RAIRO. INFORMATIQUE THÉORIQUE

PIERRE LESCANNE

Some properties of decomposition ordering, a simplification ordering to prove termination of rewriting systems

RAIRO. Informatique théorique, tome 16, nº 4 (1982), p. 331-347
<http://www.numdam.org/item?id=ITA_1982__16_4_331_0>

© AFCET, 1982, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Informatique théorique » implique l'accord avec les conditions générales d'utilisation (http://www.numdam.org/conditions). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM
Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
http://www.numdam.org/

R.A.I.R.O. Informatique théorique/Theoretical Informatics
(vol. 16, n° 4, 1982, p. 331 à 347)

SOME PROPERTIES OF DECOMPOSITION ORDERING, A SIMPLIFICATION ORDERING TO PROVE TERMINATION OF REWRITING SYSTEMS (*)

by Pierre LESCANNE (1)

Communicated by M. SINTZOFF

Abstract. — *Recursive path ordering is an ordering on terms introduced by Dershowitz to prove the termination of rewriting systems. A new ordering, called decomposition ordering, is defined. It is proved to be equivalent to the path ordering and, as corollary, a simple proof of the well-foundedness of recursive path ordering is given.*

Résumé. — *L'ordre récursif sur les chemins a été introduit par Dershowitz pour prouver la terminaison des systèmes de réécriture. On propose ici un nouvel ordre appelé ordre de décomposition et l'on montre qu'il est équivalent à l'ordre récursif sur les chemins et comme corollaire on donne une preuve simple de la bonne fondation de ce dernier.*

1. INTRODUCTION

This paper introduces a new definition of recursive path ordering when a total ordering is given on the set of symbols. Decomposition ordering is based on a preliminary and comprehensive analysis of terms. This preparation called *decomposition* arranges relevant information in such an way that ordering looks like a lexicographical ordering. The recursive path ordering or the equivalent decomposition ordering is a good tool to prove the termination of rewriting systems [6], and therefore to prove inductive properties by the Knuth and Bendix method [4, 5, 8, 13].

The second section of this paper is devoted to the definition of the recursive path ordering on terms on a partially ordered set of symbols, we explain why it is possible to focus our study only on totally ordered set of symbols if we want to prove the well-foundedness. Thus, we define properties related to such a set of symbols, namely left-weighted terms and recursive lexicographic

(*) Received September 1981.

(1) Centre de Recherche en Informatique de Nancy and Laboratory for Computer Science, Massachusetts Institute of Technology. *Present address:* Centre de Recherche en Informatique de Nancy Campus Scientifique, B.P. 239, 54506 Vandœuvre-lès-Nancy, France.

R.A.I.R.O. Informatique théorique/Theoretical Informatics, 0399-0540/1982/331/$ 5.00
© AFCET-Bordas-Dunod

ordering. In the third section, we give the proofs of the main theorems in a particular case where F is a set of symbols of monadic functions. This is an introduction to the fourth section where the decomposition ordering is defined and the main theorems are proved, namely equivalence of both ordering and their well-foundedness. In the conclusion, we sketch a comparative study of two algorithms which can be deduced from each definition.

2. THE RECURSIVE PATH ORDERING

2.1. Multiset ordering

A multiset on $\mathscr{D}$ is a mapping $S: \mathscr{D} \to N$. The set of multisets on $\mathscr{D}$ is denoted by $\mathscr{M}(\mathscr{D})$. For example, $\{1, 3, 3, 5, 5, 5\} = S$ is a multiset on N, where $S(1)=1$, $S(3)=2$, $S(5)=3$ and $S(x)=0$ for all other $x \in N$. Let $<_{\mathscr{D}}$ be an ordering on $\mathscr{D}$, and $<_N$ be the canonical ordering on N.

DEFINITION 1: The multiset ordering deduced from $<_{\mathscr{D}}$ is defined as follow:

$$S << T \quad \text{if and only if}$$

$$S \neq T \text{ and } (\forall x \in \mathscr{D})(T(x) <_N S(x) \quad \Rightarrow \quad [(\exists y \in \mathscr{D})\, x <_{\mathscr{D}} y \text{ and } (S(y) <_N T(y)]),$$

i. e., if an element x occurs more frequently in T, there exists another element y greater than x that occurs more frequently in S.

Example 1: If $\mathscr{D} = \{1, 2, 3, \ldots\} \cup \{a, b, c, \ldots\}$ and $1 < 2 < 3 < \ldots$ and $a < b < c < \ldots$ we have:

$$\{2, 2, 2, 3, 4, a, a, b, c, c\} << \{2, 3, 3, 4, b, b, d\}.$$

LEMMA 1 [18]: *If $\prec$ contains $<$ in the relation sense, then $\prec\prec$ contains $<<$. In other words:*

$$[(\forall x, y \in \mathscr{D})(x < y) \Rightarrow (x \prec y)]$$
$$\Rightarrow \quad [(\forall S, T \in \mathscr{M}(\mathscr{D}))(S << T) \Rightarrow (S \prec\prec T)]. \quad \blacksquare$$

LEMMA 2: *If $<$ is total, then $<<$ is the left to right lexicographic ordering on the decreasing sorted list of the elements.* $\blacksquare$

2.2. Recursive path ordering

The set $T(F)$ of terms on F terms without arity restrictions will be considered here. Let us consider the congruence $s \overset{*}{=} t$ recursively defined as follows:

DEFINITION 2: $f(s_1, ..., s_m) \overset{*}{=} g(t_1, ..., t_m)$ iff $f=g$ and there exists a permutation $\sigma \in S_n$ such that $s_i \overset{*}{=} t_{\sigma(i)}$.

Let $<$ be an ordering on the set F of the symbols.

DEFINITION 3: The *recursive path ordering* [1] over $T(F)$ is recursively defined as follows:

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

(3.1) $f=g$ and $\{s_1, \ldots, s_m\} \overset{*}{<} \overset{*}{<} \{t_1, \ldots, t_n\}$,

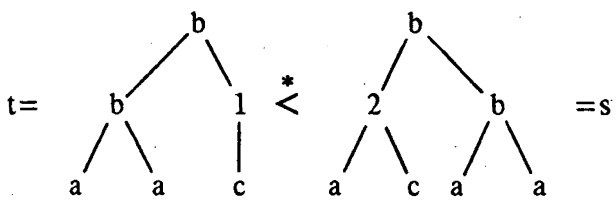
or (3.2) $f<g$ and for all s_i, $s_i \overset{*}{<} t$,

or (3.3) $\neg f \leqq g$ and for some t_j, $s \overset{*}{<} t_j$ or $s \overset{*}{=} t_j$,

this definition can be "less deterministic", changing (3.3) to:

(3.3′) for some t_j, $s \overset{*}{<} t_j$ or $s \overset{*}{=} t_j$.

Example 2: If F is the set $\mathscr{D}$ of example 1, the following figure gives two terms which are compared:



This inequality can be proved as follows:

$t \overset{*}{<} s$ *iff* (by rule 3.1) $S=\{b(a, a), 1(c)\} \overset{*}{<} \overset{*}{<} \{2(a, c), b(a, a)\}=T$, because the top operators are the same;

iff (by Definition 1 giving the multiset ordering, here $\mathscr{D}$ is $T(F)$) $[\exists y \in T(F)]\, 1(c) \overset{*}{<} y$ and $S(y) <_N T(y)$;

iff (choosing $y=2(a, c)$ and because $S(2(a, c))=0$ and $T(2(a, c))=1$):

$$1(c) \overset{*}{<} 2(a, c);$$

iff (by rule 3.2, because $1<2$) $c \overset{*}{<} 2(a, c)$;

iff (by rule 3.3′ with $j=2$) $c=c$;

iff true.

2.3. Total ordering on F is enough for proving well-foundedness

In this section we give some lemmas showing that it is possible to suppose the ordering on F to be total when we want only to prove the well-foundedness.

LEMMA 3: *Let $\prec$ be an ordering containing $<$, in the relation sense: i.e., $f < g \Rightarrow f \prec g$, then $\overset{*}{\prec}$ contains $\overset{*}{<}$. In other words, $s \overset{*}{<} t \Rightarrow s \overset{*}{\prec} t$.*

Proof: Suppose $s \overset{*}{<} t$. The reasoning is by induction on the structure of terms. If the first or second condition is applied to prove $s \overset{*}{<} t$, then trivially $s \overset{*}{\prec} t$. Otherwise, the condition (3.3′) is used and ($s \overset{*}{<} t_j$ or $s \overset{*}{=} t_j$) implies ($s \overset{*}{\prec} t_j$ or $s \overset{*}{=} t_j$) thus $s \overset{*}{\prec} t$. ■

LEMMA 4: *If $\prec$ contains $<$ and if $\overset{*}{\prec}$ is well-founded, then $\overset{*}{<}$ is well-founded.*

Proof: By Lemma 3, if $t_1 \overset{*}{>} t_2 \overset{*}{>} \ldots \overset{*}{>} t_n \overset{*}{>} \ldots$ is an infinite $\overset{*}{<}$-decreasing sequence on $T(F)$, it is an infinite $\overset{*}{\prec}$-decreasing sequence. ■

By Lemma 3, $\overset{*}{<}$ is well-founded if $<$ can be embedded in an ordering $\prec$ on F such that $\overset{*}{\prec}$ on $T(F)$ is well-founded. It will be proved constructively, that a necessary condition is that $\prec$ is total and well-founded (i.e., well-ordered). In other words, a result of this paper can be: $\overset{*}{<}$ is well-founded if $<$ can be embedded in a well-ordering.

LEMMA 5: *Every well-founded ordering can be embedded in a well-ordering.*

Proof: The proof follows from Zorn's Lemma.

The Lemma 5 allows us to translate, the previous statement into: $\overset{*}{<}$ is well-founded if $<$ is well-founded ([2]). Henceforth, $<$ will be a well-ordering, therefore a total ordering.

2.4. Left-weighted terms

The following result is easily provable.

([2]) It will be noted that Zermelo's Theorem is a particular case of Lemma 5 when the well-founded relation is the empty relation. Therefore, Lemma 5 is equivalent to the axiom of choice. In most cases, F is finite, so $<$ is well-founded and can be embedded constructively in a well-ordering, by topological sort.

LEMMA 6: *If F is totally ordered, if s and t are given, one and only one of the following assertions is always true:*

$$s \overset{*}{<} t, \qquad s \overset{*}{=} t, \qquad s \overset{*}{>} t.$$

DEFINITION 4: A term $f(s_l, \ldots, s_m)$ is called recursively path left-weighted or more simply $\overset{*}{<}$*-left-weighted* iff for all i, s_i is $\overset{*}{<}$-left-weighted and for all i and j, $i<j$ implies $s_i \overset{*}{\leqq} s_j$ (where $s \overset{*}{\leqq} t$ means $s \overset{*}{<} t$ or $s \overset{*}{=} t$).

LEMMA 7: (i) *Two* $\overset{*}{<}$*-left-weighted terms are* $\overset{*}{=}$ *congruent if and only if they are equal.*

(ii) *In each class there exists a unique* $\overset{*}{<}$*-left-weighted term.*

(iii) *Because* $<$ *is a total ordering, the restriction of the ordering* $\overset{*}{<}$ *to the* $\overset{*}{<}$*-left-weighted terms is a total ordering.*

The $\overset{*}{<}$-left-weighted terms are the canonical forms of $\overset{*}{=}$-classes. Consequently, to compare two terms, it is only needed to compare their $\overset{*}{<}$-left-weighted terms canonically associated.

2.5. Recursive lexicographic ordering

We give a simpler ordering which is based on the lexicographic ordering and which coincides with the recursive path ordering on the $\overset{*}{<}$-left-weighted terms.

DEFINITION 5: *The recursive lexicographic ordering is given as follows:*

$$s=f(s_l, \ldots, s_m) \overset{\alpha}{<} g(t_l, \ldots, t_n)=t \qquad \textit{iff}$$

$f=g$ and $\langle s_l, \ldots, s_m \rangle \overset{\alpha}{<}_{\text{lex}} \langle t_l, \ldots, t_n \rangle$,

or $f<g$ and $s_l \overset{\alpha}{<} t$,

or $f>g$ and $s \overset{\alpha}{<} t_l$ or $s=t_l$,

where $\overset{\alpha}{<}_{\text{lex}}$ *is the lexicographic ordering deduced from* $\overset{\alpha}{<}$.

LEMMA 8: *If* $<$ *is total on F, the recursive lexicographic ordering and the recursive path ordering coincide on the* $\overset{*}{<}$*-left-weighted terms, therefore the* $\overset{*}{<}$*-left-weighted terms are also the* $\overset{\alpha}{<}$*-left-weighted terms.*

Proof: By induction on the structure of terms. The case $f=g$ comes from Lemma 2. For $f<g$, we have $s_1 \overset{\alpha}{<} t$ if and only if by induction $s_1 \overset{*}{<} t$; but for all i, $s_i \overset{*}{\leqq} s_1$, because the terms s_i are $\overset{*}{<}$-left-weighted, so $s_1 \overset{\alpha}{<} t$ is equivalent to $s_i \overset{*}{<} t$, for all i. Since F is totally ordered, $\neg f \leqq g$ means $f>g$ and the result follows, by induction, from the equivalence between $s \overset{\alpha}{\leqq} t_1$ and $s \overset{*}{\leqq} t_1$ and so there exists $a\,j\,(j=1)$ with $s \overset{*}{\leqq} t_j$. ■

3. THE RECURSIVE PATH ORDERING AND THE DECOMPOSITION ORDERING ON MONADIC TERMS

F is a well-founded set and F^* is the set of words on F which can be seen as terms on a set F of symbols of monadic functions. A decomposition ordering can be defined on F^*. It introduces the decomposition ordering on terms.

DEFINITION 6: The *recursive path ordering on words* [16] is given by:

$$\mu \overset{*}{<} \nu \qquad \text{iff}$$

$\mu=\varepsilon$ and $\nu \neq \varepsilon$ (where ε is the empty word),

or $\mu=a\alpha$ and $\nu=b\beta$ and,

$a=b$ and $\alpha \overset{*}{<} \beta$,

or $a<b$ and $\alpha \overset{*}{<} \nu$,

or $\neg\, a \leqq b$ and $\mu \overset{*}{<} \beta$ or $\mu=\beta$.

This definition is valid even if F is partially ordered, but the recursive path ordering on words is total if F is totally ordered; that will be supposed in the sequel of this paper. Supposing $a<b<c$, we have ([3]):

$$abbb \overset{*}{<} abacaaa \overset{*}{<} abacab \overset{*}{<} bbcab.$$

In the case of words or linear terms it would be of no interest to distinguish between lexicographic and recursive path ordering.

([3]) For example, $abbb \overset{*}{<} abacab$ if $bbb \overset{*}{<} bacab$ i.e., if $bb \overset{*}{<} acab$ i.e., if $bb \overset{*}{\leqq} cab$ i.e., if $b \overset{*}{\leqq} cab$ i.e., if $\varepsilon \overset{*}{\leqq} cab$.

DEFINITION 7: $[a;\ \alpha';\ \alpha''] \in F \times F^* \times F^*$ is called a *decomposition* of $\alpha \in F^+$ if $\alpha = \alpha''\, a\, \alpha'$, $b > a \Rightarrow b \notin \alpha$ (i. e. a is a maximal element of α) and $b \in \alpha'' \Rightarrow \neg\, b \geqq a$.

If $<$ is total we will say "the" decomposition of α. Then a is the first occurrence of the greatest symbol in α.

DEFINITION 8: The *decomposition ordering* $\overset{\delta}{<}$ is defined by:

$$\alpha \overset{\delta}{<} \beta \qquad \text{iff}$$

$\alpha = \varepsilon$ and $\beta \neq \varepsilon$,

or $\alpha \neq \varepsilon$ and the decomposition of α (resp. β) is $[a;\ \alpha';\ \alpha'']$ (resp. $[b;\ \beta';\ \beta'']$) and,

$a < b$,

or $a = b$ and $\alpha' \overset{\delta}{<} \beta'$,

or $a = b$ and $\alpha' = \beta'$ and $\alpha'' \overset{\delta}{<} \beta''$.

The decompositions of the terms of the above example are respectively $[b;\ bb;\ a]$, $[c;\ aaa;\ aba]$, $[c;\ ab;\ aba]$, $[c;\ ab;\ bb]$ and the sorting of these words is easy; it is the same one as above.

THEOREM 1: *$\overset{\delta}{<}$ is a well-founded ordering, if $<$ is well-founded on F.*

Proof: Let us define $\alpha \succ \beta$ as: $\alpha \succ \beta$ iff $\max(\alpha) > \max(\beta)$ or $\max(\alpha) = \max(\beta) = a$ and $\operatorname{occ}(a, \alpha) > \operatorname{occ}(a, \beta)$, where $\operatorname{occ}(x, \alpha)$ is the number of occurrences of x in α.

CLAIM 1: *$\succ$ is well-founded on F^* if $>$ is well-founded on F, therefore a noetherian induction on $\succ$ is possible.*

CLAIM 2: *If $[a, \alpha', \alpha'']$ is the decomposition of α then $\alpha \succ \alpha'$ and $\alpha \succ \alpha''$.*

The induction hypothesis $IH(\beta)$ is: "There exists *no* $\overset{\delta}{<}$-decreasing sequence starting at β." Suppose now that:

(A) $(\forall \beta \prec \alpha)\ IH(\beta)$;

(B) $\neg\, IH(\alpha)$ i. e., there exists a $\overset{\delta}{<}$-decreasing sequence $\alpha_0 \overset{\delta}{>} \alpha_1 \overset{\delta}{>} \ldots \overset{\delta}{>} \alpha_n \overset{\delta}{>}, \ldots$, where $\alpha = \alpha_0$.

Three cases have to be considered:

1. There exists α_i such that $\max(\alpha_0) \neq \max(\alpha_i)$, then $\max(\alpha_0) > \max(\alpha_i)$, then $\alpha_0 \succ \alpha_i$ and α_i is the beginning of a $\overset{\delta}{<}$-decreasing sequence which contradicts (A).

2. For all n, $\max(\alpha_n) = \max(\alpha) = a$ and there exists an infinite sequence $i_1, \ldots, i_j, \ldots$, such that $\alpha'_{i_1} \overset{\delta}{>} \alpha'_{i_2} \overset{\delta}{>} \ldots \overset{\delta}{>} \alpha'_{i_j} \overset{\delta}{>} \alpha'_{i_{j+1}} \overset{\delta}{>} \ldots$ where $i_l = 0$ and $[a, \alpha'_{i_j}, \alpha''_{i_j}]$ is the decomposition of α_{i_j}. Because $\alpha'_{i_l} \prec \alpha$, this contradicts (A).

3. For all n, $\max(\alpha_n) = \max(\alpha) = a$ and there exists k such that $i \geqq k \Rightarrow \alpha'_i = \alpha'_k$, then there exists an infinite $\overset{\delta}{<}$-decreasing sequence $\alpha''_k \overset{\delta}{>} \alpha''_{k+1} \overset{\delta}{>} \alpha''_{k+2} \overset{\delta}{>} \ldots$ and because α''_k does not contain any occurrence of a then $\alpha''_k \prec \alpha$, this also contradicts (A). ■

LEMMA 9 (Monotonicity Lemma): *If* $\beta \overset{\delta}{<} \gamma$ *then* $a\beta \overset{\delta}{<} a\gamma$.

Proof: Let $\mu = a\beta$ with decomposition $[m; \mu'; \mu'']$ and $\nu = a\gamma$ with decomposition $[n; \nu'; \nu'']$. The decomposition of μ is related to the one of β and two cases are possible:

(1) $a < b$, where the decomposition of β by $[b; \beta'; \beta'']$, then $m = b$, $\mu' = \beta'$ and $\mu'' = a\beta''$;

(2) $a \geqq b$ then $m = a$, $\mu' = \beta$ and $\mu'' = \varepsilon$.

The same properties hold between ν and γ, where the decomposition of γ is $[c; \gamma', \gamma'']$.

By hypothesis $b \leqq c$, then three cases have to be considered.

(1) $a < b \leqq c$ if $b = c$ and $\beta' = \gamma'$ then $\beta'' \overset{\delta}{<} \gamma''$ (by hypothesis), therefore $m = n$ and $\mu' = \nu'$ and $\mu'' \overset{\delta}{<} \nu''$ (by induction); if $b < c$ or $b = c$ and $\beta' \overset{\delta}{<} \gamma'$, it is straightforward.

(2) $b \leqq a < c$ then $a = m < n = c$ and $\mu \overset{\delta}{<} \nu$.

(3) $b \leqq c \leqq a$ then $\mu \overset{\delta}{<} \nu$ is equivalent to $\beta \overset{\delta}{<} \gamma$. ■

LEMMA 10 (Subterm Lemma): $\beta \overset{\delta}{<} a\beta$.

Proof: This proof is similar to that of the Monotonicity Lemma and will be omitted. ■

THEOREM 2: *Decomposition ordering and recursive path ordering are equivalent on monadic terms.*

Proof: Let be $\mu \overset{*}{<} \nu$ with decomposition as in the Monotonicity Lemma. If $\mu = \varepsilon$ then $\nu \neq \varepsilon$ and $\mu \overset{*}{<} \nu$ is equivalent to $\mu \overset{\delta}{<} \nu$. If $\mu = d\beta$ and $\nu = c\gamma$, let us suppose by induction that if ζ is a word extracted from μ and η a word extracted from ν then $\zeta \overset{*}{<} \eta$ is equivalent to $\zeta \overset{\delta}{<} \eta$ and on the other hand $\beta \overset{*}{<} \nu$ is equivalent to $\beta \overset{\delta}{<} \nu$ and $\mu \overset{*}{<} \gamma$ is equivalent to $\mu \overset{\delta}{<} \gamma$. Three cases have to be considered.

(1) $d = e$. It is straightforward from the Monotonicity Lemma.

(2) $d < e$. Then $\mu \overset{*}{<} \nu$ is equivalent (by Definition 6.2) to $\beta \overset{*}{<} \nu$ and $\beta \overset{*}{<} \nu$ is equivalent (by the induction hypothesis) to $\beta \overset{\delta}{<} \nu$. Let the decomposition of $\beta = [b;\ \beta';\ \beta'']$:

– if $b \leqq d$ then $m = d$ and $m < e \leqq n$ thus $\beta \overset{\delta}{<} \nu$ is equivalent to $\mu \overset{\delta}{<} \nu$;
– if $d < b$ then $m = b$ and $\mu' = \beta'$ and $\mu'' = d\beta''$;

- if $b = n$ and $\beta' = \nu'$ and $\beta'' \overset{\delta}{<} \nu''$ which is $\beta'' \overset{*}{<} \nu''$, by induction, we have $\varepsilon \neq \nu'' = e\gamma''$ and, since $d < e$, $\mu'' \overset{*}{<} \nu''$ *i.e.*, $\mu'' \overset{\delta}{<} \nu''$ by induction hypothesis, therefore $\beta \overset{\delta}{<} \nu$ is equivalent to $\mu \overset{\delta}{<} \nu$;

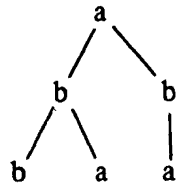
- if $b = m < n$ or $m = n$ and $\mu' = \beta' \overset{\delta}{<} \nu'$ the result is obvious.

(3) $e < d$. Then $\mu \overset{*}{<} \nu$ is equivalent, by Definition 6.3, to $\mu \overset{*}{<} \gamma$, which is equivalent, by the induction hypothesis, to $\mu \overset{\delta}{<} \gamma$, which is equivalent, by the Subterm Lemma and by transitivity, to $\mu \overset{\delta}{<} \nu$. ■

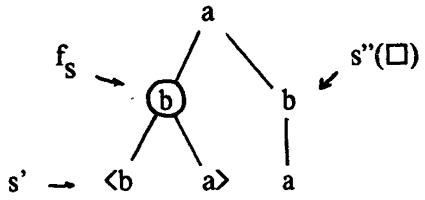
4. THE DECOMPOSITION ORDERING ON POLYADIC TERMS AND THE MAIN THEOREMS

In this section, we give a definition of the recursive path ordering, on left-weighted terms similar to the decomposition ordering on the words. Here, the *decomposition* of a left-weighted term s is the unique triple $[f_s;\ \mathbf{s}';\ s''(\square)]$ where f_s is the first occurrence of the maximal operator in s on its most left branch, $\mathbf{s}' = \langle s'_1, \ldots, s'_p \rangle$ is a sequence of terms and $s''(\square)$ is a term with a unique constant subterm denoted by the symbol $\square$; the set of such terms will be written by $T(F;\ \square)$. $s = s''(f_s(s'_1, \ldots, s'_p))$ where $s''(t)$ is the terms given by

substituting t in place of $\square$. The following term:



has the decomposition $[b; \langle b, a \rangle; a(\square, b(a))]$, described by the following figure.



This decomposition looks like the decomposition of a word, the one difference is that the second component is replaced by a sequence. It will be supposed that the term $\square$ is less than any other term. Then it must be noted that $s''(\square)$ is not $\overset{*}{<}$-left-weighted. Except the first immediate subterm, the term is $\overset{*}{<}$-left-weighted and so on recursively for each first subterm of this term. Such terms will be called almost $\overset{*}{<}$-left-weighted.

DEFINITION 9: A term s of $T(F; \square)$ is *almost* $\overset{*}{<}$*-left-weighted* iff $s = \square$ or $s = f(s_l, \ldots, s_m)$:

s_i is $\overset{*}{<}$-left-weighted for all $i \geqq 2$,

and s_l is almost $\overset{*}{<}$-left-weighted,

and for all $i, j, l < i < j \leqq m$ implies $s_i \overset{*}{\leqq} t_j$.

Note that a term is almost $\overset{*}{<}$-left-weighted if it is $\overset{*}{<}$-left-weighted; the definition of the lexicographic ordering can be easily extended to the almost $\overset{*}{<}$-left-weighted terms on $T(F; \square)$. The decomposition ordering will be given for almost $\overset{*}{<}$-left-weighted terms of $T(F; \square)$ too.

DEFINITION 10: The *decomposition ordering* on almost-$\overset{*}{<}$-left-weighted terms is given by:

$$s \overset{\delta}{<} t \qquad \text{iff}$$

$s = \square$ and $t \neq \square$,

or $f_s < f_t$,

or $f_s = f_t$ and $\langle s'_1, \ldots, s'_p \rangle \overset{\delta}{<}_{\text{lex}} \langle t'_1, \ldots, t'_q \rangle$,

or $f_s = f_t$ and $\langle s'_1, \ldots, s'_p \rangle = \langle t'_1, \ldots, t'_q \rangle$ and $s''(\square) \overset{\delta}{<} t''(\square)$.

THEOREM 3: *Decomposition ordering* $\overset{\delta}{<}$ *is a well-founded ordering on* $T(F; \square)$, *if* $<$ *is a well-ordering on* F.

Proof: As for Theorem 1, where $\text{occ}(f, s)$ counts the number of occurrence of f on the leftmost branch of s. ■

Before we prove the decomposition ordering is actually the recursive path ordering, we will prove some lemmas showing that the decomposition ordering works well on the structure of terms.

LEMMA 11 (Monotonicity Lemma): *If* $\langle s_1, \ldots, s_m \rangle \overset{\delta}{<}_{\text{lex}} \langle t_1, \ldots, t_n \rangle$ then $f(s_1, \ldots, s_m) \overset{\delta}{<} f(t_1, \ldots, t_n)$.

Proof: Similar to that of Lemma 9. Let us note $u = s_1$ and $v = t_1$, whose decompositions are $[f_u; \mathbf{u}'; u''(\square)]$ and $[f_v; \mathbf{v}'; v''(\square)]$. As in the case of words the decompositions of $s = f(s_l, \ldots, s_m)$ and u are related:

- if $f < f_u$ then $f_s = f_u$, $\mathbf{s}' = \mathbf{u}'$ and $s''(\square) = f(u'(\square), s_2, \ldots, s_n)$;
- if $f_u \leqq f$ then $f_s = f$, $\mathbf{s}' = \langle s_1, \ldots, s_n \rangle$ and $s''(\square) = \square$.

Here four cases have to be considered:

(1) $u = v$ then

$$\langle s_2, \ldots, s_m \rangle \overset{\delta}{<}_{\text{lex}} \langle t_2, \ldots, t_n \rangle$$

and

$$\langle u''(\square), s_2, \ldots, s_m \rangle \overset{\delta}{<} \langle v''(\square), t_2, \ldots, t_n \rangle$$

and by induction

$$s''(\square) = f(u''(\square), s_2, \ldots, s_m) \overset{\delta}{<} f(v''(\square), t_2, \ldots, t_n) = t''(\square).$$

On the other hand $f_u = f_s = f_t = f_v$ and $\mathbf{u}' = \mathbf{s}' = \mathbf{t}' = \mathbf{v}'$, and therefore $s \overset{\delta}{<} t$.

(2) $u \overset{\delta}{<} v$ and $f < f_u \leqq f_v$ if $f_u = f_v$ and $\mathbf{u}' = \mathbf{v}'$ then $u''(\square) \overset{\delta}{<} v''(\square)$ thus $f_s = f_t$, $\mathbf{s}' = \mathbf{t}'$ and $f(u''(\square), s_2, \ldots, s_m) \overset{\delta}{<} f(v''(\square), t_2, \ldots, t_n)$ (by induction) and $s \overset{\delta}{<} t$; if $f_u < f_v$ or $f_u = f_v$ and $\mathbf{u}' \overset{\delta}{<}{}^{\square}_{\mathrm{lex}} \mathbf{v}'$ it is obvious.

(3) $u \overset{\delta}{<} v$ and $f_u \leqq f < f_v$ then $f = f_s < f_t = f_u$ and $s \overset{\delta}{<} t$.

(4) $u \overset{\delta}{<} v$ and $f_u \leqq f_v \leqq f$ then $f_s = f = f_t$ and $\mathbf{s} \overset{\delta}{<}_{\mathrm{lex}} \mathbf{t}$ is equivalent to $s \overset{\delta}{<} t$. ■

LEMMA 12 (Subterm Lemma):

– (*Weak form*) $s_1 \overset{\delta}{<} f(s_1, \ldots, s_m)$;

– (*Strong form*) *for all* i, $0 \leqq i \leqq m$ $s_i \overset{\delta}{<} f(s_1, \ldots, s_i, \ldots, s_m)$.

Proof: (Weak form) Easy by induction. (Strong form) Consequence of the Subterm Lemma for $\overset{\alpha}{<}$ and Theorem 4 (it will not be used to prove Theorem 4). ■

THEOREM 4: *Decomposition ordering and recursive lexicographic ordering coincide on* $T(F; \square)$.

Proof: If $s = \square$, obviously $s \overset{\alpha}{<} t$ is equivalent to $s \overset{\delta}{<} t$. If $s = f(s_1, \ldots, s_m) = f(\mathbf{s})$ and $t = g(t_1, \ldots, t_n) = f(\mathbf{t})$, then as in the Monotonicity Lemma, let $u = s_1$ and $v = t_1$. By induction, assume $\mathbf{s} \overset{\alpha}{<}_{\mathrm{lex}} \mathbf{t}$ is equivalent to $\mathbf{s} \overset{\delta}{<}_{\mathrm{lex}} \mathbf{t}$, $u \overset{\alpha}{<} t$ is equivalent to $u \overset{\delta}{<} t$ and $s \overset{\alpha}{<} v$ is equivalent to $s \overset{\delta}{<} v$. Three cases have to be considered.

(1) $f = g$. Straightforward from the Monotonicity Lemma.

(2) $f < g$. Then, by Definition 5, $s \overset{\alpha}{<} t$ is equivalent to $u \overset{\alpha}{<} t$ and by induction $u \overset{\alpha}{<} t$ is equivalent to $u \overset{\delta}{<} t$;

– if $f_u \leqq f$ then $f_s = f$ and $f < g \leqq f_t$ then $u \overset{\delta}{<} t$ is equivalent to $s \overset{\delta}{<} t$;

– if $f < f_u$ then $f_s = f_u$ and $\mathbf{s}' = \mathbf{u}'$ and $s''(\square) = f(u''(\square), s_2, \ldots, s_m)$;

• if $f_u = f_t$ and $\mathbf{u}' = \mathbf{t}'$ and $u''(\square) \overset{\delta}{<} t''(\square)$ then $u''(\square) \overset{\alpha}{<} t''(\square)$, by induction, therefore, by Definition 5, $s''(\square) \overset{\alpha}{<} t''(\square)$ then, by induction, $s''(\square) \overset{\delta}{<} t''(\square)$ therefore $s \overset{\delta}{<} t$. The converse comes from the weak form of the Subterm Lemma;

- if $f_u < f_t$ or $f_u = f_t$ and $\mathbf{u}' \overset{\delta}{<}_{\text{lex}} \mathbf{t}'$, the result is obvious.

(3) $g < f$. Then $s \overset{\alpha}{<} t$ is equivalent to $s \overset{\alpha}{\leqq} v$ which is equivalent, by induction, to $s \overset{\delta}{\leqq} v$, itself equivalent to $s \overset{\delta}{<} t$ by the weak form of the Subterm Lemma and transitivity.

The proof of the induction hypothesis for $\overset{\alpha}{<}_{\text{lex}}$ and $\overset{\delta}{<}_{\text{lex}}$ presents no difficulty and will be omitted. ■

THEOREM 5: *If $<$ is well-founded on F the recursive path ordering is well-founded on $T(F)$.*

Proof: From Lemma 8 $\overset{\delta}{<}$ is $\overset{\alpha}{<}$ on $\overset{*}{<}$-left-weighted terms, i. e., $\overset{\alpha}{<}$-left-weighted terms. Therefore, by Theorem 4, $\overset{*}{<}$ is $\overset{\alpha}{<}$ on $\overset{*}{<}$-left-weighted terms, but by Lemmas 6 and 7 and Theorem 3, this means $\overset{*}{<}$ is well-founded on $T(F)$. ■

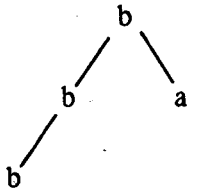
It is interesting to compare the simple arguments used in this proof with those used by Dershowitz [1, 2] which are inspired from Nash-William's proof of Kruskal's theorem [9, 14].

5. CONCLUSION

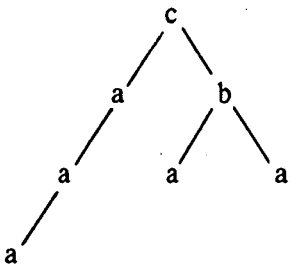
Decomposition ordering allows us to compare two terms by efficient algorithms. In this way, it would be interesting to look at the examples of Section 3 again. The words we have to compare are *abbb*, *abacaaa*, *abacab*, *bbcab*. The decompositions [*b*; *bb*; *a*] and [*c*; *aaa*; *aba*] allow us to answer which of *abbb* and *abacaa* is the greater word in one comparison. To compare *abacab* and *bbcab*, it is necessary to refine this decomposition in what we call *generalized decomposition* by replacing each word in the decomposition by its generalized decomposition. The generalized decomposition of a word will be more easily represented by a tournament i. e., a labeled binary tree where the labels are ordered. The tournament associated with the word α where α has the decomposition $[a; \alpha'; \alpha'']$, has a root labeled by the maximal letter a, its left subtree is the generalized decomposition of the part α' and its right subtree is the generalized decomposition of the part α''. In [3], Françon, Viennot and Vuillemin give a way to associate a binary tournament to a permutation represented by a word. Here, the method is quite similar. But it will be noted that the ordering between the labels is as follows: the label of each node is greater than or equal to that of its left son, the label of each node is strictly

greater than that of its right son. So the generalized decompositions are:

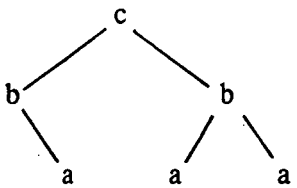
generalized decomposition of *abbb:*



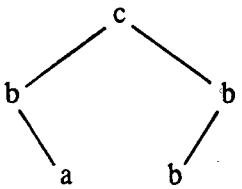
generalized decomposition of *abacaaa*:



generalized decomposition of *abacab*:



generalized decomposition of *bbcab*:



To compare two words it is sufficient to traverse the two trees in parallel in prefix order until we find the first difference. The number of comparisons of letters will be less than 2 min × (length (α), length (β)) + 1, and often much smaller, as given by the following array:

abbb	9			
abacaaa	1	15		
abacab	1	2	13	
bbcab	1	2	8	11
	abbb	*abacaaa*	*abacab*	*bbcab*

The following array gives the number of comparisons of letters to perform the comparison of the same words using the recursive path ordering. This number is greater than min (length (α), length (β)) + 1.

abbb	5			
abacaa	6	7		
abacab	6	7	7	
abbb	5	8	8	6
	abbb	*abacaaa*	*abacab*	*bbcab*

Some other results on these two algorithms can be found in [10]. [11] suggests an implementation for polyadic terms based on the same ideas.

On another hand, Fernand Reinig proposed a generalization of decomposition ordering to terms with variables on a partially ordered set of functional symbols; that gives an ordering more general than the recursive path ordering [17, 19].

A first version of the decomposition ordering took shape during conversations with Jean-Luc Remy and Fernand Reinig [12]. Then Nachum Dershowitz suggested a good improvement which gave a nicer definition. I thank them. I am grateful also to Jean-Pierre Jouannaud, Marc Shapiro and Jeannette Wing for their help at several steps of this work.

REFERENCES

1. N. DERSHOWITZ, *Ordering for Term Rewriting Systems*, Proc. 20th Symposium on Foundations of Computer Science, 1979, pp. 123-131, also in Theoritical Computer Science, Vol. 17, 1982, pp. 279-301.
2. N. DERSHOWITZ, *A note on Simplification Orderings*, Inform. Proc. Ltrs., Vol. 9, 1979, pp. 212-215.
3. J. FRANÇON, G. VIENNOT and J. VUILLEMIN, *Description and Analysis of an Efficient Priority Queues Representation*, Proc. 19th Symposium of Foundations of Computer Science, 1978, pp. 1-7.
4. G. HUET, *A Complete Proof of Correctness of the Knuth-Bendix Completion Algorithm*, Rapport INRIA 25, 1980.
5. G. HUET and J. HULLOT, *Proof by Induction in Equational Theories with Constructors*, Proc. 21th Symposium on Foundations of Computer Science, 1980.
6. G. HUET and D. C. OPPEN, *Equations and Rewrite Rules: a Survey*, in *Formal Languages perspectives and Open Problems*, R. BOOK, Ed., Academic Press, 1980.
7. D. E. KNUTH, *The Art of Computer Programming*. Vol. 1: *Fundamental Algorithms*, Addison Wesley, Reading, Mass., 1968.
8. D. E. KNUTH and P. BENDIX, *Simple Word Problems in Universal Algebra*, in *Computational Problems in Abstract Algebra*, J. LEECH, Ed., Pergamon Press, 1970, pp. 263-297.
9. J. B. KRUSKAL, *Well-Quasi-Ordering, the Tree Theorem, and Vazsonyi's Conjecture*, Trans. Amer. Math. Soc., Vol. 95, 1960, pp. 210-225.
10. P. LESCANNE, *Two Implementations of the Recursive Path Ordering on Monadic Terms*, 19th Annual Allerton Conf. on Communication, Control, and Computing, Allerton House, Monticello, Illinois, 1981.
11. P. LESCANNE, *Decomposition Ordering as a Tool to prove the Termination of Rewriting Systems*, 7th IJCAI, Vancouver, Canada, 1981, pp. 548-550.
12. P. LESCANNE and F. REINIG, *A Well-Founded Recursively Defined Ordering on First Order Terms*, Centre de Recherche en Informatique de Nancy, France, CRIN 80-R-005.
13. D. L. MUSSER, *On Proving Inductive Properties of Abstract Data Types*, Proc. 7th ACM Symposium on Principles of Programming Languages, 1980.
14. C. St. J. A. NASH-WILLIAM, *On Well-Quasi-Ordering on Finite Trees*, Proc. Cambridge Philos. Soc., Vol. 60, 1964, pp. 833-835.
15. D. PLAISTED, *Well-Founded Orderings for Proving Termination of Systems of Rewrite Rules*, Dept of Computer Science Report 78-932, U. of Illinois at Urbana-Champaign, July 1978.
16. D. PLAISTED, *A Recursively Defined Ordering for Proving Termination of Term Rewriting Systems*, Dept of Computer Science Report 78-943, U. of Illinois at Urbana-Champaign, Sept. 1978.
17. F. REINIG, *Les ordres de décomposition : un outil incrémental pour prouver la terminaison finie de systèmes de réécriture de termes*, Thèse, Université de Nancy, 1981.

18. J.-P. JOUANNAUD and P. LESCANNE, *On Multiset Orderings*, in Inform. Proc. Ltrs., Vol. 15, 1982, pp. 57-63.

19. J.-P. JOUANNAUD, P. LESCANNE and F. REINIG, *Recursive Decomposition Ordering*, IFIP Conf. on Formal Description of Programming Concepts, Garmisch-Partenkirchen, Germany, 1982.