ソフトウェア基礎論　36−1
（1990.　9.　21）

構成子を共有する項書き換えシステムの単純停止性のモジュラ性

栗原 正仁　　大内 東
北海道大学工学部

項書き換えシステムは、その停止性を証明する単純化順序が項の集合上に存在するならば、単純停止性を持つといわれる。R_0とR_1を、構成子 (constructors) を共有し得るが、演算子 (defined symbols) を共有しない項書き換えシステムとする。ここで、構成子とは、書き換え規則の左辺の最左位置に生起し得ない関数記号、それ以外の関数記号は演算子である。本論文では、$R_0 \cup R_1$が単純停止性を持つのはR_0およびR_1ともに単純停止性を持つとき、かつ、そのときに限ることを示す。

MODULARITY OF SIMPLE TERMINATION OF TERM REWRITING SYSTEMS WITH SHARED CONSTRUCTORS

Masahito Kurihara　　Azuma Ohuchi
Department of Information Engineering, Hokkaido University
Sapporo 060, Japan

A term rewriting system is said to be simply-terminating if there exists a simplification ordering (on the set of terms) showing its termination. Let R_0 and R_1 be term rewriting systems which may share constructors but do not share defined symbols. Here, a constructor is a function symbol which cannot occur at the leftmost positions of the left-hand sides of rewrite rules; the rest of the function symbols are defined symbols. In this paper, we prove that $R_0 \cup R_1$ is simply-terminating if and only if both R_0 and R_1 are so.

1 Introduction

A term rewriting system (TRS) R is a computer program expressed as a set of rewrite rules. A property P of TRSs is *modular* [9] if, $R_0 \cup R_1$ has the property P if and only if both R_0 and R_1 have the same property. Much effort has been made for finding modular properties in the direct sum case $R_0 \oplus R_1$ in which R_0 and R_1 are prohibited from sharing any function symbols. Toyama proved the modularity of confluence [14], but refuted the modularity of termination [15]. Barendregt and Klop refuted the modularity of completeness (i.e., termination plus confluence)[15]. These refutations inspired several authors to find appropriate class of systems for which termination is modular. Rusinowitch [13] proved the modularity of termination of non-collapsing systems and non-duplicant systems, respectively. His results were further extended by Middeldorp [10]. Toyama, Klop and Barendregt [16] proved the modularity of completeness of left-linear systems. Kurihara and Kaji [6] and Kurihara and Ohuchi [8] proved the modularity of rpo-termination and the modularity of simple termination, respectively, where a TRS is *rpo-terminating* (*simply-terminating*) if there exists a recursive path ordering (a simplification ordering) [3] showing its termination.

There are at least three ways of extending the research. The first is the extension to conditional TRSs, currently being pursued by Middeldorp [11]. The second is the extension to *non*-direct sums which allow sharing some function symbols. The third is the new approach to the modularity, based on modular rewriting [7] by the family (rather than the union) of TRSs.

In this paper, we pursue the second approach. When systems share some function symbols, almost nothing has been known except the Dershowitz's pioneering work [2] based on commutation. The results were extended to equational TRSs by Bachmair and Dershowitz [1], and to fair termination by Porat and Francez [12]. Unfortunately, however, difficulty in establishing commutation restricts its full use.

We prove in this paper that the modularity of simple termination, which was proved for the direct sum case [8], remains true if the systems are allowed to share constructors. Here, we define *constructors* to be function symbols which are not allowed to occur at the leftmost (outermost) positions of the left-hand sides of rewrite rules; the rest of the function symbols are *defined symbols*. Although the improvement from the direct sum case might seem moderate, the proof provides careful and elegant generalization of the techniques in [8], and the result has greatly enhanced its utility, as seen in the following example.

Consider the following systems which share the constructors A and B:

$$R_0 = \{\ F(\text{A}, \alpha, \alpha) \rightarrow F(\text{B}, \alpha, \alpha)\ \}$$
$$R_1 = \{\ g(\text{B}) \rightarrow g(\text{A})\ \}$$

Both systems can be shown to be simply-terminating by the recursive path ordering (RPO) of Dershowitz or any other appropriate simplification orderings. Therefore, by our result, $R_0 \cup R_1$ is simply-terminating. The point here is that the direct application of RPO to $R_0 \cup R_1$ fails, because the first rule requires the precedence $\mathrm{A} \succ \mathrm{B}$, while the second requires $\mathrm{B} \succ \mathrm{A}$. (This implies that rpo-termination, which is modular in the direct sum case, is not modular in the constructor-sharing case.) Note also that this example refuses the application of the Dershowitz's result[1] [2] for restricted systems, because the first rule is nonlinear.

Unfortunately, the confluence, which is modular in the direct sum case [14], is not modular in the constructor-sharing case:

$$R_0 = \{F(\alpha, \alpha) \to \mathrm{A}, \quad F(\alpha, \mathrm{H}(\alpha)) \to \mathrm{B}\}$$
$$R_1 = \{g \to \mathrm{H}(g)\}$$

In this example,[2] R_0 and R_1, which share a constructor H, are confluent, but $R_0 \cup R_1$ is not, because $F(g, g)$ has two normal forms A and B.

Of course, as we have already stated, the completeness is not modular in the direct sum case. However, once our main result was established, it is obvious that the simple completeness (i.e., simple termination plus confluence) is modular, because R_0, R_1 and $R_0 \cup R_1$ are terminating and there is no critical pair between R_0 and R_1.

2 Formal Preliminaries

2.1 Term Rewriting Systems With Constructors

Let $\mathcal{V}$ be a set of *variables*, and $\mathcal{F}$ be a set of *function symbols*. We assume that $\mathcal{F}$ is partitioned into two disjoint sets $\mathcal{D}$ and $\mathcal{C}$. The elements in $\mathcal{D}$ are called *defined symbols*, and those in $\mathcal{C}$ are called *constructors*. Each function symbol may have variable arity, or may be restricted to a fixed arity. We denote the set of terms over $\mathcal{D}$, $\mathcal{C}$ and $\mathcal{V}$ by $\mathcal{T}(\mathcal{D},\mathcal{C},\mathcal{V})$, and the set of (ground) terms over $\mathcal{D}$ and $\mathcal{C}$ by $\mathcal{T}(\mathcal{D},\mathcal{C})$. We use $\mathcal{T}$ for $\mathcal{T}(\mathcal{D},\mathcal{C},\mathcal{V})$ if it yields no ambiguity. The *root* of a term t, notation $root(t)$, is f if t is of the form $f(t_1, \ldots, t_n)$; otherwise, it is t itself. If $f(\ldots, t_i, \ldots)$ is a term, f is the *parent* of the occurrence $root(t_i)$.

Let $\Box$ be an extra constant called a *hole*. We assume that the hole is a constructor. A term C over $\mathcal{F} \cup \{\Box\}$ and $\mathcal{V}$ is called a *context* on $\mathcal{F}$. When C is a context with n holes, $C[t_1, \ldots, t_n]$ denotes the result of replacing the holes by the terms $t_1, \ldots, t_n$ from left to right.

[1] Let R and S be terminating TRS's. If R is left-linear, S is right-linear, and there is no overlap between left-hand sides of R and right-hand sides of S, then $R \cup S$ also terminates.

[2] This example was borrowed from Huet [5].

A *term rewriting system* R on $\mathcal{T}$ is a set of rewrite rules of the form $\ell \rightarrow r$, where *$root(\ell)$ must be a defined symbol* and every variable occurring in r must also occur in ℓ. Note that this definition coincides with the ordinary one if $\mathcal{C} = \emptyset$. The single-step rewriting relation by R is denoted by $\rightarrow_R$. The transitive closure of a relation, say, $\rightarrow_R$ is denoted by $\rightarrow_R^+$. In this paper, we restrict the relation $\rightarrow_R$ to be defined only on the ground terms $\mathcal{T}(\mathcal{D},\mathcal{C})$. This is just for clarifying the discussions, and our major results are easily shown to hold for the more general case.

2.2 Combined Systems With Shared Constructors

Let $\mathcal{D}_0$ and $\mathcal{D}_1$ be two sets of defined symbols, and $\mathcal{C}$ be a set of constructors. We assume that $\mathcal{D}_0$, $\mathcal{D}_1$, and $\mathcal{C}$ are pairwise disjoint. The union of the systems R_0 on $\mathcal{T}(\mathcal{D}_0,\mathcal{C},\mathcal{V})$ and R_1 on $\mathcal{T}(\mathcal{D}_1,\mathcal{C},\mathcal{V})$, which is a term rewriting system on $\mathcal{T}(\mathcal{D}_0 \cup \mathcal{D}_1, \mathcal{C}, \mathcal{V})$, is called the *combined system with shared constructors* $\mathcal{C}$. In particular, if $\mathcal{C} = \emptyset$, then it is called the *direct sum system* [14]. In the rest of this paper, we assume that $R = R_0 \cup R_1$. For mnemotechnical reasons we will paint the function symbols: the defined symbols $\mathcal{D}_0$ in *black*, and $\mathcal{D}_1$ in *white*. Each occurrence of the constructors $\mathcal{C}$ is painted depending on the surrounding context: if the occurrence has no parent, it is transparent; otherwise, its color is the same as that of its parent. (The definition applies recursively if the parent is a constructor.) A term is *root-black* (resp. *root-white, root-transparent*) if its root symbol is black (resp. white, transparent). A term is black (resp. white, transparent) if every function symbol in it is black (resp. white, transparent); otherwise, it is *mixed*. To distinguish in print among them, defined symbols are printed in upper case $F, A, \ldots$ if they are black, and in lower case $g, b, \ldots$ if they are white. Constructors are printed in small capital case $\textsc{h}$, $\textsc{c}, \ldots$. Variables are written in Greek letters $\alpha, \beta, \ldots$.

Definition 2.1 An *alien* in a term t is a nonvariable proper subterm u of t which is maximal with respect to the 'subterm' relation, such that $root(t)$ and $root(u)$ are in distinct colors.

We write $t \equiv C[\![t_1, \ldots, t_n]\!]$ if $t_1, \ldots, t_n$ are all the aliens in t (from left to right) and C is the context obtained by replacing each alien by a hole. For example, the term $t \equiv F(F(b, A), \textsc{h}(F(g(\textsc{h}(A), b), \textsc{c})))$ has the two aliens b and $g(\textsc{h}(A), b)$; thus $t \equiv C[\![b, g(\textsc{h}(A), b)]\!]$, where $C \equiv F(F(\Box, A), \textsc{h}(F(\Box, \textsc{c})))$. Note that the two occurrences of the constructors $\textsc{h}$ and $\textsc{c}$ in C are painted in black, while a single occurrence $\textsc{h}$ in the second alien is painted in white.

Since each alien t_i in t may have aliens in itself, we can identify a hierarchy of aliens:

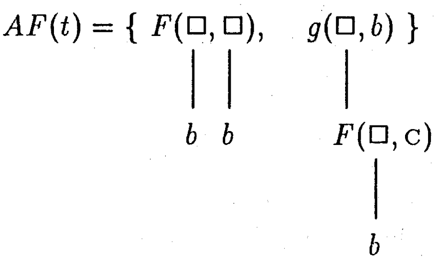
Definition 2.2 The *alien tree* $AT(t)$ of a root-black or root-white term t is the tree defined below. Note that each node is either a black context or a white context:

1. if t has no alien, then $AT(t)$ consists of a single node t, the root of the tree;
2. if $t \equiv C[\![t_1, \ldots, t_n]\!]$ $(n > 0)$, then $AT(t)$ consists of the root C and the subtrees $AT(t_i)$, $1 \leq i \leq n$.

Definition 2.3 The *alien forest* $AF(t)$ of a term t is the singleton set $\{AT(t)\}$, if t is either root-black or root-white; otherwise (if t is root-transparent), it is the set $\{AT(t_i) | 1 \leq i \leq n\}$ of the alien trees of the aliens $t_1, \ldots, t_n$ in t.

The *rank* of a term t, notation $rank(t)$, is 1 + the height of the highest alien tree in $AF(t)$. If $AF(t) = \emptyset$, then we define $rank(t) = 0$.

For example, the alien forest of a root-transparent term $t \equiv \mathrm{H}(F(b, b), g(F(b, \mathrm{C}), b))$ with rank 3 is depicted below:



The rank of a term is never increased by rewriting:

Lemma 2.4 *If $s \rightarrow_R t$ then $rank(s) \geq rank(t)$.*

Proof. Routine, using the induction on $rank(s)$. □

For example, consider the two systems:

$R_0 = \{F(\mathrm{H}(\alpha)) \rightarrow \mathrm{H}(F(\alpha))\}$

$R_1 = \{g(\alpha) \rightarrow \alpha\}$

Then we have

$$s \equiv F(\mathrm{H}(g(A))) \rightarrow_R t \equiv \mathrm{H}(F(g(A))) \rightarrow_R u \equiv \mathrm{H}(F(A))$$

and $rank(s) = rank(t) = 3 > 1 = rank(u)$. Note that it is essential for Lemma 2.4 that the outermost transparent context make no contribution to the calculation of the rank; it is invisible; otherwise, we might have had $rank(t) = 4$ in this example.

Definition 2.5 A subterm u of s is an *inner* subterm if it is a subterm of some alien in s; otherwise, u is an *outer* subterm. A reduction $s \rightarrow_R t$ is an *inner reduction* if the redex is an inner subterm of s; otherwise, it is an *outer reduction.*

3 Modularity of Simple Termination

3.1 Simple Termination

A partial ordering $\succ$ on $\mathcal{T}$ is *monotonic* if it possesses the *replacement* property: $s \succ t$ implies $f(\ldots, s, \ldots) \succ f(\ldots, t, \ldots)$. A monotonic partial ordering $\succ$ is a *simplification ordering* [3] if it possesses the *subterm* property, $f(\ldots, t, \ldots) \succ t$, and the *deletion* property, $f(\ldots, t, \ldots) \succ f(\ldots, \ldots)$.

A term rewriting system R on $\mathcal{T}$ is *simply-terminating* if there exists a simplification ordering $\succ$ on $\mathcal{T}$ such that $\rightarrow_R \subseteq \succ$. A term rewriting system R is *terminating* if there is no infinite rewrite sequence $t_0 \rightarrow_R t_1 \rightarrow_R \cdots$. It is known that every simplification ordering $\succ$ is well-founded [3], so there is no infinite reduction sequence $t_0 \succ t_1 \succ \cdots$. Therefore:

Theorem 3.1 (Dershowitz) *A simply-terminating system is terminating.*

Recall that the purpose of this paper is to prove that the combined system $R_0 \cup R_1$ with shared constructors is simply-terminating if and only if both R_0 and R_1 are so.

Definition 3.2 The relations $\rightarrow_{sub}$ and $\rightarrow_{del}$ on $\mathcal{T}$ are defined below:

$$s \rightarrow_{sub} t \quad \text{iff} \quad s \equiv C[f(\ldots, u, \ldots)] \text{ and } t \equiv C[u],$$
$$s \rightarrow_{del} t \quad \text{iff} \quad s \equiv C[f(\ldots, u, \ldots)] \text{ and } t \equiv C[f(\ldots\ , \ldots)],$$

for some C, f, and u; In both definitions, the occurrence $f(\ldots, u, \ldots)$ in s is called a *redex,* and we use the terminology 'inner' and 'outer' in the same manner as in Definition 2.5.

The following lemma, in which R can be any TRS (not restricted to a combined system), characterizes the simple termination:

Lemma 3.3 *A system R is simply-terminating if and only if $(\rightarrow_R \cup \rightarrow_{sub} \cup \rightarrow_{del})^+$ is irreflexive.*

The proof is easy, and given in [8]. To prove the *if* part, verify that $(\rightarrow_R \cup \rightarrow_{sub} \cup \rightarrow_{del})^+$ is a simplification ordering including $\rightarrow_R$. To prove the *only-if* part, let $\succ$ be a simplification ordering including $\rightarrow_R$, and show that $\rightarrow_R \cup \rightarrow_{sub} \cup \rightarrow_{del} \subseteq \succ$.

We use the following notations:

$$\rightarrow_{0sd} = \rightarrow_{R_0} \cup \rightarrow_{sub} \cup \rightarrow_{del}$$
$$\rightarrow_{01sd} = \rightarrow_{R_0} \cup \rightarrow_{R_1} \cup \rightarrow_{sub} \cup \rightarrow_{del}$$

3.2 Alien Replacement

The following definition was introduced just for technical reasons. The idea comes from the intention that, when there is a cyclic sequence

$$s \rightarrow_{01sd} \cdots \rightarrow_{01sd} s$$

of root-black *mixed-color* terms, we want to construct a cyclic sequence

$$\rho(s) \rightarrow_{0sd} \cdots \rightarrow_{0sd} \rho(s)$$

of *black* terms, thus uncovering the contradiction to the irreflexivity of $\rightarrow_{0sd}{}^{+}$.

Definition 3.4 Let $\mathrm{E} \in \mathcal{C}$ be the distinguished, variable-arity constructor, not used in the rules of $R_0 \cup R_1$. Consider a root-black finite sequence

$$s_0 \rightarrow_{01sd} s_1 \rightarrow_{01sd} \cdots \rightarrow_{01sd} s_m,$$
$$root(s_i) \in \mathcal{D}_0 \ (0 \leq i \leq m),$$

and let A and O be the finite sets of the aliens and the outer subterms, respectively, occurring in the sequence. We assume that $rank(s_0)$ is the minimal rank of the terms that may initiate cyclic sequences, so that there is no cyclic sequence $t \rightarrow_{01sd} \cdots \rightarrow_{01sd} t$ starting from t if $rank(t) < rank(s_0)$. Then the *alien replacement* ρ for this sequence is defined to be the mapping from $\mathrm{A} \cup \mathrm{O}$ to $\mathcal{T}(\mathcal{D}_0, \mathcal{C})$ determined inductively as follows:

1. $\rho(t) = \mathrm{E}(\rho(t'_1), \ldots, \rho(t'_n))$ if $t \in A$ (i.e., $root(t) \in \mathcal{D}_1$), where $\{t'_1, \ldots, t'_n\}$ is the finite set of the terms t' such that for some k $(0 \leq k < m)$ the reduction
$$s_k \equiv C[\![\ldots, t, \ldots]\!] \rightarrow_{01sd} C[\ldots, t', \ldots] \equiv s_{k+1}$$
is inner and $t \rightarrow_{01sd} t'$. (In other words, t' is an occurrence of a direct 'descendant' of the alien t.)
2. $\rho(t) = F(\rho(t_1), \ldots, \rho(t_n))$ if $t \in \mathrm{O}$ (i.e., $root(t) \in \mathcal{D}_0 \cup \mathcal{C}$), where $t \equiv F(t_1, \ldots, t_n)$.

Remark: Since this definition is recursive, readers might wonder if the recursion eventually terminates. To answer the question, first note that when the first branch of the definition applies, $t \rightarrow_{01sd} t'_i \in \mathrm{A} \cup \mathrm{O}$ is true. When the second branch applies, $t \rightarrow_{01sd} t_i \in \mathrm{A} \cup \mathrm{O}$ is true because $t \rightarrow_{sub} t_i$. Thus, the recursion starting from $\rho(t)$ will terminate, because there is no infinite decreasing sequence $t \rightarrow_R \cdots$ of the arguments of ρ. This can be easily verified in two stages; firstly, the case $rank(t) < rank(s_0)$ is trivial from the assumption and the finiteness of $\mathrm{A} \cup \mathrm{O}$; then the case $rank(t) = rank(s_0)$ should be obvious, noting that in this case, $t \in \mathrm{O}$.

Lemma 3.5 *Let* $t \equiv C[t_1, \ldots, t_n]$, *$C$ being a context on* $\mathcal{D}_0 \cup \mathcal{C}$. *Then*

$$\rho(t) = C[\rho(t_1), \ldots, \rho(t_n)].$$

Proof. Obvious. □

Lemma 3.6 *Let ρ be the alien replacement for the root-black sequence*

$$s_0 \rightarrow_{01sd} \cdots \rightarrow_{01sd} s_m.$$

Then

$$\rho(s_0) \rightarrow_{0sd} \cdots \rightarrow_{0sd} \rho(s_m).$$

Proof. We show that $\rho(s_k) \rightarrow_{0sd} \rho(s_{k+1})$, $0 \leq k < m$.

CASE 1: $s_k \rightarrow_{01sd} s_{k+1}$ is an outer reduction.

Since s_k is root-black, we have $s_k \rightarrow_{0sd} s_{k+1}$; and since s_{k+1} is also root-black,

- if $s_k \to_{R_0} s_{k+1}$ then $\rho(s_k) \to_{R_0} \rho(s_{k+1})$. (Apply to $\rho(s_k)$ the same rewrite rule that has reduced s_k. This is possible even if the rule is non-left-linear, because if $s_k \equiv C[\![\ldots, t, \ldots, t, \ldots]\!]$ then $\rho(s_k) \equiv C[\![\ldots, \rho(t), \ldots, \rho(t), \ldots]\!]$, where t and $\rho(t)$, respectively, are completely 'covered' by some variables in the left-hand side of the rule.)
- if $s_k \to_{sub} s_{k+1}$ then $\rho(s_k) \to_{sub} \rho(s_{k+1})$.
- if $s_k \to_{del} s_{k+1}$ then $\rho(s_k) \to_{del} \rho(s_{k+1})$.

Therefore, $\rho(s_k) \to_{0sd} \rho(s_{k+1})$.

CASE 2: $s_k \to_{01sd} s_{k+1}$ is an inner reduction.
Assume, without loss of generality, that the first alien was reduced:

$$s_k \equiv C[\![t_1, t_2, \ldots, t_n]\!],$$
$$s_{k+1} \equiv C[t'_1, t_2, \ldots, t_n],$$
$$\rho(s_k) \equiv C[\rho(t_1), \rho(t_2), \ldots, \rho(t_n)],$$
$$\rho(s_{k+1}) \equiv C[\rho(t'_1), \rho(t_2), \ldots, \rho(t_n)],$$
$$t_1 \to_{01sd} t'_1.$$

Since, by definition, ρ is constructed such that $\rho(t_1)$ contains $\rho(t'_1)$ as an argument, we have that $\rho(t_1) \to_{sub} \rho(t'_1)$. Therefore, $\rho(s_k) \to_{sub} \rho(s_{k+1})$. □

Example 3.7 Let $\mathcal{D}_0 = \{F, A\}$, $\mathcal{D}_1 = \{g, b\}$, $\mathcal{C} = \{\mathrm{H}, \mathrm{E}\}$, and consider the following systems sharing the constructor H:

$$R_0 = \{\ F(\mathrm{H}(\alpha), F(\alpha, \beta)) \to F(\beta, \alpha)\ \},$$
$$R_1 = \{\ g(\alpha) \to \mathrm{H}(\alpha)\ \}.$$

Consider a root-black sequence

$$\begin{array}{rcll} & s_0 & \equiv & F(g(b), F(b, A)) \\ \to_{R_1} & s_1 & \equiv & F(\mathrm{H}(b), F(b, A)) \\ \to_{R_0} & s_2 & \equiv & F(A, b) \end{array}$$

where the first reduction is inner and the second outer. The set of the aliens is $\mathrm{A} = \{g(b), b\}$, and the set of the outer subterms is $\mathrm{O} = \{s_0, s_1, s_2, A, F(b, A), \mathrm{H}(b)\}$. Since $g(b) \to_{01sd} \mathrm{H}(b)$ is inner and there is no descendant of b occurring in the sequence, we see that

$$\rho(b) = \mathrm{E},$$
$$\rho(g(b)) = \mathrm{E}(\mathrm{H}(\mathrm{E})).$$

Verify that

$$\begin{array}{rcl} \rho(s_0) & \equiv & F(\mathrm{E}(\mathrm{H}(\mathrm{E})), F(\mathrm{E}, A)) \\ \to_{sub}\ \rho(s_1) & \equiv & F(\mathrm{H}(\mathrm{E}), F(\mathrm{E}, A)) \\ \to_{R_0}\ \rho(s_2) & \equiv & F(A, \mathrm{E}). \end{array}$$

Note that the first occurrence of E introduced in $\rho(s_0)$ was removed by $\to_{sub}$, making $\rho(s_1)$ reducible by $\to_{R_0}$.

3.3 Modularity of Simple Termination

The following lemma is essentially the proof of our main result:

Lemma 3.8 *If R_0 and R_1 are simply terminating, then $R_0 \cup R_1$ is simply terminating.*

Proof. Since R_0 is simply-terminating, $\rightarrow_{0sd}{}^+$ is irreflexive (from Lemma 3.3). Since $\rightarrow_{R_0 \cup R_1} = \rightarrow_{R_0} \cup \rightarrow_{R_1}$, we have to show that $\rightarrow_{01sd}{}^+$ is irreflexive. Assume that $\rightarrow_{01sd}{}^+$ is not irreflexive. Then there is a cyclic sequence $s \equiv s_0 \rightarrow_{01sd} \cdots \rightarrow_{01sd} s_n \equiv s$, where we can assume without loss of generality that s is root-black, and $rank(s)$ is the minimal rank of the terms that initiate cyclic sequences. Thus, since the assumptions in Definition 3.4 are fulfilled, we can define the alien replacement ρ for this cyclic sequence. Then, from Lemma 3.6, we have that $\rho(s) \rightarrow_{0sd} \cdots \rightarrow_{0sd} \rho(s)$. This contradicts the irreflexivity of $\rightarrow_{0sd}{}^+$. □

Now, our main result, the modularity of simple termination, is established:

Theorem 3.9 *$R_0 \cup R_1$ is simply-terminating if and only if both R_0 and R_1 are so.*

Proof. The *only-if* part is trivial. The *if* part is direct from Lemma 3.8. □

As seen in Section 1, confluence is not modular when constructors are shared, and completeness is not modular even when no function symbols are shared. However, simple completeness (i.e., simple termination plus confluence) is modular even if constructors are shared:

Corollary 3.10 *$R_0 \cup R_1$ is simply-complete if and only if both R_0 and R_1 are so.*

Proof. Obvious from the remark in Section 1 and the fact that a terminating system is confluent iff every critical pair is convergent. □

Acknowledgments

The authors would like to thank Y. Toyama for his helpful comments. This work is partially supported by the Grants from Ministry of Education, Science and Culture of Japan, #02249104.

References

[1] L. Bachmair and N. Dershowitz, Commutation, transformation, and termination, in: *Proc. CADE '86*, Lecture Notes in Computer Science 230 (Springer, Berlin, 1986) 5–20.

[2] N. Dershowitz, Termination of linear rewriting systems (Preliminary version), in: *Lecture Notes in Computer Science* 115 (Springer, Berlin, 1981) 448–458.

[3] N. Dershowitz, Orderings for term-rewriting systems, *Theoret. Comput. Sci.* 17 (1982) 279–301.

[4] N. Dershowitz, Termination of rewriting, *J. Symbolic Computation* 3 (1987) 69–116.

[5] G. Huet, Confluent reductions: abstract properties and applications to term rewriting systems, *J. ACM* 27 (1980) 797–821.

[6] M. Kurihara and I. Kaji, Termination of the direct sum of rpo-terminating term rewriting systems, *Trans. IEICE Japan* E71 (1988) 975–977.

[7] M. Kurihara and I. Kaji, Modular term rewriting systems and the termination, *Inform. Process. Lett.* 34 (1990) 1–4.

[8] M. Kurihara, and A. Ohuchi, Termination of direct sum of simply-terminating term rewriting systems, in: *Proc. InfoJapan '90* (Elsevier, Amsterdam, 1990) to appear.

[9] A. Middeldorp, Modular aspects of properties of term rewriting systems related to normal forms, in: *Proc. RTA '89*, Lecture Notes in Computer Science 355 (Springer, Berlin, 1989) 263–278.

[10] A. Middeldorp, A sufficient condition for the termination of the direct sum of term rewriting systems, in: *Proc. 4th IEEE symp. Logic in Computer Science* (1989) 396–401.

[11] A. Middeldorp, Termination of disjoint unions of conditional term rewriting systems, Vrije Univ., Amsterdam (1990), submitting.

[12] S. Porat and N. Francez, Full-commutation and fair-termination in equational (and combined) term-rewriting systems, in: *Proc. CADE '86*, Lecture Notes in Computer Science 230 (Springer, Berlin, 1986) 21–41.

[13] M. Rusinowitch, On termination of the direct sum of term-rewriting systems, *Inform. Process. Lett.* 26 (1987) 65–70.

[14] Y. Toyama, On the Church-Rosser property for the direct sum of term rewriting systems, *J. ACM* 34 (1987) 128–143.

[15] Y. Toyama, Counterexamples to termination for the direct sum of term rewriting systems, *Inform. Process. Lett.* 25 (1987) 141–143.

[16] Y. Toyama, J. W. Klop and H. P. Barendregt, Termination for the direct sum of left-linear term rewriting systems (preliminary draft), in: *Proc. RTA '89*, Lecture Notes in Computer Science 355 (Springer, Berlin, 1989) 477–491.