Simplification Ordering for Higher-Order Rewrite Systems

MUNEHIRO IWAMI† and YOSHIHITO TOYAMA†

Simplification orderings, like the recursive path ordering and the improved recursive decomposition ordering, are widely used for proving the termination property of term rewriting systems. The recursive path ordering is known as the most useful simplification ordering. Recently Jouannaud and Rubio extended the recursive path ordering to higher-order rewrite systems by introducing an ordering on type structure. In this paper, we define the notion of simplification orderings for higher-order rewrite systems. Further, we redefine the recursive path ordering for higher-order rewrite systems and compare our approach to that of Jouannaud and Rubio.

1. Introduction

Higher-order rewriting is a natural extension of first-order term rewriting to reason with higher-order equations[7),9)]. An important application of higher-order rewrite systems (HRSs) is to model the basic mechanisms of higher-order functional programming languages like ML and Haskell and of higher-order theorem provers like TPS and Isabelle[9)].

Termination is one of the most important properties of higher-order rewriting[4)], like first-order rewriting[1),3)]. It is well known that termination is undecidable in general even for first-order rewriting. Thus, several semi-automated techniques for proving termination of term rewriting systems (TRSs) have been successfully developed. In particular, simplification orderings, like the recursive path ordering (RPO)[2),10),11)], are widely used for first-order rewriting.

Recently Jouannaud and Rubio extended the recursive path ordering on first-order terms, called algebraic terms, to that on higher-order terms[5),6)] by using a first-order interpretation on λ-terms. They showed that this ordering can prove termination of several interesting examples. However, in this recursive path ordering, two higher-order terms have to be compared by type first, then by root function symbol, before the comparison can proceed recursively on the arguments. This unnatural priority between type and function symbol restricts their ordering to only on type compatible terms.

In this paper we introduce the notion of simplification orderings on algebraic terms and propose a new powerful recursive path ordering on higher-order terms, called higher-order recursive path ordering (HRPO). Though our approach was inspired by the first-order interpretation method described in Jouannaud and Rubio[5)], we develop the higher-order recursive path ordering within a more natural framework of a simplification ordering. Our key idea in higher-order recursive path ordering is the concept of *envelopes* for typed terms that allows us to treat higher-order variables as function symbols. We clarify that the priority between type and function symbol introduced in Jouannaud and Rubio is not essential and any partial ordering on types and function symbols can be used freely to define the higher-order recursive path ordering. Thus we can remove the type compatible term limitation in Jouannaud and Rubio's ordering.

In section 2 we give the basic notations. Section 3 presents the definition of simplification orderings on algebraic terms. We define the higher-order recursive path ordering in section 4. Section 5 develops the technique of envelopes, and section 6 compares our approach to that of Jouannaud and Rubio.

2. Preliminaries

We mainly follow the basic notations in the literatures[5),7),9),12)].

2.1 Abstract Reduction Systems

An *abstract reduction system* (ARS for short) is a pair $\langle A, \rightarrow \rangle$ consisting of a set A and a binary relation $\rightarrow \subseteq A \times A$. The relation $\rightarrow^+$ is the transitive closure of $\rightarrow$, the relation $\rightarrow^*$ is the reflexive and transitive closure of $\rightarrow$, and the relation $\leftrightarrow^*$ is reflexive, symmetric and transitive closure of $\rightarrow$. If there is no element $b \in A$ such that $a \rightarrow b$, then we say $a \in A$ is a

† School of Information Science, Japan Advanced Institute of Science and Technology, Hokuriku

normal form (with respect to $\rightarrow$). If $b \in A$ is a normal form such that $a \rightarrow^* b$ then we say that b is a normal form of a. The binary relation $\rightarrow$ on A is *terminating* if there is no infinite sequence $a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow \ldots$ of elements in A. We say that ARS $\langle A, \rightarrow \rangle$ is *terminating* if the binary relation $\rightarrow$ on A is terminating.

A binary relation on a set A is called a *quasi-ordering* on A if it is a reflexive and transitive on A. The quasi-ordering is usually denoted by . A binary relation on A is called a *partial ordering* on A if it is irreflexive and transitive on A. The partial ordering is usually denoted by $>$. A partial ordering is *total* if for any a, $b \in A$ we have either $a > b$ or $a = b$ or $a < b$. A partial ordering $>$ on A is *well-founded* if $>$ has no infinite descending sequences, i.e., there is no sequence of the form $a_0 > a_1 > a_2 > \ldots$ of elements in A. A partial ordering $>$ on A is a *well-partial ordering* if for every infinite sequence $(a_i)_i$ of elements in A there are indexes $k < l$ such that $a_k \leq a_l$. Given a well-partial ordering $>$ on A, $\langle A, > \rangle$ is called a *well-partially ordered set*.

Given a binary relation $>$ on A, the *multiset extension* of $>$ is denoted by $>^{mul}$ and the *lexicographic extension* of $>$ is denoted by $>^{lex}$, following Baader and Nipkow[1)].

2.2 Simply Typed λ-Terms

Let $\mathcal{S}$ be a set of *basic types* (or *sorts*), b, b', b'',.... The set $\mathcal{T_S}$ of *types* is generated from the set of basic types by the *constructor* $\rightarrow$ as follows: $\mathcal{T_S} := \mathcal{S} \mid \mathcal{T_S} \rightarrow \mathcal{T_S}$. We use σ, τ and ρ to denote types. We use the abbreviation $\sigma_1 \rightarrow \ldots \rightarrow \sigma_n \rightarrow \tau$ for $\sigma_1 \rightarrow (\ldots \rightarrow (\sigma_n \rightarrow \tau) \ldots)$. If b is a basic type then σ_i $(i = 1,\ldots,n)$ is called *input type* and b is called *output type* of $\sigma_1 \rightarrow \ldots \rightarrow \sigma_n \rightarrow b$.

We assume a set of *variables* $\mathcal{X}_\tau$ and a set of *constants* $\mathcal{F}_\tau$ for each type $\tau \in \mathcal{T_S}$, where $\mathcal{X}_\tau \cap \mathcal{X}_{\tau'} = \mathcal{F}_\tau \cap \mathcal{F}_{\tau'} = \emptyset$ if $\tau \neq \tau'$. The set of all variables is $\mathcal{X} = \cup_{\tau \in \mathcal{T_S}} \mathcal{X}_\tau$, which is disjoint from the set of all constants $\mathcal{F} = \cup_{\tau \in \mathcal{T_S}} \mathcal{F}_\tau$. $\mathcal{F}$ is called *signature*. If $f{:}\sigma_1 \rightarrow \ldots \rightarrow \sigma_n \rightarrow b \in \mathcal{F}$ and b is a basic type then $arity(f) = n$. Arbitrary variables are denoted by x, y, z ..., free variables by upper case letters F, G, X, ... and constants by a, c, d, e, ...

The set of *untyped λ-terms* is generated from $\mathcal{F}$ and $\mathcal{X}$ according to the grammar: $\mathcal{T} := \mathcal{X} \mid \mathcal{F} \mid (\lambda\, \mathcal{X}.\, \mathcal{T}) \mid (\mathcal{T}\mathcal{T})$. Terms are denoted by l, r, s, t, ... The application of s to t is denoted by (st). We write $s(t_1,\ldots,t_n)$ for $(\ldots(st_1)\ldots t_n)$. We use $FV(t)$ for the set of *free variables* and $BV(t)$ for the set of *bound variables* of t. We may assume that bound variables are different from free ones. Further, we assume that for any free variable $X : \sigma_1 \rightarrow \ldots \rightarrow \sigma_n \rightarrow b$, b is a basic type.

A *type judgment* stating that t is of type τ is written as $t : \tau$. The following rules inductively define the set of *simply typed λ-terms* $\mathcal{T}(\mathcal{F}, \mathcal{X})$.

- $x \in \mathcal{X}_\tau$ implies $x : \tau$.
- $c \in \mathcal{F}_\tau$ implies $c : \tau$.
- $s : \sigma \rightarrow \tau$ and $t : \sigma$ imply $(st) : \tau$.
- $x : \sigma$ and $s : \tau$ imply $(\lambda\, x.s) : \sigma \rightarrow \tau$.

In the rest of this paper, simply typed λ-terms are written as *terms*. A term is *ground* if it contains no free variables. $\mathcal{T}(\mathcal{F})$ denotes the set of ground terms.

Substitutions are written as in $\{x_1 \leftarrow t_1,\ldots,x_n \leftarrow t_n\}$ where term t_i is assumed different from variable x_i and x_i and t_i have the same type $(i = 1, \ldots, n)$. We use the letter θ for substitutions. Substitutions behave as endomorphisms defined on free variables. Letting $\theta = \{x_1 \leftarrow t_1,\ldots, x_n \leftarrow t_n\}$, $dom(\theta)$ denotes the set $\{x_1,\ldots,x_n\}$ and $range(\theta)$ denotes the set $\{t_1,\ldots,t_n\}$. A substitution θ is *ground* if $range(\theta) \subseteq \mathcal{T}(\mathcal{F})$.

2.3 η-Long β-Normal Forms

Two rules originate from the λ-*calculus*, β-*reduction* and η-*expansion*:

$(\lambda\, x.s)(t) \rightarrow_\beta s\ \{x \leftarrow t\}$,

$s \rightarrow_\eta (\lambda\, x.s(x))$ if $s{:}\sigma \rightarrow \tau$, $x{:}\sigma \notin FV(s)$ and s is not an abstraction.

The simply typed λ-calculus is confluent and terminating with respect to β-reductions and η-expansions. Given a term s, we denote by $s\!\downarrow$ its unique *η-long β-normal form* (*η-long β-normal term*), defined as the β-normal form of its η-normal form. We say shortly that s is *normalized* when s is in η-long β-normal form. $\mathcal{T}(\mathcal{F}, \mathcal{X})\!\downarrow$ denotes the set of normalized terms. $\mathcal{T}(\mathcal{F})\!\downarrow$ denotes the set of ground normalized terms.

A substitution θ is *normalized* if $range(\theta) \subseteq \mathcal{T}(\mathcal{F}, \mathcal{X})\!\downarrow$. In the rest of this paper, substitution θ is normalized.

We suppose that for every basic type b there is a constant of type b not occurring $\mathcal{F}$ that is denoted by $\Box$ (called *hole*). A *context* is a term with occurrences of $\Box$. A context with only one occurrence of $\Box$ is denoted by $C[\ \ ]$. If t is a term then $C[t]$ denotes the result of replacing the hole in $C[\ \ ]$ by t.

Normalized terms have one of the following two forms[5)] : $(\lambda\, x.s)$ for some normalized term

s, or $\alpha(s_1,\ldots,s_n)$ for some $\alpha \in \mathcal{F} \cup \mathcal{X}$ and normalized terms $s_1, \ldots, s_n$.

From now on we assume that each bound variable in normarized terms has a basic type. This restriction is neccesary to gurantee the stability of ground substitutions (See example 18).

Let $C[s]$ and t be normalized terms such that s and t have the same type. Then, $C[t]$ is normalized.

2.4 Higher-Order Rewrite Systems

A normalized term t is called a *pattern* if every free occurrence of a variable X is in a subterm $X(u_1,\ldots,u_n)$ of t, such that $u_1,\ldots,u_n$ are η-equivalent to a list of distinct bound variables. Examples of patterns are $\lambda x.a(x)$, F, and $\lambda xy.F(x,y)$. Examples of non-patterns are $F(d)$, $\lambda x.F(x,x)$ and $\lambda x.G(H(x))$.

A *rewrite rule* is a pair $l \to r$ such that l and r are normalized terms with the same basic type, l is not $\beta\eta$-equivalent to free variable, l is a pattern and $FV(l) \supseteq FV(r)$. A *higher-order rewrite system* (HRS) is a set of rewrite rules. The letter $\mathcal{R}$ denotes a higher-order rewrite system. Then, the restriction $FV(l) \supseteq FV(r)$ is preserved under substitution, i.e., for any substitution θ, $FV(l) \supseteq FV(r)$ implies $FV(l\theta\downarrow) \supseteq FV(r\theta\downarrow)$ holds[7].

Given a higher-order rewrite system $\mathcal{R}$, a normalized term s is rewritten to a term t with respect to $\mathcal{R}$, written $s \to_{\mathcal{R}} t$, if $s = C[l\theta\downarrow]$ and $t = C[r\theta\downarrow]$ for some $l \to r \in \mathcal{R}$, $C[\ \ ]$ and $\theta:\mathcal{X} \to \mathcal{T}(\mathcal{F},\mathcal{X})\downarrow$. Note that t is normalized since s is so.

The *status* is a function $status:\mathcal{F} \to \{mult, lex\}$. Thus every constant has one of the following statuses: *mult* (the arguments will be compared as multiset), *lex* (lexicographical comparison from left to right).

3. Simplification Ordering for Higher-Order Rewrite Systems

In this section, we introduce the notion of simplification orderings for higher-order rewrite systems.

Definition 1 Let $\mathcal{L} = \{\lambda_{\tau\to(\sigma\to\tau)} \mid \sigma, \tau \in \mathcal{T}_S\}$ and $\mathcal{B} = \{c_\sigma \mid \sigma \in \mathcal{T}_S\}$ where $\lambda_{\tau\to(\sigma\to\tau)}:\tau \to (\sigma \to \tau)$ and $c_\sigma:\sigma$ are fresh constants. We define the new signature $\lambda\mathcal{F} = \mathcal{F} \cup \mathcal{L} \cup \mathcal{B}$. *Algebraic terms* are typed terms over $\lambda\mathcal{F}$ obtained from normalized terms over $\mathcal{F}$ through the following interpretation.

Definition 2 The *interpretation function* $\|\ \ \|$ from normalized terms over the signature $\mathcal{F} \cup \mathcal{B}$ to algebraic typed terms over the signature $\lambda\mathcal{F}$ is defined by:
$\|(\lambda x.s):\sigma\to\tau\| = \lambda_{\tau\to(\sigma\to\tau)}(\|s\{x \leftarrow c_\sigma\}\|)$.
$\|\alpha(s_1,\cdots,s_n)\| = \alpha(\|s_1\|,\cdots,\|s_n\|)$ if $\alpha \in \mathcal{F} \cup \mathcal{B} \cup \mathcal{X}$.
Then, the set of algebraic terms is $\mathcal{T}(\lambda\mathcal{F},\mathcal{X}) = \{\| t \| \mid t \in \mathcal{T}(\mathcal{F},\mathcal{X})\downarrow\}$ and the set of ground algebraic terms is $\mathcal{T}(\lambda\mathcal{F}) = \{\| t \| \mid t \in \mathcal{T}(\mathcal{F})\downarrow\}$.

Lemma 3 Let $C[s] \in \mathcal{T}(\mathcal{F})\downarrow$ and $s \in \mathcal{T}(\mathcal{F},\mathcal{X})\downarrow$. Let $\theta = \{x_1 \leftarrow c_{\sigma_1},\ldots,x_n \leftarrow c_{\sigma_n}\}$ where $x_i:c_{\sigma_i}$ $(1 \le i \le n)$ and $FV(s) = \{x_1,\ldots,x_n\}$. Then, $\| C[s] \| = \| C \|[\| s\theta \|]$.
Proof. It is straightforward by the definition of the interpretation function. □

$Fun(t)$ denotes the set of constants in an algebraic term t. The size $\mid t \mid$ is defined as the number of symbols occurring in t.

Definition 4 The *root symbol* of an algebraic term is defined by $top(\alpha(s_1,\ldots,s_m)) = \alpha$ if $\alpha \in \lambda\mathcal{F} \cup \mathcal{X}$. Note that an algebraic term $\alpha(s_1,\ldots,s_n)$ has a basic type if and only if $\alpha \notin \mathcal{L}$. An algebraic term s_i is called an *immediate subterm* of an algebraic term $s = \alpha\,(s_1, \ldots, s_i, \ldots, s_n)$. Then, $st(s) = \langle\, s_1, \ldots, s_i, \ldots, s_n \,\rangle$ and $st^m(s) = \{s_1, \ldots, s_i, \ldots, s_n\}$ denote the sequence and the multiset of immediate subterms of s, respectively. The *subterm relation*, denoted by $\rhd$, is the transitive closure of the immediate subterm relation. A partial ordering $>$ on algebraic terms has the *subterm property* if for any $s_i \in st^m(s)$, $s > s_i$ holds.

Definition 5 Let $\succ$ be a partial ordering on $\lambda\mathcal{F}$. A partial ordering $>$ is a *simplification ordering* on $\mathcal{T}(\lambda\mathcal{F})$ if it possesses the following three properties:
(1) $s > t$ implies $\alpha\,(u_1, \ldots, s, \ldots, u_n) > \alpha\,(u_1, \ldots, t, \ldots, u_n)$ for $\alpha \in \lambda\mathcal{F}$. (the *replacement property*),
(2) $s > s_i$ for any $s_i \in st^m(s)$ (the *subterm property*),
(3) $\alpha(u_1,\ldots,u_n) > \beta(u_{i1},\ldots,u_{im})$ if $\alpha, \beta \in \lambda\mathcal{F}$, $\alpha \succ \beta$, $1 \le i1 < \ldots < im \le n$, $arity(\alpha) = n$ and $arity(\beta) = m$.

Definition 6 Let $\succ$ be a partial ordering on $\lambda\mathcal{F}$. The *homeomorphic embedding relation*

$\rhd_{emb}$ on $\mathcal{T}(\lambda\mathcal{F})$ is defined inductively as follows:
$s = \alpha\ (s_1, \ldots, s_n) \rhd_{emb} \beta\ (t_1, \ldots, t_m) = t$ ($arity(\alpha) = n$ and $arity(\beta) = m$)
if and only if
(1) $\alpha \succ \beta$ and there exist indexes $j1, \ldots, jm$ such that $1 \le j1 < j2 < \ldots < jm \le n$ and $s_{ji} \unrhd_{emb} t_i$ ($i = 1, \ldots, m$), or
(2) $s_j \unrhd_{emb} t$ for some j.

Lemma 7 *Let $\succ$ be a partial ordering on $\lambda\mathcal{F}$ and $>$ a simplification ordering on $\mathcal{T}(\lambda\mathcal{F})$. Then, $\rhd_{emb} \subseteq >$ holds.*
Proof. We show that $s \rhd_{emb} t$ implies $s > t$ by induction on $|s|$.

- Basic step: $|s| = 1$. Since $\alpha, \beta \in \lambda\mathcal{F}$ and $s = \alpha \rhd_{emb} \beta = t$, $\alpha \succ \beta$. Hence, $s = \alpha > \beta = t$ holds.
- Induction step: We consider that $s = \alpha\ (s_1, \ldots, s_n) \rhd_{emb} \beta\ (t_1, \ldots, t_m) = t$.
  (1) By induction hypothesis, $s_{ji} \ge t_i$ holds ($i = 1, \ldots, m$). By replacement property, $\beta\ (s_{j1}, \ldots, s_{jm}) \ge \beta\ (t_1, \ldots, t_m)$ holds. Since $\alpha \succ \beta$ and $1 \le j1 < j2 < \ldots < jm \le n$, $\alpha\ (s_1, \ldots, s_m) > \beta\ (s_{j1}, \ldots, s_{jm})$ holds. Hence, $s = \alpha\ (s_1, \ldots, s_n) > \beta\ (t_1, \ldots, t_m) = t$ holds.
  (2) By induction hypothesis, $s_j \ge t$ for some j. By subterm property, $s > s_j$ holds. Hence, $s > t$ holds. □

Note. If we assume that $\mathcal{S}$ and $\mathcal{F}$ are finite then we can give a well-partial ordering $>$ on $\lambda\mathcal{F}$, i.e., $(\lambda\mathcal{F}, >)$ is a well-partially ordered set: See lemma 32 in appendix.

Theorem 8 *Let $\succ$ be a well-partial ordering on $\lambda\mathcal{F}$. Then, a homeomorphic embedding relation $\rhd_{emb}$ is a well-partial ordering on $\mathcal{T}(\lambda\mathcal{F})$.*
Proof. It is straightforward by Kruskal's tree theorem[3),8)]. □

Theorem 9 *Let $\succ$ be a well-partial ordering on $\lambda\mathcal{F}$. Then, simplification orderings on $\mathcal{T}(\lambda\mathcal{F})$ are well-founded.*
Proof. It is straightforward since lemma 7 and theorem 8. □

The following theorem guarantees the termination property for higher-order rewrite systems.

Theorem 10 *Let $\mathcal{R}$ be a HRS on normalized terms. Let $\succ$ be a well-partial ordering on $\lambda\mathcal{F}$ and $>$ a simplification ordering on $\mathcal{T}(\lambda\mathcal{F})$ such that $\| l\theta\!\downarrow \| > \| r\theta\!\downarrow \|$ for any ground substitution $\theta: \mathcal{X} \to \mathcal{T}(\mathcal{F} \cup \mathcal{B})\!\downarrow$ and any rewrite rule $l \to r$ in $\mathcal{R}$. Then, $\mathcal{R}$ is terminating.*
Proof. Assume that $\mathcal{R}$ is not terminating. There is an infinite rewrite sequence $t_0 \to_{\mathcal{R}} t_1 \to_{\mathcal{R}} t_2 \to_{\mathcal{R}} \ldots$. Without loss of generality, we may assume that terms in this infinite sequence are ground. By the definition of rewriting, $s \to_{\mathcal{R}} t$ if and only if there exist a rewrite rule $l \to r \in \mathcal{R}$, a substitution $\theta: \mathcal{X} \to \mathcal{T}(\mathcal{F}, \mathcal{X})\!\downarrow$ and a ground context C such that $s = C[l\theta\!\downarrow]$ and $t = C[r\theta\!\downarrow]$. $FV(l\theta\!\downarrow) = \{x_1, \ldots, x_n\}$ where $x_i{:}\sigma_i$ $(1 \le i \le n)$. Let $\theta' = \{x_1 \leftarrow c_{\sigma_1}, \ldots, x_n \leftarrow c_{\sigma_n}\}$. Note that $l\theta\theta'\!\downarrow = l\theta\!\downarrow\!\theta'$, $r\theta\theta'\!\downarrow = r\theta\!\downarrow\!\theta'$ and substitution $\theta\theta': \mathcal{X} \to \mathcal{T}(\mathcal{F} \cup \mathcal{B})\!\downarrow$ is ground. Since $l\theta\!\downarrow\!\theta'$ and $r\theta\!\downarrow\!\theta'$ are ground and the assumption, $\| l\theta\!\downarrow\!\theta' \| > \| r\theta\!\downarrow\!\theta' \|$ holds. By replacement property, $\| C \|[\| l\theta\!\downarrow\!\theta' \|] > \| C \|[\| r\theta\!\downarrow\!\theta' \|]$. By lemma 3, it follows that $\| C[l\theta\!\downarrow] \| = \| C \|[\| l\theta\!\downarrow\!\theta' \|]$ and $\| C[r\theta\!\downarrow] \| = \| C \|[\| r\theta\!\downarrow\!\theta' \|]$. Hence, $\| s \| > \| t \|$. Thus we have the infinite sequence $\| t_0 \| > \| t_1 \| > \| t_2 \| > \ldots$: contradiction to the well-foundedness of $>$ by theorem 9. Therefore, $\mathcal{R}$ is terminating. □

4. Higher-Order Recursive Path Ordering

In this section, we define the recursive path ordering for higher-order rewrite systems. First we define the recursive path ordering on algebraic terms.

Definition 11 (TRPO) Let $>_{\lambda\mathcal{F}}$ be a partial ordering on $\lambda\mathcal{F}$. Let s and t be algebraic terms. The *typed recursive path ordering* (TRPO) on $\mathcal{T}(\lambda\mathcal{F}, \mathcal{X})$ is defined as follows:
$s >_{TRPO} t$ if and only if $top(s) \notin \mathcal{X}$ and
(1) $s_i \ge_{TRPO} t$ for some $s_i \in st^m(s)$, or
(2) $top(s) >_{\lambda\mathcal{F}} top(t)$ and $s >_{TRPO} t_i$ for all $t_i \in st^m(t)$, or
(3) $top(s) = top(t)$, $status\ (top(s)) = mult$ and $st^m(s) >^{mul}_{TRPO} st^m(t)$, or
(4) $top(s) = top(t)$, $status\ (top(s)) = lex$, $st(s) >^{lex}_{TRPO} st(t)$ and $s >_{TRPO} t_i$ for all $t_i \in st^m(t)$.

We show that the TRPO is a simplification ordering on $\mathcal{T}(\lambda\mathcal{F})$ for some partial ordering $>_{\lambda\mathcal{F}}$ on $\lambda\mathcal{F}$. Note that a partial ordering $>_{\lambda\mathcal{F}}$

on $\lambda\mathcal{F}$ is given in the following lemmas.

Lemma 12 *The TRPO is a partial ordering on $\mathcal{T}(\lambda\mathcal{F})$.*
Proof. We can show the transitivity and the irreflexivity of TRPO by using the same argument as that for the recursive path ordering on first-order terms[2]. □

Lemma 13 *The TRPO has the subterm property on $\mathcal{T}(\lambda\mathcal{F})$, i.e., $s \in \mathcal{T}(\lambda\mathcal{F})$ and $s \rhd t$ imply $s >_{TRPO} t$.*
Proof. Let s be a ground algebraic term and t be a strict subterm of s. We show that $s >_{TRPO} t$ by induction on $|s|$. Let $s = \alpha(s_1, \ldots, s_m)$ $(m \geq 1)$. Since t is a subterm of some s_i, $s_i \geq_{TRPO} t$ holds by induction hypothesis. Hence, $s >_{TRPO} t$ holds by case (1) of definition 11. □

Lemma 14 *The TRPO has the replacement property on $\mathcal{T}(\lambda\mathcal{F})$, i.e., $s >_{TRPO} t$ implies $\alpha\,(u_1,\ldots, s, \ldots, u_n) >_{TRPO} \alpha\,(u_1,\ldots, t, \ldots, u_n)$ for $\alpha\,(u_1,\ldots, s, \ldots, u_n)$, $\alpha\,(u_1,\ldots, t, \ldots, u_n) \in \mathcal{T}(\lambda\mathcal{F})$.*
Proof. If $\alpha : \sigma_1 \to \ldots \to \sigma_n \to \sigma \in \lambda\mathcal{F}$ and $status(\alpha) = mult$ then $\alpha\,(u_1,\ldots,s,\ldots,u_n) >_{TRPO} \alpha\,(u_1,\ldots,t,\ldots,u_n)$ holds, since $\{u_1, \ldots, s, \ldots, u_n\} >^{mul}_{TRPO} \{u_1, \ldots, t, \ldots, u_n\}$. If $status(\alpha) = lex$ then $\alpha\,(u_1, \ldots, s, \ldots, u_n) >_{TRPO} \alpha\,(u_1, \ldots, t, \ldots, u_n)$ holds, since $\langle u_1, \ldots, s, \ldots, u_n\rangle >^{lex}_{TRPO} \langle u_1, \ldots, t, \ldots, u_n\rangle$ and $\alpha\,(u_1, \ldots, s, \ldots, u_n) >_{TRPO} u$ for all $u \in \{u_1, \ldots, t, \ldots, u_n\}$. □

Lemma 15 *If $\alpha >_{\lambda\mathcal{F}} \beta$ then $\alpha(u_1,\ldots,u_n) >_{TRPO} \beta(u_{j1},\ldots,u_{jm})$ where α, $\beta \in \lambda\mathcal{F}$, $1 \leq j1 < \ldots < jm \leq n$, $arity(\alpha) = n$ and $arity(\beta) = m$.*
Proof. It is straightforward by the definition of TRPO. □

Theorem 16 *If $>_{\lambda\mathcal{F}}$ is a partial ordering on $\lambda\mathcal{F}$ then the TRPO is a simplification ordering on $\mathcal{T}(\lambda\mathcal{F})$.*
Proof. By lemmas 12, 13, 14 and 15, the TRPO is a simplification ordering on $\mathcal{T}(\lambda\mathcal{F})$. □

We define the higher-order recursive path ordering based on the typed recursive path ordering.

Definition 17 (HRPO) Let s and t be normalized terms. The *higher-order recursive path ordering* (HRPO) $s >_{HRPO} t$ is defined by $\|s\| >_{TRPO} \|t\|$.

Example 18 We consider the following normalized terms. $s = X(\lambda xy.x)$ and $t = X(\lambda xy.y)$ where $X : Nat \to Nat \to Nat$. Then, $\|s\| = X\ (\lambda_{Nat \to Nat \to Nat}\ (\lambda_{Nat \to Nat}\ (c_{Nat}))) = \|t\|$ holds.
Let $\theta = \{X \leftarrow \lambda z.\ \ z(0, 1)\}$ where $z{:}Nat \to Nat \to Nat$. Since $s\theta\!\downarrow = 0$ and $t\theta\!\downarrow = 1$, we have $\|s\theta\!\downarrow\| \neq \|t\theta\!\downarrow\|$. Hence, it does not hold that $\|s\| = \|t\|$ implies $\|s\theta\!\downarrow\| = \|t\theta\!\downarrow\|$ for any ground substitution $\theta{:}\mathcal{X} \to \mathcal{T}(\mathcal{F} \cup \mathcal{B})\!\downarrow$ in general.

The above example explains why each bound variable is restricted into a basic type to guarantee the stability of ground substitutions.

Lemma 19 *The TRPO is stable under ground substitutions, i.e., $\|s\| >_{TRPO} \|t\|$ implies $\|s\theta\!\downarrow\| >_{TRPO} \|t\theta\!\downarrow\|$ on $\mathcal{T}(\lambda\mathcal{F})$ for any ground substitution $\theta{:}\mathcal{X} \to \mathcal{T}(\mathcal{F} \cup \mathcal{B})\!\downarrow$.*
Proof. See appendix. □

Lemma 20 *For any well-founded ordering $>_{\lambda\mathcal{F}}$ on $\lambda\mathcal{F}$ there exists a well-partial ordering $>^*_{\lambda\mathcal{F}}$ on $\lambda\mathcal{F}$ such that $>_{\lambda\mathcal{F}} \subseteq >^*_{\lambda\mathcal{F}}$ and $>_{TRPO} \subseteq >^*_{TRPO}$.*
Proof. By structural induction we can show that if $> \subseteq >^*$ then $>_{TRPO} \subseteq >^*_{TRPO}$. Further we can show that every well-founded ordering is contained in a total well-founded ordering. Since every total well-founded ordering is a well-partial ordering, there exists a well-partial ordering $>^*$ on $\lambda\mathcal{F}$. □

Theorem 21 *Let $\mathcal{R}$ be a HRS on normalized terms. Let $>_{\lambda\mathcal{F}}$ be a well-founded ordering on $\lambda\mathcal{F}$. If for any rewrite rule $l \to r$ in $\mathcal{R}$ we have $l >_{HRPO} r$, then $\mathcal{R}$ is terminating.*
Proof. Since theorem 16 and lemma 20, there exists a simplification ordering $>^*_{TRPO}$ on $\mathcal{T}(\lambda\mathcal{F})$. For any $l \to r \in \mathcal{R}$, $\|l\theta\!\downarrow\| >_{TRPO} \|r\theta\!\downarrow\|$ holds on $\mathcal{T}(\lambda\mathcal{F})$ for any ground substitution $\theta{:}\mathcal{X} \to \mathcal{T}(\mathcal{F} \cup \mathcal{B})\!\downarrow$ because of $l >_{HRPO} r$ and lemma 19. Since $>_{TRPO} \subseteq >^*_{TRPO}$, $\|l\theta\!\downarrow\| >^*_{TRPO} \|r\theta\!\downarrow\|$ holds. From theorem 10, it follows that the HRS $\mathcal{R}$ is terminating. □

Example 22 We show the termination property of the following HRS $\mathcal{R}$[7].
$\mathcal{S} = \{term,\ form\}$, $\mathcal{F} = \{\neg{:}form \to form,\ \wedge,\ \vee{:}form \to form \to form,\ \forall,\ \exists{:}(term \to form) \to form\}$, $\mathcal{X} = \{P,\ Q{:}form,\ P'{:}term \to form,\ x{:}term\}$ and

$$\mathcal{R} = \begin{cases} \neg\neg P \to P \\ \neg(P \wedge Q) \to (\neg P) \vee (\neg Q) \\ \neg(P \vee Q) \to (\neg P) \wedge (\neg Q) \\ \neg\forall(\lambda x.P'(x)) \to \exists(\lambda x.\neg P'(x)) \\ \neg\exists(\lambda x.P'(x)) \to \forall(\lambda x.\neg P'(x)). \end{cases}$$

We give the precedence $\neg >_{\lambda\mathcal{F}} \vee$, $\neg >_{\lambda\mathcal{F}} \wedge$, $\neg >_{\lambda\mathcal{F}} \exists >_{\lambda\mathcal{F}} \lambda_{form \to (term \to form)}$, $\neg >_{\lambda\mathcal{F}} \forall >_{\lambda\mathcal{F}} \lambda_{form \to (term \to form)}$. Then, $\neg\neg P >_{HRPO} P$, $\neg(P \wedge Q) >_{HRPO} (\neg P) \vee (\neg Q)$, $\neg(P \vee Q) >_{HRPO} (\neg P) \wedge (\neg Q)$, $\neg\forall(\lambda x.P'(x)) >_{HRPO} \exists(\lambda x.\neg P'(x))$ and $\neg\exists(\lambda x.P'(x)) >_{HRPO} \forall(\lambda x.\neg P'(x))$ hold. Hence, $\mathcal{R}$ is terminating by theorem 21.

Example 23 We show the termination property of the following HRS $\mathcal{R}$.
$\mathcal{S} = \{Nat,\ List\}$, $\mathcal{F} = \{f{:}Nat \to Nat,\ g{:}(List \to List) \to List \to Nat,\ h{:}List \to Nat,\ \hat{f}{:}List \to List,\ \hat{g}{:}(Nat \to Nat) \to Nat \to List,\ \hat{h}{:}Nat \to List\}$, $\mathcal{X} = \{X{:}List \to List,\ x{:}List,\ Z{:}List,\ Y{:}Nat \to Nat,\ y{:}Nat,\ W{:}Nat\}$ and

$$\mathcal{R} = \begin{cases} f(g(\lambda x.X(x), Z)) \to h(Z) \\ \hat{f}(\hat{g}(\lambda y.Y(y), W)) \to \hat{h}(W). \end{cases}$$

We give the precedence $f >_{\lambda\mathcal{F}} h$ and $\hat{f} >_{\lambda\mathcal{F}} \hat{h}$. Then, $f(g(\lambda x.X(x),Z) >_{HRPO} h(Z)$ and $\hat{f}(\hat{g}(\lambda y.Y(y),W)) >_{HRPO} \hat{h}(W)$ hold. Hence, $\mathcal{R}$ is terminating by theorem 21.
On the other hand, Jouannaud and Rubio's ordering[5),6)] cannot deal with this example. Let $_{\mathcal{T}_S}$ be a quasi-ordering on $\mathcal{T}_{\mathcal{S}}$. A normalized term $s{:}\sigma$ is *type compatible* if $t{:}\tau$ is type compatible and $\sigma\ _{\mathcal{T}_S}\ \tau$ for any subterm $t{:}\tau$ of s. Since their ordering works on only type compatible terms and $f(g(\lambda x.X(x),Z)){:}Nat \rhd \lambda x.X(x){:}List \to List$, we have $Nat\ _{\mathcal{T}_S}\ List \to List$ when the quasi-ordering $_{\mathcal{T}_S}$ is a recursive path ordering on $\mathcal{T}_{\mathcal{S}}$. Hence, we have $Nat\ _{\mathcal{S}}\ List$. Since $\hat{f}(\hat{g}(\lambda y.Y(y),W)){:}List \rhd \lambda y.Y(y){:}Nat \to Nat$ and the type compatibility, it is obtained that $List\ _{\mathcal{T}_S}\ Nat \to Nat$. Hence, we have $List\ _{\mathcal{S}}\ Nat$. So $Nat \not>_{\mathcal{T}_S} List \to List$ and $List \not>_{\mathcal{T}_S} Nat \to Nat$ by $List \sim_{\mathcal{S}}$ Nat. This is contradiction to the type compatibility. Thus termination of $\mathcal{R}$ on type compatible normalized terms cannot be shown by their ordering when $_{\mathcal{T}_S}$ is a recursive path ordering on $\mathcal{T}_{\mathcal{S}}$.

5. Envelope for Typed Terms

The envelope G_τ is a subset of $\lambda\mathcal{F}$ such that every constant symbol occurring in ground algebraic terms with the type τ is contained in G_τ, i.e, $Fun(s) \subseteq G_\tau$ for any ground algebraic term $s{:}\tau$. Let $X{:}\sigma_1 \to \ldots \to \sigma_n \to b$ be a variable with an output type b and let θ a ground normalized substitution such that $X\theta = \lambda x_1 \ldots x_n.t{:}\sigma_1 \to \ldots \to \sigma_n \to b \in \mathcal{T}(\mathcal{F})\downarrow$. Then, for any ground normalized terms $s_1{:}\sigma_1, \ldots, s_n{:}\sigma_n$, we have $Fun(\|\ X\theta\ (s_1,\ldots,s_n)\downarrow\ \|) \subseteq G_b$. This property allows us to treat the variable X as a constant symbol with the output type b in the TRPO under appropriate conditions about G_b. In fact, if we can give a precedence $>_{\lambda\mathcal{F}}$ satisfying $\alpha >_{\lambda\mathcal{F}} \beta$ for all $\beta \in G_b$, then $\alpha(t_1,\ldots,t_n) >_{TRPO} \|\ X\theta\ (s_1,\ldots,s_n)\downarrow\ \|$ follows from the definition of the TRPO. This means that X plays in the TRPO like a constant with $\alpha >_{\lambda\mathcal{F}} X$.
First, we define the envelope and the precedence between constants and free variables as follows.

Definition 24 We define the *envelope* G_τ ($\subseteq \lambda\mathcal{F}$) inductively as follows.
(1) • $f{:}\tau_1 \to \ldots \to \tau_l \to \tau \in \mathcal{F}$ implies $f \in G_\tau$.
 • $\lambda_{\sigma \to (\tau \to \sigma)},\ c_\tau \in G_{\tau \to \sigma}$.
(2) • $g{:}\sigma_1 \to \ldots \to \sigma_n \to \sigma \in G_\tau$ and $g \in \mathcal{F} \cup \mathcal{B}$ imply $G_{\sigma_1} \cup \ldots \cup G_{\sigma_n} \subseteq G_\tau$.
 • $\lambda_{\sigma \to (\eta \to \sigma)} \in G_\tau$ implies $G_\sigma \subseteq G_\tau$.

Example 25 We consider the following signature: $\mathcal{F} = \{0{:}Nat,\ s{:}Nat \to Nat,\ +{:}Nat \to Nat \to Nat,\ nil{:}List,\ cons{:}Nat \to List \to List,\ map{:}(Nat \to Nat) \to List \to List\}$. Then, $G_{Nat} = \{0,\ s,\ +\}$ and $G_{List} = \{nil,\ cons,\ map,\ \lambda_{Nat \to (Nat \to Nat)},\ c_{Nat},\ 0,\ s,\ +\}$.

Lemma 26 *Let s be a ground algebraic term with a type τ. Then, $Fun(s) \subseteq G_\tau$ holds.*
Proof. It is straightforward by the definitions of the interpretation function and of algebraic terms. □

By using envelopes we can extend the precedence $>_{\lambda\mathcal{F}}$ on $\lambda\mathcal{F}$ to that on $\lambda\mathcal{F} \cup \mathcal{X}$ as follows. Let $\alpha \in \lambda\mathcal{F}$ and let Y be a free variable with an output type b. Then we define $\alpha >_{\lambda\mathcal{F}} Y$ if $\alpha >_{\lambda\mathcal{F}} \beta$ for any $\beta \in G_b$. We say the *TRPO (HRPO) with envelopes* shortly when the TRPO is based on this extended precedence $>_{\lambda\mathcal{F}}$ on $\lambda\mathcal{F} \cup \mathcal{X}$.

Lemma 27 *Let $s = \alpha(s_1,\ldots,s_n)$ $(n \geq 0)$ be a algebraic term and X a free variable with an output type b. If $\alpha >_{\lambda\mathcal{F}} X$ then $s >_{TRPO} t$ for any ground algebraic term $t{:}b$.*
Proof. Trivial from the definitions of $\alpha >_{\lambda\mathcal{F}} X$ and of the TRPO. □

We next show that the TRPO with envelopes is stable under ground substitutions.

Lemma 28 *The TRPO with envelopes is stable under ground substitutions, i.e., $\| s \| >_{TRPO} \| t \|$ with envelopes implies $\| s\theta\downarrow \| >_{TRPO} \| t\theta\downarrow \|$ on $\mathcal{T}(\lambda\mathcal{F})$ for any ground substitution $\theta{:}\mathcal{X} \to \mathcal{T}(\mathcal{F} \cup \mathcal{B})\downarrow$.*
Proof. See appendix. □

Lemma 29 *For any well-founded ordering $>_{\lambda\mathcal{F}}$ on $\lambda\mathcal{F}$ there exists a well-partial ordering $>^*_{\lambda\mathcal{F}}$ on $\lambda\mathcal{F}$ such that $>_{\lambda\mathcal{F}} \subseteq >^*_{\lambda\mathcal{F}}$ and $>_{TRPO} \subseteq >^*_{TRPO}$ with envelopes.*
Proof. It is straightforward by the same arguent of lemma 20. □

From now on we restrict $l\theta\downarrow$ as a ground term in the reduction $C[l\theta\downarrow] \to_{\mathcal{R}} C[r\theta\downarrow]$ in the rest of this paper.

Theorem 30 *Let $\mathcal{R}$ be a HRS on normalized terms. Let $>_{\lambda\mathcal{F}}$ be a well-founded ordering on $\lambda\mathcal{F} \cup \mathcal{X}$. If for any rewrite rule $l \to r$ in $\mathcal{R}$ we have $l >_{HRPO} r$ with envelopes, then $\mathcal{R}$ is terminating.*
Proof. Since theorem 16 and lemma 29, there exists a simplification ordering $>^*_{TRPO}$ on $\mathcal{T}(\lambda\mathcal{F})$. For any $l \to r \in \mathcal{R}$ and any ground substitution $\theta:\mathcal{X} \to \mathcal{T}(\mathcal{F} \cup \mathcal{B})\downarrow$, $\| l\theta\downarrow \| >_{TRPO} \| r\theta\downarrow \|$ holds on $\mathcal{T}(\lambda\mathcal{F})$ since $l >_{HRPO} r$ and lemma 28. Since $>_{TRPO} \subseteq >^*_{TRPO}$ with envelopes, $\| l\theta\downarrow \| >^*_{TRPO} \| r\theta\downarrow \|$ holds. From theorem 10, it follows that the HRS $\mathcal{R}$ is terminating. □

The following example explains how to apply the HRPO with envelopes to prove termination.

Example 31 Consider the following basic types, signature and HRS $\mathcal{R}$: $\mathcal{S} = \{Nat, List\}$, $\mathcal{F} = \{nil{:}List,\ cons{:}Nat \to List \to List,\ map{:}(Nat \to Nat) \to List \to List,\ 0{:}Nat,\ s{:}Nat \to Nat,\ +{:}Nat \to Nat \to Nat\}$, $\mathcal{X} = \{X{:}Nat \to Nat,\ N{:}Nat,\ L{:}List,\ x{:}Nat\}$ and

$$\mathcal{R} = \begin{cases} map(\lambda x.X(x), nil) \to nil \\ map(\lambda x.X(x), cons(N, L)) \\ \quad \to cons(X(N), map(\lambda x.X(x), L)). \end{cases}$$

To prove the termination property of $\mathcal{R}$ we use the precedence: $map >_{\lambda\mathcal{F}} cons$, 0, s, $+$ and $status(map) = mult$. Since it is obvious that $map(\lambda x.X(x),nil) >_{HRPO} nil$, we consider the second rule. From definition 24 we can obtain $G_{Nat} = \{0,s,+\}$. Since $map >_{\lambda\mathcal{F}} f$ for any $f \in G_{Nat}$, we have $map >_{\lambda\mathcal{F}} X$. Hence, $map\ (\lambda x.X(x),cons\ (N,L)) >_{HRPO} X(N)$ holds. Thus $map\ (\lambda x.X(x),cons\ (N,L)) >_{HRPO} cons\ (X(N),map\ (\lambda x.X(x),L))$ follows. Therefore, HRS $\mathcal{R}$ is terminating by theorem 30.

6. Related Works

In this section, we compare our higher-order recursive path ordering with Jouannaud and Rubio's ordering[5),6)]. We consider the following HRS $\mathcal{R}$:

$\mathcal{S} = \{Nat, List\}$, $\mathcal{F} = \{app{:}(Nat \to Nat) \to Nat \to Nat,\ list{:}Nat \to List,\ twice{:}Nat \to List,\ cons{:}Nat \to List \to List,\ nil{:}List,\ 0{:}Nat,\ s{:}Nat \to Nat,\ +{:}Nat \to Nat \to Nat\}$, $\mathcal{X} = \{X{:}Nat \to Nat,\ Y{:}Nat,\ x{:}Nat\}$ and

$$\mathcal{R} = \begin{cases} list(app(\lambda x.X(x), Y)) \to twice(X(Y)) \\ twice(Y) \to cons(Y, cons(Y, nil)) \\ list(Y) \to cons(Y, nil). \end{cases}$$

We first show termination of $\mathcal{R}$ by applying our ordering with envelopes. From the definition of envelopes we have $G_{Nat} = \{app,\ 0,\ s,\ +,\ \lambda_{Nat \to (Nat \to Nat)},\ c_{Nat}\}$. Give the following precedence: $list >_{\lambda\mathcal{F}} twice >_{\lambda\mathcal{F}} cons$, nil and $list >_{\lambda\mathcal{F}} X$, i.e., $list >_{\lambda\mathcal{F}} \alpha$ for any $\alpha \in G_{Nat}$. Then, $list(app(\lambda x.X(x),Y)) >_{HRPO} twice(X(Y))$ with envelopes. From theorem 30 it follows that HRS $\mathcal{R}$ is terminating.

Jouannaud and Rubio's ordering[5),6)] cannot be applied to the above $\mathcal{R}$ for proving termination because of the type compatible term limitation. Let τ_S be a recursive path ordering on $\mathcal{T}_S$. In this case, $list(app(\lambda x.X(x),Y))$ is not a type compatible normalized term. More precisely, $app(\lambda x.X(x),Y){:}Nat \rhd \lambda x.X(x){:}Nat \to Nat$ and $Nat \not\tau_S Nat \to Nat$ by $Nat \to Nat \; \tau_S \; Nat$. Thus termination of $\mathcal{R}$ cannot be proven by their ordering.

Next, we consider the following HRS $\mathcal{R}$: $\mathcal{S} = \{b, b'\}$, $\mathcal{F} = \{f{:}(b' \to b') \to b,\ g{:}b' \to b,\ d{:}b',\ e{:}b' \to b',\ h{:}(b \to b') \to b'\}$, $\mathcal{X} = \{X{:}b' \to b',\ Y{:}b \to b',\ x{:}b',\ y{:}b\}$ and

$$\mathcal{R} = \left\{ \begin{array}{l} f(\lambda x.X(x)) \to g(X(d)) \\ e(h(\lambda y.Y(y))) \to d. \end{array} \right.$$

Our ordering can prove termination of $\mathcal{R}$. We have $G_{b'} = \{d, e, h, \lambda_{b' \to (b \to b')}, c_b\}$. Give the following precedence: $f >_{\lambda\mathcal{F}} g$, d and $e >_{\lambda\mathcal{F}} d$ and $f >_{\lambda\mathcal{F}} X$, i.e., $f >_{\lambda\mathcal{F}} \alpha$ for any $\alpha \in G_{b'}$. Then, $f(\lambda x.X(x)) >_{HRPO} g(X(d))$ holds. Hence, HRS $\mathcal{R}$ is terminating.

Jouannaud and Rubio's ordering[5),6)] again cannot prove termination of $\mathcal{R}$. Let τ_S be a recursive path ordering on $\mathcal{T}_S$. Since $e(h(\lambda y.Y(y))){:}b'$, $\lambda y.Y(y){:}b \to b'$ and $b' \not\tau_S b \to b'$, $e(h(\lambda y.Y(y)))$ is not type compatible.

Jouannaud and Rubio[5),6)] proposed the *sort ordering* on $\mathcal{T}_S$ as the other quasi-ordering τ_S. However, the sort ordering τ_S does not work for this example. We have to show $f(\lambda x.X(x)){:}b > X(d){:}b'$ in the first rule of $\mathcal{R}$. Thus we have $b >_S b'$. Then $e(h(\lambda y.Y(y)))$ is not type compatible for the sort ordering, because $e(h(\lambda y.Y(y))){:}b'$ and $\lambda y.Y(y){:}b \to b'$.

7. Conclusion

We have introduced a natural framework of a simplification ordering for analyzing termination of higher-order rewriting, and based on this framework we have proposed a powerful recursive path ordering on higher-order terms, called the higher-order recursive path ordering (HRPO). Our ordering has extended Jouannaud and Rubio's ordering, which does not allow comparing two type incompatible terms. We have shown through several examples that our ordering can be applied to prove termination of higher-order rewrite systems to which Jouannaud and Rubio's ordering cannot apply. We believe that our ordering provides very useful means of proving termination which arises in various higher-order formal systems like higher-order functional programming languages and higher-order theorem provers.

Acknowledgments The authors would like to thank professor Masahiko Sakai for his helpful comments. This work is partially supported by Grants from Ministry of Education, Science and Culture of Japan, ♯10139214 and ♯10680346.

References

1) Baader, F. and Nipkow, T.: Term rewriting and all that, Cambridge University Press (1998).
2) Dershowitz, N.: Orderings for term-rewriting systems, Theoretical Computer Science, 17, pp.279–301 (1982).
3) Dershowitz, N.: Termination of rewriting, J. Symbolic Computation, 3, pp.69–116 (1987).
4) Iwami, M., Sakai, M. and Toyama, Y.: An improved recursive decomposition ordering for higher-order rewrite systems, IEICE Transactions of Information and Systems, E81-D, pp.988–996 (1998).
5) Jouannaud, J. P. and Rubio, A.: A recursive path ordering for higher-order terms in η-long β-normal form, Proc. 7th International Conf. on Rewriting Techniques and Applications, LNCS, 1103, pp.108–122 (1996).
6) Jouannaud, J. P. and Rubio, A.: Rewrite orderings for higher-order terms in η-long β-normal form and the recursive path ordering, to appear in Theoretical Computer Science.
7) Mayr, R. and Nipkow, T.: Higher-order rewrite systems and their confluence, Theoretical Computer Science, 192, pp.3–29 (1998).
8) Middeldrop, A. and Zantema, H.: Simple termination of rewrite systems, Theoretical Computer Science, 175, pp.127–158 (1997).
9) Prehofer, C.: Solving higher-order equations from logic to programming, Progress in theoretical computer science, Birkhäuser (1998).
10) Steinbach, J.: Termination of rewriting – extension, comparison and automatic generation of simplification orderings, Ph.D. Thesis, University of Kaiserslautern (1994).
11) Steinbach, J.: Simplification ordering: history of results, Fundamenta Informaticae, 24, pp.44–87 (1995).
12) Walukiewicz, D.: A total AC-reduction ordering on higher-order terms, Proc. 25th International Conf. on Automata, Languages and Programming, LNCS, 1443, pp.530–542 (1998).

Appendix

A.1 Lemma 32.

Lemma 32 *Let $\mathcal{S}$ and $\mathcal{F}$ be finite. Then, we can give a well-partial ordering $>$ on $\lambda\mathcal{F}$, i.e., $(\lambda\mathcal{F},>)$ is a well-partially ordered set.*

Proof. We consider $\lambda\mathcal{F} = \mathcal{F} \cup \mathcal{L} \cup \mathcal{B}$ where $\mathcal{F}$ is a finite set.

(1) As $\mathcal{F}$ is finite, $(\mathcal{F},\emptyset)$ is a well-partially ordered set where $\emptyset$ is the empty relation.

(2) We show that a well-partial ordering $>_\lambda$ can be defined on $\mathcal{L}$. Let $\mathcal{S}$ be $\{b_1,\ldots,b_n\}$ where b_i is a basic type $(i = 1,\ldots,n)$. The total precedence on basic types and the constructor $\to$ is given by $b_0 < b_1 < \ldots < b_n < \to$. Then, for any τ, $\sigma \in \mathcal{T}_\mathcal{S}$, either $\tau >_{LPO} \sigma$ or $\tau <_{LPO} \sigma$ $(\tau \neq \sigma)$ holds where $>_{LPO}$ is the *lexicographic path ordering*[1] on types. Since $>_{LPO}$ is total and well-founded, it is a well-partial ordering. Hence, $(\mathcal{T}_\mathcal{S},>_{LPO})$ is a well-partially ordered set. The partial ordering $\lambda_\sigma >_\lambda \lambda_\tau$ is defined by $\sigma >_{LPO} \tau$. Thus, $(\mathcal{L},>_\lambda)$ is a well-partially ordered set.

(3) By the same argument as that in case (2), we can give a well-partial ordering $c_\sigma >_c c_\tau$ on $\mathcal{B}$ by $\sigma >_{LPO} \tau$. Thus, $(\mathcal{B},>_c)$ is a well-partially ordered set.

Therefore, $(\lambda\mathcal{F},>)$ is a well-partially ordered set where $> = >_\lambda \cup >_c$ by cases (1), (2) and (3). (Note that $\langle A_1,>_1\rangle$, $\langle A_2,>_2\rangle$ are well-partially ordered sets and $A_1 \cap A_2 = \emptyset$ then $\langle A_1 \cup A_2,>_1 \cup >_2\rangle$ is also a well-partially ordered set.) □

A.2 Proof of Lemma 19.

Lemma 33 *Let s and t be normalized terms. Then, $\| s \| = \| t \|$ implies $\| s\theta\downarrow \| = \| t\theta\downarrow \|$ for any ground substitution $\theta{:}\mathcal{X} \to \mathcal{T}(\mathcal{F} \cup \mathcal{B})\downarrow$.*

Proof. We show that $\| s \| = \| t \|$ implies $\| s\theta\downarrow \| = \| t\theta\downarrow \|$ by induction on the structure of s. In the case that $s = \alpha(s_1,\cdots,s_m)$ with $\alpha \in \mathcal{F}$ or $s = \lambda x.s_1$, the proof is straightforward. We consider the case that $s = \alpha(s_1,\cdots,s_m)$ with $\alpha \in \mathcal{X}$. Since $s = \alpha(s_1,\cdots,s_m)$ and $\| s \| = \| t \|$, we have $t = \alpha(t_1,\cdots,t_m)$ such that $\| s_i \| = \| t_i \|$ $(1 \leq i \leq m)$. Then $s\theta = (\alpha\theta)(s_1\theta,\cdots,s_m\theta)$ and $t\theta = (\alpha\theta)(t_1\theta,\cdots,t_m\theta)$. Letting $s' = \alpha\theta$, we show $\|s'(s_1\theta,\cdots,s_m\theta)\downarrow\| = \|s'(t_1\theta,\cdots,t_m\theta)\downarrow\|$ by induction on the size of s'. If $s' = \lambda x_1\cdots x_m.x_i$ then $\| s_i\theta\downarrow \| = \| t_i\theta\downarrow \|$ by induction hypothesis. Let $s' = \lambda x_1 \cdots x_m.f(u_1,\cdots,u_n)$ $(n \geq 0$ and $f \in \mathcal{F})$. Let $\gamma = \{x_1 \leftarrow s_1\theta,\cdots,x_m \leftarrow s_m\theta\}$.

$\|s'(s_1\theta,\cdots,s_m\theta)\downarrow\|$
$= f(\|u_1\gamma\downarrow\|,\cdots,\|u_n\gamma\downarrow\|)$
$= f(\ \|(\lambda\ x_1 \cdots x_m.u_1)\ (s_1\theta,\cdots,s_m\theta)\downarrow\|,\cdots,\|(\lambda\ x_1 \cdots x_m.u_n)\ (s_1\theta,\cdots,s_m\theta)\downarrow\|)$
$= f(\ \|(\lambda\ x_1 \cdots x_m.u_1)\ (t_1\theta,\cdots,t_m\theta)\downarrow\|,\cdots,\|(\lambda\ x_1 \cdots x_m.u_n)\ (t_1\theta,\cdots,t_m\theta)\downarrow\|)$

by induction hypothesis. □

Lemma 34 *Let $top(\| s \|) \notin \mathcal{X}$ and $\| t \| \in st^m(\| s \|)$. Then, $\| t\theta\downarrow \| \in st^m(\| s\theta\downarrow \|)$ for any ground substitution $\theta{:}\mathcal{X} \to \mathcal{T}(\mathcal{F} \cup \mathcal{B})\downarrow$.*

Proof. We consider the following cases.

(1) $\| \alpha(t_1,\ldots,t_n) \| = \alpha(\| t_1 \|,\ldots,\| t_n \|)$ and $\| \alpha\ (t_1,\ldots,t_n)\ \theta\downarrow \| = \alpha\ (\| t_1\theta\downarrow \|,\ldots,\| t_n\theta\downarrow \|)$ for $\alpha \in \mathcal{F} \cup \mathcal{B}$.

(2) $\| \lambda x.s{:}\tau \to \sigma \| = \lambda_{\sigma\to(\tau\to\sigma)}(\| s\{x \leftarrow c_\tau\} \|)$ and $\| (\lambda x.s)\theta\downarrow{:}\tau \to \sigma \| = \lambda_{\sigma\to(\tau\to\sigma)}(\| (s\theta\downarrow)\ \{x \leftarrow c_\tau\} \|)$. Since $x \in FV(s)$ and $x \notin dom(\theta)$, $\| (s\theta\downarrow)\ \{x \leftarrow c_\tau\} \| = \| (s\ \{x \leftarrow c_\tau\})\theta\downarrow \|$. □

Lemma 19 *The TRPO is stable under ground substitutions.*

Proof. Let s and t be normalized terms and $\theta{:}\mathcal{X} \to \mathcal{T}(\mathcal{F} \cup \mathcal{B})\downarrow$ be a ground substitution. We assume that $\| s \| >_{TRPO} \| t \|$ and show $\| s\theta\downarrow \| >_{TRPO} \| t\theta\downarrow \|$ by induction on $|\ \| s \|\ | + |\ \| t \|\ |$.

Basic step: $|\ \| s \|\ | + |\ \| t \|\ | = 2$.

If $\| s \| = \alpha$ and $\| t \| = \beta$ $(\alpha, \beta \in \lambda\mathcal{F})$, then $\| s\theta\downarrow \| >_{TRPO} \| t\theta\downarrow \|$ holds since $\| s\theta\downarrow \| = \| s \|$ and $\| t\theta\downarrow \| = \| t \|$.

Induction step:

(1) $top(\| s \|) \notin \mathcal{X}$ and $\| s_i \| \geq_{TRPO} \| t \|$ for some $\| s_i \| \in st^m(\| s \|)$.
Since induction hypothesis and lemma 33, we have $\| s_i\theta\downarrow \| \geq_{TRPO} \| t\theta\downarrow \|$. By lemma 34, $\| s_i\theta\downarrow \| \in st^m(\| s\theta\downarrow \|)$. Hence, $\| s\theta\downarrow \| >_{TRPO} \| t\theta\downarrow \|$ holds.

(2) $top(\| s \|) >_{\lambda\mathcal{F}} top(\| t \|)$, $top(\| s \|) \notin \mathcal{X}$ and $\| s \| >_{TRPO} \| t_i \|$ for any $\| t_i \| \in st^m(\| t \|)$.
By $top(\| t \|) \notin \mathcal{X}$, $top(\| t\theta\downarrow \|) = top(\| t \|)$ holds. From lemma 34, we have $\| t_i\theta\downarrow \| \in st^m(\| t\theta\downarrow \|)$. Since $\| s \| >_{TRPO} \| t_i \|$ and induction hypothesis, $\| s\theta\downarrow \| >_{TRPO} \| t_i\theta\downarrow \|$ holds for any $\| t_i\theta\downarrow \| \in st^m(\| t\theta\downarrow \|)$.

(3) $top(\| s \|) = top(\| t \|)$, $top(\| s \|) \notin \mathcal{X}$ and $st^m(\| s \|) >^{mul}_{TRPO} st^m(\| t \|)$.
Since $top(\| s \|) = top(\| t \|)$ and $top(\| s \|) \notin \mathcal{X}$, $top(\| s\theta\downarrow \|) = top(\| s \|)$ and $top(\| t\theta\downarrow \|) = top(\| t \|)$ hold. Since induction hypothesis, lemma 33 and lemma 34, $st^m(\| s\theta\downarrow \|) >^{mul}_{TRPO} st^m(\| t\theta\downarrow \|)$ holds.

(4) $top(\| s \|) = top(\| t \|)$, $top(\| s \|) \notin \mathcal{X}$, $st(\| s \|) >^{lex}_{TRPO} st(\| t \|)$ and $\| s \| >_{TRPO} \| t_i \|$ for any $\| t_i \| \in st^m(\| t \|)$.
Since $top(\| s \|) = top(\| t \|)$ and $top(\| s \|) \notin \mathcal{X}$, $top(\| s\theta\downarrow \|) = top(\| s \|)$ and $top(\| t\theta\downarrow \|) = top(\| t \|)$ hold. Since induction hypothesis, lemma 33 and lemma 34, $st(\| s\theta\downarrow \|) >^{lex}_{TRPO} st(\| t\theta\downarrow \|)$ and $\| s\theta\downarrow \| >_{TRPO} \| t_i\theta\downarrow \|$ for any $\| t_i\theta\downarrow \| \in st^m(\| t\theta\downarrow \|)$ hold. □

A.3 Proof of Lemma 28.

Lemma 28 *The TRPO with envelopes is stable under ground substitutions.*

Proof. Let s and t be normalized terms and $\theta{:}\mathcal{X} \to \mathcal{T}(\mathcal{F} \cup \mathcal{B})\downarrow$ be a ground substitution. We assume that $\| s \| >_{TRPO} \| t \|$ with envelopes and show $\| s\theta\downarrow \| >_{TRPO} \| t\theta\downarrow \|$ by induction on $| \| s \| | + | \| t \| |$.

Basic step: $| \| s \| | + | \| t \| | = 2$.

If $\| s \| = \alpha$ and $\| t \| = \beta$ ($\alpha, \beta \in \lambda\mathcal{F}$), then $\| s\theta\downarrow \| >_{TRPO} \| t\theta\downarrow \|$ holds since $\| s\theta\downarrow \| = \| s \|$ and $\| t\theta\downarrow \| = \| t \|$.

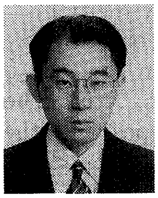
If $\| s \| = \alpha$ and $\| t \| = X$ with an output type b ($\alpha \in \lambda\mathcal{F}$, $X \in \mathcal{X}$), then $\alpha >_{\lambda\mathcal{F}} X$. From $\| X\theta\downarrow \|{:}b$ and lemma 27 it follows that $\| s\theta\downarrow \| = \alpha >_{TRPO} \| X\theta\downarrow \| = \| t\theta\downarrow \|$.

Induction step: The cases (1), (3) and (4) in the definition of TRPO are straightforward by the proof of lemma 19. We only show the case (2) in TRPO.

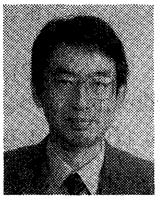
Case (2) $top(\| s \|) >_{\lambda\mathcal{F}} top(\| t \|)$, $top(\| s \|) \notin \mathcal{X}$ and $\| s \| >_{TRPO} \| t_i \|$ for any $\| t_i \| \in st^m(\| t \|)$.

If $top(\| t \|) \notin \mathcal{X}$, the claim follows by the same argument as that for the case (2) of lemma 19. Thus, we assume $top(\| t \|) \in \mathcal{X}$. Let $top(\| s \|) = \alpha$ and $t = X(t_1,\ldots,t_n)$ where $X{:}\sigma_1 \to \ldots \to \sigma_n \to b$. Then for any ground algebraic term $\| t\theta\downarrow \|{:}b$, we have $\| s\theta\downarrow \| >_{TRPO} \| t\theta\downarrow \|$ because of $top(\| s\theta\downarrow \|) = top(\| s \|) = \alpha$ and lemma 27. □

(Received October 16, 1998)
(Accepted December 16, 1998)



Munehiro Iwami was born on September 22 1970. He received the B.S. in mathematics from Tokai University in 1994. He received the M.S. in information science from JAIST, Hokuriku in 1996. He is now a doctoral student of the School of Information Science in JAIST, Hokuriku. He is interested in term rewriting systems. He is a member of JSSST.



Yoshihito Toyama received B.E. from Niigata University in 1975, M.E. and D.E. from Tohoku University in 1977 and 1990. From 1977 to 1993 he worked at NTT Laboratories. Since 1993, he is a a professor of School of information Science, Japan Advanced Institute of Science and Technology (JAIST). His research interests include term rewriting systems, program theory, and automated theorem proving. He is a member of IPSJ, IEICE, JSSST, ACM, EATCS.