

Semi-completeness of Hierarchical and Super-hierarchical Combinations of Term Rewriting Systems

M. R. K. Krishna Rao*

Computer Science Group
Tata Institute of Fundamental Research
Colaba, BOMBAY 400 005, INDIA
e-mail: krishna@tifrvax.bitnet

Abstract. In this paper, we study modular aspects of hierarchical and super hierarchical combinations of term rewriting systems. In particular, a sufficient condition for modularity of semi-completeness of hierarchical and super hierarchical combinations is proposed. We first establish modularity of weak normalization for this class (defined by the sufficient condition) and modularity of semi-completeness for a class of crosswise independent unions. From these results, we obtain modularity of semi-completeness for a class of hierarchical and super hierarchical combinations. Our results generalize the semi-completeness results of Ohlebusch [14] and Middeldorp and Toyama [13]. The notion of crosswise independent unions is a generalization of both constructor sharing unions as well as Plump's crosswise disjoint unions.

1 Introduction

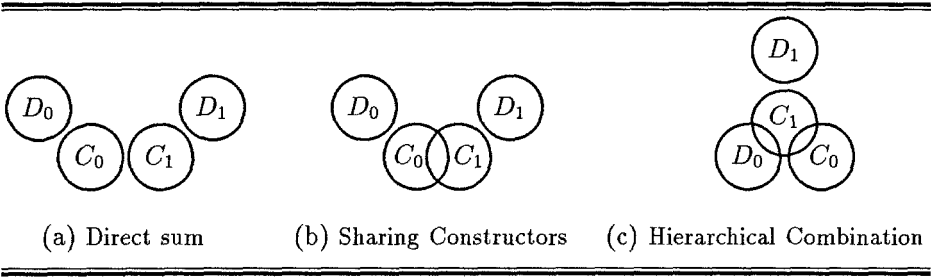
In the last few decades, term rewriting systems (TRS, for short) have played a fundamental role in the analysis and implementation of abstract data type specifications, decidability of word problems, theorem proving, computability theory, design of functional programming languages (e.g. Miranda), integration of functional programming and logic programming paradigms, etc. The study of properties which are preserved under combinations of term rewriting systems (called modular properties) is of both theoretical and practical importance. In particular, the modularity results facilitate (i) incrementality in the synthesis of systems and (ii) divide-and-conquer approach in the analysis of systems. One of the important properties of TRSs is semi-completeness (confluence + weak normalization). This property is very useful in establishing consistency of equality theories. In this paper, we study modularity of semi-completeness.

Starting with a seminal paper [18] by Toyama, modularity has been receiving a lot of attention in the last ten years. Since none of the interesting properties are modular under arbitrary combinations, a start is made by considering unions of

* Some of the details were worked out during the author's stay at Max-Planck-Institut für Informatik, Saarbrücken.

TRSS with disjoint alphabets (such unions are called direct-sums). After a good number of results were obtained (in among others, [18, 16, 11, 9]), researchers (see a.o. [10, 13, 4, 14]) considered unions of systems sharing constructor symbols — a function symbol is a constructor in a given TRS if it does not occur at the outermost level in the left-hand side of any rewrite rule in the system; otherwise it is called a defined symbol. Very recently, more practical unions called hierarchical combinations — where one system is allowed to use defined symbols of the other system as constructors (or built-ins) — are considered in [6, 7, 2]. See [12] for a comprehensive study of modularity results for direct-sums and (extended version of) [6] for a brief survey of the very recent results.

The following diagram gives the pictorial view of direct-sums, constructor sharing systems and hierarchical combinations. The sets of defined and constructor symbols of \mathcal{R}_i are denoted by D_i and C_i respectively.



In this paper, we deal with modular aspects of hierarchical combinations (in fact, a larger class called super hierarchical combinations), in particular *semi-completeness*. A TRS is semi-complete if it is both weakly-normalizing and confluent. It's known that confluence is not modular even for constructor sharing systems. Recently, Ohlebusch [14] established that confluence is modular for constructor sharing systems provided the constituent systems are weakly-normalizing. Since weak-normalization is known to be modular for constructor sharing systems, this amounts to showing modularity of semi-completeness for constructor sharing systems. However, neither semi-completeness nor weak-normalization is modular for hierarchical combinations. In this paper, we propose a set of sufficient conditions for modularity of semi-completeness of hierarchical combinations. It may be noted that techniques of neither [14] nor [6, 7, 2] are applicable in establishing our results.

We first establish modularity of weak normalization for a class of super hierarchical combinations by giving an algorithm to find a normal form of a given term. This algorithm is described by a strongly normalizing relation, called \rightarrow_a . It is shown that \rightarrow_a is confluent if the two component systems are semi-complete. Then, we establish modularity of semi-completeness for a class of crosswise independent unions. From these results, we obtain modularity of semi-completeness for a class of hierarchical and super hierarchical combinations using Staples' lemma.

The rest of the paper is organized as follows. The next section gives preliminary definitions and results needed later. In section 3, some classes of hierarchical and super hierarchical combinations (nice-extensions and nice-extensions*) are defined. Section 4 establishes modularity of weak normalization for nice extensions*. Using this result and Staples lemma, modularity of semi-completeness for a class of nice extensions* is established in section 5. Section 6 discusses some extensions of these results which are omitted due to space restrictions. Section 7 concludes with a summary.

2 Preliminaries

We assume that the reader is familiar with the basic terminology of term rewriting systems and give definitions only when they are required. The notations not defined in the paper can be found in Dershowitz and Jouannaud [1], Klop [5] or Middeldorp [12].

Definition 1 (critical pairs)

Let $l_1 \rightarrow r_1$ and $l_2 \rightarrow r_2$ be renamed versions of rewrite rules of a TRS \mathcal{R} such that they have no variables in common. Suppose $l_{1|p}$ is not a variable for some position p and $l_{1|p}$ unifies with l_2 through a most general unifier σ . The pair of terms $\langle l_1[r_2]_p\sigma, r_1\sigma \rangle$ is called a critical pair of \mathcal{R} . If $l_1 \rightarrow r_1$ and $l_2 \rightarrow r_2$ are renamed versions of the same rewrite rule, we do not consider the case $p = \epsilon$. A critical pair $\langle l_1[r_2]_p\sigma, r_1\sigma \rangle$ with $p = \epsilon$ is called an overlay and a critical pair $\langle s, t \rangle$ is trivial if $s \equiv t$.

In the following, $\mathcal{T}(\mathcal{F}, \mathcal{X})$ denotes the set of terms constructed from set of function symbols \mathcal{F} and set of variables \mathcal{X} , and $F(t)$ denotes the set of function symbols occurring in term t . $root(t) = f$ if $t \equiv f(s_1, \dots, s_n)$, and $root(t) = t$ if $t \in \mathcal{X}$.

Definition 2 The set $D_{\mathcal{R}}$ of *defined* symbols of a TRS $\mathcal{R}(\mathcal{F}, R)$ is defined as $\{root(l) \mid l \rightarrow r \in R\}$ and the set $C_{\mathcal{R}}$ of *constructor* symbols of $\mathcal{R}(\mathcal{F}, R)$ is defined as $\mathcal{F} - D_{\mathcal{R}}$.

We need the following definitions and lemmas in the sequel.

Definition 3 The dependency relation of TRS $\mathcal{R}(D, C, R)$ is defined as:

1. We say that a defined symbol $f \in D$ *directly depends on* defined symbol $g \in D$ (denoted as $f \rightarrow_d g$) if there is a rewrite rule $l \rightarrow r \in R$ such that $f \equiv root(l)$ and $g \in F(r)$.
2. The dependency relation of \mathcal{R} is the transitive reflexive closure of \rightarrow_d . We say that h depends on f if $h \rightarrow_d^* f$.
3. The set of symbols depending on a set of symbols S is defined as $\{f \mid f \rightarrow_d^* g \text{ and } g \in S\}$.

The following fundamental result due to Staples is very useful in establishing confluence (and hence semi-completeness) of relations.

Definition 4 (Staples [17])

Let S be a set and \rightarrow_a and \rightarrow_b be two binary relations on S . We say that the relation \rightarrow_b is a *refinement of the relation \rightarrow_a* if $\rightarrow_a \subseteq \rightarrow_b^*$. A refinement \rightarrow_b of \rightarrow_a is a *compatible refinement* of \rightarrow_a if $\forall s, t, u \in S \ \exists v \in S \ (s \rightarrow_b t \rightarrow_a^* u \implies u \rightarrow_a^* v \ \& \ s \rightarrow_a^* v)$.

Lemma 1 (Staples [17])

Let S be a set and \rightarrow_a and \rightarrow_b be two binary relations on S such that \rightarrow_b is a *compatible refinement* of \rightarrow_a . Then, \rightarrow_a is *confluent* if and only if \rightarrow_b is *confluent*.

When \rightarrow_a is confluent, the following lemma exploiting this fact is useful.

Lemma 2 Let S be a set and \rightarrow_a and \rightarrow_b be two binary relations on S such that \rightarrow_b is a refinement of \rightarrow_a and \rightarrow_a is confluent. Then, \rightarrow_b is a compatible refinement of \rightarrow_a if $\forall s, t \in S \ \exists u \in S \ (s \rightarrow_b t \implies s \rightarrow_a^* u \ \& \ t \rightarrow_a^* u)$.

3 Hierarchical and Super-hierarchical Combinations

In this section, we define a class (called, nice-extensions) of hierarchical combinations for which modularity of weak-normalization and semi-completeness properties is studied in later sections. The term nice-extension was first introduced by Krishna Rao [6] where modularity of completeness was established. In [7], Krishna Rao proved modularity of simple termination for a very similar (but slightly different) class of hierarchical combination using techniques very different from those of [6]. This lead to some confusion about nice-extensions. One of the contributions of this section is to clarify this confusion and give a definition of nice-extensions which includes both the classes considered in [6] and [7]. Before defining this class, we show that weak-normalization and semi-completeness are not modular for hierarchical combinations in general.

Example 1 It is easy to see that the following two systems \mathcal{R}_0 and \mathcal{R}_1 are *complete* and hence both weakly-normalizing and semi-complete.

$$\begin{array}{ll} \mathcal{R}_0 : & f(x) \rightarrow x \\ & \mathcal{R}_1 : \quad g(y) \rightarrow f(y) \\ & \quad h(a) \rightarrow h(g(a)) \end{array}$$

To wit, the combined system is not weakly-normalizing (and hence not semi-complete). The term $h(a)$ has no normal form with respect to the combined system — note the cyclic derivation:

$$h(a) \Rightarrow_{\mathcal{R}_1} h(g(a)) \Rightarrow_{\mathcal{R}_1} h(f(a)) \Rightarrow_{\mathcal{R}_0} h(a) \cdots$$

□

In the above example, the combined system is confluent. This might give an impression that confluence is modular for hierarchical combinations. However, it is not the case even for constructor sharing systems as pointed out by Kurihara and Ohuchi [10].

Example 2 The following two systems with a shared constructor, h , are confluent.

$$\begin{array}{ll} \mathcal{R}_0 : & f(x, x) \rightarrow a \\ & f(x, h(x)) \rightarrow b \end{array} \quad \mathcal{R}_1 : \quad g \rightarrow h(g)$$

But $\mathcal{R}_0 \cup \mathcal{R}_1$ is not confluent; $f(g, g)$ has two different normal forms, a and b . \square

For discussions in the sequel, it is convenient to classify defined symbols in D_1 into two sets (i) $D_1^0 = \{ f \mid f \in D_1 \text{ and } f \rightarrow_d^* D_0 \}$ consisting of function symbols depending on D_0 and (ii) $D_1^1 = D_1 - D_1^0$ consisting of function symbols not depending on D_0 .

The following definition characterizes the class of hierarchical combinations we are interested in.

term rewriting system

Definition 5 A term rewriting system $\mathcal{R}_1(D_1, C_1, R_1)$ is a *nice-extension* of another system $\mathcal{R}_0(D_0, C_0, R_0)$ if the following conditions are satisfied:

1. $D_0 \cap D_1 = C_0 \cap D_1 = \phi$ (i.e., $\mathcal{R}_0 \cup \mathcal{R}_1$ is a hierarchical combination).
2. Each rewrite rule $l \rightarrow r \in \mathcal{R}_1$ satisfies the following condition:
(H1): For every subterm s of r , if $root(s) \in D_1^0$, then s contains no function symbol depending on D_0 except at the outermost level (of s).

The second (and the main) condition essentially says that nesting of defined symbols from D_1^0 is not allowed in the right-hand side terms of rules and no symbol from D_0 occurs below D_1^0 -symbols.

Example 3 The following system \mathcal{R}_1 is a nice-extension of \mathcal{R}_0 .

$$\begin{array}{ll} \mathcal{R}_0 : & \text{add}(0, x) \rightarrow x \\ & \text{add}(S(x), y) \rightarrow S(\text{add}(x, y)) \end{array} \quad \mathcal{R}_1 : \quad \begin{array}{l} \text{mult}(0, x) \rightarrow 0 \\ \text{mult}(S(x), y) \rightarrow \text{add}(y, \text{mult}(x, y)) \end{array}$$

3.1 Super-hierarchical Combinations

In hierarchical combinations, defined symbols of \mathcal{R}_1 are not allowed to occur in \mathcal{R}_0 . In a few (very rare) situations, it may not be possible to divide a system into two subsystems \mathcal{R}_0 and \mathcal{R}_1 such that the combination is hierarchical, but it might be possible to divide that system into two subsystems \mathcal{R}_0 and \mathcal{R}_1 such that the defined symbols of \mathcal{R}_0 do not depend on the defined symbols of \mathcal{R}_1 . Basically, the defined symbols of \mathcal{R}_1 are allowed to occur in the left-hand side (but not in the right-hand side) terms of \mathcal{R}_0 and defined symbols of \mathcal{R}_0 can occur in both the left and the right-hand side terms of \mathcal{R}_1 . Such combinations are called super-hierarchical combinations. It may be noted that such a situation can occur with TRSS generated by completion procedures.

Definition 6 A term rewriting system $\mathcal{R}_1(D_1, C_1, R_1)$ is a *nice-extension** of another system $\mathcal{R}_0(D_0, C_0, R_0)$ if condition (H1) of definition 5 and the following condition are satisfied (here, \rightarrow_a^* is the dependency relation of the combined system):

$$\forall f \in D_0, \forall g \in D_1, f \not\rightarrow_a^* g \text{ (i.e., } \mathcal{R}_0 \cup \mathcal{R}_1 \text{ is a super-hier. combination).}$$

Notation: Until further notice, we deal with nice-extensions*. We denote the set of constructors $(C_0 \cup C_1) - (D_0 \cup D_1)$ of the combined system by *Constr*. T_i denotes $\mathcal{T}(D_i \cup \text{Constr}, \mathcal{X})$ and \mathcal{C}_i denotes the set of contexts of $D_i \cup \text{Constr}$, i.e., terms in $\mathcal{T}(D_i \cup \text{Constr} \cup \{\square\}, \mathcal{X})$. By \mathcal{C}_0^1 , we denote the set of contexts of $\text{Constr} \cup D_0 \cup D_1^1$.

The following lemma characterizes the rewrite rules in nice-extensions*.

Lemma 3 *If \mathcal{R}_1 is a nice-extension* of \mathcal{R}_0 then for each rule $l \rightarrow r \in \mathcal{R}_1$, r is of the form $C[t_1, \dots, t_n]$, where $C \in \mathcal{C}_0^1$, $\text{root}(t_i) \in D_1^0$ and $t_i \in T_1$, $1 \leq i \leq n$ ($n \geq 0$). Further, no proper subterm of t_i contains any function symbol depending on D_0 .*

Proof: Follows from the condition (H1) of Definition 5. □

4 Weak Normalization

In this section, we establish modularity of weak normalization (WN) property for nice-extensions*. We basically have to show that every term has a normal form with respect to the combined system. We do more than required by giving an algorithm to compute a normal form of any given term. This algorithm is described by the following relation \rightarrow_a on terms. In the following, we assume that \mathcal{R}_0 and \mathcal{R}_1 are weakly normalizing (WN).

Definition 7 The relation \rightarrow_a on terms is defined as follows:

$$C[s] \rightarrow_a C[t]$$

if (i) s is an innermost redex in $C[s]$ and (ii) t is a normal form of s w.r.t. \mathcal{R}_i if $\text{root}(s) \in D_i$.

We show that this relation gives an effective algorithm by establishing strong normalization (termination) of \rightarrow_a . Towards this purpose, we need the following lemmas.

Lemma 4 *If \mathcal{R}_1 is a nice-extension* of \mathcal{R}_0 and t is a term such that no subterm (say, s) of t with $\text{root}(s) \in D_1$ is reducible by $\mathcal{R}_0 \cup \mathcal{R}_1$, then any normal form of t with respect to \mathcal{R}_0 is also a normal form of t with respect to $\mathcal{R}_0 \cup \mathcal{R}_1$.*

Proof: Follows from the fact that the defined symbols of \mathcal{R}_1 do not occur in the right-hand side terms of \mathcal{R}_0 . □

The above lemma is useful in establishing that normalization of an innermost redex w.r.t. \mathcal{R}_0 decreases the depth of innermost redexes. The following two lemmas are useful in establishing that normalization of an innermost redex w.r.t. \mathcal{R}_1 decreases the depth of innermost reducible terms with root in D_1^0 .

Lemma 5 If \mathcal{R}_1 is a nice-extension* of \mathcal{R}_0 and t is a term of the form $C[t_1, \dots, t_n]$, $C \in \mathcal{C}_0^1$, $\text{root}(t_i) \in D_1^0$ such that no proper subterm (say, s) of t_i , $1 \leq i \leq n$ with $\text{root}(s) \in (D_0 \cup D_1^0)$ is reducible by $\mathcal{R}_0 \cup \mathcal{R}_1$ and $t \Rightarrow_{\mathcal{R}_1} t'$, then t' is of the form $C'[s_1, \dots, s_m]$, $C' \in \mathcal{C}_0^1$, $\text{root}(s_i) \in D_1^0$ such that no proper subterm (say, s) of s_i , $1 \leq i \leq m$ with $\text{root}(s) \in (D_0 \cup D_1^0)$ is reducible by $\mathcal{R}_0 \cup \mathcal{R}_1$.

Proof: Let $l \rightarrow r$ be the rewrite rule applied in $t \Rightarrow_{\mathcal{R}_1} t'$ and σ be the applied matching substitution. There are two cases.

Case (a): $\text{root}(l) \in D_1^1$. There are two subcases. (1) the reduction took place in C . By definition, no function symbol in D_1^0 occurs in r and hence, t' is of the form $C'[s_1, \dots, s_m]$, $C' \in \mathcal{C}_0^1$, $\text{root}(s_i) \in D_1^0$ such that each s_i is a subterm of some t_j . The lemma holds. (2) the reduction took place in some t_i and $t' \equiv C[t_1, \dots, t_{i-1}, t'_i, t_{i+1}, \dots, t_n]$. Let t_i be of the form $C_1[u_1, \dots, u_m]$ such that $\text{root}(u_j) \in (D_0 \cup D_1^0)$ and no symbol from $(D_0 \cup D_1^0)$ occurs in C_1 except at the root. By hypothesis, each u_k is irreducible. Since no function symbol from $(D_0 \cup D_1^0)$ occurs in r , t'_i is of the form $C'_1[s_1, \dots, s_{m'}]$ such that $\text{root}(s_j) \in (D_0 \cup D_1^0)$ and no symbol from $(D_0 \cup D_1^0)$ occurs in C'_1 except at the root. Further, each s_j is a subterm of some u_k and hence irreducible and the lemma holds.

Case (b): $\text{root}(l) \in D_1^0$. In this case, the reduction should take place at the root of some t_i . It follows from lemma 3 and irreducibility of proper subterms (with root in $(D_0 \cup D_1^0)$) of t_i that $r\sigma$ is of the form $C'[u_1, \dots, u_m]$, $C' \in \mathcal{C}_0^1$, $\text{root}(u_i) \in D_1^0$ such that no proper subterm (say, s) of u_i , $1 \leq i \leq m$ with $\text{root}(s) \in (D_0 \cup D_1^0)$ is reducible by $\mathcal{R}_0 \cup \mathcal{R}_1$. It is easy to see that $t' \equiv C[t_1, \dots, t_{i-1}, r\sigma, t_{i+1}, \dots, t_n]$ can be written as $C''[s_1, \dots, s_{m+n-1}]$, $C'' \in \mathcal{C}_0^1$, $\text{root}(s_i) \in D_1^0$ such that no proper subterm (say, s) of s_i , $1 \leq i \leq m+n-1$ with $\text{root}(s) \in (D_0 \cup D_1^0)$ is reducible by $\mathcal{R}_0 \cup \mathcal{R}_1$. \square

The following lemma about normal forms easily follows from this lemma.

Lemma 6 If \mathcal{R}_1 is a nice-extension* of \mathcal{R}_0 and t is a term of the form $C[t_1, \dots, t_n]$, $C \in \mathcal{C}_0^1$, $\text{root}(t_i) \in D_1^0$ such that no proper subterm (say, s) of t_i , $1 \leq i \leq n$ with $\text{root}(s) \in (D_0 \cup D_1^0)$ is reducible by $\mathcal{R}_0 \cup \mathcal{R}_1$ and t' is a normal form of t with respect to \mathcal{R}_1 , then t' is of the form $C'[s_1, \dots, s_m]$, $C' \in \mathcal{C}_0^1$, $\text{root}(s_i) \in D_1^0$ such that each s_i , $1 \leq i \leq m$ is irreducible by $\mathcal{R}_0 \cup \mathcal{R}_1$.

Now, we are in a position to establish strong normalization of \rightarrow_a .

Theorem 1 If \mathcal{R}_1 is a nice-extension* of \mathcal{R}_0 and \rightarrow_a is the relation defined above, then \rightarrow_a is strongly normalizing.

Proof: We basically prove this using a terminating function f . For any given term t , $f(t)$ is an ordered pair $\langle f_1(t), f_2(t) \rangle$, where f_1 and f_2 are defined as follows: $f_1(t)$ is the set $\{p \mid \text{root}(t|_p) \in D_1^0, t|_p \text{ is reducible and no proper subterm } s \text{ of } t|_p \text{ with } \text{root}(s) \in D_1^0 \text{ is reducible}\}$ of deepest reducible D_1^0 -positions in t and $f_2(t)$ is the set of innermost redex positions. It may be noted that p need not be a redex position but $t|_p$ is reducible.

We use the lexicographic ordering \prec induced by two ordering \prec_1 and \prec_2 on the two components as the well-founded ordering. The orderings \prec_1 and \prec_2 are multiset orderings induced by the prefix ordering over positions.

To establish strong normalization of \rightarrow_a , it is enough to show that $f(t) \succ f(t')$ whenever $t \rightarrow_a t'$. Now consider $t \equiv C[u] \rightarrow_a C[v] \equiv t'$ such that u is the innermost redex normalized to v . There are two cases: (a) $\text{root}(u) \in D_0$ and (b) $\text{root}(u) \in D_1$.

In case (a), it's obvious that $f_2(t) \succ_2 f_2(t')$ as v is a normal form of the combined system by lemma 4. Let p be the nearest D_1^0 position above u in $C[u]$. By definition, p is in $f_1(t)$. It's easy to see that no position below p can occur in $f_1(t')$ and in fact p itself may not be there in $f_1(t')$ — $t|_p$ may not be reducible. This clearly shows $f(t) \succ f(t')$.

In case (b) there are two subcases: (i) $\text{root}(u) \in D_1^0$ and (ii) $\text{root}(u) \in D_1^1$. In subcase (i) it's obvious that $f_1(t) \succ_1 f_1(t')$ by lemma 6. In subcase (ii), v is also a normal form of \mathcal{R}_0 as D_1^1 symbols do not depend on D_0 and hence $f_2(t) \succ_2 f_2(t')$. Further, $f_1(t) \succeq_1 f_1(t')$ as in case(a). Hence, $f(t) \succ f(t')$. \square

From this lemma, it follows that weak normalization (WN) is modular for the class of nice-extensions*.

Theorem 2 *Let \mathcal{R}_0 and \mathcal{R}_1 be two TRSSs such that \mathcal{R}_1 is a nice-extension* of \mathcal{R}_0 . Then, the combined system $\mathcal{R}_0 \cup \mathcal{R}_1$ is weakly normalizing (WN) if \mathcal{R}_0 and \mathcal{R}_1 are weakly normalizing (WN).*

In general, weak normalization (WN) is not modular for hierarchical combinations as demonstrated by Example 1.

5 Semi-completeness

In this section, we study modularity of semi-completeness for nice-extensions*. Unlike weak normalization, semi-completeness is not modular for the whole class of nice-extensions* as shown by the following counterexample.

Example 4 The following systems are semi-complete and \mathcal{R}_1 is a nice-extension* of \mathcal{R}_0 .

$$\mathcal{R}_0 : g(x, y) \rightarrow y \qquad \mathcal{R}_1 : f(g(x, y)) \rightarrow x$$

However, $\mathcal{R}_0 \cup \mathcal{R}_1$ is not semi-complete as the term $f(g(x, y))$ has two different normal forms x and $f(y)$. \square

In the following, we give a class of nice-extensions* for which semi-completeness is modular. Since weak normalization (WN) is shown to be modular for nice-extensions*, it is enough to show that the combined system is confluent whenever the two components are semi-complete. We establish this using Staples' lemma by establishing that the rewrite relation of the combined system is a compatible

refinement of \rightarrow_a and \rightarrow_a is confluent. In the rest of this section, we assume that \mathcal{R}_0 and \mathcal{R}_1 are semi-complete.

The following theorem establishes that \rightarrow_a is confluent for semi-complete systems.

Theorem 3 *The above relation \rightarrow_a is confluent if \mathcal{R}_0 and \mathcal{R}_1 are semi-complete.*

Proof: Since \rightarrow_a is strongly normalizing by theorem 1, it is enough to show that \rightarrow_a is locally confluent. Let u, v, w be terms such that $u \rightarrow_a v$ and $u \rightarrow_a w$. We have to show that v and w are joinable by \rightarrow_a . By the definition of \rightarrow_a , the two reductions must take place either (i) at two disjoint positions or (ii) at one position. In case (i) joinability is obvious and in case (ii) semi-completeness of \mathcal{R}_0 and \mathcal{R}_1 implies that $v \equiv w$. Therefore, \rightarrow_a is confluent. \square

In establishing that the rewrite relation of the combined system is a compatible refinement of \rightarrow_a , it is useful to first establish modularity of semi-completeness for a smaller class of combinations.

5.1 Semi-completeness of crosswise independent unions

In this section, we study modularity of semi-completeness for crosswise independent unions. The notion of crosswise independent unions is a generalization of constructor sharing unions and Plump's crosswise disjoint unions [15].

Definition 8 We say that two TRSS $\mathcal{R}_0(D_0, C_0, R_0)$ and $\mathcal{R}_1(D_1, C_1, R_1)$ are *crosswise independent* if $f_i \not\vdash_d^* f_{1-i}$ for each $f_i \in D_i$ and $f_{1-i} \in D_{1-i}$, where $i \in \{0, 1\}$. We say that $\mathcal{R}_0 \cup \mathcal{R}_1$ is a crosswise independent union if \mathcal{R}_0 and \mathcal{R}_1 are crosswise independent.

Now, we establish that the rewrite relation of the crosswise independent union is a compatible refinement of \rightarrow_a for the following class of systems.

Theorem 4 Let \mathcal{R}_0 and \mathcal{R}_1 be two crosswise independent semi-complete TRSS such that

if $l \rightarrow r \in \mathcal{R}_i$ and s is a subterm of l with $\text{root}(s) \in D_{1-i}$ then no nonvariable subterm of s unifies with the left-hand side term of any rule in $\mathcal{R}_0 \cup \mathcal{R}_1$.

Then, the rewrite relation $\Rightarrow_{\mathcal{R}_0 \cup \mathcal{R}_1}$ (denoted by \rightarrow_b henceforth) of the combined system is a compatible refinement of the relation \rightarrow_a defined above.

Proof: Consider a reduction $s \equiv C[l\sigma] \rightarrow_b C[r\sigma] \equiv t$. To prove that \rightarrow_b is a compatible refinement of \rightarrow_a , we have to show that s and t are joinable by \rightarrow_a . That's, $\exists u (s \rightarrow_a^* u \text{ and } t \rightarrow_a^* u)$.

Let σ' be the irreducible substitution obtained from σ by reducing the terms in its range to normal forms with respect to \rightarrow_a . Since \rightarrow_a is confluent and strongly normalizing, σ' is unique. It is obvious that $s \rightarrow_a^* s' \equiv C[l\sigma']$ and

$t \rightarrow_a^* t' \equiv C[r\sigma']$. Without any loss of generality assume that $l \rightarrow r \in \mathcal{R}_0$. Now, let v be the normal form of $r\sigma'$ with respect to \mathcal{R}_0 . Since D_0 -symbols do not depend on D_1 , it follows that no rule from \mathcal{R}_1 is applicable on $r\sigma'$ or on any term derived from $r\sigma'$. Therefore, $r\sigma' \rightarrow_a^* v$ (every time we reduce an innermost redex to its normal form w.r.t. \mathcal{R}_0). By semi-completeness of \mathcal{R}_0 it follows that v is the normal form of $l\sigma'$ as well and hence $l\sigma' \Rightarrow_{\mathcal{R}_0}^* v$. Since \mathcal{R}_0 and \mathcal{R}_1 satisfy the above condition, each subterm s of $l\sigma'$ with $\text{root}(s) \in D_1$ is a normal form. Again due to crosswise independence, no rule of \mathcal{R}_1 is applicable on any term derived from $l\sigma'$ and hence $l\sigma' \rightarrow_a^* v$. Therefore, \rightarrow_b is a compatible refinement of \rightarrow_a as $s \rightarrow_a^* s' \equiv C[l\sigma'] \rightarrow_a^* C[v] \equiv u$ and $t \rightarrow_a^* t' \equiv C[r\sigma'] \rightarrow_a^* C[v] \equiv u$. \square

The modularity of semi-completeness follows from the above theorem and Staples' lemma.

Theorem 5 Let \mathcal{R}_0 and \mathcal{R}_1 be two crosswise independent semi-complete TRSS such that

if $l \rightarrow r \in \mathcal{R}_i$ and s is a subterm of l with $\text{root}(s) \in D_{1-i}$ then no nonvariable subterm of s unifies with the left-hand side of any rule in $\mathcal{R}_0 \cup \mathcal{R}_1$.

Then, $\mathcal{R}_0 \cup \mathcal{R}_1$ is semi-complete too.

Proof: Confluence of $\mathcal{R}_0 \cup \mathcal{R}_1$ follows from the above theorem and Staples' lemma. Weak normalization of $\mathcal{R}_0 \cup \mathcal{R}_1$ follows from theorem 1. \square

This theorem is a generalization Ohlebusch's result (cf. [14]) on the modularity of semi-completeness for constructor sharing systems as the constructor sharing systems (i) are clearly crosswise independent and (ii) since D_i symbols do not occur in \mathcal{R}_{1-i} , the above condition is vacuously satisfied.

5.2 Semi-completeness of nice-extensions*

Theorem 6 Let \mathcal{R}_0 and \mathcal{R}_1 be two semi-complete TRS such that

1. \mathcal{R}_1 is a nice-extension* of \mathcal{R}_0 ,
2. if $l \rightarrow r \in \mathcal{R}_i$ and s is a subterm of l with $\text{root}(s) \in D_{1-i}$ then no nonvariable subterm of s unifies with the left-hand side of any rule in $\mathcal{R}_0 \cup \mathcal{R}_1$ and
3. if $l \rightarrow r \in \mathcal{R}_1$ and $\text{root}(l) \in D_1^0$ then all the critical pairs involving this rule are overlays.

Then, the rewrite relation $\Rightarrow_{\mathcal{R}_0} \cup \Rightarrow_{\mathcal{R}_1}$ (denoted by \rightarrow_b henceforth) of the combined system is a compatible refinement of the relation \rightarrow_a defined in the previous section.

Proof: Consider a reduction $s \equiv C[l\sigma] \rightarrow_b C[r\sigma] \equiv t$. We have to show that $\exists u (s \rightarrow_a^* u \text{ and } t \rightarrow_a^* u)$. Let σ' be the irreducible substitution obtained from σ by reducing the terms in its range to normal forms with respect to \rightarrow_a . Since \rightarrow_a is confluent and strongly normalizing, σ' is unique. It is obvious that $s \rightarrow_a^*$

$s' \equiv C[l\sigma']$ and $t \rightarrow_a^* t' \equiv C[r\sigma']$. Now, we have two cases: (i) $l \rightarrow r \in \mathcal{R}_0$ and (ii) $l \rightarrow r \in \mathcal{R}_1$.

Case (i). Let v be the normal form of $l\sigma'$ with respect to \mathcal{R}_0 . By semi-completeness of \mathcal{R}_0 it follows that v is the normal form of $r\sigma'$ as well. By condition 2, every subterm s of $l\sigma'$ with $\text{root}(s) \in D_1$ is irreducible. Further, since D_0 -symbols do not depend on D_1 , it follows that no rule from \mathcal{R}_1 is applicable on any term derived from $l\sigma'$. Therefore, $l\sigma' \rightarrow_a^* v$ and $r\sigma' \rightarrow_a^* v$. Hence, $s \rightarrow_a^* s' \equiv C[l\sigma'] \rightarrow_a^* C[v] \equiv u$ and $t \rightarrow_a^* t' \equiv C[r\sigma'] \rightarrow_a^* C[v] \equiv u$.

Case (ii). Let v be the normal form of $l\sigma'$ with respect to \mathcal{R}_1 . By semi-completeness of \mathcal{R}_1 it follows that v is the normal form of $r\sigma'$ as well. There are two subcases: (a) $\text{root}(l) \in D_1^1$ and (b) $\text{root}(l) \in D_1^0$. By conditions 2 and 3 of the theorem, in case (a) it follows that no rule $l' \rightarrow r'$ with $\text{root}(l') \in (D_0 \cup D_1^0)$ is applicable on $l\sigma'$. Since D_1^1 -symbols do not depend on $(D_0 \cup D_1^0)$ these rules are not applicable on any term derived from $l\sigma'$. Therefore, $l\sigma' \rightarrow_a^* v$ and $r\sigma' \rightarrow_a^* v$. Hence, $s \rightarrow_a^* s' \equiv C[l\sigma'] \rightarrow_a^* C[v] \equiv u$ and $t \rightarrow_a^* t' \equiv C[r\sigma'] \rightarrow_a^* C[v] \equiv u$.

In subcase (b), by conditions 2 and 3, it follows that $l\sigma'$ is an innermost redex and hence $l\sigma' \rightarrow_a v$. By lemma 3, r is of the form $C'[s_1, \dots, s_n]$, $C' \in \mathcal{C}_0^1$, $\text{root}(s_i) \in D_1^0$ and no proper subterm of s_i has root in $(D_1^0 \cup D_0)$. Therefore, \mathcal{R}_0 is not applicable on any proper subterm of $s_i\sigma'$ or on any term derived from a proper subterm of $s_i\sigma'$. Hence it follows that $s_i\sigma' \rightarrow_a^* t_i$, where t_i is the normal form of $s_i\sigma'$ w.r.t. \mathcal{R}_1 (in $s_i\sigma' \rightarrow_a^* t_i$, every time we normalize an innermost redex w.r.t. \mathcal{R}_1). By lemma 6, t_i is of the form $C_i[t_{i_1}, \dots, t_{i_{k_i}}]$, $C_i \in \mathcal{C}_0^1$, $\text{root}(t_{i_j}) \in D_1^0$ and t_{i_j} is a normal form of $\mathcal{R}_0 \cup \mathcal{R}_1$. By reducing each $s_i\sigma'$ to t_i , we get $r\sigma' \rightarrow_a^* C'[t_1, \dots, t_n] \equiv v'$ and $r\sigma' \Rightarrow_{\mathcal{R}_1}^* v'$. By semi-completeness of \mathcal{R}_1 , it follows that v is the normal form of v' w.r.t. \mathcal{R}_1 and $v' \Rightarrow_{\mathcal{R}_1}^* v$. The term v' can be written as $C''[t_{1_1}, \dots, t_{1_{k_1}}, \dots, t_{n_1}, \dots, t_{n_{k_n}}]$ such that $C'' \in \mathcal{C}_0^1$, $\text{root}(t_{i_j}) \in D_1^0$, where $C'' \equiv C'[C_1, \dots, C_n]$. Since each t_{i_j} is a normal form of $\mathcal{R}_0 \cup \mathcal{R}_1$, no rule $l' \rightarrow r'$ with $\text{root}(l') \in D_1^0$ is applicable on v' or on any term derived from v' (since $C'' \in \mathcal{C}_0^1$). By a similar reasoning it follows that no rule $l' \rightarrow r'$ with $\text{root}(l') \in D_1^0$ is applicable on v or on any term derived from v .

The situation can be paraphrased as follows: (i) $v' \Rightarrow_{\mathcal{R}_1}^* v$ and (ii) the only rules applicable on v and v' and any term derived from them are from $\mathcal{R}_0 \cup \mathcal{R}'_1$, where $\mathcal{R}'_1 = \{l \rightarrow r \in \mathcal{R}_1 \mid \text{root}(l) \in D_1^1\}$. The system \mathcal{R}'_1 is semi-complete since no symbol in D_1^1 depends on D_1^0 . Since \mathcal{R}_0 and \mathcal{R}'_1 are crosswise independent, $\mathcal{R}_0 \cup \mathcal{R}'_1$ is semi-complete by theorem 5. Therefore, v and v' have the same normal form (say, u) w.r.t. $\mathcal{R}_0 \cup \mathcal{R}'_1$ and $v \rightarrow_a^* u$ and $v' \rightarrow_a^* u$ (every time we normalize an innermost redex w.r.t. \mathcal{R}_0 or \mathcal{R}'_1). Hence $s \rightarrow_a^* s' \equiv C[l\sigma'] \rightarrow_a C[v] \rightarrow_a^* C[u]$ and $t \rightarrow_a^* t' \equiv C[r\sigma'] \rightarrow_a^* C[v'] \rightarrow_a^* C[u]$. \square

Remark: note that, $l\sigma' \rightarrow_a v$ need not be true in case (i) and subcase (a) of case (ii) in the proof of the above Theorem — in particular $l\sigma'$ may not be an innermost redex.

The modularity of semi-completeness follows from the above theorem and Staples' lemma.

Theorem 7 Let \mathcal{R}_0 and \mathcal{R}_1 be two semi-complete TRS such that

1. \mathcal{R}_1 is a nice-extension* of \mathcal{R}_0 ,
2. if $l \rightarrow r \in \mathcal{R}_i$ and s is a subterm of l with $\text{root}(s) \in D_{1-i}$ then no nonvariable subterm of s unifies with the left-hand side of any rule in $\mathcal{R}_0 \cup \mathcal{R}_1$ and
3. if $l \rightarrow r \in \mathcal{R}_1$ and $\text{root}(l) \in D_1^0$ then all the critical pairs involving this rule are overlays.

Then, the combined system $\mathcal{R}_0 \cup \mathcal{R}_1$ is semi-complete as well.

6 Extending the results

In this section, we discuss the extensions of the above results in two directions. These (extensions of the) results are established in the larger version of the paper, but omitted here to meet the space restrictions.

In the above, we assume that the two component systems do not share any rules and defined symbols, i.e., $D_0 \cap D_1 = \emptyset$. However, we can allow two systems to share defined symbols provided the rules defining these symbols are the same in both the systems, i.e., we can consider the following situation: two systems $\mathcal{R}_0(D_0 \cup D, C_1, R_1)$ and $\mathcal{R}_1(D_1 \cup D, C_1, R_1)$ sharing defined symbols D and rules $R_0 \cap R_1 = \{l \rightarrow r \mid \text{root}(l) \in D\}$. This generalization is in particular useful while studying properties like weak normalization, innermost normalization, confluence and semi-completeness, which do not have the following hereditary property: *if R has property P and R' is a subsystem of R then R' has property P* . Lack of this property forces us to allow two components to share some rules while studying (and proving) these properties in a modular way.

The notion of generalized nice-extension* allows such sharing. The notations of D_1^1 and D_1^0 need slight changes: $D_1^0 = \{f \mid f \in (D_1 \cup D) \text{ and } f \rightarrow_a^* D_0\}$ and $D_1^1 = (D_1 \cup D) - D_1^0$.

Definition 9 A TRS $\mathcal{R}_1(D_1 \cup D, C_1, R_1)$ is a *generalized nice-extension** of another TRS $\mathcal{R}_0(D_0 \cup D, C_0, R_0)$ if the following conditions are satisfied:

1. $D_0 \cap D_1 = \emptyset$ and $R_0 \cap R_1 = \{l \rightarrow r \mid \text{root}(l) \in D\}$.
2. $\forall f \in (D_0 \cup D), \forall g \in D_1, f \not\rightarrow_a g$ (i.e., $\mathcal{R}_0 \cup \mathcal{R}_1$ is a super-hierarchical combination).
3. Each rewrite rule $l \rightarrow r \in R_1$ satisfies the following condition:
(H1'): For every subterm s of r , if $\text{root}(s) \in (D_1^0 - D)$, then s contains no function symbol depending on D_0 except at the outermost level (of s).

All theorems in the above sections can be established for generalized nice-extensions*. However, the definition of \rightarrow_a needs a modification to handle the shared defined symbols (and rules). The proofs of all the above results can be done using just the techniques used above, but they need much more detailed case analysis. The reader is referred to [8] for the proofs of the generalized results. The main result for generalized nice-extensions* is the following.

Theorem 8 Let \mathcal{R}_0 and \mathcal{R}_1 be two semi-complete TRS such that

1. \mathcal{R}_1 is a generalized nice-extension* of \mathcal{R}_0 ,
2. if $l \rightarrow r \in \mathcal{R}_i$ and s is a subterm of l with $\text{root}(s) \in D_{1-i}$ then no nonvariable subterm of s unifies with the left-hand side of any rule in $\mathcal{R}_0 \cup \mathcal{R}_1$ and
3. if $l \rightarrow r \in \mathcal{R}_1$ and $\text{root}(l) \in D_1^0$ then all the critical pairs involving this rule are overlays.

Then, the combined system $\mathcal{R}_0 \cup \mathcal{R}_1$ is semi-complete as well.

Using this theorem, we can extend the results (as in [6, 7]) to the class of generalized proper-extensions*, where nesting of D_1^0 symbols is allowed in the right-hand side terms.

Definition 10 A TRS $\mathcal{R}_1(D_1 \cup D, C_1, R_1)$ is a *generalized proper-extension** of another TRS $\mathcal{R}_0(D_0 \cup D, C_0, R_0)$ if conditions 1 and 2 of the above definition and the following condition are satisfied:

- 3 Each rewrite rule $l \rightarrow r \in R_1$ satisfies the following condition:
(H2): For every subterm s of r , if $\text{root}(s) \in (D_1^0 - D)$ and $\text{root}(s) \rightarrow_d^* \text{root}(l)$, then s contains no function symbol depending on D_0 except at the outermost level (of s).

To relate the notion of generalized proper-extension* with generalized nice-extension*, we need the following definition.

Definition 11 From the dependency relation, \rightarrow_d^* (see Def. 3) of \mathcal{R}_1 , we define

1. Equivalence relation \approx ($f \approx g$ if $f \rightarrow_d^* g$ and $g \rightarrow_d^* f$) on the set of defined symbols D_1^0 . We denote the equivalence class containing f by $[f]$.
2. Partial ordering \sqsupset ($[f] \sqsupset [g]$ if $f \rightarrow_d^* g$ and $g \not\rightarrow_d^* f$) on the set of equivalence classes.

Assumption: In the following we assume that the relation \sqsupset on D_1^0 is noetherian.

Since signature of any TRS is a countable set, the equivalence relation \approx partitions D_1^0 into a countable set E of equivalence classes. Since relation \sqsupset is noetherian, one can easily extend it to a well-ordering of order type λ , where λ is a countable ordinal.

Notation: For any ordinal α , we denote the α^{th} element in the above well-ordering by E_α (for all ordinals $\alpha > \lambda$, we let $E_\alpha = \phi$) and the TRS $\{l \rightarrow r \in \mathcal{R}_1 \mid \text{root}(l) \in D \cup D_1^1 \cup (\bigcup_{\beta \leq \alpha} E_\beta)\}$ by R_α and the combined system $(\bigcup_{\beta < \alpha} R_\beta) \cup \mathcal{R}_0$ by S_α . In particular, S_0 is \mathcal{R}_0 and S_κ is $\mathcal{R}_0 \cup \mathcal{R}_1$ for any ordinal κ above λ .

The following theorem relates generalized proper-extensions* and generalized nice-extensions*.

Theorem 9 Let \mathcal{R}_0 and \mathcal{R}_1 be two TRSs such that \mathcal{R}_1 is a generalized proper-extension* of \mathcal{R}_0 and \sqsupset is noetherian. Then, R_α is a generalized nice-extension* of S_α for every ordinal α , where R_α and S_α denote the objects explained in the above notation.

It is easy to see that R_α is semi-complete and satisfies conditions 2 and 3 of Theorem 9 if \mathcal{R}_1 satisfies them. Therefore by induction, we have:

Theorem 10 Let \mathcal{R}_0 and \mathcal{R}_1 be two semi-complete TRS such that

1. \mathcal{R}_1 is a generalized proper-extension* of \mathcal{R}_0 and \sqsupset is noetherian,
2. if $l \rightarrow r \in \mathcal{R}_i$ and s is a subterm of l with $root(s) \in D_{1-i}$ then no nonvariable subterm of s unifies with the left-hand side of any rule in $\mathcal{R}_0 \cup \mathcal{R}_1$ and
3. if $l \rightarrow r \in \mathcal{R}_1$ and $root(l) \in D_1^0$ then all the critical pairs involving this rule are overlays.

Then, the combined system $\mathcal{R}_0 \cup \mathcal{R}_1$ is semi-complete as well.

7 Conclusion

In this paper, the modular aspects of hierarchical and super hierarchical combinations are investigated. We identified some classes of hierarchical and super hierarchical combinations for which *semi-completeness* property is modular. It is also shown that *semi-completeness* property is modular for a class of crosswise independent unions. The crosswise independent unions are generalization of constructor sharing unions as well as Plump's crosswise disjoint unions.

References

1. N. Dershowitz and J.-P. Jouannaud (1990), *Rewrite Systems*, In J. van Leeuwen (ed.), *Handbook of Theoretical Computer Science*, Vol. B, pp. 243-320, North-Holland.
2. N. Dershowitz (1995), *Hierarchical termination*, to appear in Proc. of CTRS'94, forthcoming LNCS, Springer-Verlag.
3. B. Gramlich (1992), *Relating innermost, weak, uniform and modular termination of term rewrite systems*, Proc. of Logic Prog. and Automated Reasoning, LPAR'92, Lecture Notes in Computer Science 624, pp. 285-296, Springer-Verlag.
4. B. Gramlich (1994), *Generalized sufficient conditions for modular termination of rewriting*, in AAECC (Applicable Algebra in Engineering, Communication and Computing) 5, pp. 131-158.
5. J.W. Klop (1992), *Term Rewriting Systems*, tech rep. CS-R9073, CWI, Amsterdam. Also appears as a chapter in S. Abramsky, D. Gabbay and T. Maibaum (ed.), *Handbook of Logic in Computer Science*, Vol. 2, Oxford University Press.

6. M.R.K. Krishna Rao (1993), *Completeness of hierarchical combinations of term rewriting systems*, Proc. of 13th conference on Foundations of Software Technology and Theoretical Computer Science, FST&TCS'93, Lecture Notes in Computer Science **761**, pp. 125-138, Springer-Verlag.
7. M.R.K. Krishna Rao (1994), *Simple termination of hierarchical combinations of term rewriting systems*, Proc. of Theoretical Aspects of Computer Science, TACS'94, Lecture Notes in Computer Science **789**, pp. 203-223, Springer-Verlag.
8. M.R.K. Krishna Rao (1994), *Semi-completeness of hierarchical and super hierarchical combinations of term rewriting systems*, Technical report, TIFR, Bombay, October.
9. M. Kurihara and A. Ohuchi (1990), *Modularity of simple termination of term rewriting systems*, Journal of IPS, Japan **34**, pp. 632-642.
10. M. Kurihara and A. Ohuchi (1992), *Modularity of simple termination of term rewriting systems with shared constructors*, Theoretical Computer Science **103**, pp. 273-282.
11. A. Middeldorp (1989), *A sufficient condition for the termination of the direct sum of term rewriting systems*, Proc. of LICS'89, pp. 396-401.
12. A. Middeldorp (1990), *Modular properties of term rewriting systems*, Ph.D. Thesis, Free University, Amsterdam.
13. A. Middeldorp and Y. Toyama (1991), *Completeness of combinations of constructor systems*, Proc. of RTA'91, Lecture Notes in Computer Science **488**, pp. 188-199, Springer-Verlag. Also appears in J. Symb. Comp. **15**, pp. 331-348.
14. E. Ohlebusch (1994), *On the modularity of confluence of constructor-sharing term rewriting systems*, Proc. of CAAP'94, Lecture Notes in Computer Science **787**, pp. 261-275, Springer-Verlag.
15. D. Plump (1993), *Evaluation of functional expressions by hypergraph rewriting*, Ph.D. Thesis, University of Bremen.
16. M. Rusinowitch (1987), *On termination of the direct sum of term rewriting systems*, Information Processing Letters, IPL **26**, pp. 65-70.
17. J. Staples (1975), *Church-Rosser Theorems for Replacement Systems*, in J. Crosley (ed.), *Algebra and Logic*, Lecture Notes in Mathematics, Vol **450**, pp. 291-307, Springer-Verlag.
18. Y. Toyama (1987), *On the Church-Rosser property for the direct sum of term rewriting systems*, JACM **34**, pp. 128-143.
19. Y. Toyama (1987), *Counterexamples to termination for the direct sum of term rewriting systems*, Information Processing Letters, IPL **25**, pp. 141-143.
20. Y. Toyama, J.W. Klop and H.P. Barendse (1989), *Termination for the direct sum of left-linear term rewriting systems*, Proc. of RTA'89, Lecture Notes in Computer Science **355**, pp. 477-491, Springer-Verlag.