J. Symbolic Computation (1985) **1**, 329–349

Termination Orderings for Associative-commutative Rewriting Systems†

LEO BACHMAIR

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

AND

DAVID A. PLAISTED

Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, North Carolina 27514, *U.S.A.*

(*Received* 21 *December* 1984)

In this paper we describe a new class of orderings—*associative path orderings*—for proving *termination* of *associative-commutative term rewriting systems*. These orderings are based on the concept of *simplification orderings* and extend the well-known *recursive path orderings* to *E*-congruence classes, where E is an equational theory consisting of associativity and commutativity axioms. Associative path orderings are applicable to term rewriting systems for which a *precedence ordering* on the set of operator symbols can be defined that satisfies a certain condition, the *associative path condition*. The precedence ordering can often be derived from the structure of the reduction rules. We include termination proofs for various term rewriting systems (for rings, boolean algebra, etc.) and, in addition, point out ways to handle situations where the associative path condition is too restrictive.

1. Introduction

Term rewriting systems provide a simple mechanism for computing with equations. The Knuth–Bendix completion method and its extensions to equational term rewriting systems have been applied to a wide variety of problems, including the word problem in universal algebra (Knuth & Bendix, 1970), algebraic specification of data types (Guttag *et al.*, 1978), theorem proving in first order logic (Hsiang, 1985), and computing with rewrite programs (Dershowitz, 1985*b*). Termination of the reduction relation corresponding to a given equational term rewriting system (R, E) is a prerequisite for these completion methods. For the classical case, i.e. $E = \emptyset$, various powerful techniques for proving termination have been developed, among them the *recursive path ordering*, the *recursive decomposition ordering* and other *simplification orderings*. These termination orderings have in common that they extend a *precedence ordering* on the set of operator symbols to the set of terms. They are not directly applicable to the more general problem of proving termination of a reduction relation modulo a non-empty set of equations E. The only methods suitable for the general E-termination problem are polynomial interpretations, which are sometimes applicable to sets of equations E that consist of

† This research was supported in part by the National Science Foundation under grant MCS 83–07755.

0747–7171/85/040329+21 \$03.00/0

associativity and commutativity axioms. The problem with polynomial orderings is to find appropriate polynomials, a task which can be quite tiresome.

In this paper we introduce a new class of simplification orderings, called *associative path orderings*, that are applicable to the problem of proving termination of a reduction relation modulo a set of associativity and commutativity axioms. Associative path orderings are based on the idea of transforming terms and generalise recursive path orderings. The transform that is used can be described by a reduction relation. It is simple enough to allow efficient implementations, yet is applicable to a variety of theories.

Termination orderings based on transforms were first proposed in Dershowitz *et al.* (1983). The transform used in Dershowitz *et al.* (1983) is rather complex, however, and it was not possible to lift the corresponding ordering to non-ground terms. Our ordering is conceptually simpler, more efficient, and, most important, can be lifted to non-ground term $t=f(t_1, \ldots, t_n)$ is f. The *size* $|t|$ of a term t is the number of operator symbols and variables it contains. For example, the term $t=f(x, f(x, y))$ has size $|t|=5$. To enhance readability we will often use infix notation. For instance, we may write $0+I(x+y)$ instead of $+(0, I(+(x, y)))$.

In sections 2 and 3 we give a brief overview of term rewriting systems and termination orderings. In sections 4 and 5 we discuss E-termination, in particular for the case of associative-communitative theories and adapt general results on termination to the problem of E-termination. In sections 6 and 7 we describe associative path orderings. Termination proofs for various term rewriting systems (for abelian groups, rings, boolean algebra, etc.) are included in section 8.

2. Definitions

We assume that the reader is familiar with the basic concepts concerning reduction relations and term rewriting systems. We briefly summarise the most important definitions below and refer to Huet & Oppen (1980), Huet (1980) and Jouannaud & Kirchner (1984) for more details.

A *signature* or *arity function* is a function $a: F \rightarrow \mathbf{N}$, where $\mathbf{N}$ is the set of non-negative integers. Elements of F are called *function* or *operator symbols* and denoted $f, g, h, \cdots$. An operator symbol f of arity $a(f)=0$ is called a *constant*. Let V be a (denumerable) set disjoint from F. Elements of V are called *variables* and denoted $x, y, z, \cdots$. The set $T(F, V)$ of *terms* over F and V is defined as being the smallest set such that $f(t_1, \ldots, t_n) \in T(F, V)$ if $a(f)=n$ and $t_1, \cdots, t_n \in T(F, V)$ (we implicitly assume that $a(x)=0$ for all variables x). We will abbreviate $T(F, V)$ by T and denote terms by $s, t, u, v, w \ldots$. The set of all *ground terms*, i.e. terms containing no variables, is denoted by G.

For any term t, $V(t)$ denotes the set of variables occurring in t. A term in which no variable appears twice is called a *linear* term. The (top-level) *operator symbol* $op(t)$ of a term $t=f(t_1, \ldots, t_n)$ is f. The *size* $|t|$ of a term t is the number of operator symbols and variables it contains. For example, the term $t=f(x, f(x, y))$ has size $|t|=5$. To enhance readability we will often use infix notation. For instance, we may write $0+I(x+y)$ instead of $+(0, I(+(x, y)))$.

A *substitution* is a homomorphism σ from T into itself, i.e. $\sigma(f(t_1, \ldots, t_n)) = f(\sigma(t_1), \ldots, \sigma(t_n))$, for all terms $f(t_1, \ldots, t_n)$. A substitution σ is totally determined by specifying its values for all variables. The set $D(\sigma)=\{x: \sigma(x) \neq x\}$ is called the *domain* of σ. We will only consider substitutions with a finite domain. A substitution is called *ground* if $\sigma(x)$ is a ground term, for all x in the domain of σ. Let p denote the position of a subterm of s. Then $s[p \leftarrow t]$ denotes the term that results from *replacing* the subterm of s at position p by t (positions may be represented as sequences of integers, see, for instance, Huet (1980) for a formal treatment).

A binary relation is called a (strict) *partial ordering* if it is transitive and irreflexive. A partial ordering $>$ is *well-founded* if there is no infinite chain $t_1, t_2, t_3, \cdots$ such that $t_1 > t_2 > t_3 > \cdots$. A relation $>$ on T is *monotonic* if $s > t$ implies $f(\cdots s \cdots) > f(\cdots t \cdots)$, for all terms $s, t, f(\cdots s \cdots)$ and $f(\cdots t \cdots)$ in T.

An *equational theory* is a set $E \subset T \times T$. Elements (s, t) in E are called *equations* and written $s = t$. Groups, for example, are an equational theory:

$$\{0 + x = x,\ I(x) + x = 0,\ (x + y) + z = x + (y + z)\}.$$

We define $\mid\!\mid_E$ to be the smallest symmetric relation that contains E and is closed under monotonicity and substitution. The reflexive-transitive closure of $\mid\!\mid_E$ is denoted by $\sim_E$. A relation $>$ is *compatible* with E if $s \sim_E s' > t' \sim_E t$ implies $s > t$.

A *term rewriting system* R over F is any set $\{(l_i, r_i) : l_i, r_i \in T,\ V(r_i) \subset V(l_i)\}$. The pairs (l_i, r_i) are called *rewriting* or *reduction rules* and written $l_i \rightarrow r_i$. If all the left-hand sides l_i are linear, then R is called a *left-linear* rewriting system. The *reduction relation* $\rightarrow_R$ on T is defined as being the smallest relation containing R that is closed under monotonicity and substitution. We say that s *reduces* to t (in one step) if $s \rightarrow_R t$. Occasionally we will abbreviate $\rightarrow_R$ by $\rightarrow$ and write $t \leftarrow_R s$ instead of $s \rightarrow_R t$. The symbols $\rightarrow^+$, $\rightarrow^*$ and $\leftrightarrow$ denote the transitive, transitive-reflexive, and symmetric closure of $\rightarrow$, respectively.

An *equational term rewriting system* is a tuple (R, E), also written R/E, where R is a term rewriting system and E an equational theory. The reduction relation $\rightarrow_{R/E}$ is defined by $\sim_E \circ \rightarrow_R \circ \sim_E$, where $\circ$ denotes composition of relations.

Let R be a term rewriting system and E be an equational theory. A term rewriting system R is *E-terminating* if there is no infinite sequence $t_1, t_2, t_3, \cdots$ such that $t_1 \rightarrow_{R/E} t_2 \rightarrow_{R/E} t_3 \rightarrow_{R/E} \cdots$. Note that R is E-terminating if and only if $\rightarrow^+_{R/E}$ is a well-founded partial ordering on T. We say that R is *terminating* or *noetherian* if it is E-terminating for $E = \emptyset$. A term t is *R-reducible* if there is a term t' such that $t \rightarrow_R t'$. Otherwise, t is said to be *R-irreducible* or in *R-normal form*. If t' is irreducible and $t \rightarrow^*_R t'$, then t' is called an *R-normal form of* t, written $t \rightarrow^!_R t'$. We will omit the prefix R if it is clear from the context.

R is *Church–Rosser* modulo E if, for all terms s and t, $s \sim_{R \cup E} t$ implies that there exists a term u such that $s \rightarrow^*_{R/E} u \leftarrow^*_{R/E} t$.

We say that two terms s and t *converge modulo E*, written $s \downarrow_{R,E} t$, if there exist terms s' and t' such that $s \rightarrow^*_R s' \sim_E t' \leftarrow^*_R t$. If $E = \emptyset$ we say that s and t *converge* and write $s \downarrow_R t$. R is called *confluent modulo E* if whenever $s \leftarrow^*_R u \rightarrow^*_R t$, then $s \downarrow_{R,E} t$. R is *locally confluent modulo E* if whenever $s \leftarrow_R u \rightarrow_R t$, then $s \downarrow_{R,E} t$. R is (locally) confluent if it is (locally) confluent modulo E for $E = \emptyset$. Note that confluence of R/E and confluence modulo E of R are two different conditions. R/E is confluent if and only if R is Church–Rosser modulo E.

R is called *coherent modulo E* if $s \leftarrow^*_R u \sim_E t$ implies $s \downarrow_{R,E} t$. R is *locally coherent modulo E* if $s' \leftarrow_R s \mid\!\mid_E t'$ implies $s' \downarrow_{R,E} t'$. Coherence is vacuously true if E is empty.

The following theorem shows how these properties are connected.

THEOREM 1 (Jouannaud & Kirchner, 1984). *Let (R, E) be an equational term rewriting system. If R is E-terminating then the following conditions are equivalent:*

(*a*) *R is Church–Rosser modulo E.*

(*b*) *R is confluent modulo E and coherent modulo E.*
(*c*) *R is locally confluent modulo E and locally coherent modulo E.*
(*d*) *For all terms s, s', t and t',* $s' \leftarrow^!_R s \sim_{R \cup E} t \rightarrow^!_R t'$ *implies* $s' \sim_E t'$.

Confluence is decidable for finite noetherian term rewriting systems. A decision procedure based on *critical pairs* was given by Knuth & Bendix (1970). Let $l \rightarrow r$ and $l' \rightarrow r'$ be two reduction rules and p be a position in l such that l/p is not a variable and l/p and l' are *unifiable*, and let θ be the most general unifier of l/p and l'. The *superposition* of $l \rightarrow r$ onto $l' \rightarrow r'$ determines a critical pair $(s, t) = (\theta(r), \theta(l)[p \leftarrow \theta(r')])$. The terms s and t are the two outcomes of reducing $\theta(l)$ by $l \rightarrow r$ and $l' \rightarrow r'$, respectively. A critical pair (s, t) converges if s and t converge, otherwise it diverges. The following lemma holds:

LEMMA 1 (Knuth & Bendix, 1970; see also Huet, 1980). *Let R be a noetherian term rewriting system. R is confluent if and only if no critical pair* (s, t) *between rules in R diverges.*

Knuth & Bendix (1970) partially solved the problem of constructing, for a given non-confluent term rewriting system, an equivalent confluent one, by designing a so-called *completion method*. The basic idea of completion is, to add, for every diverging critical pair (s, t), a new reduction rule $s' \rightarrow t'$ or $t' \rightarrow s'$, where s' and t' are normal forms of s and t, respectively. The method is not guaranteed to succeed and need not terminate. It may also abort when s' and t' are incomparable and cannot be made into a directed rewrite rule without violating termination. Both the critical pair lemma and the completion method have been generalised to rewriting modulo equations. In the general case additional critical pairs between rules in R and equations in E have to be considered. These results are discussed in detail in Jouannaud & Kirchner (1984).

Termination is a prerequisite for the completion method. Techniques for proving E-termination are therefore of great importance in practice. Although various powerful methods have been developed for proving termination when $E = \emptyset$, very little is known about the more general problem of E-termination for non-empty equational theories E. In this paper we address the latter problem. Since our approach is based on and extends techniques for proving ordinary termination ($E = \emptyset$), we give a brief review of such methods in the next section.

3. Termination Orderings

Proving termination of term rewriting systems is a non-trivial task, in general, and the uniform halting problem for term rewriting systems is undecidable, even though it is decidable for ground systems, i.e. systems where the reduction rules consist of ground terms only (see Huet & Lankford, 1978). A survey of various techniques for proving termination of term rewriting systems is given in Dershowitz (1985*a*). We will briefly describe several important techniques in this section. We will assume from now on that F, the set of operator symbols, is finite, and G, the set of ground terms, is non-empty.

Let R be a term rewriting system over T. First note that R is noetherian if and only if there is no ground term t having an infinite sequence of reductions. In other words, R is noetherian if and only if $\rightarrow^+_R$ defines a well-founded partial ordering on G. Therefore R is noetherian, if we can find a well-founded partial ordering $>$ on G such that, for all ground terms s and t, $s \rightarrow_R t$ implies $s > t$. The problem with this method is that it quantifies over all possible reductions (on ground terms) $s \rightarrow_R t$. An important refinement is based on the concept of simplification orderings, which was introduced by Dershowitz (1979).

DEFINITION 1. A partial ordering $>$ on G is called a *simplification ordering* if it is monotonic and has the following *subterm property:* $f(\cdots t \cdots) > t$, for all terms t and $f(\cdots t \cdots)$. We say that a term $s = f(s_1, \cdots, s_m)$ is *homeomorphically embedded* in a term $t = g(t_1, \cdots, t_n)$, written $s \preceq t$, if and only if

(a) $f = g$ and, for some permutation π, $s_i \preceq t_{\pi(i)}$, *for all* i, $1 \leqslant i \leqslant m$, or
(b) $s \preceq t_j$ for some j, $1 \leqslant j \leqslant n$.

For example, $I(x)+0 \preceq 0+(I(x+y)+I(0+z))$. The following lemma holds:

LEMMA 2. *Let $>$ be a simplification ordering on G and let s and t be ground terms. Then $s \preceq t$ implies $s \not> t$.*

Non-termination may be characterised as follows:

THEOREM 2 (Kruskal, 1960). *In any infinite sequence of ground terms $t_1, t_2, \cdots$ over a finite set of function symbols there exist i and j, $i < j$, such that $t_i \preceq t_j$.*

The following termination theorem is based on Kruskal's Theorem:

THEOREM 3 (Dershowitz, 1979). *A term rewriting system R terminates if there exists a simplification ordering $>$ on G such that $\sigma(l) > \sigma(r)$, for all reduction rules $l \to r$ in R and for every ground substitution σ with $V(l) \subset D(\sigma)$.*

PROOF. Let $>$ be a simplification ordering satisfying the stated requirements. It can easily be proved that $>$ contains $\to_R^+$. Suppose that R does not terminate. Then there is an infinite sequence of ground terms $t_1 \to_R t_2 \to_R t_3 \cdots$. By Theorem 2 there exist i and j, $i < j$, such that $t_i \preceq t_j$, and therefore, by Lemma 1, $t_i \not> t_j$. But since $>$ contains $\to_R^+$ we may also infer $t_i > t_j$, which is a contradiction.

An important class of simplification orderings are *recursive path orderings* (see Dershowitz, 1982), which extend a given partial ordering $>$, also called a *precedence ordering*, on the set of operator symbols F to the set of ground terms G.

DEFINITION 2. Let $>$ be a partial ordering on F. The *recursive path ordering* $>_{rpo}$ corresponding to $>$ is defined recursively as follows:

$$s = f(s_1, \cdots, s_m) >_{rpo} t = g(t_1, \cdots, t_n)$$

if and only if

(a) $f = g$ and $\{s_1, \cdots, s_m\} \gg_{rpo} \{t_1, \cdots, t_n\}$, or
(b) $f > g$ and $s >_{rpo} t_i$, for all i, $1 \leqslant i \leqslant n$, or
(c) $f \not> g$ and $s_i \gtrsim_{rpo} t$, for some i, $1 \leqslant i \leqslant m$.

Here $\gg_{rpo}$ is the extension of $>_{rpo}$ to multisets and $s \gtrsim_{rpo} t$ means $s >_{rpo} t$ or $s \sim t$, where the *permutation equivalence* $\sim$ is defined by

$$s = f(s_1, \ldots, s_n) \sim t = f(t_1, \cdots, t_n)$$

if and only if for some permutation π, $s_i \sim t_{\pi(i)}$, for all i, $1 \leqslant i \leqslant n$.

A multiset is an unordered collection of elements, where elements may appear more than once. For any partial ordering $>$ on a set S, the multiset ordering $\gg$ on the set of all finite multisets over S, defined by

$$M \gg N \text{ if and only if, } M \neq N \text{ and, for every } y \in N - M, \text{ there exists } x \in M - N, \text{ such that } x > y,$$

is also a partial ordering. If $M(x)$ and $N(x)$ denote the number of occurrences of x in M and N, respectively, then the number of occurrences $(M-N)(x)$ of x in $M-N$ is defined as $max(0, M(x)-N(x))$. The ordering $\gg$ is well founded if and only if $>$ is well founded. For details see Dershowitz & Manna (1979).

THEOREM 4 (Dershowitz, 1982). *Any recursive path ordering is a simplification ordering.*

The ordering, as given above, is defined on ground terms, but can be extended to non-ground terms by including the following clause:

$$s = f(s_1, \ldots, s_n) >_{rpo} x \text{ if and only if } x \in V(s).$$

This means that variables are treated as constants that are unrelated to other operator symbols in the precedence ordering $>$.

LEMMA 3 (Huet & Oppen, 1980). *Let s and t be terms in T. Then $s >_{rpo} t$ implies $\sigma(s) >_{rpo} \sigma(t)$, for every ground substitution σ.*

Simplification orderings are discussed in detail in Dershowitz (1979, 1982). Extensions of the recursive path ordering that incorporate the semantics of operators were proposed by Kamin & Levy (1980). Other simplification orderings include the "path of subterms" ordering (Plaisted, 1978*a*, *b*), the recursive decomposition ordering (Lescanne, 1981, 1982; Jouannaud *et al.*, 1983), and another path ordering similar to the (extended) path of subterms ordering (Kapur *et al.*, 1985). The recursive path ordering is used in the RRL rewrite rule laboratory, along with the lexicographic recursive path ordering by Kamin & Levy, see Kapur & Sivakumar (1983). It was also used in the first version of the REVE rewrite rule laboratory (Lescanne, 1983). In recent versions of REVE the recursive decomposition ordering is used as the main tool for proving termination.

All these orderings have in common that they extend a precedence ordering $>$ on function symbols to an ordering on terms. Dershowitz (1982) remarks that whenever $>$ is a total ordering then the recursive path ordering is essentially the path of subterms ordering, and whenever $>$ is a partial ordering, then the recursive path ordering is contained in an obvious extension of the path of subterms orderings. The recursive decomposition ordering also contains the recursive path ordering, but both orderings are the same when the precedence ordering is a total ordering. A detailed analysis of the relationship between these orderings may be found in Rusinowitch (1985).

Manna & Ness (1970) (see also Lankford, 1979) proposed a different technique that is based on the idea of associating with every n-ary operator f an n-ary function $\tau(f)$ over the natural numbers. Every function $\tau(f)$ has to be monotonically increasing in each argument, i.e. if $x > y$ then $\tau(f)(\cdots x \cdots) > \tau(f)(\cdots y \cdots)$. The function τ can be extended to terms in the obvious way. A term rewriting system R is noetherian if $\tau(\sigma(l)) > \tau(\sigma(r))$ for every rule $l \rightarrow r$ in R and every ground substitution σ whose domain contains all variables in l. Usually this method is applied with polynomials over the non-negative integers as interpretations for the operators in F.

4. Equational Term Rewriting Systems

A basic idea underlying the concept of rewriting systems is that of reducing a given term to a simpler one. An equation is converted into a directed rewriting rule in such a way that the right-hand side of the rule is "simpler" than the left-hand side. However, there are equations where left-hand side and right-hand side are intrinsically incomparable. For example, using the commutativity axiom $x+y=y+z$ as a directed rewriting rule results in a non-terminating term rewriting system. One way to handle theories that contain such equations is to partition the given equational theory in two sets R and E, where E is used as a set of equations and R as a set of directed rewriting rules. The reduction relation $\rightarrow_{R/E}$ corresponding to the equational term rewriting system R/E allows reductions modulo the equations in E. The classical Knuth–Bendix completion procedure has first been extended to rewriting modulo equations by Lankford & Ballantyne (1977), for the case of sets of equations that consist of permutativity axioms, for instance, associativity and commutativity. A different approach by Huet (1980) deals with left-linear rewriting systems. The method by Peterson & Stickel (1981) may be applied to linear theories E with a finite and complete unification algorithm. These approaches have been unified in a more general framework by Jouannaud & Kirchner (1984).

E-termination is a prerequisite for all these methods. The termination orderings described in the previous section are not directly applicable to the problem of proving E-termination, however. Polynomial interpretations may be used for proving termination of associative-commutative rewriting systems, provided the proposed interpretation is compatible with the given associativity and commutativity axioms. The class of polynomials that satisfy these conditions consists of all polynomials $p(x, y)$ of the form $axy+b(x+y)+c$, where $ac+b=b^2$. For example, the polynomials $p(x, y)\equiv x+y+c$ and $p(x, y)\equiv cxy$ are associative and commutative in x and y, but $p(x, y)\equiv xy+1$ is not. In the following sections we introduce a new class of orderings that are also applicable to associative-commutative rewriting systems, but provide a conceptually simpler alternative to polynomial interpretations. Before we formally describe the ordering we adapt the general results on termination from the previous section to the case of equational term rewriting systems.

THEOREM 5. *An equational term rewriting system R/E terminates if there is a simplification ordering $>$ on G such that $s\rightarrow_{R/E}t$ implies $s>t$, for all ground terms s and t.*

The proof of the theorem is similar to the proof of Theorem 3. The theorem quantifies over all possible reductions $s\rightarrow_{R/E}t$ on ground terms. This requirement can be refined. If $>$ is compatible with E, then $>$ contains $\rightarrow^+_{R/E}$ if and only if it contains $\rightarrow^+_R$. To see this assume that $>$ contains $\rightarrow^+_R$, i.e. $u\rightarrow_R v$ implies $u>v$, for all terms u and v. Let s, s', t and t' be such that $s\sim_E s'\rightarrow_R t'\sim_E t$. Then $s'>t'$ and therefore, since $>$ is compatible with E, also $s>t$. Thus $>$ contains $\rightarrow^+_{R/E}$. This suggests the following theorem:

THEOREM 6. *An equational term rewriting system R/E terminates if there exists a simplification ordering $>$ on G that is compatible with E, such that $\sigma(l)>\sigma(r)$, for every rule $l\rightarrow r$ of R and every ground substitution σ with $V(l)\subset D(\sigma)$.*

PROOF. Let $>$ be a simplification ordering satisfying the stated requirement. By Theorem

5 and the remarks above it suffices to show that $s \rightarrow_R t$ implies $s > t$, for all ground terms s and t. But this can be proved easily under the given assumptions.

Our approach to the problem of E-termination is based on this theorem. A different approach, based on the following theorem, has been suggested by Jouannaud & Munoz (1984).

THEOREM 7 (Munoz, 1983). *Let R/E be an equational term rewriting system and let $\rightarrow$ be a reduction relation with $R \subset \rightarrow \subset R/E$. If $\rightarrow$ terminates and is E-commuting then R is E-terminating.*

A reduction relation $\rightarrow$ is *E-commuting* if and only if for all terms s, s' and t with $s' \sim_E s \rightarrow^+ t$, there exists a term t' such that $s' \rightarrow^+ t' \sim_E t$. If a reduction relation is not E-commuting it can sometimes be extended to an E-commuting relation by adding appropriate rules to R. This process is similar to the completion process for term rewriting systems and need not always terminate. The termination property of the extended (E-commuting) set of rules can be proved using existing methods for proving termination of $\rightarrow$. Candidates for $\rightarrow$ are $\rightarrow_R$ and $\rightarrow_{R,E}$. The reduction relation $\rightarrow_{R,E}$ is defined as follows: $s \rightarrow_{R,E} t$ if and only if there exist an occurrence p of s, a rule $l \rightarrow r$ in R, and a substitution σ, such that s/p is not a variable, $s/p \sim_E \sigma(l)$ and $t = s[p \leftarrow \sigma(r)]$. Note that $\rightarrow_{R,E}$ is different from $\rightarrow_{R/E}$, since $\rightarrow_{R,E}$ allows E-equality to be applied only to the subterm at occurrence p, whereas $\rightarrow_{R/E}$ allows unrestricted E-equality steps. Consequently it should be simpler to prove termination of $\rightarrow_{R,E}$ than of $\rightarrow_{R/E}$. However, at the present time no specific orderings are known for proving termination of $\rightarrow_{R,E}$. Of course, if E is an associative-commutative theory, the associative path orderings we will describe below can be used for that purpose.

5. Associative-commutative Term Rewriting Systems

We will now give a formal definition of associative-commutative rewriting systems and introduce a formalism for representing associative-commutative terms. Let f be some operator symbol. A *commutativity axiom* for f is an equation of the form $f(x, y) = f(y, x)$, an *associativity axiom* for f is an equation of the form $f(x, f(y, z)) = f(f(x, y), z)$ or $f(f(x, y), z) = f(x, f(y, z))$. An equational theory E is called an *associative-commutative (AC) theory* if every equation in E is either an associativity or commutativity axiom. An equational term rewriting system (R, E) is called an *AC rewriting system* if E is an AC theory. Let E be an AC theory. $F_C(F_A)$ denotes the set of all operators f such that E contains a commutativity (associativity) axiom for f. Elements of F_C and F_A are called *commutative* and *associative* operator symbols, respectively. $F_{AC} = F_A \cap F_C$ is called the set of *associative-commutative* (AC) operators. A term $f(s, t)$, where f is an AC operator, is called an *AC term*.

We will use varyadic terms for representing AC terms. Given a set of operator symbols F, a *varyadic signature* over F is a function $a: F \rightarrow 2^{\mathbf{N}}$, where $2^{\mathbf{N}}$ is the set of all subsets of $\mathbf{N}$. This basically means that we allow operators with variable arity. For example, if $a(f) = \{2, 3\}$ then both $f(a, b)$ and $f(a, b, c)$ are syntactically correct terms. More precisely, the set $TV(F, V)$ of *varyadic terms* is defined as being the smallest set such that $f(t_1, \ldots, t_n)$ is in $TV(F, V)$ if $f \in F$, $n \in a(f)$ and $t_1, \ldots, t_n \in TV(F, V)$. (We assume that $a(x) = \{0\}$ for every variable x.) Let a be an (ordinary) signature over some set of function symbols F and let E be an AC theory over T. The varyadic signature a' corresponding to

a is defined as follows: $a'(f) := \{a(f)\}$ if $f \notin F_A$, and $a(f) := \mathbf{N} - \{0\}$, otherwise. That is, associative operators may take any positive number of arguments, whereas non-associative operators have fixed arity. Associative path orderings are based on transforms that map terms over an ordinary signature a into terms over the corresponding varyadic signature a'. The results on recursive path orderings from section 3 can easily be extended to varyadic terms, see Dershowitz (1982).

AC terms are conveniently represented as *flattened* terms, i.e. terms that have no nested occurrences of AC operators and where the order of subterms does not matter. We will use a *multiset notation* to denote such terms (see also Plaisted, 1983). That is, we will use the expression $f(\tilde{N})$, where N is the multiset $\{t_1, \ldots, t_n\}$, to denote a term $f(t_1, \ldots, t_n)$. Let now Fl be the set of all reduction rules (on varyadic terms) of the form $f(\tilde{X}, f(\tilde{Y})) \rightarrow f(\tilde{Z})$, where f is an AC operator, X and Y are disjoint sets of distinct variables, and $Z = X \cup Y$. (A similar notation, using vectors instead of multisets, can be used for operators that are associative but not commutative.) The rewriting system Fl is terminating because the number of AC operators occurring in a term decreases with each application of a flattening rule. Furthermore, all critical pairs (s, t) between rules in Fl are convergent modulo the permutation equivalence $\sim$, and Fl therefore is confluent modulo $\sim$. The Fl-normal form of a term t, also called the *flattened version* of t, is denoted by $\bar{t}$. An Fl-irreducible term t is also called *flattened*. Whenever two terms are equal under associativity and commutativity then their flattened versions are equivalent under permutation, that is, $s \sim_E t$ implies $\bar{s} \sim \bar{t}$, for any AC theory E. If N is a multiset of terms $\{t_1, \ldots, t_n\}$, then $\bar{N}$ denotes the multiset $\{\bar{t_1}, \ldots, \bar{t_n}\}$. Also, given multisets $N_1, \ldots, N_k$, we will write $f(N_1, \ldots, N_k)$ to denote the multiset $\{f(t_1, \ldots, t_k) : t_i \in N_i, \text{ for } 1 \leqslant i \leqslant k\}$.

EXAMPLE 1. Let f be an associative operator and t be the term $f(a, f(b, f(c, d)))$. Then $\bar{t} = f(a, b, c, d)$ and may also be denoted by $f(\tilde{N})$, where $N = \{a, b, c, d\}$. The multiset expression $f(\{a, b\}, \{c, d\})$ denotes the multiset $\{f(a, c), f(b, c), f(a, d), f(b, d)\}$. Frequently, we will delete brackets of singleton multisets and write, for example, $f(\{a, b\}, c)$ instead of $f(\{a, b\}, \{c\})$. Suppose that $M = g(N)$ is a multiset of flattened terms, then flattening $f(f(a, b), f(\tilde{M}))$ yields $f(\tilde{P})$, where $P = \{a, b\} \cup g(N)$.

6. Associative Path Orderings

In this section we will outline how to use the recursive path ordering in combination with a transform for termination proofs of AC rewriting systems. Let (R, AC) be an associative-commutative rewriting system. Note that the case of commutative operators is already handled by the recursive path ordering, since two terms equivalent under commutativity are equivalent under permutation and $s \sim s' >_{rpo} t' \sim t$ implies $s >_{rpo} t$. But the recursive path ordering cannot handle associative operators. For example, if $a > b > c$ then $f(a, f(b, c)) \sim_{AC} f(f(a, b), c) >_{rpo} f(a, f(b, c))$. Therefore the recursive path ordering is not compatible with AC and not well-founded for R/AC-reductions, in general. Since $s \sim_{AC} t$ implies that $\bar{s}$ and $\bar{t}$ are permutationally equivalent, we might try to apply the recursive path ordering to flattened terms. Let A be the transform given by $A(t) = \bar{t}$. We define the *associative path ordering* $>_{rpo}$ by: $s >_{apo} t$ if and only if $A(s) >_{rpo} A(t)$. But this ordering is not monotonic, as the following counterexample illustrates. Let f and g be AC operators with $f > g$, and let a be a constant. Then $f(a, a) >_{apo} g(a, a)$, but $f(a, f(a, a)) >_{apo} f(a, g(a, a))$ is false. In fact, we even have $f(a, g(a, a)) >_{rpo} f(a, a, a)$

$=\overline{f(a, f(a, a))}$, and therefore $f(a, g(a, a)) >_{apo} f(a, f(a, a))$. Thus the monotonic extension of $>_{apo}$ is not well founded.

The example above indicates that monotonicity is violated for those terms in which an AC operator is embedded within a larger AC operator. We will design a transform that maps T into a subset $T' \subset T$ that contains no such "critical" terms. For example, we will transform the term $f(a, g(a, a))$ to $g(f(a, a), f(a, a))$. This transformation process can be conveniently characterised using a rewriting system. In order to simplify the presentation we will assume from now on that every associative operator in F is commutative, and every commutative operator is associative, that is, $F_{AC} = F_A = F_C$. Our results can easily be adapted to the case where F also contains functions that are associative but not commutative.

Let $>$ be a precedence ordering on the set of operator symbols F. The set of *distributivity* rules D corresponding to $>$ consists of all reduction rules of the form $f(g(x, y), z) \to g(f(x, z), f(y, z))$ or $f(x, g(y, z)) \to g(f(x, y), f(x, z))$, where f and g are AC operators with $f > g$. In order to be able to compute with D, we have to prove that D/AC is terminating and confluent. Then D-normal forms exist and are unique up to associativity and commutativity. Confluence is not satisfied for sets of distributivity rules, in general, but we can enforce it by properly restricting the precedence ordering $>$. We say that $>$ satisfies the *associative path condition* if for all AC operators f either

(a) f is minimal in F, or
(b) there exists an AC operator g such that f is minimal in $F - \{g\}$.

For example, let f and g be AC operators. Then the precedence ordering $>$, defined by $f > g$ and $h > g$, where h may be any operator, satisfies the associative path condition, whereas a precedence ordering with $f > g$ and $f > h$ does not (see Figs 1 and 2). The ordering $>$, defined by $h > f > g$, only satisfies the condition if h is not an AC operator (see Fig. 3).

LEMMA 4. *Let $>$ be a precedence ordering that satisfies the associative path condition, and D be the corresponding set of distributivity rules. Then D/AC is confluent and terminating.*

PROOF. AC-termination of D can easily be proved by using the following polynomial interpretation:

$\tau(f) = \lambda x\ y\,.\,x \times y$, if f is a non-minimal AC operator,

$\tau(g) = \lambda x\ y\,.\,x + y + 1$, if g is a minimal AC operator, and

$\tau(x) = 2$, if x is a variable.

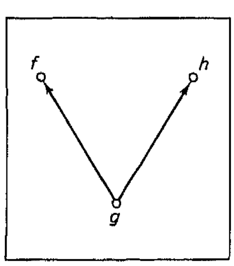

Fig. 1

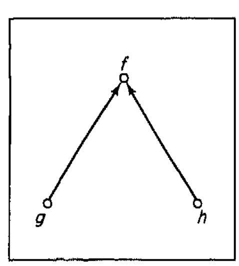

Fig. 2

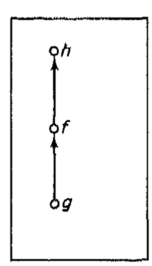

Fig. 3

Given that D is AC-terminating, by Theorem 1, D/AC is confluent if it is locally confluent modulo AC and locally coherent modulo AC. To prove local confluence modulo AC and local coherence modulo AC we use the following results by Huet (1980) for left-linear rewriting systems: a rewriting system R is locally confluent modulo E if, for every critical pair (s, t) between rules in R, $s{\downarrow_{R,E}}t$; it is locally coherent modulo E if, for every critical pair (s, t) between a rule in R and an equation in E, $s{\downarrow_{R,E}}t$.

We first prove local confluence modulo AC. The only non-trivial critical pairs result from superposing the rules

$$f(g(x, y), z)) \to g(f(x, z), f(y, z))$$

and

$$f(x, g(y, z)) \to g(f(x, y), f(x, z)).$$

The corresponding critical pair is (s, t), where

$$s = g(f(x, g(x', y')), f(y, g(x', y')))$$

and

$$t = g(f(g(x, y), x'), f(g(x, y), y')).$$

We may reduce s and t as follows:

$$s \to_D g(g(f(x, x'), f(x, y')), g(f(y, x'), f(y, y'))) = s'$$

and

$$t \to_D g(g(f(x, x'), f(y, x')), g(f(x, y'), f(y, y'))) = t'.$$

Obviously, $s' \sim_{AC} t'$, therefore $s{\downarrow_{D,AC}}t$.

For local coherence we have to consider critical pairs between rules in D and equations in AC. A typical critical pair results from superposing

$$f(x, g(y, z)) \to g(f(x, y), f(x, z))$$

and

$$f(f(x, y), z) = f(x, f(y, z)).$$

The corresponding critical pair is (s, t), where

$$s = g(f(f(x, y), x'), f(f(x, y), y'))$$

and

$$t = f(x, f(y, g(x', y'))).$$

We may reduce t as follows:

$$t \to_D f(x, g(f(y, x'), f(y, y'))) \to_D g(f(x, f(y, x')), f(x, f(y, y'))) = t'.$$

We see that $s \sim_{AC} t'$, therefore $s{\downarrow_{D,AC}}t$. Other critical pairs are handled similarly.

We may define now the transform A by $A(t) = \overline{t'}$, where t' is a D-normal form of t. We will define associative path orderings in a slightly more general way than we indicated above. Namely, if two terms have the same transform we may still compare them using some given partial ordering $\rhd$ that is monotonic, compatible with AC, and well founded on every set $[t] = \{s \mid A(s) \sim A(t)\}$. We call such a partial ordering *admissible for the transform A*.

Definition 3. Let $>$ be a precedence ordering on F that satisfies the associative path condition and $\rhd$ be a partial ordering that is admissible for the transform A associated with $>$. The corresponding *associative path ordering* $>_{apo}$ on G is defined by

$$s >_{apo} t \text{ if and only if } A(s) >_{rpo} A(t) \text{ or } A(s) \sim A(t) \text{ and } s \rhd t.$$

Example 2. Let f and g be AC-operators with $f > g$, and let s and t be the terms

$f(a, f(a, a))$ and $f(a, g(a, a))$, respectively. Then

$$A(s) = f(a, a, a) >_{rpo} g(f(a, a), f(a, a)) = A(t),$$

and therefore $s >_{apo} t$.

Comparing terms by the inverse of their sizes is an admissible ordering for our transform. More precisely, let $\rhd$ be defined by

$s \rhd t$ if and only if $|s| \prec |t|$ and no variable appears more often in s than in t.

(Here $\prec$ is the natural order on integers.) The ordering $\rhd$ is monotonic and compatible with AC. Furthermore, there is a bound on the size of the terms in $[t]$, i.e. there is a constant k such that $s \in [t]$ implies $|s| \prec k$. As a consequence, there can be no infinite sequence $t_1 \rhd t_2 \rhd \cdots$ with $A(t_1) \sim A(t_2) \sim \cdots$. The ordering $\rhd$ is therefore admissible. For example, let s and t be the terms $f(a, g(a, a))$ and $g(f(a, a), f(a, a))$, respectively. Then $A(s) = A(t)$, but $s \rhd t$, and therefore $s >_{apo} t$.

The use of distributivity rules for transforming terms was proposed by Plaisted (1983). The termination orderings in Dershowitz *et al.* (1983) are also based on the idea of transforming terms. Here terms are transformed into multisets of terms and no requirement is imposed on the precedence ordering $>$. Our ordering is conceptually simpler and more efficient, since we transform terms into terms. But the advantage of our ordering lies not only in the improved efficiency (at the cost of imposing restrictions on the precedence ordering), more important, our ordering can be lifted to non-ground terms, which was not possible for the ordering in Dershowitz *et al.* (1983). This is of course crucial for practical applications of the Knuth–Bendix completion procedure.

We will sketch now another way to characterise the transform A. Again let $>$ be a precedence ordering that satisfies the associative path condition. Let DV be the set of all reduction rules (on varyadic terms) of the form $f(\tilde{X}, g(\tilde{Y})) \to g(\tilde{N})$, where f and g are AC operators with $f > g$, $X = \{x_1, \ldots, x_n\}$ and Y are disjoint sets of distinct variables, and $N = f(x_1, \ldots, x_n, Y)$. An instance of a rule in DV is, for example,

$$f(x, y, g(z, z')) \to g(f(x, y, z), f(x, y, z')).$$

The rewriting system DV contains all rules in D and the reduction rules in DV may be viewed as "generalised distributivity rules". A term t is DV-irreducible if it contains no subterm $\tilde{f}(N)$, where f is an AC operator and $f > op(u)$ for some term u in N.

The combined system $DV \cup Fl$ is terminating, since for every rule $l \to r$ we have $l >_{rpo} r$, where $>_{rpo}$ is the recursive path ordering corresponding to $>$. The rewriting system $DV \cup Fl$ is also confluent (modulo the permutation equivalence $\sim$), as can be seen by checking all possible critical pairs. (This can be done in a way similar to the proof of Lemma 4). The following lemma can easily be proved:

LEMMA 5. *Let t be a term in T and t^* its $(DV \cup Fl)$-normal form. Then $A(t) \sim t^*$.*

Lemma 5 allows us to extend the transform A to varyadic terms by defining $A(t) = t^*$, where t^* is a $(DV \cup Fl)$-normal form of t.

We will state now our main results on associative path orderings. We assume from now on that $>$ is a precedence ordering that satisfies the associative path condition, and that $\rhd$ is an admissible ordering for the transform A.

LEMMA 6. *Any associative path ordering is compatible with AC.*

PROOF. Follows from the fact that $s \sim_{AC} t$ implies $\overline{s'} \sim \overline{t'}$, where s' and t' are D-normal forms of s and t, respectively.

THEOREM 8. *Any associative path ordering is a simplification ordering.*

The proof of the theorem is given below. We combine Theorems 6 and 8 and Lemma 6 to get the following termination theorem:

THEOREM 9. *A term rewriting system R is AC-terminating if there exists an associative path ordering $>_{apo}$ on G such that $\sigma(l) >_{apo} \sigma(r)$, for each rule $l \to r$ of R-D and every ground substitution σ whose domain contains all variables in l.*

The following lemmata are needed for the proof of Theorem 8.

LEMMA 7. *Let $s, t, f(\cdots s \cdots)$ and $f(\cdots t \cdots)$ be DV-irreducible terms such that $s >_{apo} t$. Then $f(\cdots s \cdots) >_{apo} f(\cdots t \cdots)$.*

PROOF. Suppose $s >_{apo} t$, that is $\bar{s} >_{rpo} \bar{t}$. Let u and v be such that $u = f(\cdots s \cdots)$ and $v = f(\cdots t \cdots)$. We have to prove $\bar{u} >_{rpo} \bar{v}$. The only problematic case is when s has top-level AC operator symbol f and the operator symbol h of t is different from f. We observe that in this case $f \not> h$. This is vacuously true if f is a minimal operator. It is also true if f is minimal in $F - \{g\}$, because if t has top-level operator g then v would be DV-reducible, which would contradict our assumption. But $f \not> h$ and $\bar{s} = f(s_1, \ldots, s_n) >_{rpo} \bar{t}$ imply that there is a subterm s_i of $\bar{s}$ such that $s_i >_{rpo} \bar{t}$. Therefore

$$\bar{u} = f(\cdots s_1, \cdots, s_i, \cdots, s_n \cdots) >_{rpo} f(\cdots t \cdots) = \bar{v}.$$

As an immediate corollary we get:

COROLLARY 1. *The associative path ordering restricted to D-irreducible terms is monotonic.*

The next lemma characterises transforms of certain terms.

LEMMA 8. *Let t_1 and t_2 be D-irreducible terms and f be an AC operator. Then the term $t = f(t_1, t_2)$ is either D-irreducible or $A(t) = g(\tilde{N})$, where g is an AC operator with $f > g$, $N = \overline{f(T_1, T_2)}$, and T_i $(i = 1, 2)$ is such that $T_i = \{\overline{t_i}\}$, if $f \not> op(t_i)$, and $\overline{t_i} = \tilde{g}(T_i)$, if $f > op(t_i)$.*

PROOF. Suppose t is reducible. Then t_1 or t_2 has top-level operator g, where $f > g$. Without loss of generality we may assume that $op(t_1) = g$. Let T_1 be such that $\overline{t_1} = g(\tilde{T}_1)$. Now,

$$s = f(t_1, t_2) \to^*_{FI} f(\overline{t_1}, \overline{t_2}) = f(g(\tilde{T}_1), \overline{t_2}) \to^+_{DV} g(\tilde{M}),$$

where $M = f(T_1, T_2)$ and T_2 is such that $T_2 = \{\overline{t_2}\}$, if $op(t_2) \neq g$, and $\overline{t_2} = g(\tilde{T}_2)$, if $op(t_2) = g$. Finally, $g(\tilde{M}) \to^!_{DV \cup FI} g(\tilde{N}) = A(s)$, where $N = \overline{f(T_1, T_2)}$. This proves our assertion.

EXAMPLE 3. Let s be $f(t_1, t_2)$, where $t_1 = g(g(a, a), a)$, $t_2 = b$, and $f > g$. Then $T_1 = \{a, a, a\}$,

$T_2 = \{b\}$, and s has D-normal form $u = g(g(f(a, b), f(a, b)), f(a, b))$. We see that, in fact,

$$A(s) = g(f(a, b), f(a, b), f(a, b)) = g(\tilde{N}),$$

where

$$N = \overline{f(T_1, T_2)}.$$

We will next prove Theorem 8.

PROOF OF THEOREM 8. Let $>$ be a precedence ordering that satisfies the associative path condition and $\rhd$ be an admissible ordering. We have to show that the corresponding associative path ordering is a monotonic partial ordering and has the subterm property.

Irreflexivity and *transitivity* follow immediately from the corresponding properties for the recursive path ordering and the ordering $\rhd$.

Subterm property. We have to show $s = f(\cdots t \cdots) >_{apo} t$, for all terms s and t. We may assume, without loss of generality, that all top-level subterms of s, including t, are D-irreducible. If s is D-irreducible, then obviously $\bar{s} >_{rpo} \bar{t}$, and thus $s >_{apo} t$. Otherwise, s has to be of the form $f(u, t)$ (or $f(t, u)$), where f is a non-minimal AC operator. Let g be an AC operator such that $f > g$. By Lemma 8, $A(s) = g(\tilde{N})$, where $N = \overline{f(U, T)}$ and T is such that $\bar{t} = g(\tilde{T})$, if t has top-level operator g, and $T = \{\bar{t}\}$, otherwise. It can easily be seen that in either case $A(s) >_{rpo} \bar{t} = A(t)$, and therefore $s >_{apo} t$.

Monotonicity. We have to prove that $s >_{apo} t$ implies $f(\cdots s \cdots) >_{apo} f(\cdots t \cdots)$, for any operator symbol f and terms $s, t, u = f(\cdots s \cdots)$ and $v = f(\cdots t \cdots)$.

If $A(s) \sim A(t)$ and $s \rhd t$, then $A(u) \sim A(v)$ and the assertion follows from the monotonicity of $\rhd$. Let us therefore assume $A(s) >_{rpo} A(t)$, that is, $\bar{s} >_{rpo} \bar{t}$. We show $A(u) >_{rpo} A(v)$. Without loss of generality we may assume that all top-level subterms of u and v, including s and t, are D-irreducible.

If u and v are both D-irreducible then the assertion follows from Corollary 1. Otherwise u or v is reducible and f is a non-minimal AC operator. Let g be a minimal AC operator with $f > g$ and suppose that u is of the form $f(w, s)$ and v of the form $f(w, t)$.

We consider first the case that v only is reducible. Then w cannot have top-level operator symbol g because otherwise u would be reducible. By Lemma 8, $A(v) = g(\tilde{N})$, where $N = \overline{f(w, T)}$ and $\bar{t} = g(\tilde{T})$. From Lemma 7 and $\bar{s} >_{rpo} \bar{t}$ we may conclude that, for all terms t_i in T, $\overline{f(w, s)} >_{rpo} \overline{f(w, t_i)}$. This implies $\overline{f(w, s)} \gg_{rpo} \overline{f(w, T)} = N$ and, since $f > g$, also

$$A(u) = \bar{u} = \overline{f(w, s)} >_{rpo} g(\tilde{N}) = A(v).$$

Next suppose that u only is reducible. Then $A(u) = g(\tilde{M})$, where $M = \overline{f(w, S)}$ and $\bar{s} = \tilde{g}(S)$. Since g is a minimal operator and $\bar{s} >_{rpo} \bar{t}$, there must be some term s_i in S such that $s_i >_{rpo} \bar{t}$. Then, by Lemma 7, $\overline{f(w, s_i)} >_{rpo} \overline{f(w, t)}$, and therefore

$$A(u) = g(\tilde{M}) >_{rpo} \overline{f(w, s_i)} >_{rpo} \overline{f(w, t)} = A(v).$$

Finally, let us assume that both u and v are reducible. Then $A(u) = g(\tilde{M})$ and $A(v) = g(\tilde{N})$, where $M = \overline{f(W, S)}$, $N = \overline{f(W, T)}$, and the multisets S and T depend on s and t, respectively, as described in Lemma 8. It can easily be shown that $S \gg_{rpo} T$ holds. This implies $S' = \overline{f(W, S)} \gg_{rpo} \overline{f(W, T)} = T'$ and, as a further consequence, $A(u) >_{rpo} A(v)$. To prove $S' \gg_{rpo} T'$ we assume that $\overline{f(w_i, t_j)}$ is some element in $T' - S'$. Then t_j is contained in $T - S$, and therefore, since $S \gg_{rpo} T$, $S - T$ must contain an element s_k such that $s_k >_{rpo} t_j$. From Lemma 7 we obtain $\overline{f(w_i, s_k)} >_{rpo} \overline{f(w_i, t_j)}$, where $\overline{f(w_i, s_k)}$ is in $S' - T'$. In summary, $S' \gg T'$. This completes the proof of monotonicity and of Theorem 8.

7. Lifting the Ordering to Non-Ground Terms

We will next give a method for comparing terms with variables in the associative path ordering. When applying Theorem 9 we have to compare, for given terms s and t, $\sigma(s)$ and $\sigma(t)$, for every ground substitution σ whose domain contains all variables in s and t. We will generalise the associative path ordering in such a way that only a finite number of the potentially infinitely many ground substitutions σ have to be checked.

In the recursive path ordering terms that contain variables are compared by treating variables like constants. The equivalent of Lemma 3 does not hold for the associative path ordering, however. The corresponding method for associative path orderings is more complicated since the result of transforming a term may depend on the particular substitution and, in general, $A(\sigma(t)) \neq \sigma(A(t))$. For example, given a term $f(x, c)$, where c is a constant, we may substitute $g(c, c)$ for x to obtain $f(g(c, c), c)$, a term that can be reduced to $g(f(c, c), f(c, c))$ (assuming f and g are AC operators with $f > g$). However, if we substitute c for x we get the irreducible term $g(c, c)$. When comparing $g(x, c)$ with some other term we have to consider both cases. In general, we say that a variable x has a *critical occurrence* for a (minimal) AC operator g in a term s if s contains a subterm of the form $f(u, x)$ or $f(x, u)$, where f is an AC operator with $f > g$. Note that a variable may have critical occurrences with respect to more than one AC operator.

A substitution σ is called *normalised* for a rewriting system R (or R-normalised) if $\sigma(x)$ is R-irreducible, for all x. Let s and t be terms and σ be a substitution. The substitution σ_L *associated with* σ, s, *and* t is defined as follows:

$$\sigma_L(x) = g(x_1, x_2), \text{ if } op(\sigma(x)) = g \text{ and } x \text{ has a critical occurrence for } g \text{ in } s \text{ or } t,$$
$$\sigma_L(x) = x, \text{ otherwise.}$$

The following lemma shows why we are interested in associated substitutions.

LEMMA 9. *Let s and t be terms, σ a D-normalised ground substitution whose domain contains all variables in s and t, and σ_L its associated substitution. Then $A(\sigma_L(s)) >_{rpo} A(\sigma_L(t))$ implies $A(\sigma(s)) >_{rpo} A(\sigma(t))$.*

PROOF. Let u and v be D-normal forms of $\sigma_L(s)$ and $\sigma_L(t)$, respectively. Then $A(\sigma_L(s)) = \bar{u}$ and $A(\sigma_L(t)) = \bar{v}$. We have to show that $\bar{u} >_{rpo} \bar{v}$ implies $A(\sigma(s)) >_{rpo} A(\sigma(t))$. Let ρ be a substitution such that $\sigma = \sigma_L \circ \rho$. Then

$$\sigma(s) = \rho(\sigma_L(s)) \to_D^* \rho(u) \to_{Fl}^* \rho(\bar{u}),$$

and similarly

$$\sigma(t) = \rho(\sigma_L(t)) \to_D^* \rho(v) \to_{Fl}^* \rho(\bar{v}).$$

Let $\rho'(x) = \overline{\rho(x)}$ and let u' and v' be $(DV-Fl)$-normal forms of $\rho'(\bar{u})$ and $\rho'(\bar{v})$, respectively. Obviously, $A(\sigma(s)) = A(\rho(\bar{u})) = u'$ and $A(\sigma(t)) = A(\rho(\bar{v})) = v'$. Therefore, $A(\sigma(s)) >_{rpo} A(\sigma(t))$ if and only if $u' >_{rpo} v'$. In order to show that $\bar{u} >_{rpo} \bar{v}$ implies $u' >_{rpo} v'$ we first analyse the connection between $\bar{u}$ and u' and between $\bar{v}$ and v', respectively. Suppose $w = f(\tilde{M}, x)$ is some subterm of $\bar{u}$ or $\bar{v}$ and $\rho'(x) = g(\tilde{N})$, where $N = \{t_1, \ldots, t_n\}$.

(a) If f is an AC operator and $f = g$, then $\rho'(w) = f(\tilde{M}, f(\tilde{N})) \to_{Fl} f(\tilde{M}, \tilde{N})$.

(b) If f is an AC operator and $f > g$, then w must be embedded in $\bar{u}$ or $\bar{v}$ in a term $w' = g(\tilde{P}, w)$. Then

$$\rho(w') = g(\tilde{P}, f(\tilde{M}, g(\tilde{N}))) \to_{DV} g(\tilde{P}, g(f(\tilde{M}, t_1), \ldots, f(\tilde{M}, t_n)))$$
$$\to_{DV} g(\tilde{P}, f(\tilde{M}, t_1), \ldots, f(\tilde{M}, t_n)) = g(\tilde{P}, f(\tilde{M}, N)).$$

(c) In any other case, no reduction rule in DV or Fl is applicable to $\rho'(w)$ at the position of $\rho'(x)$.

To summarise, u' and v' may be obtained from $\bar{u}$ and $\bar{v}$, respectively, by replacing each occurrence of a variable x either by $\rho'(x)$ or, depending on the particular context, by $\tilde{N}$ or N, where $\rho'(x) = g(\tilde{N})$. Using these facts we may prove that $\bar{u} >_{rpo} \bar{v}$ implies $u' >_{rpo} v'$. This part of the proof is similar to the proof of Lemma 3 (induction on the size of terms) and is omitted here.

EXAMPLE 4. Let

$$s = f(x, a), \quad t = g(x, a) \quad \text{and} \quad \sigma(x) = g(b, g(c, c)).$$

Using the same notation as above,

$$\sigma_L(x) = g(x_1, x_2), \quad \rho(x_1) = b = \rho'(x_1), \quad \text{and} \quad \rho(x_2) = g(c, c) = \rho'(x_2).$$

Now,

$$A(\sigma_L(s)) = A(f(g(x_1, x_2), a)) = g(f(x_1, a), f(x_2, a))$$
$$>_{rpo} g(x_1, x_2, a) = A(g(g(x_1, x_2), a)) = A(\sigma_L(t))$$

and, in fact,

$$A(\sigma(s)) = A(f(g(b, g(c, c)), a)) = g(f(b, a), f(c, a), f(c, a))$$
$$>_{rpo} g(b, c, c, a) = A(g(g(b, g(c, c)), a)) = A(\sigma(t)).$$

We see that $g(f(b, a), f(c, a), f(c, a))$ may be obtained from $g(f(x_1, a), f(x_2, a))$ by replacing x_1 by $\{b\}$ and replacing x_2 by $\{c, c\}$.

Let $\Lambda(s, t)$ denote the set of all substitutions σ such that $\sigma(x) = x$ or $\sigma(x) = g(x_1, x_2)$, if x has a critical occurrence for g in s or t. Clearly, for any ground substitution σ, the substitution σ_L associated with σ, s, and t is contained in $\Lambda(s, t)$. We require that the ordering $\rhd$, which is used for comparing terms s and t with $A(s) \sim A(t)$, can be extended to non-ground terms and is *compatible with substitution:* whenever $A(s) \sim A(t)$ and $s \rhd t$ then $\sigma(s) \rhd \sigma(t)$, for all ground substitutions σ. We assume from now on that $\rhd$ is an admissible ordering that is compatible with substitution. The ordering given in Example 3 is compatible with substitution.

THEOREM 10. *An AC rewriting system R/AC terminates if there exists an associative path ordering $>_{apo}$ such that, for every rule $l \to r$ in R-D, either*

(a) *$A(l) \sim A(r)$ and $l \rhd r$, or*
(b) *$A(\sigma_L(l)) >_{apo} A(\sigma_L(r))$, for every substitution σ_L in $\Lambda(l, r)$.*

PROOF. Let R/AC be an AC rewriting system and let $>_{apo}$ be an associative path ordering that satisfies the given conditions. By Theorem 9, R/AC is terminating if $\sigma(l) >_{apo} \sigma(r)$ for every rule $l \to r$ in R but not in D and for every ground substitution σ. Let $l \to r$ be a rule in $R - D$ and σ be a ground substitution.

If $A(l) \sim A(r)$ then, by our assumption, $l \rhd r$, and therefore, since $\rhd$ is compatible with substitution, $\sigma(l) \rhd \sigma(r)$. This, together with $A(\sigma(l)) \sim A(\sigma(r))$, implies $\sigma(l) >_{apo} \sigma(r)$.

Suppose next that $A(l)$ and $A(r)$ are not permutationally equivalent. Let $\sigma'(x)$ be the D-normal form of $\sigma(x)$, and let σ_L be the substitution associated with σ', l and r. The substitution σ_L is contained in $\Lambda(l, r)$ and therefore, by our assumption, $A(\sigma_L(l)) >_{rpo} A(\sigma_L(r))$. By Lemma 9, $A(\sigma'(l)) >_{rpo} A(\sigma'(r))$, which implies $A(\sigma(l)) >_{rpo} A(\sigma(r))$.

EXAMPLE 5. Suppose s is $f(x, y)$ and t is $g(x, y)$, where $f > g$. Both x and y have critical occurrences for g in s. Therefore $\Lambda(s, t)$ contains the substitution σ with $\sigma(x) = g(x_1, x_2)$ and $\sigma(y) = g(y_1, y_2)$. Then

$$A(\sigma(s)) = g(f(x_1, y_1), f(x_1, y_2), f(x_2, y_1), f(x_2, y_2)),$$
$$A(\sigma(t)) = g(x_1, x_2, y_1, y_2),$$

and we see that $A(\sigma(s)) >_{rpo} A(\sigma(t))$.

Terms may be compared more efficiently in the associative path ordering if we replace x by $g(\tilde{S}_x)$ instead of $g(x_1, x_2)$, where the symbolic multiset S_x represents $\{x_1, x_2\}$. For instance, in the example above, we have $\sigma(x) = g(\tilde{S}_x)$ and $\sigma(y) = g(\tilde{S}_y)$, and obtain

$$A(\sigma(s)) = A(f(g(\tilde{S}_x), g(\tilde{S}_y))) = g(\tilde{N}),$$

where $N = f(S_x, S_y)$, and

$$A(\sigma(t)) = A(g(g(\tilde{S}_x), g(\tilde{S}_y))) = g(\tilde{S}_x, \tilde{S}_y).$$

Theorem 10 provides the basis for our method of proving termination of AC rewriting systems. In the next section we will demonstrate the method with several examples.

8. Examples

EXAMPLE 6. *Abelian group theory.* Let AC be the set of equations

$$\{x + y = y + z, (x + y) + z = x + (y + z)\}$$

and let R consist of the following rules:

(R1) $x + 0 \to x$,
(R2) $x + I(x) \to 0$,
(R3) $I(0) \to 0$,
(R4) $I(I(x)) \to x$,
(R5) $I(x + y) \to I(x) + I(y)$.

R/AC is a confluent equational rewriting system for abelian group theory. In this example the associative path ordering reduces to the recursive path ordering since there is only one AC operator and all terms in R are already flattened. Rules (R1), (R3) and (R4) are ordered correctly by the subterm property. For rule (R5) we need $I > +$. Rule (R2) is ordered correctly since 0 is a minimal constant. The resulting ordering $>$ satisfies the associative path condition, therefore R is AC-terminating.

EXAMPLE 7. *Boolean algebra.* The following confluent equational rewriting system for Boolean algebra is taken from Hsiang (1985). R contains the following rules:

(R1) $x + 0 \to x$,
(R2) $x \cdot 0 \to 0$,
(R3) $x \cdot 1 \to x$,
(R4) $x \cdot x \to x$,
(R5) $(x + y) \cdot z \to x \cdot z + y \cdot z$,
(R6) $x + x \to 0$,
(R7) $x \vee y \to (x \cdot y) + (x + y)$,
(R8) $x \supset y \to (x \cdot y) + (x + 1)$,
(R9) $x \equiv y \to (x + y) + 1$,
(R10) $-x \to x + 1$.

AC consists of the associativity and commutativity laws for $\cdot$ and $+$. 0 and 1 are constants, they stand for FALSE and TRUE, respectively. The operators $\cdot$, $+$, $\vee$, $\supset$, $\equiv$ and $-$ denote conjunction, exclusive disjunction (EXCLUSIVE-OR), disjunction, implication, equivalence and negation, respectively. We have to construct an appropriate precedence ordering. Rules (R1)–(R4) can be ordered correctly by the subterm property. Rule (R5) is a distributivity rule corresponding to $\cdot > +$. For rule (R6) we need $1 > 0$; for rules (R7) and (R8) $\vee > \cdot$ and $\supset > \cdot$; for rule (R9) $\equiv > +$ and $\equiv > 1$; and for rule (R10) $- > +$ and $- > 1$. The ordering $>$ in fact satisfies the associative path condition, therefore R/AC is terminating.

EXAMPLE 8. The following example is taken from Huet (1980). F consists of operators 0, 1, e, $+$, $*$, where 0 and 1 are constants, e is unary, and $+$ and $*$ are binary. Let AC be the set of equations

$$\{(x+y)+z = x+(y+z),\ x+y = y+x,\ (x*y)*z = x*(y*z),\ x*y = y*x\}$$

and let R consist of the following rules:

(R1) $x+0 \rightarrow x$,
(R2) $x*1 \rightarrow x$,
(R3) $e(0) \rightarrow 1$,
(R4) $e(x+y) \rightarrow e(x)*e(y)$,

The associative path ordering corresponding to the following precedence ordering on F can be used to prove termination: $e > *$ and $e > 1$. A different termination proof, based on extending the set of reduction rules rather than on transforms, is given by Jouannaud & Munoz (1984). They have to design a rather complex *ad hoc* ordering for proving termination of $\rightarrow_{R',E}$, where R' is a certain extension of R.

EXAMPLE 9. *Rings*. The following example is taken from Hullot (1980). Let R be the following set of rules: rules (R1)–(R5) for abelian group theory from Example 6 and the following additional rules:

(R6) $x*(y+z) \rightarrow (x*y)+(x*z)$,
(R7) $(x+y)*z \rightarrow (x*z)+(y*z)$,
(R8) $x*0 \rightarrow 0$,
(R9) $0*x \rightarrow 0$,
(R10) $x*I(y) \rightarrow I(x*y)$,
(R11) $I(x)*y \rightarrow I(x*y)$.

AC contains the associativity and commutativity axiom for $+$. R/AC is a confluent equational rewriting system for rings. AC-termination may be shown using the associative path ordering corresponding to $>$, where $* > I > +$. The following polynomial interpretation can also be used for proving termination:

$\tau(0) = 2$,
$\tau(-) = \lambda x\,.\,2 \times (x+1)$,
$\tau(+) = \lambda x\ \ y\,.\,x+y+5$,
$\tau(*) = \lambda x\ \ y\,.\,x \times y+1$.

It clearly requires more expertise to find an appropriate polynomial interpretation than to find a suitable precedence ordering $>$.

EXAMPLE 10. The following is a confluent equational rewriting system R/AC for associative-commutative rings with unit. AC consists of the associativity and commutativity axioms for $+$ and $*$. R consists of the following rules:

(R1) $x+0 \to x$,
(R2) $x+I(x) \to 0$,
(R3) $I(0) \to 0$,
(R4) $I(I(x)) \to x$,
(R5) $I(x+y) \to I(x)+I(y)$,
(R6) $x*(y+z) \to (x*y)+(x*z)$,
(R7) $x*0 \to 0$,
(R8) $x*I(y) \to I(x*y)$,
(R9) $x*1 \to x$.

We cannot use the ordering $>$ given above, since now both $+$ and $*$ are AC operators, and therefore the associative path condition is not satisfied anymore. We introduce a different confluent equational rewriting system for the same structure. The idea is to introduce a new constant c which represents $I(1)$. Using AC completion we obtain the following system:

(R0) $I(x) \to c*x$,
(R1) $x+0 \to x$,
(R2') $x+(c*x) \to 0$,
(R2'') $1+c \to 0$,
(R4') $c*c \to 1$,
(R6) $x*(y+z) \to (x*y)+(x*z)$,
(R7) $x*0 \to 0$,
(R9) $x*1 \to x$.

Termination of this equational rewriting system can be proved using the associative path ordering corresponding to the precedence ordering $>$, with $I > * > +$, $I > c > 0$, and $c > 1$.

EXAMPLE 11. Let us consider the standard confluent equational term rewriting system R for associative-commutative rings again. We cannot use an associative path ordering since the AC operator $*$ has to be greater than both I and $+$, which violates the associative path condition. One way to deal with this situation is to consider the operators I and $+$ as being equivalent. That is, we preprocess R by introducing a new operator f that replaces both I and $+$. Let R' denote the resulting rewriting system, AC' denote the AC theory corresponding to the new set of operator symbols, assuming f is an AC operator, and let t' denote the term resulting from t by replacing all occurrences of I and $+$ by f. Then $s \sim_{AC} t$ implies $s' \sim_{AC'} t'$ and $s \to_R t$ implies $s' \to_{R'} t'$. Therefore, for any infinite sequence of terms $t_1 \to_{R/AC} t_2 \to_{R/AC} t_3 \cdots$ there is an infinite sequence of terms $t'_1 \to_{R'/AC'} t'_2 \to_{R'/AC'} t'_3 \cdots$. Thus AC'-termination of R' implies AC-termination of R. For our example we get the following set of rules R':

(R1') $f(x, 0) \to x$,
(R2') $f(x, f(x)) \to 0$,
(R3') $f(0) \to 0$,
(R4') $f(f(x)) \to x$,
(R5') $f(f(x, y)) \to f(f(x), f(y))$,

(R6') $x*f(y, z) \to f(x*y, z*z)$,
(R7') $x*0 \to 0$,
(R8') $x*f(y) \to f(x*y)$,
(R9') $x*1 \to x$.

Rules (R1'), (R3'), (R4'), (R7') and (R9') can be ordered correctly by the subterm property, (R2') if 0 is minimal among all constants, that is, if $1 > 0$. Rules (R6') and (R8') are distributivity rules for $* > f$. Finally, in (R5') both terms are transformed to $f(x, y)$, but they may be ordered by inverse of size. The resulting precedence ordering satisfies the associative path condition. This proves that R'/AC' and therefore R/AC is terminating.

The technique outlined in the last example is applicable in general, subject to the restriction that any newly introduced operator f that replaces an AC operator has to be an AC operator too. The new operator f may be of variable arity, as in the example above. In general, if F_D denotes the set of operators that occur in one of the distributivity rules in D, then F_{AC} must be a subset of F_D in order to ensure that the corresponding associative path ordering is compatible with AC. But F_D may also contain other operators that are not in F_{AC}.

9. Summary

We have introduced a class of termination orderings for associative-commutative term rewriting systems, called associative path orderings, that are based on the idea of transforming terms. Associative path orderings extend the well-known recursive path orderings and provide a conceptually simpler alternative to polynomial interpretations. They may be applied in any context in which a precedence ordering that satisfies a certain condition, the associative path condition, can be specified. The precedence ordering can often be derived from the structure of the reduction rules. We have shown how to compare non-ground terms in this ordering and have demonstrated the usefulness of the ordering with a number of examples. In addition, we pointed out ways to deal with situations where the associative path condition is too restrictive.

We believe that our approach of using transforms for termination orderings is also applicable to equational theories other than associativity-commutativity. Transforms should also be applicable in connection with orderings other than the recursive path ordering: the recursive decomposition ordering or a semantic path ordering, for instance. For example, if we include an associativity rule in R rather than in E, this rule cannot be ordered using a recursive path ordering, although it can be ordered using the lexicographic path ordering of Kamin & Levy (1980). Some of these problems are explored in Bachmair & Dershowitz (1985).

We would like to thank Nachum Dershowitz for his valuable suggestions and are also thankful to the referees for their comments.

References

Bachmair, L., Dershowitz, N. (1985). Commutation, transformation, and termination. (Submitted).
Dershowitz, N. (1979). A note on simplification orderings. *Inf. Proc. Lett.* **9**, 212–215.
Dershowitz, N. (1982). Orderings for term-rewriting systems. *Theor. Comp. Sci.* **17**, 279–301.
Dershowitz, N. (1985*a*). Termination, *Proc. First Int. Conf. on Rewriting Techniques and Applications*, Dijon, France, Lect. Notes in Comp. Scie., Springer.
Dershowitz, N. (1985*b*). Computing with rewrite systems. *Inf. Control* (to appear).

Dershowitz, N., Manna, Z. (1979). Proving termination with multiset orderings. *Commun ACM* **22**, 465–476.

Dershowitz, N., Hsiang, J., Josephson, N. A., Plaisted, D. A. (1983). Associative-commutative rewriting. *Proc. 8th IJCAI*, pp. 940–944. Karlsruhe.

Guttag, J. V., Kapur, D., Musser, D. R. (1978). Abstract data types and software validation. *Commun. ACM* **21**, 1048–1064.

Hsiang, J. (1985). Refutational theorem proving using term-rewriting systems. *Artif. Intell.* **25**, 255–300.

Huet, G. (1980). Confluent reductions: abstract properties and applications to term rewriting systems. *J. Assoc. Comp. Mach.* **27**, 797–821.

Huet, G., Lankford, D. S. (1978). On the uniform halting problem for term rewriting systems. Rapport Laboria 283, IRIA.

Huet, G., Oppen, D. C. (1980). Equations and rewrite rules: a survey. In: (Book, R. ed.) *Formal Languages: Perspectives and Open Problems*, pp. 349–405. New York: Academic Press.

Hullot, J.-M. (1980). A catalogue of canonical term rewriting systems. Technical Report CSL-113, SRI International, Menlo Park, Calif.

Jouannaud, J.-P., Kirchner, H. (1984). Completion of a set of rules modulo a set of equations. *11th ACM Symp. on Principles of Programming Languages*. Salt Lake City, Utah, January 15–18, pp. 83–92.

Jouannaud, J.-P., Lescanne, P., Reinig, F. (1983). Recursive decomposition ordering. In: (Bjorner, D., ed.) *IFIP Working Conf. on Formal Description of Programming Concepts II*. Amsterdam: North-Holland.

Jouannaud, J.-P., Munoz, M. (1984). Termination of a set of rules modulo a set of equations. In: (Shostak, R., ed.) Proc. 7th Int. Conf. on Automated Deduction. *Springer Lec. Notes Comp. Sci.*, **170**, 175–193.

Kamin, S., Levy, J. J. (1980). *Two Generalizations of the Recursive Path Ordering*. Unpublished memo, Univ. of Illinois at Urbana-Champaign.

Kapur, D., Narendran, P., Sivakumar, G. (1985). A path ordering for proving termination of term rewriting systems. *Proc. 10th Colloquium on Trees in Algebra and Programming*.

Kapur, D., Sivakumar, G. (1983). Architecture of and experiments with RRL, a rewrite rule laboratory. *Proc. NSF Workshop on the Rewrite Rule Laboratory*, pp. 33–56. Rensellaerville, New York.

Knuth, D., Bendix, P. (1970). Simple word problems in universal algebras. In: (Leech, J., ed.) *Computational Problems in Abstract Algebra*, pp. 263–297. Oxford: Pergamon Press.

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

Lankford, D. (1979). *On Proving Term Rewriting Systems are Noetherian*. Memo MTP-3, Mathematics Department, Lousiana Technical University, Ruston, LA.

Lankford, D., Ballantyne, A. (1977). *Decision Procedures for Simple Equational Theories with Associative Commutative Axioms: Complete Sets of Associative Commutative Reductions*. Technical Report, Univ. of Texas at Austin, Dept. of Math. and Comp. Scie.

Lescanne, P. (1981). Decomposition ordering as a tool to prove the termination of rewriting systems. *Proc. 7th IJCAI*, pp. 548–550. Vancouver, Canada.

Lescanne, P. (1982). Some properties of decomposition ordering, a simplification ordering to prove termination of rewriting systems. *R.A.I.R.O. Informatique Theoretique/Theoretical Informatics* **16**, 331–347.

Lescanne, P. (1983). Computer experiments with the REVE term rewriting system. *Proc. 10th ACM Symp. on Principles of Programming Languages*.

Manna, Z., Ness, S. (1970). On the termination of Markov algorithms. *Proc. 3rd Hawaii Int. Conf. on System Science*, pp. 789–792. Honolulu, Hawaii.

Munoz, M. (1983). *Probleme de Terminaison Finie des Systemes de Reécriture Equationnels*. Thesis, Université Nancy 1.

Peterson, G. E., Stickel, M. E. (1981). Complete sets of reductions for some equational theories. *J. Assoc. Comp. Mach.* **28**, 233–264.

Plaisted, D. A. (1978*a*). *Well-founded Orderings for Proving Termination of Systems of Rewrite Rules*. Dept. of Computer Science Report 78-932, Univ. of Illinois at Urbana-Champaign.

Plaisted, D. A. (1978*b*). *A Recursively Defined Ordering for Proving Termination for Term Rewriting Systems*. Dept. of Computer Science Report 78–943, Univ. of Illinois at Urbana-Champaign.

Plaisted, D. A. (1983). An associative path ordering. *Proc. NSF Workshop on the Rewrite Rule Laboratory*, pp. 123–126. Rensellaerville, New York.

Rusinowitch, M. (1985). Plaisted ordering and recursive decomposition ordering revisited. *Proc. First Int. Conf. on Rewriting Techniques and Applications*, Dijon, France. *Springer Lec. Notes Comp. Sci.* (in press).