j.

d) U^1 is a subsequence of proper subterms of $\{u_j\}_j$.

By lemma 6, for some i, some $u \in U^i$, $msub'_f(u)$ exists. However, this cannot be true, by property c) above.

Case 2: Assume case 1 does not apply. Then some infinite descending subsequence $\{u_j\}_j$ of proper subterms of $\{t_i\}_i$ has w occurring infinitely often in the sequence $\{Msub_g(u_j)\}_j$. Since a main sequence must be non-increasing, some suffix of $\{u_j\}_j$ has $Msub_g(u) \sim w$ for all u in the suffix. Assume without loss of generality that $w \sim Msub_g(u_j)$ for all $j \geq 1$. If v is a proper subterm of u_j, then $Msub_g(v) \leq Msub_g(u_j)$. Hence if $\{v_j\}_j$ is an infinite descending subsequence of proper subterms of $\{u_j\}_j$, then $Msub_g(v_j) \sim w$ for all $j \geq 1$. Hence w, w, w ... is a main sequence of $\{v_j\}_j$ also.

Let f be the smallest function symbol such that $Msub'_f(u_1) \sim Msub_g(u_1)$. It is easy to show that f occurs in u_1 and that $f < g$. Also, $\{Msub'_f(u_j)\}_j$ is an infinite non-increasing sequence, hence $Msub'_f(u_j) \sim Msub_g(u_j) \sim w$ for all $j \geq 1$. Similarly, if $\{v_j\}_j$ is an infinite descending subsequence of proper subterms of $\{u_j\}_j$, then $Msub'_f(v_j) \sim Msub_g(v_j) \sim w$ for all $j \geq 1$. We have two subcases to consider:

Case 2a: The sequence $\{u_j\}_j$ itself or some subsequence $\{v_j\}_j$ of proper subterms of $\{u_j\}_j$ has the following properties:

a) $\{v_j\}_j$ is an infinite descending sequence.

b) No infinite descending subsequence of proper subterms of $\{v_j\}_j$ has f as a top-level function symbol.

In this case, it is easy to show that f is a coarseness bound for $\{v_j\}_j$. Hence the coarseness of $\{v_j\}_j$ is strictly less than that of $\{t_j\}_j$.

Case 2b: The sequence $\{u_j\}_j$ itself and every infinite descending subsequence of proper subterms of $\{u_j\}_j$ has an infinite descending subsequence of proper subterms with f as a top-level function symbol. In this case, we can show that there exist sequences $U^i = \{u_j^i\}_j$ of ground terms with the following properties for all $i \geq 1$:

a) U^i is an infinite descending sequence.

b) U^{i+1} is a subsequence of proper subterms of U^i.

c) For all $i \geq 1$, $j \geq 1$, the top-level function symbol of u_j^i is f.

d) U^1 is a subsequence of proper subterms of $\{u_j\}_j$.

By lemma 6, for some $i \geq 1$, for some $u \in U^i$, $msub_f'(u)$ exists and for some $j \geq 1$, $\delta \geq 1$, $msub_f'(u) \geq u_j^{i+\delta}$. However, $msub_f'(u) \sim Msub_f'(u) \sim w$ so $w \geq u_j^{i+\delta}$. Also, $Msub_f'(u_j^{i+\delta}) \sim w$ and $w < u_j^{i+\delta}$. This is because $u_j^{i+\delta}$ has f as a top-level function symbol, which implies that $Msub_f'(u_j^{i+\delta})$ is a proper subterm of $u_j^{i+\delta}$. Hence $w \geq u_j^{i+\delta}$ and $w < u_j^{i+\delta}$, contradiction. Thus case 2b cannot be true. This completes the proof.

<u>Corollary 1</u>: The path of subterms ordering on ground terms is well-founded.

<u>Proof</u>: If not, an infinite descending sequence of ground terms would exist. Then we could construct an infinite descending sequence of coarsenesses, using the theorem. However, this is impossible since the function symbol ordering is a well-founded ordering.

<u>Corollary 2</u>: The path of subterms ordering on arbitrary terms is a well-ordering.

5. Computing the Ordering on Ground Terms

We now give a reasonably efficient procedure for computing the path of subterms ordering on ground terms. In fact, the procedure will sort a set of ground terms in non-decreasing order in the path of subterms ordering. The procedure will also determine which of the terms are equivalent to within a permutation of arguments. Later we present methods for computing the ordering on non-ground terms. The algorithm makes use of ideas and notation from [9].

Theorem 5.1. Suppose s and t are ground terms of the form $f(s_1, \ldots, s_n)$ and $f(t_1, \ldots, t_n)$, respectively. Suppose $\alpha \in$ PSPaths(s) and $\beta \in$ PSPaths(t) and α and β are identical to within a permutation of arguments. Then there exists i, j, $1 \le i \le n$, $1 \le j \le n$ such that s_i and t_j are identical to within a permutation of arguments.

Proof: Choose i, j such that s_i is the first term in the path α and t_j is the first term in the path β. Since α and β are identical to within a permutation of arguments, so are s_i and t_j.

Corollary: Let α_i be the maximal element of SPaths(s_i) for $1 \le i \le n$ and let β_i be the maximal element of SPaths(t_i) for $1 \le i \le n$. Let A be the multiset $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ and let B be the multiset $\{\beta_1, \beta_2, \ldots, \beta_n\}$. Let $\{\gamma_1, \gamma_2, \ldots, \gamma_n\}$ be a listing of A in non-increasing order and let $\{\delta_1, \delta_2, \ldots, \delta_n\}$ be a listing of B in non-increasing order. Assume that s, t are not identical to within a permutation of arguments. Then s < t in the intermediate path of subterms ordering iff $\gamma_j < \delta_j$ where $j = \min\{i\colon \gamma_i, \delta_i$ are not identical to within a permutation of arguments$\}$.

<u>Proof</u>: $PSPaths(s) = \Psi_{i=1}^{n} SPaths(s_i)$ and $PSPaths(t) = \Psi_{i=1}^{n} SPaths(t_i)$. Let $\{u_1, \ldots, u_n\}$ be a listing of $\{s_1, \ldots, s_n\}$ such that $\gamma_i = \max(SPaths(u_i))$ and let $\{v_1, \ldots, v_n\}$ be a listing of $\{t_1, \ldots, t_n\}$ such that $\delta_i = \max(SPaths(v_i))$. Then u_i and v_i are identical to within a permutation of arguments, for all $i < j$. Hence the ordering of PSPaths(s) and PSPaths(t) is the same as the ordering of $\Psi_{i=j}^{n} SPaths(u_i)$ and $\Psi_{i=j}^{n} SPaths(v_i)$. But γ_j is a maximal element of $\Psi_{i=j}^{n} SPaths(u_i)$ and δ_j is a maximal element of $\Psi_{i=j}^{n} SPaths(v_i)$. Also, we know that γ_j and δ_j are not identical to within a permutation of arguments. Hence $\Psi_{i=j}^{n} SPaths(u_i) < \Psi_{i=j}^{n} SPaths(v_i)$ iff $\gamma_j < \delta_j$. Hence $s < t$ in the intermediate path of subterms ordering iff $\gamma_j < \delta_j$.

The following procedure will output a multiset $\{t_1, \ldots, t_p\}$ of ground terms in non-decreasing order in the path of subterms ordering. It actually outputs all subterms of t_i, $1 \le i \le p$, along with the terms t_i themselves, in non-decreasing order. For purposes of this algorithm, we consider different occurrences of the same subterm to be different subterms. We assume as usual that the function and constant symbols of the terms t_i are identified with the integers $\{1, 2, 3, \ldots, k\}$. The symbol ordering is then the usual arithmetic ordering on integers. Thus k is the largest function symbol and 1 is the smallest. We say a term t is <u>eligible</u> if all proper subterms of t have been output already but t itself has not been output. We make use of queues $Q_1, Q_2, \ldots, Q_k$ with queue operations indicated as usual. Thus $t \leftarrow Q_i$ means remove an element from the front of Q_i and assign it to t. Also, $Q_i \leftarrow t$ means add t to the back of queue Q_i. By "Delete S from Q_i" we mean to remove elements of S from Q_i, wherever they occur in Q_i (not necessarily at the front). The meaning of "Delete T from Q_j" is similar.

The running time of this algorithm is $O(L(\log D + \log k))$ where L is $\sum_{i=1}^{p}$ Size(t_i) and D is the maximum number of occurrences of any subterm in $\{t_1, t_2, \ldots, t_p\}$, including constants. Note that $D \leq L$. The factor logD appears because of the necessity to sort the list T. Perhaps the algorithm can be restructured to avoid this factor of logD. This analysis assumes that all subterms can be output in unit time. The whole subterm does not actually need to be written out, but only enough information to identify it. Hence this assumption seems to be realistic. To make this sorting possible, we keep integer indices with each subterm telling its position in the queue relative to other subterms.

```
procedure sortt2({t_1, t_2, ..., t_p});
   for i = 1 to k do Q_i ← Λ;
   for i = 1 to p do
      for all subterms t of t_i do
         (f ← top-level function symbol of t;
          Q_f ⇐ t);
   while (∃j)(Q_j contains an eligible term) do
      (i ← min{j: Q_j contains an eligible term};
         [use a priority queue to find i]
      S ← the multiset of elements of Q_i all of which are identical
          to the first eligible term in Q_i, up to a permutation of
          arguments;
      Delete S from Q_i;
      for j = 1 to k do
      if S ≠ ∅ then
      repeat [T ← list of the multiset of elements of Q_j having at
           least one element of S as a top-level subterm. Also,
           T is sorted in the same order as the elements were in Q_j;
           Delete T from Q_j and append it to the end of Q_j;
           for all t ∈ T(let s ∈ S be some top-level subterm of t.
           Delete s from S and output s)] until T = Λ);
end sortt2;
```

The reason that this algorithm works is the following:

Let $Q_1Q_2...Q_k$ be a concatenation of the queues, listed from front to back. At the start of the "while" statement, the following is always true:

1. If u and v are both eligible and u occurs before v in the list $Q_1Q_2...Q_k$, then the maximal path of subterms of u is less than that of v, or else u and v are identical up to a permutation of arguments.
2. Suppose u and v are arbitrary terms in Q_j and u occurs before v in Q_j. Let $\{u_1, ..., u_m\}$ be the multiset of top-level subterms of u that have already been output, and let $\{v_1, ..., v_n\}$ be the multiset of top-level subterms of v that have already been output. Then $\{u_1, ..., u_m\} \leq \{v_1, ..., v_n\}$ in the multiset ordering induced by the path of subterms ordering on terms.
3. Eligible terms occur in non-decreasing order in the path of subterms ordering in the list $Q_1Q_2...Q_k$.

It is not difficult to show that 1. implies 3. By 3., we know that terms will be output in the desired order. By 1., we know that the procedure sortt2 simulates the path ordering algorithm "sortb" of [9] that scans paths from back to front. Thus we know that terms will be output in non-decreasing order according to the ordering on their maximal paths of subterms. By 2., we know that whenever a term becomes eligible, it will be in the proper place in the list $Q_1Q_2...Q_k$. Also, it is not difficult to see that 2. is preserved by the operations within the "repeat" statement. We show this as follows:

Suppose u occurs in Q_j and has an element s of S as a top-level subterm. Then, since s is larger in the path of subterms ordering than any term output so far, the multiset $\{s, u_1, \ldots, u_m\}$ will be larger than any multiset of a term in Q_j <u>not</u> having an element of S as a top-level subterm. Also, if u and v <u>both</u> have s as a top-level subterm, then the ordering on $\{s, u_1, \ldots, u_m\}$ and $\{s, v_1, \ldots, v_n\}$ is the same as the ordering on $\{u_1, \ldots, u_m\}$ and $\{v_1, \ldots, v_n\}$. Hence deleting T from Q_j, appending it (appropriately ordered) to the end of Q_j and outputing an appropriate sub-multiset of S preserves condition 2 for Q_j. We may have to repeat more than once for Q_j if some term has <u>more than one</u> element of S as a top-level subterm. Note that non-eligible terms may appear before eligible terms in Q_j. For example, if no top-level subterms of u have been output, then u may appear first in Q_j.

We have not specified how to determine if two terms are identical up to a permutation of arguments. We can easily modify the above algorithm to keep track of which subterms are identical to within a permutation of arguments. To do this, we keep a bit with each term v as follows:

Suppose $v \in Q_j$ and $\{v_1, \ldots, v_n\}$ is the multiset of top-level subterms of v output so far. Suppose $u \in Q_j$ and u occurs immediately before v in the queue. Let $\{u_1, \ldots, u_m\}$ be the multiset of top-level subterms of u that have been output so far. Then the bit for v is 1 if $\{v_1, \ldots, v_n\}$ is identical to $\{u_1, \ldots, u_m\}$ up to a permutation of arguments, and the bit is 0 otherwise. That is, the bit is 1 if $m = n$ and $u_i \sim v_i$ for $1 \le i \le m$, and 0 otherwise, assuming that $\{u_1, \ldots, u_m\}$ and $\{v_1, \ldots, v_n\}$ are both listed in non-increasing order. It is easy to update these bits when top-level subterms of u or v are output, making use of the fact that all elements of S are identical up to a permutation of arguments. In particular, $\{v_1, \ldots, v_n\} \sim \{u_1, \ldots, u_m\}$ unless at some time so far, T has included one of u and v but not the other. Also, if u and v are both eligible, and u occurs just before v in a queue, then $u \sim v$ iff the bit for v is 1. Hence the set S can easily be obtained using these bits. Finally, two subterms of terms in $\{t_1, t_2, \ldots, t_p\}$ are equivalent up to a permutation of arguments iff they are output in the same set S. Thus we can decide if two terms are identical up to a permutation of arguments in time $O(L(\log D + \log k))$, the same time complexity as the algorithm to sort terms in the path of subterms ordering.

6. Computing the Ordering on Non-ground Terms

We now give algorithms to compute the path of subterms ordering on terms with variables in them. We do not yet have a completely general procedure but we present methods that will work on many replacements occurring in practice, and give ideas for extending these methods. Our methods always terminate and if they yield an answer it is correct. However, sometimes the algorithm returns "don't know."

Suppose α and β are Paths of subterms and $mtf(\alpha) < mtf(\beta)$ in the symbol ordering. Then $\alpha < \beta$ in the path ordering, since the maximal term in α will be less than the maximal term in β in the intermediate path of subterms ordering.

<u>Definition</u>: If α is a path of subterms and f is a function or constant symbol, let $\#_f(\alpha)$ be the number of terms in α whose top-level function symbol is f.

<u>Theorem 6.1.</u> Suppose α and β are maximal paths of subterms from ground terms s and t, respectively. Suppose that $mtf(\alpha) = mtf(\beta)$ and $\#_f(\alpha) < \#_f(\beta)$ where $f = mtf(\alpha)$. Then $\alpha < \beta$ in the path ordering.

<u>Proof</u>: Let f be $mtf(\alpha)$. Now, α and β will be ordered by the intermediate path of subterms ordering on $mt(\alpha)$ and $mt(\beta)$, if $mt(\alpha)$ and $mt(\beta)$ are not identical to within a permutation of arguments. Let u and v be $mt(\alpha)$ and $mt(\beta)$, respectively. (Perhaps u is s or v is t.) Let α' be max(PSPaths(u)) and let β' be max(PSPaths(v)). Note that α' and β' are suffixes of α and β, respectively. Also, $\#_f(\alpha') = \#_f(\alpha) - 1$ and $\#_f(\beta') = \#_f(\beta) - 1$. Hence $\#_f(\alpha') \neq \#_f(\beta')$ so α' and β' are not identical to within a permutation of arguments. Hence u and v are not identical to within a permutation of arguments. Also, $u < v$ in the intermediate path of subterms ordering iff $\alpha' < \beta'$ in the path ordering.

If $\#_f(\alpha') = 0$ then $\alpha' < \beta'$ in the path ordering since $mtf(\alpha') < mtf(\beta')$. If $\#_f(\alpha') \neq 0$ then $\#_f(\alpha') < \#_f(\beta')$ and so we can apply the theorem inductively to show that $\alpha' < \beta'$ in the path ordering. Hence $u < v$ in the intermediate path of subterms ordering, and so $\alpha < \beta$ in the path ordering, as desired.

<u>Theorem 6.2.</u> Suppose s and t are two ground terms and α and β are maximal paths of subterms from s and t, respectively. Let $s_1, s_2, \ldots, s_m$ be the maximal <u>descending</u> subsequence of α and let $t_1, t_2, \ldots, t_n$ be the maximal <u>descending</u> subsequence of β. Let f_i be the top-level function symbol of s_i for $1 \leq i \leq m$ and let g_i be the top-level function symbol of t_i for $1 \leq i \leq n$. Then $\alpha < \beta$ in the path ordering if the sequence $f_1 f_2 \ldots f_m$ is less than the sequence $g_1 g_2 \ldots g_n$ in the lexicographic ordering.

This theorem can be extended to give a general method of comparing maximal paths, as follows:

Definition: Suppose that s and t are two ground terms and α and β are arbitrary paths of subterms from s and t, respectively. Then the surface ordering on α and β is defined as follows:

Let γ_1 and γ_2 be the longest possible suffixes of α and β, respectively, such that $\gamma_1 \sim \gamma_2$. Let α be $\alpha'\gamma_1$ and let β be $\beta'\gamma_2$. Let $s_1, s_2, \ldots, s_m$ be the maximal descending sequence of α' and let $t_1, t_2, \ldots, t_n$ be the maximal descending subsequence of β'. Let f_i be the top-level function symbol of s_i for $1 \leq i \leq m$ and let g_i be the top-level function symbol of t_i for $1 \leq i \leq n$. Then $\alpha < \beta$ in the surface ordering iff

a) the sequence $f_1 f_2 \ldots f_m$ is less than the sequence $g_1 g_2 \ldots g_n$ in the lexicographic ordering on sequences of function symbols, or

b) $m = n$ and f_i is identical to g_i for $1 \leq i \leq m$, and $s_m < t_m$ in the path of subterms ordering on terms.

(Note that s_m and t_m are not the same to within a permutation of arguments because s_m occurs immediately before γ_1 in α and t_m occurs immediately before γ_2 in β. Hence either $s_m < t_m$ or $s_m > t_m$ in the path of subterms ordering on terms.)

Theorem 6.3. Suppose that s_1 and s_2 are two ground terms and α_1 and α_2 are maximal paths of subterms from s_1 and s_2, respectively. (By "maximal" we mean maximal in the subsequence ordering.) Suppose $\alpha_1 > \alpha_2$ in the subsequence ordering. Then $\alpha_1 > \alpha_2$ in the surface ordering.

<u>Proof</u>: Let α_1 be $\alpha_1'\gamma_1$ and let α_2 be $\alpha_2'\gamma_2$ where $\gamma_1 \sim \gamma_2$ and γ_1 and γ_2 are the longest suffixes of α_1 and α_2 with this property. Assume inductively that the theorem is true for all paths of subterms δ_1, δ_2 as in the theorem for which $\delta_1 < \alpha_1$ and $\delta_2 < \alpha_2$ in the subsequence ordering.

Let α_1' be $\beta_1 v_1 \beta_1'$ where $v_1 = mt(\alpha_1')$ and let α_2' be $\beta_2 v_2 \beta_2'$ where $v_2 = mt(\alpha_2')$. We cannot have $v_1 \sim v_2$ by definition of γ_1 and γ_2. Hence $v_1 > v_2$ in the intermediate path of subterms ordering.

Let f_1 and f_2 be the top-level function symbols of v_1 and v_2, respectively. If f_1 and f_2 are not identical, then $f_1 > f_2$. It follows that $\alpha_1 > \alpha_2$ in the surface ordering.

Suppose that f_1 and f_2 are identical. Since γ_1 and γ_2 are the longest suffixes of α_1 and α_2 such that $\gamma_1 \sim \gamma_2$, we cannot have $\beta_1' \sim \beta_2'$ unless β_1' and β_2' are both empty. If β_1' and β_2' are empty, the surface ordering on α_1 and α_2 is the path of subterms ordering on v_1 and v_2. Since $v_1 > v_2$ in the path of subterms ordering, $\alpha_1 > \alpha_2$ in the surface ordering.

Suppose β_1' and β_2' are not both empty. Then we cannot have $\beta_1' \sim \beta_2'$. Hence we cannot have $\beta_1'\gamma_1 \sim \beta_2'\gamma_2$. Now, $\beta_1'\gamma_1$ is a maximal element of PSPaths(v_1) and $\beta_2'\gamma_2$ is a maximal element of PSPaths(v_2) in the subsequence ordering. Since $v_1 > v_2$ in the path of subterms ordering, $\beta_1'\gamma_1 > \beta_2'\gamma_2$ in the subsequence ordering. We can assume by induction, then, that $\beta_1'\gamma_1 > \beta_2'\gamma_2$ in the surface ordering. Hence $\alpha_1 > \alpha_2$ in the surface ordering. This completes the proof.

<u>Proof</u>: Let α_1 be $\alpha_1'\gamma_1$ and let α_2 be $\alpha_2'\gamma_2$ where $\gamma_1 \sim \gamma_2$ and γ_1 and γ_2 are the longest suffixes of α_1 and α_2 with this property. Assume inductively that the theorem is true for all paths of subterms δ_1, δ_2 as in the theorem for which $\delta_1 < \alpha_1$ and $\delta_2 < \alpha_2$ in the subsequence ordering.

Let α_1' be $\beta_1 v_1 \beta_1'$ where $v_1 = mt(\alpha_1')$ and let α_2' be $\beta_2 v_2 \beta_2'$ where $v_2 = mt(\alpha_2')$. We cannot have $v_1 \sim v_2$ by definition of γ_1 and γ_2. Hence $v_1 > v_2$ in the intermediate path of subterms ordering.

Let f_1 and f_2 be the top-level function symbols of v_1 and v_2, respectively. If f_1 and f_2 are not identical, then $f_1 > f_2$. It follows that $\alpha_1 > \alpha_2$ in the surface ordering.

Suppose that f_1 and f_2 are identical. Since γ_1 and γ_2 are the longest suffixes of α_1 and α_2 such that $\gamma_1 \sim \gamma_2$, we cannot have $\beta_1' \sim \beta_2'$ unless β_1' and β_2' are both empty. If β_1' and β_2' are empty, the surface ordering on α_1 and α_2 is the path of subterms ordering on v_1 and v_2. Since $v_1 > v_2$ in the path of subterms ordering, $\alpha_1 > \alpha_2$ in the surface ordering.

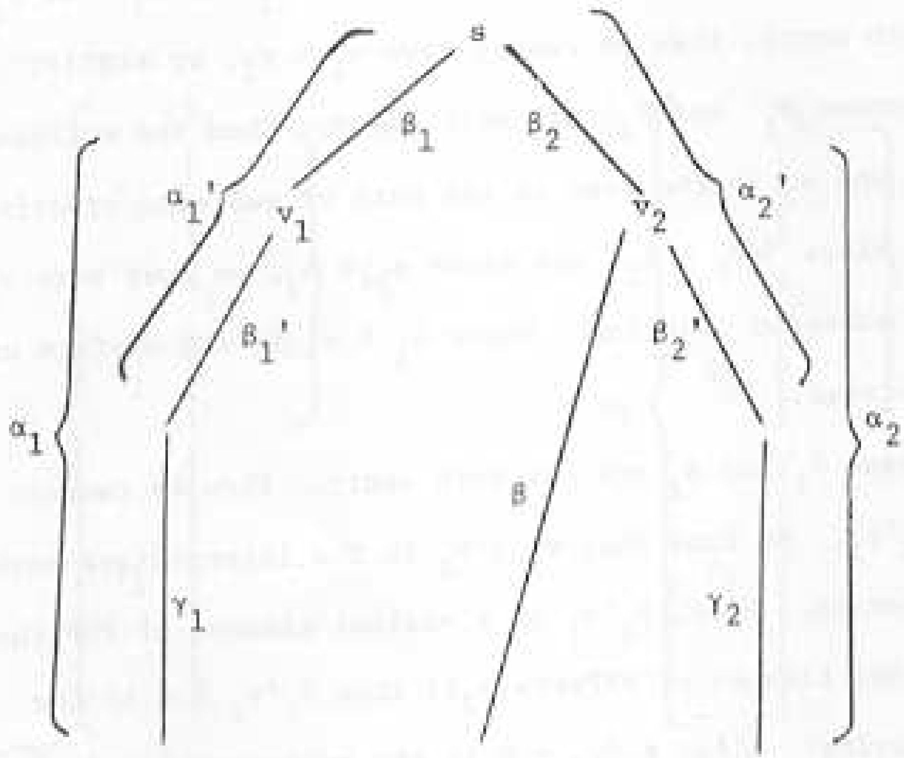
Suppose β_1' and β_2' are not both empty. Then we cannot have $\beta_1' \sim \beta_2'$. Hence we cannot have $\beta_1'\gamma_1 \sim \beta_2'\gamma_2$. Now, $\beta_1'\gamma_1$ is a maximal element of PSPaths(v_1) and $\beta_2'\gamma_2$ is a maximal element of PSPaths(v_2) in the subsequence ordering. Since $v_1 > v_2$ in the path of subterms ordering, $\beta_1'\gamma_1 > \beta_2'\gamma_2$ in the subsequence ordering. We can assume by induction, then, that $\beta_1'\gamma_1 > \beta_2'\gamma_2$ in the surface ordering. Hence $\alpha_1 > \alpha_2$ in the surface ordering. This completes the proof.

Suppose f_1 and f_2 are identical. We cannot have $\beta_1'\gamma_1 \sim \beta_2'\gamma_2$ unless β_1' and β_2' are both empty, by choice of γ_1 and γ_2. If β_1' and β_2' are both empty, then we cannot have $v_1 \sim v_2$, by similar reasoning. Suppose β_1' and β_2' are both empty. Then the surface ordering on α_1 and α_2 is the same as the path of subterms ordering on v_1 and v_2. Since $\neg(v_1 \sim v_2)$ and since $\alpha_1 > \alpha_2$, we must have $v_1 > v_2$ in the path of subterms ordering. Hence $\alpha_1 > \alpha_2$ in the surface ordering on paths of subterms.

Suppose β_1' and β_2' are not both empty. Then we cannot have $\beta_1'\gamma_1 \sim \beta_2'\gamma_2$. We know that $v_1 \geq v_2$ in the intermediate path of subterms ordering. Also, $\beta_1'\gamma_1$ is a maximal element of PSPaths(v_1). Let β be a maximal element of PSPaths(v_2); thus $\beta_1'\gamma_1 \geq \beta$ in the subsequence ordering. Also, $\beta_2'\gamma_2 \leq \beta$ in the subsequence ordering. If $\beta_1'\gamma_1 \sim \beta$ then $\beta_2'\gamma_2 < \beta$ and we can apply the theorem inductively to obtain that $\beta_2'\gamma_2 < \beta_1'\gamma_1$ in the surface ordering. Hence $\alpha_1 > \alpha_2$ in the surface ordering.

If $\beta_1'\gamma_1 > \beta$ then we know that $\beta_1'\gamma_1 > \beta$ in the surface ordering since $\beta_1'\gamma_1$ and β are maximal paths of v_1 and v_2, respectively. Also, applying the theorem inductively to β and $\beta_2'\gamma_2$ we obtain that $\beta \geq \beta_2'\gamma_2$ in the surface ordering. Hence $\beta_1'\gamma_1 > \beta_2'\gamma_2$ in the surface ordering and $\alpha_1 > \alpha_2$ in the surface ordering. This completes the proof.

Diagram:



<u>Theorem 6.5.</u> Suppose s and t are ground terms. Then s > t in the path of subterms ordering iff SPaths(s) > SPaths(t) in the multiset ordering induced by the surface ordering on paths of subterms.

<u>Proof</u>: Suppose s and t have different top-level function symbols. Let α be the maximal element of SPaths(s) in the subsequence ordering, and let β be the maximal element of SPaths (t) in the subsequence ordering. Then α > β in the subsequence ordering. Now, α and β are also maximal elements of SPaths(s) and SPaths(t) in the surface ordering, by Theorem 6.4. Also, α > β in the surface ordering by Theorem 6.3. Hence SPaths(s) > SPaths(t) in the multiset ordering induced by the surface ordering on paths of subterms.

Suppose s and t have the same top-level function symbol f. Let s be of the form $f(s_1, \ldots, s_n)$ and let t be of the form $f(t_1, \ldots, t_n)$. Let S be the multiset $\{s_1, \ldots, s_n\}$ and let T be the multiset $\{t_1, \ldots, t_n\}$. Let S1 and T1 be maximal sub-multisets of S and T satisfying the following condition:

There is a 1-1 correspondence between elements of S1 and elements of T1 such that if $\alpha \in S1$, $\beta \in T1$ and α corresponds to β then $\alpha \sim \beta$.

Since s and t have the same top-level function symbol, we know that PSPaths(s) > PSPaths(t) in the multiset ordering induced by the subsequence ordering on paths of subterms. (That is, s > t in the intermediate path of subterms ordering.) It follows that $\Psi\{SPaths(s_i):s_i \in S-S1\} > \Psi\{SPaths(t_i):t_i \in T-T1\}$. Also, the <u>maximal</u> elements of these multisets of paths differ, and so by reasoning as before we get that $\Psi\{SPaths(s_i):s_i \in S-S1\} > \Psi\{SPaths(t_i):t_i \in T-T1\}$ in the multiset ordering induced by the surface ordering on paths of subterms. Hence, by choice of S1 and T1, PSPaths(s) > PSPaths(t) in the multiset ordering induced by the surface ordering on paths of subterms. Since s > t in the intermediate path of subterms ordering, SPaths(s) > SPaths(t) in the multiset ordering induced by the surface ordering on paths of subterms. This completes the proof.

This result implies that it does not matter whether we use the surface ordering or the subsequence ordering on paths of subterms, when computing the path of subterms ordering on ground terms. The surface ordering is often easier to work with, especially when dealing with non-ground terms, so we sometimes use it instead of the subsequence ordering on paths.

<u>Definition</u>: Suppose that s and t are two not necessarily ground terms and α and β are paths of subterms from s and t, respectively. Suppose that α', β', γ_1, γ_2, $s_1, \ldots, s_m$, $t_1, \ldots, t_n$, $f_1, \ldots, f_m$, $g_1, \ldots,$ are as in the definition of the surface ordering for paths from ground terms. Then $\alpha < \beta$ in the surface ordering iff

a) the sequence $f_1 f_2 \ldots f_m$ is less than the sequence $g_1 g_2 \cdots g_n$ in the lexicographic ordering, or

b) the sequence $f_1 f_2 \ldots f_m$ is identical to the sequence $g_1 g_2 \cdots g_n$, and $s_m < t_m$ in the path of subterms ordering.

The only difference from the previous definition is that s_m and t_m need not be ground terms. Note that if $\alpha < \beta$ in the surface ordering as defined here, then for all substitutions θ, $\alpha\theta < \beta\theta$ in the surface ordering.

Suppose we are given terms t_1 and t_2, possibly containing variables, and we want to decide if $t_1 < t_2$ in the path of subterms ordering. Given paths α_1 and α_2 we write $R_2(\alpha_1, \alpha_2)$ to denote that $\alpha_1 < \alpha_2$ in the surface ordering. We write $R_1(\alpha_1, \alpha_2)$ to denote the following relation:

Let γ_1 and γ_2 be maximal suffixes of α_1 and α_2, respectively, such that $\gamma_1 \sim \gamma_2$. Let α_1 be $\alpha_1'\gamma_1$ and let α_2 be $\alpha_2'\gamma_2$. Let $f_1, f_2, \ldots, f_m$ be top-level function symbols obtained from α_1' as in the definition of the surface ordering, and let $g_1, g_2, \ldots, g_n$ be top-level function symbols obtained from α_2' in the same way. We say that $R_1(\alpha_1, \alpha_2)$ is true iff the sequence $f_1 f_2 \ldots f_m$ is less than the sequence $g_1 g_2 \cdots g_n$ in the lexicographic ordering. This completes the definition of R_1. Note that $R_1(\alpha_1, \alpha_2)$ implies $R_2(\alpha_1, \alpha_2)$ but not vice versa.

Let T_1 and T_2 be maximal sub-multisets of SPaths(t_1) and SPaths(t_2), respectively, satisfying the following condition:

There is a 1-1 correspondence between T_1 and T_2 such that

for all $\alpha_1 \in T_1$, if $\alpha_2 \in T_2$ corresponds to α_1 then $\alpha_1 \sim \alpha_2$.

Let $x_1 \ldots x_k$ be the variables occurring in t_1 and t_2. For $1 \le j \le k$, $i \in \{1, 2\}$, let S_j^i be $\{\alpha : \alpha x_j \in \text{SPaths}(t_i) - T_i\}$. Thus S_j^i is the set of paths in SPaths(t_i) $- T_i$ that end in the variable x_j, with the x_j deleted from the end.

Theorem 6.6. If for all $\alpha \in \text{SPaths}(t_1)-T_1$ there exists $\beta \in \text{SPaths}(t_2)-T_2$ such that $R_2(\alpha,\beta)$, then $t_1 < t_2$ in the path of subterms ordering.

Theorem 6.7. With notation as above, if for some j, $1 \le j \le k$, for all $\alpha \in S_j^1$ there exists $\beta \in S_j^2$ such that $R_2(\alpha, \beta)$, then t_2 is not less than t_1 in the path of subterms ordering. (However, t_1 and t_2 may be unrelated in the ordering.)

In many cases, the above results can be used to efficiently determine whether $t_1 < t_2$, $t_2 < t_1$, or t_1 and t_2 are unrelated in the path of subterms ordering. In some cases, the above theorems will not yield any information about the relative ordering of t_1 and t_2.

A direct application of these theorems requires that we recursively compute, for some subterms s_1 of t_1 and s_2 of t_2, whether $s_1 < s_2$ in the path of subterms ordering. This is because of the definition of the surface ordering. We can avoid this and gain efficiency by using R_1 instead of R_2 in the above theorems. Since $R_1(\alpha, \beta) \supset R_2(\alpha, \beta)$, the theorems are still true, but now yield less information than before. In fact, with this modification we can no longer show that x*y + x*z is simpler than x*(y+z) in the path of subterms ordering.

7. A General Characterization of a Class of Simplifications

We have the following characterization of a frequently occurring class of replacements, all of which are simplifications in the path of subterms ordering.

Theorem 7. Suppose t_1, t_2, $\ldots t_n$ and t are not necessarily ground terms such that $t_i < t$ in the path of subterms ordering, for $1 \leq i \leq n$. Suppose that $\{f_1, f_2, \ldots, f_k\}$ are function and constant symbols, all of which are less than the top-level function symbol of t in the symbol ordering. Let u be any term formed from any number of occurrences of t_1, t_2, ..., t_n and any number of occurrences of the symbols $\{f_1, f_2, \ldots, f_k\}$. Then $u < t$ in the path of subterms ordering.

Example: Let t be x*(y+z) and let t_1 be x*y, t_2 be x*z, and f_1 be "+". Assume "+" < "*" in the symbol ordering. We have $t_1 < t$ and $t_2 < t$ in the path of subterms ordering. Let u be (x*z) + (y*z). By the above theorem, u < t in the path of subterms ordering. Thus $x*(y+z) \rightarrow x*z + y*z$ is a simplification in the path of subterms ordering.

8. Examples

We now give some examples of systems of rewrite rules whose termination can be proven using the path of subterms ordering. That is, all of the replacements are simplifications in the path of subterms ordering, when the function symbol ordering is as specified.

Example 1: Let the function symbol ordering be "**" > "*" > binary "-" > unary "-" > "+" > 1 > 0 and assume 2 is represented as 1 + 1, 3 as (1 + 1)+1 etc. Here "**" represents exponentiation.

```
x + 0 → x
x - y → x + (-y)
-(-(x)) → x
x + (-x) → 0
(-x) + x → 0
-(x + y) → (-x) + (-y)

x * 1 → x
1 * x → x
x*(-y) → -(x*y)
(-x) * y → -(x*y)

x*(y + z) → x*y + x*z
(y+z)*x → y*x + z*x

x**1 → x
x**(y+1) → x*(x**y)

(x*y)**z → (x**z)*(y**z)
x**(y+z) → (x**y)*(x**z)

(x+y)**2 → x**2 + 2*x*y + y**2
(x+y)**3 → x**3 + 3*(x**2)*y + 3*x*(y**2) + (y**3)

et cetera
```

We could also allow the conditional replacement <u>if</u> x ≠ 0 <u>then</u> x**0 → 1, since x**0 → 1 is a simplification. However, we are not able to include the following replacements:

```
(x**y)**z → x**(y*z)
x+(y+z) → (x+y)+z
x*(y*z) → (x*y)*z
```

<u>Example 2</u>: For this example, it suffices to choose any ordering in which "D" is the largest function symbol. This example is intended to illustrate some of the operations of symbolic differentiation.

```
D(c) → 0
D(c*x) → c*D(x)
D(x+y) → D(x) + D(y)
D(x*y) → x*D(y) + y*D(x)
D(x/y) → (y*D(x) - x*D(y)) / (y**2)

D(x**n) → n*(x**(n-1)*D(x)
```

In fact, these replacements, together with those of Example 1, can be proven to terminate by choosing the ordering of Example 1 with "D" added as more complex than any other symbol.

Example 3: This example illustrates definitions of primitive recursive functions, a topic to which we shall return later. Assume "fact" > "*" > "+" > "s" > "0".

```
fact(s(x)) → s(x)*fact(x)
fact(0) → s(0)
s(x)*y → y+(x*y)
0*y → 0
s(x)+y → s(x+y)
0+y → y
```

Example 4: This example illustrates an application of the path of subterms ordering to LISP functions. Here pairs $((x_1, \ldots, x_m), (y_1, \ldots, y_n))$ is the multiset $\{cons(x_i, y_j) : 1 \le i \le m, 1 \le j \le n\}$ represented as a list. Order function symbols by "pairs" > "pairl" > "append" > "cons" > "NIL".

```
pairs(cons(x, y), z) → append(pairl(x, z), pairs(y, z))
pairs(NIL, z) → NIL
pairl(x, cons(y, z)) → cons(cons(x, y), pairl(x, z))
pairl(x, NIL) → NIL
pairs(append(x, y), z) → append(pairs(x, z), pairs(y, z))
pairs(x, append(y, z)) → append(pairs(x, y), pairs(x, z))
append(NIL, z) → z
append(cons(x, y), z) → cons(x, append(y, z))
```

Example 5: We show the termination of a set of rewrite rules for converting a Boolean formula to disjunctive normal form. Assume "≡" > "⊃" > "¬" > "∧" > "∨" > (all propositional variables).

$$x \equiv y \to (x \wedge y) \vee (\neg x \wedge \neg y)$$
$$x \supset y \to (\neg x) \vee y$$
$$\neg(x \wedge y) \to (\neg x) \vee (\neg y)$$
$$\neg(x \vee y) \to (\neg x) \wedge (\neg y)$$
$$\neg\neg x \to x$$
$$x \wedge (y \vee z) \to (x \wedge y) \vee (x \wedge z)$$

The idea given in Example 3 of defining a primitive recursive function by a set of simplifications in the path of subterms ordering works in general. We now discuss relations between the path of subterms ordering and the class of primitive recursive functions.

Definition: Suppose T is a set of rewrite rules. Let R_T be the relation on terms defined by $R_T(s, t)$ iff a) t can be obtained from s by a finite sequence of replacements using the rewrite rules in T and b) no other terms can be obtained from t by using the rewrite rules in T. Suppose that if $R_T(s, t_1)$ and $R_T(s, t_2)$ are both true, and s is a ground term, then t_1 and t_2 are identical. In this case, we call the mapping from ground term s into term t such that $R_T(s, t)$, the function computed by T.

Theorem 8: Suppose we represent non-negative integer n as $s^n(0)$. Thus 1 is represented as s(0), 2 as s(s(0)), et cetera. Then for every primitive recursive function h, there is a set T of rewrite rules, all of which are simplifications in the path of subterms ordering, such that h is the function computed by T. That is, $h(a_1, \ldots, a_k) = b$ iff $R_T(\text{"}h(s^{a_1}(0), \ldots, s^{a_k}(0))\text{"}, \text{"}s^b(0)\text{"})$ is true.

Proof: By definition of primitive recursive functions and properties of the path of subterms ordering. We order the primitive recursive function symbols in the order that they are defined. Thus functions defined last are considered more complex in the symbol ordering. Note that if $g < f$ in the symbol ordering, then the replacement $f(x_1, \ldots, x_n, s(y)) \to g(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y))$ is a simplification in the path of subterms ordering by the preceding characterization.

We conjecture that if T is a set of rewrite rules all of which are simplifications in the path of subterms ordering, and if T computes some function h, then h is primitive recursive. If true, this would give us a new characterization of the primitive recursive functions. We could try to prove this by bounding the number of replacements that can be done on a starting term s, using a bound that is primitive recursive in s. A consequence of this conjecture would be that the path of subterms ordering could never be sufficient to prove termination of rewrite rules for computing a non-primitive recursive function. However, since non-primitive recursive functions cannot be computed in polynomial (or exponential) time, this does not seem to be much of a limitation in practice.

9. More Results

The following results may be of some interest in devising better ways to compute the path of subterms ordering for terms with variables in them.

Theorem 9. Suppose s and t are two not necessarily ground terms, such that there exist substitutions θ_1 and θ_2 such that $s\theta_1$ and $t\theta_1$ are identical to within a permutation of arguments but $s\theta_2 < t\theta_2$ in the path of subterms ordering. Also, suppose that for no substitution θ do we have $s\theta > t\theta$ in the path of subterms ordering. Then either

a) s is the minimal constant symbol and t is a variable or

b) s is of form $f(s_1, \ldots, s_n)$ and t is of form $f(t_1, \ldots, t_n)$ for some function symbol f, and there exist i, j with $1 \leq i, j \leq n$ such that s_i and t_j also satisfy the hypotheses of the theorem.

<u>Proof</u>: If s is a variable, then this variable must occur in t or else we could find θ such that $s\theta > t\theta$. However, if this variable occurs in t then either s and t are identical, in which case θ_2 cannot exist, or else s is a proper subterm of t, in which case θ_1 cannot exist. Hence s is not a variable.

If t is a variable, then s must be the minimal constant symbol or else we could find θ such that $s\theta > t\theta$ (replace t by the minimal constant symbol).

Otherwise, neither s nor t is a variable. Since $s\theta_1$ and $t\theta_1$ are identical, both s and t must have the same top-level function symbol. Suppose s is of the form $f(s_1, \ldots, s_n)$ and t is of the form $f(t_1, \ldots, t_n)$. Assume without loss of generality that $s\theta_2$ and $t\theta_2$ are ground terms.

Let $u_1 \ldots u_n$ and $v_1 \ldots v_n$ be listings of the multisets $\{s_1\theta_2, \ldots, s_n\theta_2\}$ and $\{t_1\theta_2, \ldots, t_n\theta_2\}$ respectively, in non-increasing order. Let k be $\min\{\ell: u_\ell$ and v_ℓ are not identical to within a permutation of arguments$\}$. Then $u_k < v_k$ in the path of subterms ordering. Let j be such that $t_j\theta_2$ is v_k. Let i be such that $s_i\theta_1$ and $t_j\theta_1$ are identical to within a permutation of arguments. If $s_i\theta_2 < t_j\theta_2$ then we have i and j as desired. If not, then delete $s_i\theta_2$ and $t_j\theta_2$ from the lists $u_1 \ldots u_n$ and $v_1 \ldots v_n$, respectively, and repeat this argument. Eventually i and j as desired will be obtained. This can be done because of the following facts:

a) The multiset $\{u_1, \ldots, u_n\}$ is less than the multiset $\{v_1, \ldots, v_n\}$ in the ordering induced by the path of subterms ordering on terms.

b) If $s_i\theta_2 \geq t_j\theta_2$, then the multiset $\{u_1, \ldots, u_n\} - \{s_i\theta_2\}$ is still less than the multiset $\{v_1, \ldots, v_n\} - \{t_j\theta_2\}$ in the ordering on multisets of terms.

Corollary 1: Suppose that s, t, and θ_1 are as in the above theorem. Then θ_1 must replace at least one variable of t by the minimal constant symbol.

Proof: If a) of the above theorem is true, this is immediate. If b) is true, we can use a simple inductive argument.

Corollary 2: Suppose s and t are not necessarily ground terms, neither of which contain the minimal constant symbol. Suppose that there exists substitution θ_1 such that $s\theta_1$ and $t\theta_1$ are identical to within a permutation of arguments. Then either s and t are identical to within a permutation of arguments, or there exist substitutions θ_2 and θ_3 such that $s\theta_2 < t\theta_2$ and $s\theta_3 > t\theta_3$ in the path of subterms ordering.

It may simplify the computation of the path of subterms ordering on non-ground terms to assume that there is a minimal constant symbol that never actually appears in any of the terms. This assumption causes us to fail to recognize some replacements as simplifications, however. For example, the replacement

$$f(g(x),d) \rightarrow f(g(c),c)$$

will not be recognized as a simplification if c is the minimal constant symbol actually appearing in terms.

10. Conclusions

We have exhibited a new partial ordering on terms and have shown that it is a simplification ordering. Also, we have given efficient algorithms for computing this ordering on ground terms. The algorithms for non-ground terms are not so efficient and are not completely general. In fact, we do not have any algorithm that will compute the ordering on non-ground terms in all cases. However, we do have a characterization of a class of simplifications including many that involve non-ground terms. This characterization appears to apply to commonly occurring sets of rewrite rules. We have given several examples to illustrate the wide applicability of this "path of subterms" ordering for proving termination of systems of rewrite rules. These examples involve algebraic simplification, symbolic differentiation, number theoretic functions, LISP functions, and Boolean formulae. In fact, we have shown that any primitive recursive function can be computed by a system of rewrite rules for which termination can be proven using the path of subterms ordering. We conjecture that a function is primitive recursive iff it can be computed by a system of rewrite rules for which termination can be proven using the path of subterms ordering.
It would be interesting to see if there is a simple description of a class of systems of rewrite rules such that a function can be computed in polynomial time iff it can be computed by a system of rewrite rules in the class. Perhaps this can be done for other time or space bounds on computations. In future work, we hope to give a general method for combining this ordering with various specialized orderings to get a proof technique of even greater applicability.

References

[1] Dershowitz, N. and Manna, Zohar, Proving termination with multiset orderings, Stanford Artificial Intelligence Laboratory Memo AIM-310 (1978).

[2] Huet, Gerard, Confluent reductions: abstract properties and applications to term rewriting systems. Proceedings of the 18th Annual Symposium on Foundations of Computer Science (1977).

[3] Huet, Gerard and Lankford, Dallas, On the uniform halting problem for term rewriting systems. Report No. 283, IRIA (March 1978).

[4] Iturriaga, R., Contributions to mechanical mathematics. Ph.D. Thesis, Carnegie-Mellon University (1967).

[5] Knuth, D. E., and Bendix, P. B., Simple word problems in universal algebras. Computational Problems in Abstract Algebra, Leech, Ed., Pergamon Press (1970), pp. 263-297.

[6] Lankford, Dallas S., Canonical algebraic simplification in computational logic. Report No. ATP-25, Southwestern University, Department of Mathematics, Georgetown, Texas (May 1975).

[7] Lipton, R. J., and Snyder, L., On the halting of tree replacement systems. Proceedings of a Conference on Theoretical Computer Science, University of Waterloo, Canada (1977).

[8] Manna, Z., and Ness, S., On the termination of Markov algorithms. Proceedings of the Third Hawaii International Conference on System Sciences (1970).

[9] Plaisted, D., Well-founded orderings for proving termination of systems of rewrite rules. University of Illinois Report No. UIUCDCS-R-78-932 (1978).