MIT/LCS/TM-219

RECURSIVE DECOMPOSITION ORDERING AND MULTISET ORDERINGS

Jean-Pierre Jouannaud

Pierre Lescanne

Fernand Reinig

June 1982

Recursive Decomposition Ordering
and
Multiset Orderings

Jean-Pierre JOUANNAUD
Pierre LESCANNE
Fernand REINIG

This report contains two papers Recursive Decomposition Ordering *presented at the Conference on Formal description of Programming Concepts sponsored by the T.C. 2.2 of IFIPS (Garmish-Partenkirchen Germany June 1-4, 1982) and* On Multiset Orderings *which will be published in Information Processing Letters.*

Abstract: The Recursive Decomposition Ordering, a simplification ordering on terms, is useful to prove termination of term rewriting systems. In this paper we give the definition of the decomposition ordering and prove that it is a well-founded simplification ordering containing Dershowitz's Recursive Path Ordering. We also show that the Recursive Decomposition Ordering has a very interesting incremental property. In the second paper, we propose two well-founded orderings on multisets that extend the Dershowitz-Manna ordering. Unlike the Dershowitz-Manna ordering, ours do not have a natural monotonicity property. This lack of monotonicity suggests using monotonicity to provide a new characterization of the Dershowitz-Manna ordering. Section 5 proposes an efficient and correct implementation of that ordering.

Résumé: L'ordre récursif de décomposition est un ordre de simplification utilisé pour prouver la terminaison des systèmes de réécriture de termes. Dans cette communication, nous donnons la définition de l'ordre récursif de décomposition, nous prouvons qu'il est bien fondé, qu'il est un ordre de simplification et qu'il contient l'ordre récursif sur les chemins de Dershowitz. Nous montrons aussi que l'ordre récursif de décomposition a une intéressante propriété d'incrémentalité. Dans le second article, on propose deux ordres bien fondés qui étendent l'ordre de Dershowitz et Manna. Ces ordres ne vérifient pas une propriété naturelle de monotonicité que nous définissons. Aussi cela suggère d'utiliser la monotonicité comme une nouvelle caractérisation de l'ordre de Dershowitz et Manna. La cinquième section de cette note propose une implantion efficace et correcte de cet ordre.

Key-words: Termination, term rewriting systems, equational theories, well-founded ordering, simplification, multiset ordering.

Recursive Decomposition Ordering

Jean-Pierre JOUANNAUD

Centre de Recherche en Informatique de Nancy
Campus Scientifique, BP 239
540506 Vandoeuvre-lès-Nancy,FRANCE

Pierre LESCANNE
Centre de Recherche en Informatique de Nancy
and
Laboratory for Computer Science, Massachusetts Institute of Technology,
545 Technology Square
Cambridge, Massachusetts, 02139, U.S.A.

Fernand REINIG
Centre de Recherche en Informatique de Nancy
Campus Scientifique, BP 239
540506 Vandoeuvre-lès-Nancy,FRANCE

Abstract: The Recursive Decomposition Ordering, a simplification ordering on terms, is useful to prove termination of term rewriting systems. In this paper we give the definition of the decomposition ordering and prove that it is a well-founded simplification ordering containing Dershowitz's Recursive Path Ordering. We also show that the Recursive Decomposition Ordering has a very interesting incremental property.

Résumé: L'ordre récursif de décomposition est un ordre de simplification utilisé pour prouver la terminaison des systèmes de réécriture de termes. Dans cette communication, nous donnons la définition de l'ordre récursif de décomposition, nous prouvons qu'il est bien fondé, qu'il est un ordre de simplification et qu'il contient l'ordre récursif sur les chemins de Dershowitz. Nous montrons aussi que l'ordre récursif de décomposition a une intéressante propriété d'incrémentalité.

1. Introduction

Term rewriting systems are an important model for non deterministic computations [21]. Therefore, methods for proving termination of term rewriting systems can provide a method for proving termination in other areas of programming. Term rewriting systems have also become a major tool in many fields related to programming, like abstract data type specifications (e.g., to establish their completeness by the Knuth-Bendix superposition procedure [13, 6]), program verification, theorem provers, and decision procedures for equational theories [1, 5, 24]. The Knuth-Bendix completion algorithm completes a non confluent set of term rewriting rules into a confluent (uniquely terminating) one and is used to prove the equivalence of abstract data type specifications via consistency of theories [3, 8, 18]. The Knuth-Bendix completion algorithm requires a universal method for proving finite termination, as in Huet's proof [7]. In other words, if the termination of the final set of rules is proved using a noetherian ordering, this ordering must be sufficient to prove the termination property of all the intermediate sets of term rewriting rules generated by the algorithm. Unfortunately, Huet and Lankford have shown that the finite termination of term rewriting systems is undecidable [9]. Thus, it is impossible to find a universal procedure to check for finite termination of any system, and people have been forced to look for specific techniques (see [10] for a survey).

In that vein Guttag, Kapur, and Musser [4] proposed a method based on superposition of terms which is similar to that used by Knuth and Bendix to prove confluence. Here we are mostly interested in simplification orderings. These are orderings compatible with the structure of terms and which have the subterm property. Dershowitz established that simplification orderings are powerful tools for proving termination and proposed recursive path-ordering [2] after Plaisted's recursive path of subterms ordering [20]. These methods use an ordering on the set of function symbols. In [15] and [16] a new ordering was used to prove simply the well-foundedness of the recursive path ordering when the ordering on function symbols is total. In [22] and [23] it was shown that a similar ordering could also be defined when the ordering on function symbols is only partial, capturing easily the case of terms with variables. This ordering is a well-founded simplification ordering which has the additional useful properties. First, it contains strictly the path recursive ordering. Second, it is monotonic with respect to the ordering on the function symbols, i.e., if one increases the ordering on function symbols then one increases the ordering on terms. We call the third major property incrementality: it is easy to find an expansion of the ordering on the function symbols when a given pair of terms needs to be ordered. This idea might be used in the Knuth-Bendix completion algorithm to build the required universal ordering in an incremental and automatic way as the set of rules is completed.

In the second section of this paper we give classical definitions and notations about terms and orderings [10]. In the case of a total ordering on function symbols the decomposition ordering is based on a decomposition of terms into three parts which are compared lexicographically [16]. In the case of a partial ordering on function symbols these decompositions are quadruples, and, instead of one decomposition for a term, a set of decompositions is associated with each term and comparisons of sets of decompositions provide the decomposition ordering [22, 23]. The third chapter is devoted to extending these concepts to ground terms (i.e., terms without variables). In Section 5 and Section 6 the decomposition ordering is proved to be a simplification ordering and a well-founded ordering. In Section 7 we prove that the decomposition ordering is more powerful than the recursive path ordering. An extension of the decomposition ordering to non-ground terms is given in Section 8. The incrementality property is illustrated in the conclusion.

2. Orderings and Terms

2.1 Set and Multiset Ordering

An ordering $<$ on a set E can be extended to the set $\mathcal{P}(E)$ of sets on E by:

$S \ll T$ iff $S \neq T$ and $\forall x\ (x \in S \text{ and } x \notin T) \Rightarrow \exists y\ (y \in T \text{ and } y \notin S \text{ and } x < y)$.

Intuitively, a multiset on E is an unordered collection of elements of E, with possibly many occurrences of given elements. A multiset can be seen as a mapping $E \rightarrow \mathcal{N}$ where $\mathcal{N}$ is the set of natural numbers. Let $\mathcal{M}(E)$ be the set of all the finite multisets on E, i.e., the multisets M such that their support $\{x \in E \mid M(x) \neq 0\}$ is finite. The empty multiset $\{\}$ is the multiset such that $\{\}(x) = 0$, for all x in E. A set is a particular case of a multiset such that S(x) is 0 or 1. Usually multisets are written as lists $\{x_1,\ldots,x_m\}$ with a straightforward interpretation. If M is a multiset, $x \in M$ means $M(x) >_{\mathcal{N}} 0$. An ordering on E can be extended to multisets [11] by

$M \ll N$ iff $M \neq N$ and $\forall y\ ([N(y) <_{\mathcal{N}} M(y)] \Rightarrow [(\exists x \in E)\ y < x \text{ and } M(x) <_{\mathcal{N}} N(x)])$.

The extension to sets is a particular case of the extension to multisets. If the ordering is well-founded on E the extensions are well-founded on $\mathcal{M}(E)$ and $\mathcal{P}(E)$.

2.2 Terms and Occurrences

In this paper we will deal with terms with fixed arity function symbols. But all the results can easily be extended to varyadic terms. Suppose a set F of function symbols and a function $ar{:}F \rightarrow \mathcal{N}$ is given. T(F,X) is the set of terms on F with variables in X. $s \in T(F,X)$ is either a variable or of the form $f(s_1,\ldots,s_m)$ with $f \in F$ such that $ar(f) = m$ and $s_1,\ldots,s_m$ are in T(F,X). $T(F,X,\square)$ is the set of box terms. A box term is either the symbol $\square$ or has the form $f(s_1,\ldots,s_m)$ for $f \in F$ such that $ar(f) = m$ and there exists $i \in [1..m]$ with $s_i \in T(F,X,\square)$ and, for $i \neq j$, $s_j \in T(F,X)$. Intuitively, $T(F, X, \square)$ is the set of terms with one terminal occurrence of $\Box$. The symbol $\Box$ may be viewed as the empty term. It is used to deal with function symbols having fixed arities. If X is empty, we will write T(F) and T(F,X) instead of T(F,$\Box$) and T(F,X,$\Box$) and we call these terms, *ground terms*.

We assume by convention that $ar(\Box) = ar(x) = 0$ for all $x \in X$. Terms may be viewed as labeled trees in the following way. A term is a partial function of $\mathcal{N}_+^*$ (the monoid over $\mathcal{N}_+$ with ε as empty word) in F $\cup$ X, such that the domain or *set of occurrences* $Occ(t) = \{u \in \mathcal{N}_+^* \mid t(u) \text{ is defined}\}$ verifies:

(1) $\varepsilon \in Occ(t)$

(2) $ui \in Occ(f(\ldots t_i \ldots))$ iff $\forall i\ 1 \leq i \leq ar(f) \Rightarrow u \in Occ(t_i)$.

If u and v belongs to $\mathcal{N}_+^*$ then u/v is a $w \in \mathcal{N}_+^*$ such that $vw = u$. In the following, $|t| = |\{u \in Occ(t) \mid t/u \neq \Box\}|$ where t/u is the subterm of t at the occurrence u. $t[u \leftarrow t']$ is the term obtained by replacing t/u by t' in t. We define the set of *paths of s* as the set $Path(s) = \{p \in Occ(s) \mid ar(s(p)) = 0\}$. Given a path p, the set of *prefixes of p in s* is $Prefix(s, p) = \{p \in Occ(s) \mid u \leq p\}$ if $s/p \neq \Box$ and $Prefix(s, p) = \{p \in Occ(s) \mid u < p\}$ if $s/p = \Box$. Given a path p and a prefix u of p, we define $succ(u, p)$ as ui if $ui \in Prefix(s, p)$ and $succ(p, p) = \infty$, we will state $t/\infty = \Box$, thus $t/succ(p, p) = \Box$. A *substitution* is a mapping σ: X$\rightarrow$ T(F,X) such that $\sigma(x) = x$ except for a finite number of variables $x \in X$. It can be extended to a mapping σ: T(F,X)$\rightarrow$ T(F,X) by $\sigma(f(s_1,\ldots,s_m)) = f(\sigma(s_1),\ldots,\sigma(s_m))$.

2.3 Simplification Orderings

An ordering $\prec$ on T(F,X) is a *simplification ordering* if it has the properties:

Subterm Property: $t \prec f(\ldots,t,\ldots)$.

Compatibility Property: $t_1 \prec t_2 \Rightarrow f(\ldots,t_1,\ldots) \prec f(\ldots,t_2,\ldots)$.

Dershowitz's Theorem [2]: A term rewriting system $R = \{g_i \rightarrow d_i \mid i \in \mathcal{I}\}$ with a finite number of symbols is finitely terminating if there exist a simplification ordering $\prec$ such that for all i in $\mathcal{I}$ and for all substitution σ, $\sigma(g_i) \succ \sigma(d_i)$.

3. Decomposition Ordering for Ground Terms

We define first the concept of elementary decomposition of terms in T(F, $\Box$).

Definition 1: Given a term t, a path $p \in Path(t)$ and an occurrence $u \in Prefix(t,p)$, the *elementary decomposition* $d_u^p(t)$ of t in u along the path p is the quadruple $\langle g, t', \mathcal{T}, t''\rangle$ where

$g = t(u)$,

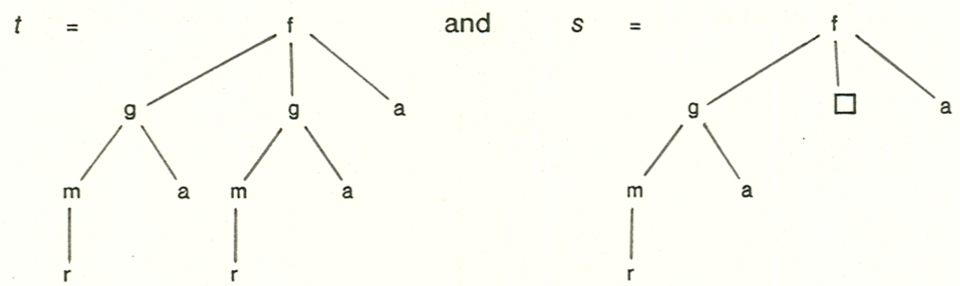
$t' = t/succ(u,p)$,

$\mathcal{T}$ is the multiset $\{t/uj \mid 1 \leq j \leq ar(t(u)),\ j \neq succ(u,p)\}$

t'' is the "box term" $t[u \leftarrow \Box]$.

In the following, we never refer to the elementary decomposition in u where $t/u = \square$, thus we do not define such a decomposition.

Example 1: Let



then $d_2^{211}(t) = \langle g; \begin{array}{c} m \\ | \\ r \end{array}; \{a\}; s\rangle$.

Assume now that a partial function $Q: T(F) \times \mathcal{N}_+^* \to \mathcal{P}(\mathcal{N}_+^*)$, called an *occurrence choice*, such that $Q(t,p)$ is defined if $p \in Path(t)$ and such that $Q(t,p) \subseteq Prefix(s, p)$ is given. We extend the previous definition to the set $Q(t,p)$ and we obtain the set $d_Q^p(t)$ (or more simply $d^p(t)$ if this is not ambiguous) that we call *path decomposition* of t along p:

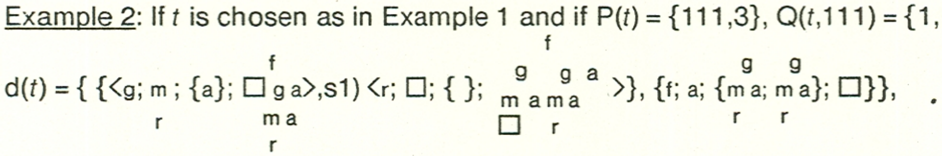
$$d_Q^p(t) = \{d_u^p(t) \mid u \in Q(t,p)\}.$$

Note that for any Q and any p, $d_Q^p(\square)$ is the empty set.

In the same way, assume a partial function $P: T(F) \to \mathcal{P}(\mathcal{N}_+^*)$, called *path choice*, such that $P(t) \subseteq Path(t)$. We extend once more the definition and obtain a set of sets of decompositions, $d_Q^P(t)$ (or more simply $d(t)$ if no ambiguity on P and Q), that we call a *decomposition* of t:

$$d_Q^P(t) = \{d_Q^p(t) \mid p \in P(t)\}.$$

Example 2: If t is chosen as in Example 1 and if $P(t) = \{111,3\}$, $Q(t,111) = \{1,111\}$ and $Q(t,3) = \{\varepsilon\}$ then



We are now able to define the decomposition ordering. Notice that in addition to the ordering $<_F$ on F, this definition uses two other orderings $\overset{d}{\nless}_{QP}$ and $<_{QP}$. $\overset{d}{\nless}_{QP}$ is an ordering on T(F) and $<_{QP}$ is an ordering on decompositions which depends upon the choices P and Q. In order to simplify notations, the multiset extension of $<_{QP}$ and $\overset{d}{\nless}_{QP}$ will be written $\ll_{QP}$ and $\overset{dd}{\nless\nless}_{QP}$ instead of $<_{QP}<_{QP}$ and $\overset{d}{\nless}_{QP}\overset{d}{\nless}_{QP}$.

Main Definition: Given a partial ordering $<_F$ on F, a path choice P and an occurrence choice Q, we define the *recursive decomposition ordering* (or more simply decomposition ordering) $\overset{d}{<}$ in the following way:

$$s \overset{d}{<}_{QP} t \quad \text{iff} \quad d^P_Q(s) \ll_{QP} \ll_{QP} d^P_Q(t)$$

with

$$d^p_u(s) = \langle f; s'; \mathcal{S}; s'' \rangle \qquad <_{QP} \qquad d^q_v(t) = \langle g; t'; \mathcal{T}; t'' \rangle$$

iff in a lexicographical way:

(dec.1) $f <_F g$

(dec.2) $d_Q^{p/succ(u,p)}(s') \ll_{QP} d_Q^{q/succ(v,q)}(t')$

(dec.3) $\mathcal{S} \overset{d\,d}{\ll}_{QP} \mathcal{T}$

(dec.4) $d^u_Q(s'') \ll_{QP} d^v_Q(t'')$.

Remarks: 1) In general, we will have $p \in P(s)$, $u \in Q(s, p)$, $q \in P(t)$ and $q \in Q(t, q)$. Notice, however, that $d^p_u(s) <_{QP} d^q_v(t)$ has a meaning although $p \in Path(s) - P(s)$, $u \in Prefix(s, p) - Q(s,p)$, $q \in Path(t) - P(t)$, $v \in Prefix(t, q) - Q(t,q)$. We will use this fact in further proofs.

2) Cases (dec.2) and (dec.4) in the main definition do not use a path choice, because the path is fixed. Therefore it is not necessary to extend the concept of path choice for terms in $T(F,\square)$.

Full examples are given in Appendix.

Theorem 1: $\overset{d}{<}_{QP}$ and $<_{QP}$ are strict orderings.

Proof: We prove the property for $<_{QP}$. It will be true for $\overset{d}{<}_{QP}$ which is a multiset extension of $<_{QP}$. The proof is easily done by induction, since the extensions of orderings as lexicographical ordering or set ordering preserve together irreflexivity and transitivity. ∎

4. Choices in Decomposition Ordering

The choice of P and Q in the previous definitions seems to be a main point. In this section, we study two possible choices: the first one consists of taking all the occurrences and paths, the second one consists of keeping only the maximum occurrences and paths (in a sense we will make precise later). The first one provides easier proofs and the second one leads to more efficient implementations, but they define the same decomposition ordering.

4.1 Entire Choice

Here we define two choices P_* and Q_*. They are called the entire choice because they correspond to choosing all the paths or occurrences in the term. P_* is defined by $P_*(t) = Path(t)$. Q_* is defined by $Q_*(t, p) = Prefix(t, p)$. We will write d^u_* and d^*_* for the associated sets and sets of sets, instead of $d^u_{Q_*}$ and $d^P_{Q_*^*}$. The associated ordering will be written $\overset{d}{<}_{**}$ and $<_{**}$.

In the following, we will use also the ordering $<_{*P}$ and $\overset{d}{<}_{*P}$ associated with the choices Q_* and P.

4.2 Maximal Choice

We define here the maximal choice which corresponds to selecting from among the paths and the occurrences the maximal ones.

Definition 3, *Maximal Paths*: The set $Mp(t)$ of the maximal paths of a term $t = g(t_1,...,t_n)$ is defined by:

(1) $Mp(t) = \varepsilon$ if $n = 0$

(2) $Mp(t) = \bigcup_{i \in \mathfrak{I}} i.Mp(t_i)$

such that $T = \{t_i \mid i \in \mathfrak{I}\}$ is a minimal and complete set of maximal elements of $S = \{t_1,...,t_n\}$, i.e.,

Minimality $(\forall t_i \in T)(\forall t_j \in T)\ \neg(t_j \overset{d}{\leq}_{**} t_i)$.

Completeness $(\forall t_j \in S)(\exists t_i \in T)\ t_j \overset{d}{\leq}_{**} t_i$.

Let now define P_+ and Q_+.

$P_+(t) = Mp(t)$

$Q_+(t, p) = \{v \in Prefix(t, p) \mid (\forall w)\ [w < v \Rightarrow \neg(t(v) \leq_F t(w))] \text{ and } [v < w \leq p \Rightarrow \neg(t(v) <_F t(w))]\}$

We will write $d^p_+(t)$ and $d^+_+(t)$ instead of $d^p_{Q_+}$ and $d^P_{Q_+^+}$. $\overset{d}{<}_{++}$ and $<_{++}$ will be the associated orderings.

4.3 Decomposition Ordering is Almost Independant of the Choices

Our aim now is to prove that $\overset{d}{<}_{++}$ and $\overset{d}{<}_{**}$ are the same ordering. More precisely, we want to prove that $\overset{d}{<}_{**}$ and $\overset{d}{<}_{QP}$ are the same ordering if the choices exhibit the following *minimality condition*:

$P(t) \supseteq P_+(t)$.

$Q(t, p) \supseteq Q_+(t, p)$ for all $p \in P_+(t)$.

Lemma 1: Let $p \in Path(s)$ and $u \in Prefix(s, p)$. Then there exists $u' \in Q_+(s, p)$ such that $d^p_u(s) \leq_{QP} d^p_{u'}(s)$, for any choice that satisfies the minimality conditions.

Proof: If $u \in Q_+(s, p)$ then $u' = u$.

If $u \notin Q_+(s, p)$ then there exits $u' \in Q_+(s, p)$ such that $s(u) \leq s(u')$.

If $s(u) < s(u')$ the result is true.

If $s(u) = s(u')$ then $u' < u$ and $s/succ(u, p)$ is a strict subterm of $s/succ(u', p)$. The result follows from (dec.2) and Lemma 2. ▮

Lemma 2: Let $p \in Path(s)$ and $k \leq p$. Then $d_Q^{p/k}(s/k) \ll_{QP} d_Q^p(s)$, for any choices P and Q which satisfy the minimality condition. The inequality is strict if $k \neq \varepsilon$ that means s/k is a strict subterm of s.

Proof: By induction on $|p| - |k|$

Basic Case: If $|p| - |k| = 0$, then s/k is a term reduced either to $\square$ then $d_Q^{s/k}(s/k) = \{\ \}$ or to the symbol $s(p)$ which occurs in s. The result will be true if a symbol greater or equal to $s(p)$ appears in the decomposition of s, that comes then from the minimality condition.

General Case: Let $u \in Q(s/k, p/k)$. For any decomposition of s/k in u along p/k, we want to find a greater decomposition of s in v along p. Two cases may happen:

- $ku \in Q(s,p)$. Then $v = ku$ works. The two decompositions are compared using case (dec.4) of the main definition applied to $d_Q^u(s/k[u \leftarrow \square])$ and $d_Q^{ku}(s[ku \leftarrow \square])$. The result is obtained from the induction hypothesis because the path ku in $s[ku \leftarrow \square]$ has length $|ku| - 1 <_{\mathcal{N}} |p|$, the subterm occurrence being the same (that is k).

- $ku \notin Q(s, p)$. Thus there exists $v \in Q_+(s, p) = Q_+(t, p) \subseteq Q(s, p)$ such that $s(v) \geq s(ku)$. This case divides into two subcases:
  - $s(v) > s(ku)$, then the result is straightforward.
  - $s(v) = s(ku)$, then $v < ku$ and the result is obtained using case (dec.2) of the main definition applied to $d_Q^{p/k/succ(u, p/k)}(s/k/succ(u, p/k)$ and $d_Q^{p/succ(v, p)}(p/succ(v, p))$. On one hand, $p/k/succ(u, p/k) = p/succ(ku, p)$ is a strict subterm of $p/succ(v, p)$ because $v < ku$. On the other hand,

    $$|p| - (|v| + 1) - [(|ku| + 1]) - (|v| + 1)] = |p| - |k| - |u| - 1 <_{\mathcal{N}} |p| - |k|.$$

    The wished result is thus true by using induction hypothesis. ▮

Proposition 1: $<_{QP}$ and $<_{*P}$ are the same ordering if the choices verify the minimality condition.

Proof: We prove $s <_{QP} t$ if and only if $s <_{*P} t$ by induction on $|s| + |t|$. Both $\Leftarrow$ and $\Rightarrow$ ways use Lemma 1 in order to delete the supplementary computation (for the $\Leftarrow$ way) or to add the missing ones (for the $\Rightarrow$ way). ▮

We now want to prove the equivalence of $<_{*P}$ and $<_{**}$. Once more the method is based upon a lemma which proves that the supplementary computation performed by $<_{**}$ is useless.

Lemma 3: Let s be a term in T(F) and p a path in s. If $p \notin P_+(s)$ then there exist $q \in P_+(s)$ such that:

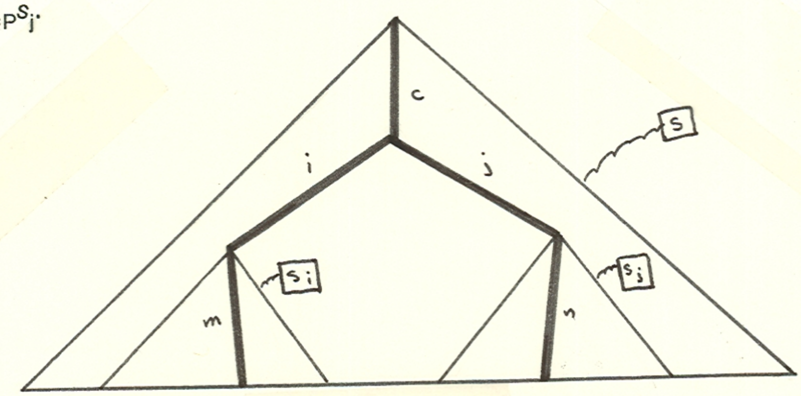
$d_*^p(s) \ll_{*P} d_*^q(s)$.

Proof: By induction on $|s|$. If $p \in Path(s) - P_+(s)$, then there exists c, $m \in \mathcal{N}_+^*$ and $i \in \mathcal{N}^*$ such that:

- $p = c.i.m$.
- the subterm s/c belongs to a maximal class of the subterms at the depth c.
- $s_i = s/c.i$ does not belong to a maximal class of the subterm of s/c.

In addition there exists $j \in \mathcal{N}^*$ such that

$s_j = s/c.j$ and $s_i \overset{d}{<}_{*P} s_j$.



Therefore there exists $n \in P(s_j)$ such that $d_*^m(s_i) \ll_{*P} d_*^n(s_j)$. By induction, n can be supposed to belong to $P^+(s_j)$. Clearly $c.j.n$ belongs to $P_+(s)$. Let q be $c.j.n$, then the result

$d_*^{c.j.m}(s) \ll_{*P} d_*^{c.j.n}(s)$

is obtained by the following lemma. ∎

Lemma 4 Let s be a term in T(F) and c, n, m be in $\mathcal{N}_+^*$ and i,j be in $\mathcal{N}_+$ such that $c.i.n$ and $c.j.m$ are paths in s. If $d_*^m(s/c.i) \ll_{*P} d_*^n(s/c.j)$ then $d_*^{c.i.m}(s) \ll_{*P} d_*^{c.j.n}(s)$.

Proof:By induction on $|c| + |m|$. For all prefix u of $c.i.m$, one must find a prefix v of $c.j.n$ such that

$d_u^{c.i.m}(s) <_{*P} d_v^{c.j.n}(c)$.

Three cases can be distinguished:

1) $u < c$: then suppose $u = v$. The result is true by case (dec.2) by using the inequality

$d_*^{succ(u,c.i.m)}(s/succ(u,c.i.m)) \ll_{*P} d_*^{succ(u,c.j.m)}(s/succ(u,c.j.m))$

which is true by induction hypothesis applied to $s/succ(u, c)$ with $|c/succ(u, c)| + |m| <_{\mathcal{N}} |c| + |m|$.

2) $u = c$. Then $v = c$ and the result is true by case (dec.2) and the hypothesis $d_*^m(s/c.i) \ll_{*P} d_*^n(s/c.j)$.

3) $u<c$. Then there exists $h\in\mathcal{N}^*_+$ such that $u = c.\mathrm{i}.h$ and $k\in\mathcal{N}^*_+$ such that: $\mathrm{d}^m_h(s/c.\mathrm{i})\overset{d}{<}_{*P}\mathrm{d}^n_k(s/c.\mathrm{j})$. Let v be $c.\mathrm{j}.k$ and let us prove $\mathrm{d}^{c.\mathrm{i}.m}_{c.\mathrm{i}.h}(s)<_{*P}\mathrm{d}^{c.\mathrm{j}.n}_{c.\mathrm{j}.k}(s)$. If (dec.1), (dec.2) or (dec.3) are used, the result is straightforward. If (dec.4) is used, that leads to prove $\mathrm{d}^{c.\mathrm{i}.h}_*(s[c.\mathrm{i}.h\leftarrow\square] \ll_{*P} \mathrm{d}^{c.\mathrm{j}.k}_*(s[c.\mathrm{j}.k\leftarrow\square]$ which results from $\mathrm{d}^h_*(s/c.\mathrm{i}[h\leftarrow\square]) \ll_{*P} \mathrm{d}^k_*(s/c.\mathrm{j}[k\leftarrow\square])$ by using the induction hypothesis with $|c| + |h| <_{\mathcal{N}} |c| + |m|$. ∎

Proposition 2: $<_{*P}$ and $<_{**}$ are a same ordering if P verifies the minimality condition.

Proof: We prove $s\overset{d}{<}_{*P}t \Leftrightarrow s\overset{d}{<}_{**}t$ by induction $|s| + |t|$. Both $\Leftarrow$ and $\Rightarrow$ ways use lemma 3 in order to delete the supplementary computation (for the $\Leftarrow$ way) or add the missing ones (for the $\Rightarrow$ way). ∎

Theorem 2: If the choices confirm to the minimality condition, then the ordering $\overset{d}{<}_{QP}$ and $\overset{d}{<}_{**}$ are the same. In particular, the orderings $\overset{d}{<}_{++}$ and $\overset{d}{<}_{**}$ are the same.

Proof: We use successively Proposition 1 and Proposition 2. ∎

The definition of $\overset{d}{<}_{++}$ can now be made intrinsic, that means that instead of using $<_{**}$ in the definition of the maximal path and the maximal occurrence, we may use $<_{++}$ itself without changing the ordering as it is proved in [22].

In the following, we write $\overset{d}{<}$ for any decomposition ordering whose choices verify the minimality conditions. We will use $\overset{d}{<}_{**}$ for the most proofs.

5. Decomposition Ordering is a Simplification Ordering

Subterm Lemma: $t \overset{d}{<} \mathrm{f}(\ldots,t,\ldots)$. Proof: By lemma 2. ∎

Compatibility Lemma: $t_1 \overset{d}{<} t_2 \Rightarrow \mathrm{f}(\ldots,t_1,\ldots)\overset{d}{<}\mathrm{f}(\ldots,t_2,\ldots)$.

Proof: By induction on $|\mathrm{f}(\ldots,t_1,\ldots)| + |\mathrm{f}(\ldots,t_2,\ldots)|$. Let $p \in Path(\mathrm{f}(\ldots,t_1,\ldots))$ and $u\in Prefix(\mathrm{f}(\ldots,t_1,\ldots),p)$. Two cases may happen.

Case 1: $p = kq$ and $\mathrm{f}(\ldots,t_1,\ldots)/k \neq t_1$ or $u = \varepsilon$. We obtain easily the result $\mathrm{d}^p_u(\mathrm{f}(\ldots,t_1,\ldots))<\mathrm{d}^p_u(\mathrm{f}(\ldots,t_2,\ldots))$ by using case (dec.4) of the main definition and the induction hypothesis.

Case 2: $p = kq$ and $\mathrm{f}(\ldots,t_1,\ldots)/k = t_1$ and $u\neq\varepsilon$. Thus $q \in Path(t_1)$. As $t_1\overset{d}{<}t_2$, there exists $q' \in Path(t_2)$ such that $(\forall\ v\leq q)\ (\exists\ v' \leq q')\ \mathrm{d}^q_v(t_1) < \mathrm{d}^{q'}_{v'}(t_2)$. If the proof of the last inequality is by case (dec.1), (dec.2) or (dec.3) of the main definition, the result is straightforward. If the proof is by case (dec.4) of definition, the result is achieved by using the induction hypothesis. ∎

Corollary: $\overset{d}{<}$ is a simplification ordering.

6. Decomposition Ordering improves over Recursive Path Ordering

Let us recall Dershowitz's definition of the Recursive Path Ordering [2].

Definition *Congruence of Permutation*: $f(s_1,\dots,s_n) \overset{*}{=} g(t_1,\dots,t_n)$ iff $f = g$ and there exists a permutation $\sigma \in S_n$ such that $s_i \overset{*}{=} t_{\sigma(i)}$.

Definition: The *recursive path ordering* over T(F) is recursively defined as follows:

$$s = f(s_1,\dots,s_m) \overset{*}{<} g(t_1,\dots,t_n) = t \qquad \text{iff}$$

(rpo.1) $f = g$ and $\{s_1,\dots,s_m\} \overset{*}{<}\overset{*}{<} \{t_1,\dots,t_n\}$

or (rpo.2) $f <_F g$ and for all s_i, $\quad s_i \overset{*}{<} t$

or (rpo.3) $\neg f \leq_F g$ and for some t_j, $\quad s \overset{*}{<} t_j$ or $s \overset{*}{=} t_j$

this definition can be made "less deterministic", by changing (rpo.3) to:

(rpo.3') for some t_j, $\quad s \overset{*}{<} t_j$ or $s \overset{*}{=} t_j$

Theorem 3 (Dershowitz [2]): $\overset{*}{<}$ is a simplification ordering. If $<_F$ is is well founded on F, then $\overset{*}{<}$ is well-founded on T(F). If $<_F$ is total on F, then $\overset{*}{<}$ restricted to $T(F)/\overset{*}{=}$ is total.

We prove now that $\overset{d}{<}$ contains $\overset{*}{<}$. We first prove some technical lemmas, which prove actually that $\overset{d}{<}$ is a fixed point of the functional which defines $\overset{*}{<}$.

Lemma 5: $d^p(s_i) \ll d^q(g(t_1,\dots,t_n))$ and $f <_F g$ imply $d^{ip}(f(s_1,\dots,s_m)) \ll d^q(g(t_1,\dots,t_n))$.

Proof: By induction upon $|s_i|$.

Basic Case: $|s_i| = 0$, that is $s_i = \square$. The result is true because the only possible decomposition takes place in ε and because $f <_F g$.

General Case: Let $u \leq ip$. Two cases may be distinguished:

– $u = \varepsilon$ then $d_u^{ip}(f(\dots,s_i,\dots)) < d_\varepsilon^v(t)$ because $f <_F g$.

– $u = iv$, there exist $w \leq q$ such that $d_v^p(s_i) < d_w^q(t)$. If the proof is performed by case (dec.1), (dec.2) or (dec.3)of the main definition, then $d_{iv}^{ip}(s) < d_w^q(t)$ in the same way.

– If the proof is performed by case (dec.4), then we have : $d^v(s_i[v \leftarrow \square]) \ll d^w(t[w \leftarrow \square])$. By the induction hypothesis, we get $d^{iv}(s[iv \leftarrow \square]) \ll d^w(t[w \leftarrow \square])$, which proves the desired result by case (dec.4) of the main definition. ∎

Lemma 6: For all i, $s_i \overset{d}{<} t$ and $f <_F g$ implies $f(s_1,\ldots,s_m) \overset{d}{<} t$.

Proof: Straightforward from Lemma 5. ∎

Lemma 7: $d^u(s_i) \ll d^v(t_j)$ implies $d^{iu}(f(\ldots,s_i,\ldots)) \ll d^{jv}(f(\ldots,t_j,\ldots))$.

Proof: By induction on $|s_i|$. We have to prove that for any $p \leq iu$ there exist $q \leq jv$ such that $d^{iu}_p(f(\ldots,s_i,\ldots)) < d^{jv}_q(f(\ldots,t_j,\ldots))$. Two case must be distinguished.

- $p = \varepsilon$, then $q = \varepsilon$. The result follows from the hypothesis using (dec.2).
- $p = ip'$. Then there exists q' such that $d^u_{p'}(s_i) < d^v_{q'}(t_j)$. If it is proved by (dec.1), (dec.2) or (dec.3) of definition, then the desired result is proved in the same way. If it is proved by case (dec.4), then we obtain $d^{p'}(s_i[p' \leftarrow \Box]) \ll d^{q'}(t_j[q' \leftarrow \Box])$ which proves the desired result by (dec.4). ∎

Lemma 8: $\{\ldots,s_i,\ldots\} \overset{d}{<}\overset{d}{<} \{\ldots,t_j,\ldots\} \Rightarrow f(\ldots,t_i,\ldots) \overset{d}{<} f(\ldots,t_j,\ldots)$.

Proof: By applying Lemma 7. . ∎

Lemma 9: If $<_F$ is total on F, $\overset{d}{<}$ is total on $T(F)/\overset{*}{=}$.

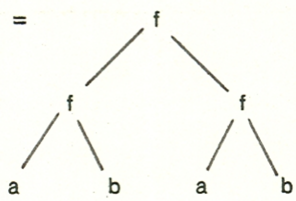
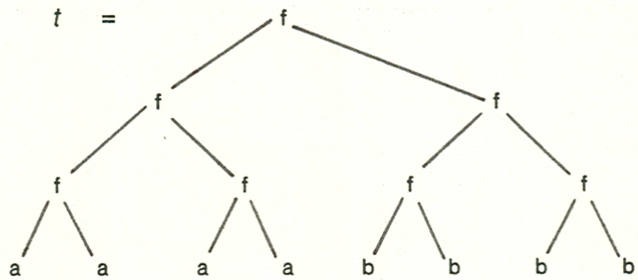
Proof: By induction on $\max(|s|, |t|)$. Suppose $<_F$ is total and neither $s \overset{d}{<} t$ nor $t \overset{d}{<} s$, let us prove that $s \overset{*}{=} t$. By the induction hypothesis, there exist in s a path p and an occurrence u such that $d^{u'}_{p'}(s) \overset{d}{\leq} d^u_p(s)$, for all other paths p' and occurrences u'. The same thing happens for q and v in t. Let d^u_p be $\langle f, s', \mathcal{S}, s''\rangle$ and $d^v_q(t)$ be $\langle g, t', \mathcal{T}, t''\rangle$. Because neither $s \overset{d}{<} t$ nor $t \overset{d}{<} s$ and by the induction hypothesis, $f = g$, $s' \overset{*}{=} t'$, $\mathcal{S} \overset{*}{=}\overset{*}{=} \mathcal{T}$ (where $\overset{*}{=}\overset{*}{=}$ is the congruence on multisets deduced from $\overset{*}{=}$) and $s'' \overset{*}{=} t''$. Then it is easy to see that $s \overset{*}{=} t$. ∎

Theorem 4: Given a partial ordering $<_F$ on F, we have $\overset{*}{<} \subseteq \overset{d}{<}$. If $<_F$ is total then $\overset{*}{<} = \overset{d}{<}$. Otherwise the inclusion is strict (whenever there exists a function symbol $f \in F$ such that $ar(f) \geq 2$).

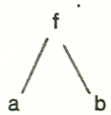
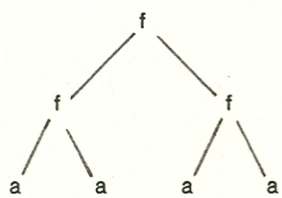
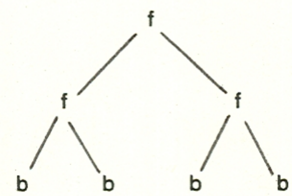
Proof: We can replace "iff" by "if" and $\overset{*}{<}$ by $\overset{d}{<}$ in the definition of recursive path ordering. By lemma 8 $\overset{d}{<}$ verifies (rpo.1), by lemma 6 $\overset{d}{<}$ verifies (rpo.2), by subterm lemma and transitivity $\overset{d}{<}$ verifies (rpo.3). Then $\overset{*}{<} \subseteq \overset{d}{<}$ is a consequence of the least fixed point property of $\overset{*}{<}$. That ends the first part of the proof.

To prove that $\overset{*}{<} = \overset{d}{<}$ if $<_F$ is total, we remark that both $\overset{*}{<}$ and $\overset{d}{<}$ are total ordering on $T(F)/\overset{*}{=}$ and do not compare terms s and t such that $s \overset{*}{=} t$. As $\overset{*}{<} \subseteq \overset{d}{<}$, we necessarily have $\overset{*}{<} = \overset{d}{<}$ in this case.

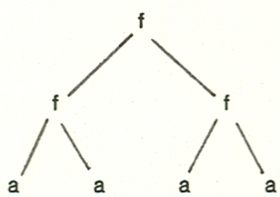
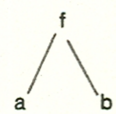
To prove that the inclusion is strict if $<_F$ is not total, we give a counter example. To build this counter example, we only need a binary function symbol f. Assume now a and b are incomparable and let

s =  and t = 

We assume without loss of generality that a and b are symbols of arity 0. If it is not the case we replace a by a(...,c,...) and b by b(...,c,...). It is not possible to compare s and t using $\overset{*}{\lesssim}$ because

 $\overset{*}{\lesssim}$  or  $\overset{*}{\lesssim}$ 

is false. On the other hand,

 $\overset{*}{\lesssim}$ 

is also false. However, we have $s \overset{d}{\lesssim} t$ because $d^{11}(s) < d^{111}(t)$ and $d^{12}(s) < d^{211}(t)$. ∎

7. Well foundednes of $\overset{d}{\lesssim}$

The well-foundedness is based on the following lemma.

Monotonicity Lemma: $\prec_F \subseteq <_F \Rightarrow \overset{d}{\precsim} \subseteq \overset{d}{\lesssim}$.

Proof: Easy, see [22]. ∎

Theorem 5: $\overset{d}{\lesssim}$ is well-founded if and only if $<_F$ is well-founded.

Proof: Assume $\overset{d}{\lessapprox}$ is not well-founded. Thus, there exists an infinite decreasing sequence $s_1 \overset{d}{\gtrapprox} s_2 \overset{d}{\gtrapprox} \ldots \overset{d}{\gtrapprox} s_n \overset{d}{\gtrapprox} \ldots$ Let now $\prec_F$ be a total well-founded ordering (i.e., a well-ordering) on F which contains $<_F$ (such an ordering exists by a variant of Zermelo's Theorem which can be seen as a transfinite topological sort). Using the monotonicity lemma, we obtain $s_1 \overset{d}{\succapprox} s_2 \overset{d}{\succapprox} \ldots \overset{d}{\succapprox} s_n \overset{d}{\succapprox} \ldots$ But $\overset{d}{\precapprox}$ = $\overset{*}{\lessapprox}$ by Theorem 3, which contradicts the well-foundedness of the recursive path ordering [2, 15]. ∎.

8. Extension of the decomposition orderings to non-ground terms

We will now define two formally different extensions of the decomposition ordering to non-ground terms. These two extensions are proved to be equivalent . The first one is more tractable for proofs, the second one leads to more efficient implementations. Moreover, these extensions are coherent with the definition of the decomposition ordering on ground terms, i.e.,

$$s \overset{d}{\lessapprox} t \Rightarrow \sigma(s) \overset{d}{\lessapprox} \sigma(t) \text{ for any substitution } \sigma.$$

Definition *by extension of the basic ordering*: Let $<_F$ a partial ordering on F. The decomposition ordering $\overset{d}{\lessapprox}$ on T(F) is extended to T(F, X) by simply extending $<_F$ to F∪X in the following way:

$$a <_{F \cup X} b \text{ iff } a \in F,\ b \in F \text{ and } a <_F b.$$

In other words, the $<_{F \cup X}$ ordering is the same as $<_F$ for functions symbols. Variable symbols can be compared with no other symbols using $<_{F \cup X}$. The orderings $\overset{d}{\lessapprox}$ and $<$ deduced from this definition will be written $\overset{d}{\lessapprox}_1$ and $<_1$ for the time being. This definition of the decomposition ordering leads to inefficient computations. For instance, let us suppose that $s/p \in X$ and $t/q \in X$ and $s/p \neq t/q$. It is quite obvious in this case that the two decomposition sets $d^p(s)$ and $d^q(t)$ cannot be compared using the new $\overset{d}{\lessapprox}$. However they will be recognized to be incomparable after a lot of useless computations. We give now a new definition of $\overset{d}{\lessapprox}$ on T(F, X) which avoids this drawback. The basic idea is to modify the definition of the multiset extension $\ll$ in order to compare sets of decompositions $d^p(s)$ and $d^q(t)$ only when it is necessary.

Definition *by extension of the decomposition definition*: $d^p(s) \ll_X d^q(t)$ iff

(1) $s/p \notin X$ and $d^p(s) \ll d^q(t)$

(2) $s/p \in X$ and $t/q = s/p$ and $d^p(s) \ll d^q(t)$.

In the following, we write $\ll$ instead of $\ll_X$. Using this definition of $\ll$, it is now possible to decrease the size of the set Q(s, p) of given occurrences in s along the path p, by ruling out the occurrences p if s/p is a variable.

Definition: $Q(s, p) \subseteq \textit{Prefix}(s, p)$, if $s/p \notin X$

$Q(s, p) \subseteq \{u \in Occ(s) \mid u < p\}$, if $s/p \in X$

$Q_+(s, p) = \{v \in Prefix(s, p) \mid (\forall\, w < v),\ \neg(s(v) \leq s(w))\ \&\ (\forall v < w \leq p)\ \neg(s(v) < s(w))\}$, if $s/p \notin X$

$Q_+(s, p) = \{v < p \mid (\forall\, w < v)\ \neg s(v) \leq s(w)\ \&\ (\forall w > v)\ \neg s(v) < s(w)\}$, if $s/p \in X$.

Notice the analogy between the definitions of $Q(s, p)$ and $Q_+(s, p)$ when $s/p = \square$ and when $s/p \in X$. The orderings $\overset{d}{<}$ and $<$ deduced from this definition will be written $\overset{d}{<}_2$ and $<_2$. It is clear that the ordering $\overset{d}{<}_1$ does not depend upon the choices P and Q, as stated by Theorem 2. But it is not so obvious for the ordering $\overset{d}{<}_2$. So we will prove that $\overset{d}{<}_1$ and $\overset{d}{<}_2$ are the same ordering, which will prove the property for $\overset{d}{<}_2$.

Theorem 6: $\overset{d}{<}_1 = \overset{d}{<}_2$.

Proof: Let us use the same choice for both orderings. In fact, the choice Q_1 of the ordering $\overset{d}{<}_1$ is not exactly the choice Q_2 of the ordering $\overset{d}{<}_2$, because if $s/p \in X$ then $p \in Q_1(s, p)$ and $p \notin Q_2(s, p)$. Both choices are the same in all the other cases.

Let us now prove that $s \overset{d}{<}_1 t \Rightarrow s \overset{d}{<}_2 t$, by induction on $|s| + |t|$. Let $p \in P_1(s)$ and $q \in P_1(t)$ such that $d^p(s) \ll_1 d^q(t)$. Two cases can occur.

Case 1: If $s/p = x \in X$ then $t/q = x$ because $p \in Q_1(s, p)$ and x is incomparable with any other symbol and therefore x must appear in a decomposition along the path q in t. Thus it is possible to compare $d^p(s)$ and $d^q(t)$ with $\ll_2$. Let $u \in Q_2(s, p)$. Then $u \in Q_1(s, p)$ and there exists $v \neq q$ and $v \in Q_1(t, q)$ such that $d^p_u(s) <_1 d^q_v(t)$. Using now the four cases of the definition and the induction hypothesis, we obtain $d^p_u(s) <_2 d^q_v(t)$ and thus $d^p(s) \ll_2 d^q(t)$.

Case 2: If $s/p \notin X$, the $Q_1(s, p) = Q_2(s, p)$. If $t/q \notin X$, there is no problem because $Q_1(t, q) = Q_2(t, q)$. If $t/q \in X$ then $q \in Q_1(t, q)$ and $q \notin Q_2(t, q)$. However, $d^p_u(s) <_1 d^q_v(t)$ implies that $v \neq q$ because $t(q)$ is not comparable with $s(u) \notin X$. The result is easily obtained by induction as in Case 1.

Let us prove now that $s \overset{d}{<}_2 t$ implies $s \overset{d}{<}_1 t$ by induction on $|s| + |t|$. In the same way as before, $d^p_u(s) <_1 d^q_v(t)$ follows from $d^p_u(s) <_2 d^q_v(t)$ if $s/p \notin X$. If $s/p \in X$ and $t/q \in X$, we have to prove the inequality:

$$\langle s(p); \square; \{\}; s[p \leftarrow \square] \rangle <_1 \langle t(q); \square; \{\}; t[q \leftarrow \square] \rangle$$

and $d^p(s[p \leftarrow \square]) \ll_1 d^q(t[q \leftarrow \square])$ follows in the same way as before from $d^p(s) \ll_2 d^q(t)$. ▮

Notice that all theorems proved in previous sections remain valid because of the definition $\overset{d}{<}_1$.

Theorem 7: $\overset{d}{<}$ is closed under instantiation, i.e., $s \overset{d}{<} t \Rightarrow \sigma(s) \overset{d}{<} \sigma(t)$, for any substitution σ.

Proof: Straightforward using definition by extension of the decomposition definition. ∎.

9. Conclusion

A major advantage of the decomposition ordering is its utility in easily building, from a set of rewrite rules to be oriented, an ordering on the set F of function symbols. We will illustrate this property with an example from Dershowitz [2, 20], a system which provides normal disjunctive forms of propositional expressions:

(i) $\neg\neg x \to x$

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

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

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

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

The termination of the rule (i) is immediate by the subterm property. Let us prove the termination of the rule (ii). That means

$$s = \neg x \wedge \neg y \overset{d}{<} \neg(x \vee y) = t.$$

The decompositions are:

$$d_*^{11}(t) = \{\langle \neg; x \vee y; \{\,\}; \square\rangle, \langle \vee; x; \{y\}; \neg\, \square\rangle\}$$
$$d_*^{12}(t) = \{\langle \neg; x \vee y; \{\,\}; \square\rangle, \langle \vee; y; \{x\}; \neg\, \square\rangle\}$$
$$d_*^{11}(s) = \{\langle \wedge; \neg x; \{y\}; \square\rangle, \langle \neg; x; \{\,\}; \square \wedge \neg\, y\rangle\}$$
$$d_*^{21}(s) = \{\langle \wedge; \neg x; \{y\}; \square\rangle, \langle \neg; y; \{\,\}; \neg x \wedge \square\rangle\}$$

Then we will get $d_*^{11}(s) \overset{dd}{\ll} d_*^{12}(t)$, and $d_*^{21}(s) \overset{dd}{\ll} d_*^{12}(t)$, only if $\wedge <_F \neg$ or $\wedge <_F \vee$. By exchanging the symbols $\wedge$ and $\vee$ we will get the condition "$\vee <_F \neg$ or $\vee <_F \wedge$" from (iii). Let us now orient the rule (iv).

$$s = (x \wedge y) \vee (x \wedge z) \overset{d}{<} x \wedge (y \vee z) = t$$
$$d_*^{11}(s) = \{\langle \vee; x \wedge y; \{x \wedge z\}; \square\rangle, \langle \wedge; x; \{y\};\ (\square \wedge y) \vee (x \wedge z)\rangle\}.$$
$$d_*^{12}(s) = \{\langle \vee; x \wedge y; \{x \wedge z\}; \square\rangle, \langle \wedge; y; \{x\}; \ldots\rangle\}$$
$$d_*^{21}(s) = \{\langle \vee; x \wedge z; \{x \wedge y\}; \square\rangle, \langle \wedge; x; \{z\}; \ldots\rangle\}$$
$$d_*^{22}(s) = \{\langle \vee; x \wedge z; \{x \wedge y\}; \square\rangle, \langle \wedge; z; \{x\}; \ldots\rangle\}$$

and

$$d_*^{1}(t) = \{\langle \wedge; x; \{y \vee z\}; \square\rangle\}$$
$$d_*^{21}(t) = \{\langle \wedge; y \vee z; \{x\}; \square\rangle; \langle \vee; y; \{z\}; \ldots\rangle\}$$
$$d_*^{21}(t) = \{\langle \wedge; y \vee z; \{x\}; \square\rangle; \langle \vee; z; \{y\}; \ldots\rangle\}.$$

In order to get $d_*^{11}(s) \overset{dd}{\ll} d_*^{1}(t)$ and $d_*^{21}(s) \overset{dd}{\ll} d_*^{1}(t)$ we need $\vee <_F \wedge$. This condition provides successfully the comparison of s and t. The rule (v) can be oriented by the same condition. From those conditions we get easily the following ordering on F: $\vee <_F \wedge <_F \neg$. Such a process can obviously be performed starting from a given partial ordering on F.

This property of the recursive decomposition ordering which leads to the automatic construction of the right ordering $<_F$ on the function symbols is a consequence of our definition when two symbols f and g are incomparable. In that case the two decompositions $\langle f; s'; \mathfrak{S}; s''\rangle$ and $\langle g; t'; \mathfrak{T}; t''\rangle$ are incomparable. Thus the comparison process stops whenever two such decompositions are required to be comparable. The idea is then to add at this step the pair $\langle f,g\rangle$ to the ordering $<_F$ in order to get comparable decompositions. Such a technique does not work with the recursive path ordering because the comparison fails when it exhausts one of the two terms. Because of this essential feature, our ordering is more suitable than Dershowitz's to any application which requires automatic proofs of termination. Our ordering is thus useful in implementing the Knuth-Bendix completion algorithm. A non-incremental version of the decomposition ordering is now implemented and we are currently implementing the incremental one.

Acknowledgment: We would like to thank Nachum Dershowitz and Jean-Luc Remy for their helpful suggestions and John Guttag for reading the manuscript.

10. References

1. Boyer R. S., Moore J S., *A Computational Logic*, Academic Press (1979).
2. Dershowitz N., *Ordering for Term Rewriting Systems*, Proc. 20th Symposium on Foundations of Computer Science (1979), 123-131.
3. Goguen J.A., *How to prove Algebraic Inductive Hypothesis Without Induction*, 5th conf. on Automated Deduction, Lecture Notes in Computer Science, **87** (1980), 356-373.
4. Guttag J.V., Kapur D., Musser D.R., *On Proving Uniform Termination and Restricted Termination of Rewriting Systems*, 9th Int. Coll. on Automata, Languages and Programming, Aarhus, Denmark (1982).
5. Hsiang J., *Refutational Theorem Proving using Term Rewriting Systems*, Dept of Computer Science, University of Illinois at Urbana-Champaign (1981).
6. Huet G., *Confluent Reduction: Abstract Properties and Applications to Term Rewriting Systems*, J. ACM, **27** (1980), 797-821.
7. Huet G., *A complete proof of correctness of the Knuth-Bendix Completion algorithm*, J. Comp. Sys. Sc., **23** (1981), 11-21.
8. Huet G., Hullot J., *Proof by Induction in Equational Theories with Constructors*, Proc. 21th Symposium on Foundations of Computer Science (1980).

9. Huet G., Lankford D.S., *On the Uniform Halting Problem for Term Rewriting Systems*, Rapport Laboria 283, IRIA, Mars 1978, INRIA Rocquencourt, France.

10. Huet G., Oppen D.C., *Equations and Rewrite Rules: a Survey*, in *Formal Languages perspectives and Open Problems*, Ed. Book R., Academic Press (1980).

11. Jouannaud J.P., Lescanne P., *On Multiset Orderings*, to appear in Inform. Proc. Ltrs.

12. Kamin S., Lévy J.J., *Attempts for generalizing the Recursive Path Ordering* to appear.

13. Knuth D.E., Bendix P., *Simple Word Problems in Universal Algebra*, in *Computational Problems in Abstract Algebra*, Ed. Leech J., Pergamon Press (1970), 263-297.

14. Lescanne P., *Two Implementations of the Recursive Path Ordering on Monadic Terms*, 19th Annual Allerton Conf. on Communication, Control, and Computing, Allerton House, Monticello, Illinois (1981), 634-643.

15. Lescanne P., *Some properties of Decomposition Ordering*, Symposium AFCET "The Mathematics for Computer Science", Paris, (march 16-18, 1982).

16. Lescanne P., *Decomposition Ordering as a Tool to prove the Termination of Rewriting Systems*, 7th IJCAI, Vancouver, Canada (1981), 548-550.

17. Lescanne P., Reinig F., *A Well-Founded Recursively Defined Ordering on First Order Terms*, Centre de Recherche en Informatique de Nancy, France, CRIN 80-R-005 (1980).

18. Musser D.L., *On Proving Inductive Properties of Abstract Data Types*, Proc. 7th ACM Symposium on Principles of Programming Languages (1980), 154-162.

19. Plaisted D., *Well-Founded Orderings for Proving Termination of Systems of Rewrite Rules*, Dept of Computer Science Report 78-932, University of Illinois at Urbana-Champaign, July 1978.

20. Plaisted D., *A Recursively Defined Ordering for Proving Termination of Term Rewriting Systems*, Dept of Computer Science Report 78-943, University of Illinois at Urbana-Champaign, Sept. 1978.

21. Raoult J-C., Vuillemin J., *Operational and Semantic Equivalence Between Recursive Programs*, J. ACM., **27** (1980), 772-796.

22. Reinig F., *Les Ordres de Décomposition: un outil incrémental pour prouver la terminaison finie de systèmes de réécriture de termes*, Thèse Université de Nancy, Octobre 1981.

23. Reinig F., Jouannaud J.P., *Decomposition Orderings: a new family of decompostion orderings*, Centre de Recherche en Informatique de Nancy, France, CRIN 81-R-040.

24. Thompson D. H., ed, *AFFIRM Reference Manual*, USC Information Science Institute (1979).

APPENDIX: Examples of decompositions

Let f < g <h and a < b.

```
s =          f                      t =          f
           / | \                               / | \
          g  g  a                             h  h  a
         / \ / \                              |  |
        m  a m  a                             m  b
        |    |                                |  |
        r    r                                r  b
                                              |
                                              a
```

P(s) = {111, 12}
Q(s, 111) = {1, 11, 111}
Q(s, 12) = {1, 12}

P(t) = {111, 2111}
Q(t, 111) = {1, 11, 111}
Q(t, 2111} = {2, 21}

```
     111                    111
    d    (s)               d    (t)
     Q                      Q
```

```
                                      f                                             f
 111               m                / | \          111              m             / | \
d    (s) = <g;     |  ;  {a}; □    g   a >        d    (t) = <h;    |  ;  {} ; □  h  a >
 1                 r               / \              1               r               |
                                  m   a                                             b
                                  |                                                 |
                                  r                                                 b
                                                                                    |
                                                                                    a
```

```
                                  f                                           f
 111                            / | \            111                        / | \
d    (s) = <m; r; {};          g  g  a  >       d    (t) = <m; r; {};      h  h  a >
 11                           / | / \            11                        |  |
                             □  a m  a                                     □  b
                                  |                                           |
                                  r                                           b
                                                                              |
                                                                              a
```

```
                                  f                                           f
 111                            / | \            111                        / | \
d    (s) = <r;  ; {};          g  g  a  >       d    (t) = <r;  ; {};      h  h  a >
 111                          / | / \            111                       |  |
                             m  a m  a                                     m  b
                             |    |                                        |  |
                             □    r                                        □  b
                                                                              |
                                                                              a
```

$d^{12}_{Q}(s)$ $\qquad\qquad$ $d^{2111}_{Q}(t)$

```
                                       f                                      b            f
   12                      m         / | \                2111                |          / | \
  d  (s) = <g;  a;  {|};  □   g    a   >               d     (t) = <h;  b;  {};  h  □   a >
   1                       r         /   \                 2                  |      |
                                    m     a                                   a      m
                                    |                                                |
                                    r                                                r

                                  f                                             f
   12                           / | \                    2111                 / | \
  d  (s) = <a;  ;  {};  g   g    a >                    d     (t) = <b;  b;  {};  h   h   a >
   12                   / \  | \                          21               |      |   |
                       m   □ m  a                                          a      m   □
                       |     |                                                    |
                       r     r                                                    r
```

On Multiset Orderings

Jean-Pierre JOUANNAUD[1]
and
Pierre LESCANNE[2]

Abstract: We propose two well-founded orderings on multisets that extend the Dershowitz-Manna ordering. Unlike the Dershowitz-Manna ordering, ours do not have a natural monotonicity property. This lack of monotonicity suggests using monotonicity to provide a new characterization of the Dershowitz-Manna ordering. Section 5 proposes an efficient and correct implementation of that ordering.

Résumé: Dans cette note, on propose deux ordres bien fondés qui étendent l'ordre de Dershowitz et Manna. Ces ordres ne vérifient pas une propriété naturelle de monotonicité que nous définissons. Aussi cela suggère d'utiliser la monotonicité comme une nouvelle caractérisation de l'ordre de Dershowitz et Manna. La cinquième section de cette note propose une implantion efficace et correcte de cet ordre.

1. Introduction

The multiset ordering proposed by Dershowitz and Manna [2] is a basis for many orderings used for proving termination of programs and term rewriting systems [1], and it would be nice to have an efficient implementation of it. Often, deriving an algorithm directly from a mathematical definition is not the best way. Thus a more suitable definition and a proof that this new program implements the desired function must be found. Our approach was as follows. We tried two definitions, both having efficient implementations but both failing to be equivalent. In fact, they are stronger than the Dershowitz-Manna multiset ordering but do not have a monotonicity property. As an explanation of these facts, we give a new definition of the Dershowitz-Manna ordering based on the main characterization of this ordering: no stronger monotonic ordering exists on multisets. In section 5 of this paper, we propose a correct and efficient implementation of the Dershowitz-Manna ordering.

2. The Dershowitz-Manna ordering

Intuitively, a multiset M (or bag) on E is an unordered collection of elements of E, with possibly multiple occurrences of elements. More formally M is a mapping $E \rightarrow \mathcal{N}$, where $\mathcal{N}$ is the set of natural numbers, associating with each value in E the number of times it occurs in the multiset. For example, x is in M if $M(x)>0$. $\mathcal{M}(E)$ denotes the set of all the finite multisets on E, i.e. the multisets M with finite carrier $\{x \in E \mid M(x) \neq 0\}$. The empty multiset $\{\ \}$ has $\{\ \}(x) = 0$, for all x in E. A set is a particular case of a multiset such that $S(x) \leq 1$ for each x in E.

Definition: *Sum of multisets*. The sum of two multisets M and N is the multiset M + N such that $M + N(x) = M(x) + N(x)$.

---

1. Centre de Recherche en Informatique de Nancy, CO 140, F54037 Nancy, France
2. Centre de Recherche en Informatique de Nancy, CO 140, F54037 Nancy, France and Laboratory for Computer Science, Massachusetts Institute of Technology, 545 Technology Square, Cambridge, Massachusetts, U.S.A.

$M+N$ is an associative and commutative operation on $\mathcal{M}(E)$ with neutral element $\{\}$. If $M_1, M_2, \ldots, M_p$ are multisets, $\sum_{i=1}^{p} M_i$ is the multiset such that $(\sum_{i=1}^{p} M_i)(x) = \sum_{i=1}^{p} M_i(x)$. $M+N$ is a set only if M and N are disjoint sets; in this case $+$ is the classical disjoint union or direct sum of sets.

Definition: *Inclusion of multisets*. Multiset M is included into multiset N, (written $M \subseteq N$), if and only if $(\forall x \in E)\ M(x) \leq N(x)$.

Definition: *Difference of multisets*. If $M \subseteq N$, the difference $N-M$ is defined by $(N-M)(x) = N(x) - M(x)$.

In this paper, an ordering $\prec$ on a set is a partial or total strict ordering i.e. an irreflexive and transitive relation on E. We use the notation $x \# y$ to mean $\neg(y \prec x \text{ or } x = y \text{ or } x \prec y)$. Assume throughout that E is ordered by $\prec$.

Definition: *The Dershowitz-Manna Ordering*.
$M \lll N$ if there exist two multisets X and Y in $\mathcal{M}(E)$ satisfying

(i) $\{\} \neq X \subseteq N$
(ii) $M = (N-X) + Y$
(iii) X dominates Y: $(\forall y \in Y)(\exists x \in X)\ y \prec x$.

Using this definition, it may be difficult to prove that two multisets are not related by $\lll$. In [3], Huet and Oppen give a different and more tractable definition.

Definition: *The Huet-Oppen Definition*.

$$M \lll N \text{ iff } M \neq N \ \&\ [M(y) > N(y) \Rightarrow (\exists x \in E)\ y \prec x \ \&\ M(x) < N(x)]. \qquad \text{(HO)}$$

Lemma 1: The Dershowitz-Manna definition is equivalent to the Huet-Oppen definition.

Proof: Let $\lll_{DM}$ and $\lll_{HO}$ be the Dershowitz-Manna and Huet-Oppen orderings, respectively.

Assume $M \lll_{HO} N$, and define X and Y as follows:

$X(x) = \max\{N(x) - M(x), 0\}$
$Y(y) = \max\{M(y) - N(y), 0\}$.

Let us prove (i)

1) $X \subseteq N$ is clear by definition.
2) Because $M \neq N$ there exists z such that $M(z) \neq N(z)$. If $M(z) < N(z)$ then $z \in X$, if $M(z) > N(z)$, by (HO) there exists x, such that $z \prec x$ and $M(x) < N(x)$. Hence $x \in X$. In both cases $X \neq \{\}$.

(ii) is true by construction. To prove (iii), let $y \in Y$. By hypothesis, there exists x satisfying $y \prec x$ and $M(x) < N(x)$. Hence $(\exists x \in X)\ y \prec x$.

Suppose $M \lll_{DM} N$. $M \neq N$ because $X \neq \{\}$ and $Y \neq X$. Without losing the generality we may assume X and Y disjoint i.e. $X(x) > 0 \Rightarrow Y(x) = 0$ & $Y(x) > 0 \Rightarrow X(x) = 0$. Otherwise X would be replaced by $X - Z$ and Y by $Y - Z$ where $Z(x) = \min\{X(x), Y(x)\}$. Assume $M(y) > N(y)$. By (ii) $M(y) = N(y) + Y(y) - X(y)$ that implies $Y(y) > X(y) \geq 0$ which means $y \in Y$. By (iii) there exists $x \in X$, i.e. $X(x) > 0$, such that $y \prec x$. The value of M in x is $M(x) = (N(x) - X(x)) + Y(x) = N(x) - X(x)$ because $Y(x) = 0$. Thus $M(x) < N(x)$. ∎

Another important property of the Dershowitz-Manna ordering is monotonicity.

Definition: *Monotonicity*. Let $\prec$ be a partial ordering on E and τ a mapping from $E \times E$ into $\mathcal{M}(E) \times \mathcal{M}(E)$. $\tau(\prec)$ is said to be a monotonic extension of $\prec$ iff:

(1) $\tau(\prec)$ is an ordering.

(2) τ is monotonic i.e. $\prec \subseteq < \Rightarrow \tau(\prec) \subseteq \tau(<)$.

Lemma 2: Monotonicity Lemma. The Dershowitz-Manna ordering $\prec\!\!\prec$ is a monotonic extension of $\prec$.

Proof: Straightforward using the Huet-Oppen definition. ∎

3. Partition based orderings

We now define two multiset orderings based on partitioning a multiset. We say that $\{M_i \mid i = 1,\ldots,p\}$ is a partition of a multiset iff $M = \sum_{i=1}^{p} M_i$. Assume now, we are able to compare the M_i using an ordering $\prec$, and thus to sort them such that $M_p \preceq M_{p-1} \preceq \ldots \preceq M_1$. It is now easy to define a new ordering $\prec_{\mathcal{g}}$ for comparing the multisets $M = \sum_{i=1}^{p} M_i$ and $N = \sum_{i=1}^{q} N_i$ using a lexicographical extension of $\prec$:

$$M \prec_{\mathcal{g}} N \text{ iff } M_1 M_2 \ldots M_p \prec^{lex} N_1 N_2 \ldots N_q.$$

In practice, we have to define the basic ordering $\prec$ and the method for constructing the partition of a given multiset.

3.1 The multiset ordering $\prec\!\!\prec_{\mathcal{M}}$

Assume that the partition $\tilde{M} = (M_i \mid i = 1,\ldots,p)$ of the multiset M satisfies the following properties:

(1) $x \in M_i \Rightarrow M_i(x) = M(x)$.

(2) $x \in M_i$ and $y \in M_i \Rightarrow$ x and y are incomparable.

(3) $(\forall\, i \in [2..p])\quad x \in M_i \Rightarrow (\exists y \in M_{i-1})\ x \prec y$.

Intuitively the partition is built by first computing the multiset M_1 of all the maximal elements and then recursively computing the partition of $M - M_1$.

Example 1: $M = \{a, a, 2, 2, b, 1, 1\}$ with $a \prec b$, $1 \prec 2$. $M_1 = \{2, 2, b\}$ and $M_2 = \{a, a, 1, 1\}$.

Let $\prec_{\mathcal{M}}$ be the following basic ordering on multisets.

$M \prec_{\mathcal{M}} N$ iff $M \neq N$ and $(\forall x \in E)\ M(x) \leq N(x)$ or $(\exists y \in N)\ x \prec y$.

Let $\mathcal{M}'(E)$ be the subset of $\mathcal{M}(E)$ such that if $M \in \mathcal{M}'(E)$ and $x \in M$ and $y \in M$ then x and y are incomparable. $\prec_{\mathcal{M}}$ is an ordering on $\mathcal{M}'(E)$ equivalent to the restriction of the Dershowitz-Manna ordering to $\mathcal{M}'(E)$.

Definition: Let M and N be multisets. We say that $M \prec\!\!\prec_{\mathcal{M}} N$ iff $\tilde{M} \prec^{lex}_{\mathcal{M}} \tilde{N}$.

Example 2: If $a \prec b$ and $M = \{1, a, b\}$ then $M_1 = \{1, b\}$, $M_2 = \{a\}$. If $N = \{1,1,b\}$ then $N_1 = \{1,1,b\}$ and $M \prec\!\!\prec_{\mathcal{M}} N$.

It is easy to see that $\prec\!\!\prec_{\mathcal{M}}$ is an ordering because lexicographical extension preserves the orderings. We will show that $\prec\!\!\prec_{\mathcal{M}}$ is more powerful than $\prec\!\!\prec$.

Lemma 3: $M \prec\!\!\prec N$ implies $M \prec\!\!\prec_{\mathcal{M}} N$.

Proof: Suppose $M \prec\!\!\prec N$, and $\tilde{M} = M_1 ... M_p$, and $\tilde{N} = N_1 ... N_q$. Then we have $M \prec\!\!\prec_{\mathcal{M}} N$. The proof is by induction on $p+q$. The result is obvious if $M = \{\ \}$. Assume that $p \neq 0$ and $q \neq 0$. Three cases have to be distinguished.

- $M_1 \prec_{\mathcal{M}} N_1$. The result is straightforward.
- $M_1 = N_1$. Thus $M_2 ... M_p \prec\!\!\prec_{\mathcal{M}} N_2 ... N_q$ and the result follows by the induction hypothesis.
- $N_1 \prec_{\mathcal{M}} M_1$ or M_1 and N_1 are incomparable. There must exist an $x \in M_1$ such that $M_1(x) > N_1(x)$ and $(\forall y \in N_1)\ x \not\prec y$. This contradicts the hypothesis $M \prec\!\!\prec N$. ∎

We show that the converse is false by using the previous example. We see that $M(a) = 1 > N(a) = 0$. However, the only element in N greater than a, i.e. b, has one occurrence in both M and N. Thus M and N are incomparable using the Dershowitz-Manna ordering. On the other hand, $\prec\!\!\prec_{\mathcal{M}}$ has a serious drawback, which prevents it from being used in some usual cases, requiring incremental orderings; it is not monotonic. Let us go back to Example 2 and assume now $1 \prec a \prec b$. This increases the basic ordering by adding the new pair $1 \prec a$. We now get:

$M_1 = \{b\}$, $M_2 = \{a\}$, $M_3 = \{1\}$
$N_1 = \{b\}$, $N_2 = \{1,1\}$.

Thus $N \prec\!\!\prec_{\mathcal{M}} M$!!

3.2 The multiset ordering $\prec\!\!\prec_{\mathcal{S}}$

A different way to construct a partition of a multiset is to require that each multiset in the partition be a set. Thus the partition $\bar{M} = (S_i \mid i = 1...p)$ must satisfy the following properties:

(1) S_i is a set, that is $S_i(x) \leq 1$.
(2) $x \in S_i$ and $y \in S_i$ implies x and y are incomparable.
(3) $(\forall\, i \in [2..p])\ x \in S_i$ implies $(\exists\, y \in S_{i-1})\ x \preceq y$.

The only difference between the partitions created by $\prec\!\!\prec_{\mathcal{M}}$ and $\prec\!\!\prec_{\mathcal{S}}$ is in condition (1). As with $\prec\!\!\prec_{\mathcal{M}}$, the partition here is built by first computing the set S_1 of maximal elements, and then recursively computing the partition of $M - S_1$.

Example 3: $M = \{a,a,2,2,b,1,1\}$ with $a \prec b$ and $1 \prec 2$ and $S_1 = \{2,b\}$, $S_2 = \{2,a\}$, $S_3 = \{1,a\}$, $S_4 = \{1\}$.

Let $\prec_{\mathcal{S}}$ be the following ordering on sets:

$S \prec_{\mathcal{S}} T$ iff $S \neq T$ and $(\forall\, x \in S)\ (\exists\, y \in T)\ x \preceq y$.

In the following, $\bar{N} = (T_i \mid i = 1..q)$ will be the partition of N. If sets are considered as a particular case of multisets, $\prec_{\mathcal{S}}$ is $\prec\!\!\prec$ on the sets of incomparable elements.

Definition: Let M and N be multisets. We say that $M \prec\!\!\prec_{\mathcal{S}} N$ iff $\bar{M} \prec_{\mathcal{S}}^{lex} \bar{N}$.

Example 4: Suppose a and b incomparable. If $N = \{a,b\}$ then $T_1 = \{a,b\}$, if $M = \{b,b\}$ then $S_1 = \{b\}$, $S_2 = \{b\}$ and $M \prec\!\!\prec_{\mathcal{S}} N$.

Once more, it is easy to see that $\prec\!\!\prec_{\mathcal{S}}$ is an ordering. Let us show that it is more powerful that $\prec\!\!\prec$.

Lemma 4: $M \prec\!\!\prec N$ implies $M \prec\!\!\prec_{\mathcal{S}} N$.

<u>Proof</u>: by induction on p and q. The result is straightforward if $M = \{\}$. Else, we have to distinguish three cases.

- $S_1 \prec_{\mathcal{S}} T_1$, the result is straightforward.
- $S_1 = T_1$, by induction hypothesis.
- $T_1 \prec_{\mathcal{S}} S_1$ or S_1 and T_1 are incomparable. There must exist an $x \in S_1$ such that $(\forall\, y \in T_1)$, $\neg x \preceq y$. This contradicts the hypothesis $M \prec\!\!\prec N$. ∎

Once again, the converse is false, in the same way as shown in the previous example: $M = \{b,b\}$ and $N = \{a,b\}$.

Suppose now $a \prec b$ and M and N are as in example 4. We get $S_1 = \{b\}$, $S_2 = \{b\}$ and $T_1 = \{b\}$, $T_2 = \{a\}$. Thus $N \prec\!\!\prec_{\mathcal{S}} M$. Therefore $\prec\!\!\prec_{\mathcal{S}}$ is not a monotonic ordering.

Now, let try to compare $\prec\!\!\prec_{\mathcal{S}}$ and $\prec\!\!\prec_{\mathcal{M}}$, using two examples with $a \prec b$:

- $M = \{1,a,b\}$, $N = \{1,1,b\}$, $M \prec\!\!\prec_{\mathcal{M}} N$ and $\neg\, M \prec\!\!\prec_{\mathcal{S}} N$.
- $M = \{b,b\}$, $N = \{1,b\}$, $M \prec\!\!\prec_{\mathcal{S}} N$ and $\neg\, M \prec\!\!\prec_{\mathcal{M}} N$.

Thus $\prec\!\!\prec_{\mathcal{S}}$ and $\prec\!\!\prec_{\mathcal{M}}$ are not comparable.

3.3 Well-foundedness

We have the following theorem.

Theorem 1: If $\prec$ is well-founded on E, then $\prec\!\!\prec$, $\prec\!\!\prec_{\mathcal{M}}$ and $\prec\!\!\prec_{\mathcal{S}}$ are well-founded on $\mathcal{M}(E)$.

<u>Proof</u>: [2] contains a proof of the well-foundedness of $\prec\!\!\prec$. Note that a proof of well-foundedness of $\prec\!\!\prec_{\mathcal{S}}$ or $\prec\!\!\prec_{\mathcal{M}}$ is also a proof of well-foundedness of $\prec\!\!\prec$ by Lemma 3 and 4. A proof of well-foundedness of $\prec\!\!\prec_{\mathcal{S}}$ and $\prec\!\!\prec_{\mathcal{M}}$ can be easily obtained by proving that $\prec_{\mathcal{S}}$ and $\prec_{\mathcal{M}}$ are well-founded. This can be done by using König Lemma as in [2]. On the other hand, note that $\prec_{\mathcal{S}}$ and $\prec_{\mathcal{M}}$ are particular cases of $\prec\!\!\prec$. ∎

4. A property of maximality of the Dershowitz-Manna Ordering

In the previous section, we showed two non monotonic orderings containing the Dershowitz-Manna ordering $\prec\!\!\prec$. A natural question arises: Do monotonic orderings exist on $\mathcal{M}(E)$ that contain $\prec\!\!\prec$? The answer is negative and provides a new important characterization of $\prec\!\!\prec$. Let us first prove an important lemma.

Lemma 5: Let $\prec$ be a partial ordering on E and let M and N be two multisets on E such that $N \not\preceq\!\!\prec M$ that is $\neg(N = M$ or $N \prec\!\!\prec M)$. Then there exists a partial ordering $<$ on E such that $< \,\supseteq\, \prec$ (that is $y \prec x \Rightarrow y < x$) and $M \ll N$.

Proof: By induction on the set $D = \{(x,y) \in M \times N \mid x \# y\}$.
Basic case: Let $D = \emptyset$. Then $N \not\preceq\!\!\!\prec M \Rightarrow M \prec\!\!\prec N$.
General case: Let D be not empty. Then either $M \prec\!\!\prec N$ and the result is proved with $< = \prec$, or $N \not\preceq\!\!\!\prec M$ and there must exist a pair (x,y) such that

(1) $M(x) > N(x)$ and $(\forall z \in E)\ x \prec z \Rightarrow N(z) \leq M(z)$.

(2) $M(y) < N(y)$ and $(\forall z \in E)\ y \prec z \Rightarrow M(z) \leq N(z)$.

It follows from (1) and (2) that $x \# y$ and thus $(x,y) \in D$. Let now $<$ the transitive closure of the relation union of $\prec$ and the pair (x,y). $<$ is clearly an ordering strictly containing $\prec$. Therefore $\ll$ is an ordering strictly containing $\prec\!\!\prec$. Since $x \prec y$, either $M \ll N$ and the result is true or M and N are incomparable according to $\ll$ and the result follows from the induction hypothesis used with a new D, whose the cardinal is strictly less than the previous one. ∎

Theorem 2 *Maximality*: Let $\prec$ a partial ordering on E and $\tau(\prec)$ a monotonic extension of $\prec$ such that $\prec\!\!\prec \subseteq \tau(\prec)$. Then $\prec\!\!\prec$ and $\tau(\prec)$ are the same ordering.

Proof: Assume first $\prec$ is total. Then $\prec\!\!\prec$ is total on $\mathcal{M}(E)$ and must coincide with $\tau(\prec)$. Assume now that $\prec$ is partial on E. Then $\prec\!\!\prec$ is partial on $\mathcal{M}(E)$. Suppose that $\tau(\prec) \supset \prec\!\!\prec$. Then there must exist two multisets M and N such that $M\ \tau(\prec)\ N$ and $M \# N$ for the ordering $\prec\!\!\prec$. Using Lemma 5 there exists $< \supseteq \prec$ such that $N \ll M$, which implies $N\ \tau(<)\ M$ by hypothesis. Using now the monotonicity of the multiset extension τ and the hypothesis $M\ \tau(\prec)\ N$, we get $M\ \tau(<)\ N$, which is a contradiction. ∎

This main property of the Dershowitz-Manna ordering can be used to give a simple proof of equivalence of (HO) and (DM). In the following, we use this technique to present and prove a new definition of $\prec\!\!\prec$. If $\prec$ is a total ordering on E, $\prec^{lex}$ is a total ordering on the ordered lists on E which provides a simple definition of the Dershowitz-Manna ordering in that particular case: let $list(M) = \{x_1, x_2, \ldots, x_n\}$ with $j > i \Rightarrow x_j \preceq x_i$ be the ordered list representation of multiset M. Then $M \prec\!\!\prec N$ iff $list(M) \prec^{lex} list(N)$.

Let us now define a new multiset ordering $\prec\!\!\prec_*$ in the following way:

Definition: Given a partial ordering $\prec$ on E, let $M \prec\!\!\prec_* N$ iff for all $<$ that are total orderings containing $\prec$, $list(M) <^{lex} list(N)$.

It is easy to prove that this new ordering is exactly the Dershowitz-Manna ordering, as an application of Theorem 2.

Lemma 6: $\prec\!\!\prec_*$ is a monotonic ordering.

Proof: Follows obviously from the definition. ∎

Lemma 7: $\prec\!\!\prec \subseteq \prec\!\!\prec_*$.

Proof: Suppose $M \prec\!\!\prec N$ and $\neg(M \prec\!\!\prec_* N)$. There exists a total ordering $<$ such that $< \supseteq \prec$ and $\neg(list(N) <^{lex} list(M))$. Thus there exists y such that $M(y) > N(y)$ and $y < z \Rightarrow N(z) = M(z)$, which implies $y \prec z \Rightarrow N(z) = M(z)$. Combining this result and (HO), we infer $\neg(M \prec\!\!\prec N)$, which is a contradiction. ∎

Theorem 3: $\prec\!\!\prec_*$ and $\prec\!\!\prec$ are the same multiset ordering.

Proof: It follows from Theorem2, Lemma 6 and Lemma 7.

5. An efficient implementation of the Dershowitz-Manna ordering

It is easy to derive an implementation of the Dershowitz-Manna ordering from the Huet-Oppen definition but it is not efficient because a comparison is performed for each pair of items. Moreover it leads to an algorithm that does not work symmetrically on the data. We propose an implementation based on the following idea. Given a pair of multisets M_k and N_k, build a new pair M_{k+1} and N_{k+1} such that at least one of the two multisets gets smaller and the value of the comparison comp(M_k, N_k) does not change, which means comp(M_k, N_k) = comp(M_{k+1}, N_{k+1}). The process is repeated until it is possible to decide easily whether M is greater than or less than or equal to or incomparable to N. To decrease M_k and N_k choose a pair (a,b) in $M_k \times N_k$ and do the following:

1. If a = b and $M_k(a) = N_k(b)$, a is removed from M_k and N_k.
2. If $a \prec b$ or a = b and $M_k(a) < N_k(b)$ then we would like to remove a from M_k without changing the value of comp(M_k,N_k). This is possible if a is maximal in M_k, in which case $a \prec b$ or a = b and $M(a) < N(b)$ implies $M_k \prec\prec N_k$ or M_k and N_k are incomparable (written $M_k \# N_k$). Thus comp(M_k, N_k) is not changed by removing a from M_k. Note that in this case one may remove with a all the elements in M_k which are less than a.

Thus instead of choosing any elements a and b in M_k and N_k, choose maximal elements in M_k and N_k. Thus after removing a from M_k, we have to compute the set of maximal elements of M_{k+1}. This is not difficult. Let us first define the function *succ* as

$(y \in succ(x) \Rightarrow y \prec x)$ & $(y \prec x \Rightarrow [\exists\, z \in succ(x)]\; y \preceq z)$,

and then

In case (1), $M_{k+1} = M_k - \{a\}$, then

$\text{Maximal}(M_{k+1}) = \text{Maximal}(M_k) - \{a\} + \{x \mid x \in succ(a)\ \&\ x \notin succ(a') \text{ for } a \neq a'\}$.

In case (2), $M_{k+1} = M_k - \{x \mid x \in succ^*(a)\}$ where $succ^*(a) = \sum_{i=1}^{n} succ^{\,i}(a)$, then

$\text{Maximal}(M_{k+1}) = \text{Maximal}(M_k) - \{a\}$.

This suggests representing a multiset M as a directed acyclic graph representing the relation *succ*. Each node contains a triple: the value x of the element, M(x), the number of antecedents nb_ant(x,M) of x in M i.e. card{y | *succ*(y) = x}. If this last number is 0, x belongs to Maximal(M).

The arrows deduced by transitivity are not necessary and the algorithm is more efficient if they are not present. For example if the definition of *succ* is the following minimal one:

$y \in succ(x) \Leftrightarrow y \prec x\ \&\ \neg\,(\exists z \in M)\; x \prec z \prec y$.

For example if $c \prec a$, $d \prec a$, $d \prec b$, $e \prec c$, $f \prec c$, $f \prec d$ and M = {a,a,b,b,b,c,c,d,e,f,f}, a representation of M is

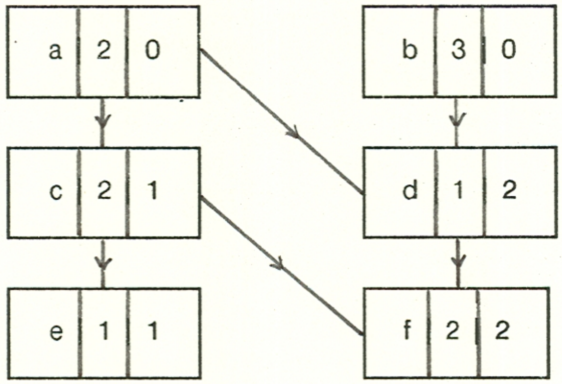


Figure 1 gives an algorithm describing our implementation. Note that if it is not possible to choose a new pair in the body of the loop, then all the elements present in Maximal(M) and Maximal(N) are incomparable. Then easily $M \#_{\mathcal{M}b} N$ and by Lemma 3, $M \# N$. The only problem is to prove termination of the algorithm, but it is easy to see that, although Maximal(M) and Maximal(N) can increase, they remain included in M and N which do decrease. Thus the algorithm terminates for any choice which computes a new pair (a,b) at each iteration.

Acknowledgment: We would like to thank the referees for their helpful comments and Sriram Atreya for reading a version of the manuscript.

6. References

1. Dershowitz N.: *Ordering for Term Rewriting Systems*, Proc. 20th Symposium on Foundations of Computer Science, (1979), 123-131.

2. Dershowitz N., Manna Z.: *Proving Termination with Multiset Orderings*, Comm. ACM, **22**, (1979), 465-476.

3. Huet G., Oppen D.C.: *Equations and Rewrite Rules: a Survey*, in *Formal Languages perspectives and Open Problems*, Ed. Book R., Academic Press, (1980).

Fig. 1. An algorithm to implement the Dershowitz-Manna ordering

```
while "possible" do
        choose a new pair (a,b) in Maximal(M) × Maximal(N)
        if b-<a or [a = b and M(a)>N(b)] then
                for each x in succ*(b,N) do remove(x,N)
                end for
        end if

        if a-<b or [a = b and M(a)<N(b)] then
                for each x in succ*(a,M) do remove(x,M)
                end for
        end if

        if a = b and M(a) = N(b) then
                for each x in succ(a,M) do
                    nb_ant(x,N): = nb_ant(x,N) - 1
                    if nb_ant(x,N) = 0 then ad(x,Maximal(M))
                end for

                remove(a,Maximal(M))

                for each x in succ(b,N) do
                    nb_ant(x,N): = nb_ant(x,N) - 1
                    if nb_ant(x,N) = 0 then ad(x,Maximal(N))
                end for

                remove(b,Maximal(N))
        end if

end while

if Maximal(M) = { } then if Maximal(N) = { } then return(" M = N ")
                                              else return(" M-<<N ") end if
                    else if Maximal(N) = { } then return("N-<<M")
                                              else return(" M # N ") end if
end if
```