Descendants and Head Normalization of Higher-Order Rewrite Systems

Hideto Kasuya[1], Masahiko Sakai[2] and Kiyoshi Agusa[2]

[1] Faculty of Information Science and Technology, Aichi Prefectural University, 1522–3 Ibaragabasama, Kumabari Nagakute-cho, Aichi, 480–1198 Japan
[2] Department of Information Engineering, Nagoya University, Furo-cho, Chikusa-ku Nagoya, 464–8603 Japan

Abstract. This paper describes an extension of head-needed rewriting on term rewriting systems to higher-order rewrite systems. The main difficulty of this extension is caused by the β-reductions induced from the higher-order reductions. In order to overcome this difficulty, we define a new descendant of higher-order rewrite systems. This paper shows the new definition of descendant, its properties and head normalization of head-needed rewriting on orthogonal higher-order rewrite systems.

1 Introduction

Higher-order rewrite systems (HRSs), an extension of term rewriting systems (TRSs) to higher-order terms, are used in functional programming, logic programming, and theorem proving as a model that contains the notion of λ-calculus. Properties of HRSs such as termination and confluence have been studied [11, 4, 5, 6, 9, 10]. On the other hand, there are a lot of works on reduction strategies of TRSs. Huet and Lévy show the following theorem on optimal normalizing strategy of orthogonal TRS [3]: a reducible term having a normal form contains at least one needed-redex to be reduced in every reduction sequence to a normal form. They also show that we always obtain a normal form of a given term by repeated reduction of the needed redex. Middeldorp generalized this result to head-needed reduction which computes head normal forms of terms [7].

In this paper, we discuss about head-needed reduction of HRSs. Since β-reductions induced by substitutions to higher-order variables change the structure of the higher-order term, the definition of descendant for TRSs [3, 2] works correctly no more. We give a definition of descendants on higher-order reduction, which represents where a redex is transferred to by a reduction. We prove that the strategy that reduces a head-needed redex is a head normalizing strategy in a higher-order setting.

2 Preliminaries

Let S be a set of basic types. The set τ_s of types is generated from S by the function space constructor $\rightarrow$ as follows:

$$\tau_s \supseteq S$$
$$\tau_s \supseteq \{\alpha \rightarrow \alpha' | \alpha, \alpha' \in \tau_s\}$$

Let $\mathcal{X}_\alpha$ be a set of variables with type α and $\mathcal{F}_\alpha$ be a set of function symbols with type α. The set of all variables with types is denoted by $\mathcal{X} = \bigcup_{\alpha \in \tau_s} \mathcal{X}_\alpha$, and the set of all function symbols with types is denoted by $\mathcal{F} = \bigcup_{\alpha \in \tau_s} \mathcal{F}_\alpha$. Simply typed λ-terms are defined by following inference rules:

$$\frac{x \in \mathcal{X}_\alpha}{x : \alpha} \quad \frac{f \in \mathcal{F}_\alpha}{f : \alpha} \quad \frac{s : \alpha \to \alpha' \;\; t : \alpha}{(st) : \alpha'} \quad \frac{x : \alpha \;\; s : \alpha'}{(\lambda x.s) : \alpha \to \alpha'}$$

If $t : \alpha$ is inferred from the rules then t is a simply typed λ-term with type α denoted by $\mathcal{T}_\alpha$. The set $\bigcup_{\alpha \in \tau_s} \mathcal{T}_\alpha$ of all λ-terms is denoted by $\mathcal{T}$. A simply typed λ-term is called a higher-order term or a term. We use well-known notion of bound variables and free variables. The sets of bound and free variables occurring in a term t are denoted by $BV(t)$ and $FV(t)$, respectively. The set $FV(t) \cup BV(t)$ is denoted by $Var(t)$. A higher-order term without free variables is called a ground term. If a term s is generated by renaming bound variables in a term t then s and t are equivalent and denoted by $s \equiv t$. We use X, Y and Z for free variables, and x, y and z for bound variables unless it is known to be free or bound from other conditions. We sometimes write $\mathbf{x}$ for a sequence $x_1 x_2 \cdots x_m$ ($m \geq 0$).

A mapping $\sigma : \mathcal{X} \mapsto \mathcal{T}$ from variables to higher-order terms of the same type is called substitution, if the domain $Dom(\sigma) = \{X | X \neq \sigma(X)\}$ is finite. If $Dom(\sigma) = \{X_1, \ldots, X_n\}$ and $\sigma(X_i) = t_i$, we also write it as $\sigma = \{X_1 \mapsto t_1, \ldots, X_n \mapsto t_n\}$. Let W be a set of variables and σ be a substitution. We write $\sigma|_W$ for a substitution generated by restricting the domain of σ to W. The set of free variables of σ is defined by $FV(\sigma) = \bigcup_{X \in Dom(\sigma)} FV(\sigma(X))$. Substitutions are extended to mappings σ' from higher-order terms to higher-order terms as follows:

$$\begin{array}{ll}
\sigma'(X) \equiv \sigma(X) & \text{if } X \in \mathcal{X} \\
\sigma'(f) \equiv f & \text{if } f \in \mathcal{F} \\
\sigma'((t_1 t_2)) \equiv (\sigma'(t_1)\sigma'(t_2)) & \text{if } t \equiv (t_1 t_2) \\
\sigma'(\lambda x.t_1) \equiv \lambda x.\sigma'|_{\overline{\{x\}}}(t_1) & \text{if } t \equiv \lambda x.t_1 \text{ and } x \notin FV(\sigma)
\end{array}$$

where $\overline{W}$ denotes the complement $\mathcal{X} - W$ of the set W of variables. We identify σ' with σ, and write $t\sigma$ instead of $\sigma'(t)$.

β-reduction is the operation that replaces $(\lambda x.s)t$ by $s\{x \mapsto t\}$. Let s be a term with type $\alpha \to \alpha'$, and $x \notin Var(s)$ be a variable with type α. Then, η-expansion is an operation that replaces s by $\lambda x.(sx)$. We say a term is η-long, if it is in normal form with respect to η-expansion. We also say a term is normalized if it is in β η-long normal form. A normalized term of t is denoted by $t\!\downarrow$. It is known that every higher-order term has a unique normalized term [5]. We say a substitution σ is normalized if $\sigma(X)$ is normalized for all $X \in Dom(\sigma)$.

Every normalized term can be represented by the form $\lambda x_1 \cdots x_m.(\cdots(at_1)\cdots t_n)$ where $m, n \geq 0$, $a \in \mathcal{F} \cup \mathcal{X}$ and $(\cdots(at_1)\cdots t_n)$ is basic type [5]. In this paper, we represent this term t by $\lambda x_1 \cdots x_m.a(t_1, \ldots, t_n)$. Note that $a(t_1, \ldots, t_n)$ is basic type. The top symbol of t is defined as $top(t) \equiv a$.

We define the position of a normalized term based on the form of $\lambda x_1 \cdots x_m.a(t_1,\ldots,t_n)$. In order to simplify the definition of descendants in the following section, we consider $\lambda \mathbf{x}.a$ in $\lambda \mathbf{x}.a(\mathbf{t})$ single symbol when defining positions. A position of a normalized term is a sequence of natural numbers. The set of positions in $t \equiv \lambda \mathbf{x}.a(t_1,\ldots,t_n)$ is defined by $Pos(t) = \{\varepsilon\} \cup \{i \cdot p \mid 1 \leq i \leq n, p \in Pos(t_i)\}$. Let p, q and r be positions. We write $p \leq r$ if $pq = r$. Moreover, if $q \neq \varepsilon$ that is $p \neq r$, we write $p < r$. If $p \nleq r$ and $p \ngeq r$, we write $p|r$.

$$(\lambda \mathbf{x}.a(t_1,\ldots,t_n))|_p \equiv \begin{cases} a(t_1,\ldots,t_n) & \text{if } p = \varepsilon \\ t_i|_q & \text{if } p = iq \end{cases}$$

$Pos_{\mathcal{X}}(t)$ indicates the set of positions $p \in Pos(t)$ such that $t|_p$ is a free variable in a normalized term t. $t[u]_p$ represents the term obtained by replacing $t|_p$ in a normalized term t by normalized term u having the same basic type as $t|_p$. This is defined as follows:

$$(\lambda \mathbf{x}.a(t_1,\ldots,t_n))[u]_p \equiv \begin{cases} \lambda \mathbf{x}.u & \text{if } p = \varepsilon \\ \lambda \mathbf{x}.a(\ldots,t_i[u]_q,\ldots) & \text{if } p = iq \end{cases}$$

$BV_p(t)$ that indicates the set of the positions $p \in Pos(t)$ of bound variables in normalized term t is defined as follows:

$$BV_p(\lambda x_1 \cdots x_m.a(t_1,\ldots,t_n)) \equiv \begin{cases} \emptyset & \text{if } p = \varepsilon \\ \{x_1,\ldots,x_m\} \cup BV_q(t_i) & \text{if } p = iq \end{cases}$$

Let t be a normalized term whose η-normal form is not a variable. We say t is a pattern if $u_1{\downarrow_\eta},\ldots,u_n{\downarrow_\eta}$ are different bound variables for any subterm $F(u_1,\ldots,u_n)$ of t such that $F \in FV(t)$ [8]. Let α be a basic type, $l : \alpha$ be a pattern and $r : \alpha$ be a normalized term such that $FV(l) \supseteq FV(r)$. Then, $l \rhd r : \alpha$ is called a higher-order rewrite rule of type α. A higher-order rewrite system (HRS) is a set of higher-order rewrite rules.

Let $\mathcal{R}$ be a HRS, $l \rhd r$ be a rewrite rule of $\mathcal{R}$ and σ be a substitution. Then, we say $l\sigma\downarrow$ is a redex. If p is a position such that $s \equiv s[l\sigma\downarrow]_p$ and $t \equiv s[r\sigma\downarrow]_p$ then s is reduced to t, denoted by $s \to_R t$ or simply $s \to t$. In case of $p = \varepsilon$, the reduction is denoted by $s \xrightarrow{\varepsilon} t$. In case of $p > \varepsilon$, it is denoted by $s \xrightarrow{>\varepsilon} t$. Since all rewrite rules are with basic type, t is normalized if s is so[5].

The reflexive transitive closure of the reduction relation $\to$ is denoted by $\to^*$. If there exists an infinite reduction sequence $t \equiv t_0 \to t_1 \to \cdots$ from t, we say t has an infinite reduction sequence. If there exists no term which has an infinite reduction sequence, we say $\to$ is terminating. If $\to_R$ is terminating, we also say HRS $\mathcal{R}$ is terminating. We sometimes refer the reduction sequence $A : t_0 \to t_1 \to \cdots \to t_n$ by attaching label A.

Let $l \rhd r$ and $l' \rhd r'$ be rewrite rules. If there exist substitutions σ , σ' and a position $p \in Pos_{\mathcal{X}}(l')$ such that $l\sigma\downarrow \equiv l'|_p(\sigma'|_{\overline{BV_p(l')}})\downarrow$, then we say these rewrite rules overlap[3]. If HRS $\mathcal{R}$ has overlapping rules, we say $\mathcal{R}$ is overlapping. When $\mathcal{R}$ is not overlapping and every rule of $\mathcal{R}$ is left-liner, we say it is orthogonal.

3 Descendant

Considering a reduction $s \to t$, t is obtained by replacing a redex in s by a term. Since the other redexes in s may appear in possibly different positions in t, we must follow the redex positions in order to argue the needed-redex. Thus, the notion of descendants was proposed[3]. From now, we extend the definition of descendants on TRSs to that on HRSs.

In TRSs, it is easy to follow descendants of redexes. However, it is not easy in HRSs because the positions of redexes move considerably by β-reductions taken in a reduction as the following shows.

Example 1. Consider the following HRS $\mathcal{R}_1$,

$$\mathcal{R}_1 = \begin{cases} apply(\lambda x.F(x), X) \rhd F(X) \\ a \rhd b, \end{cases}$$

and a reduction $A_1 : t \equiv apply(\lambda x.f(g(x), x), a) \to f(g(a), a) \equiv s$. Descendants of a redex position a that occurs at position 2 of t are positions 2 and 11 of s as shown in Fig. 1.

In order to follow positions of redexes correctly, we give mutually recursive functions PV and PT each of which returns a set of positions. The function PV that takes a term t, a substitution σ, a variable F and a position p as arguments computes the set of positions of $(F\sigma)|_p$ that occurs in $t\sigma\downarrow$. The function PT that takes a term t, a substitution σ and a position p as arguments computes the set of positions of a $t|_p$ that occurs in $t\sigma\downarrow$. In the previous example, we have $PV(F(X), \{F \mapsto \lambda x.f(g(x), x), X \mapsto a\}, X, \varepsilon) = \{11, 2\}$.

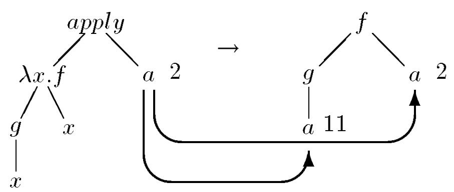

Fig. 1. Descendants

[3] The original definition of overlapping [9] is formal but complicated because $\overline{x_k}$ lifter is used to prohibit the substitution to free variables in a subterm that is bound in the original term.

Definition 1 PV. Let t be a normalized term, σ be a normalized substitution, F be a variable and $p \in Pos(F\sigma)$ be a position.

$$PV(t,\sigma,F,p) = \begin{cases} \{p\} & \text{if } t \equiv F \quad (1) \\ \{iq \mid q \in PV(t_i,\sigma,F,p)\} & \text{if } t \equiv f(t_1,\dots,t_n) \text{ and } n>0 \quad (2) \\ PV(t',\sigma|_{\overline{\{x_1,\dots,x_n\}}},F,p) & \text{if } t \equiv \lambda x_1 \cdots x_n.t' \text{ and } n>0 \quad (3) \\ \bigcup_i \bigcup_{q\in PV(t_i,\sigma,F,p)} PV(t',\sigma',y_i,q) & \text{if } t \equiv G(t_1,\dots,t_n), n>0 \text{ and } F \neq G \\ & \text{where } G\sigma \equiv \lambda y_1 \dots y_n.t' \\ & \text{s.t. } \sigma' = \{y_1 \mapsto t_1\sigma\downarrow,\dots,y_n \mapsto t_n\sigma\downarrow\} \quad (4) \\ (\bigcup_i \bigcup_{q\in PV(t_i,\sigma,F,p)} PV(t',\sigma',y_i,q)) \cup PT(t',\sigma',p) & \\ & \text{if } t \equiv F(t_1,\dots,t_n) \text{ and } n>0 \\ & \text{where } F\sigma \equiv \lambda y_1 \dots y_n.t' \\ & \text{s.t. } \sigma' = \{y_1 \mapsto t_1\sigma\downarrow,\dots,y_n \mapsto t_n\sigma\downarrow\} \quad (5) \\ \emptyset & \text{otherwise} \quad (6) \end{cases}$$

Definition 2 PT. Let t be a normalized term, σ be a normalized substitution, F be a variable and $p \in Pos(t)$ be a position.

$$PT(t,\sigma,p) = \begin{cases} \{\varepsilon\} & \text{if } p = \varepsilon \quad (1) \\ \{iq \mid q \in PT(t_i,\sigma,p')\} & \text{if } t \equiv f(t_1,\dots,t_n), n>0 \text{ and } p = ip' \quad (2) \\ \bigcup_{q\in PT(t_i,\sigma,p')} PV(t',\sigma',y_i,q) & \text{if } t \equiv G(t_1,\dots,t_n), n>0 \text{ and } p = ip' \\ & \text{where } G\sigma \equiv \lambda y_1 \dots y_n.t' \\ & \text{s.t. } \sigma' = \{y_1 \mapsto t_1\sigma\downarrow,\dots,y_n \mapsto t_n\sigma\downarrow\} \quad (3) \end{cases}$$

Example 2. Followings are examples of PV and PT.

1. $PT(f(y),\sigma,\varepsilon) = \{\varepsilon\}$ for any σ by the definition of PT (1).
2. $PV(f(y),\sigma,y,11)$ for any σ,
   $= \{1q \mid q \in PV(y,\sigma,y,11)\}$ by $PV(2)$,
   $= \{111\}$.
3. $PV(F(x),\sigma,F,\varepsilon)$ where $\sigma = \{F \mapsto \lambda y.f(y)\}$,
   $= (\bigcup_{q\in PV(x,\sigma,F,\varepsilon)} PV(f(y),\sigma',y,q)) \cup PT(f(y),\sigma',\varepsilon)$ where $\sigma' = \{y \mapsto x\}$ by PV (5),
   $= (\bigcup_{q\in\emptyset} PV(f(y),\sigma',y,q)) \cup \{\varepsilon\}$ by PV (6) and from 1 of this example,
   $= \{\varepsilon\}$.
4. $PV(g(\lambda x.h(F(x))),\sigma,F,\varepsilon)$ where $\sigma = \{F \mapsto \lambda y.f(y)\}$,
   $= \{1q \mid q \in PV(\lambda x.h(F(x)),\sigma,F,\varepsilon)\}$ by PV (2),
   $= \{1q \mid q \in PV(h(F(x)),\sigma,F,\varepsilon)\}$ by PV (3),
   $= \{11q \mid q \in PV(F(x),\sigma,F,\varepsilon)\}$ by PV (2),
   $= \{11\}$ from 3 of this example.
5. $PV(F(g(\lambda x.h(F(x)))),\sigma,F,\varepsilon)$ where $\sigma = \{F \mapsto \lambda y.f(y)\}$,
   $= (\bigcup_{q\in PV(g(\lambda x.h(F(x))),\sigma,F,\varepsilon)} PV(f(y),\sigma',y,q)) \cup PT(f(y),\sigma',\varepsilon)$ where $\sigma' = \{y \mapsto g(\lambda x.h(f(x)))\}$ by PV (5),
   $= PV(f(y),\sigma',y,11) \cup PT(f(y),\sigma',\varepsilon)$ from 4 of this example,
   $= \{111,\varepsilon\}$ from 1 and 2 of this example.

Definition 3 descendants of HRSs. Let $A : s[l\sigma\downarrow]_u \to_{l \rhd r} s[r\sigma\downarrow]_u$ be a reduction for a rewrite rule $l \rhd r \in \mathcal{R}$, a substitution σ, a term s and positions u, v in s. Then, the set of descendants of v by A is defined as follows:

$$v\backslash A = \begin{cases} \{v\} & \text{if } v \mid u \text{ or } v \prec u \\ \{up_3 \mid p_3 \in PV(r, \sigma, top(l|_{p_1}), p_2)\} & \text{if } v = up_1p_2 \text{ and } p_1 \in Pos_v(l) \\ \emptyset & \text{otherwise.} \end{cases}$$

For a set D of positions, $D\backslash A$ denotes the set $\bigcup_{v \in D} v\backslash A$. For a reduction sequence $B : s \to^* t$, $D\backslash B$ is naturally extended from $D\backslash A$.

Example 3. Consider the HRS $\mathcal{R}_1$ and the reduction sequence A_1 in Example 1. The descendants $2\backslash A_1$ of redex position 2 of t are following:

$2\backslash A_1 = PV(F(X), \{F \mapsto \lambda x.f(g(x), x), X \mapsto a\}, X, \varepsilon)$
$= \bigcup_{q \in PV(X, \{F \mapsto \lambda x.f(g(x),x), X \mapsto a\}, X, \varepsilon)} PV(f(g(x), x), \{x \mapsto a\}, x, q)$ by PV (4)
$= PV(f(g(x), x), \{x \mapsto a\}, x, \varepsilon)$ by PV (1)
$= \{1q' \mid q' \in PV(g(x), \{x \mapsto a\}, x, \varepsilon)\} \cup \{2q'' \mid q'' \in PV(x, \{x \mapsto a\}, x, \varepsilon)\}$ by PV (2)
$= \{11q''' \mid q''' \in PV(x, \{x \mapsto a\}, x, \varepsilon)\} \cup \{2\}$ by PV (1) and (2)
$= \{11, 2\}$ by PV (1)

Example 4. Consider the following HRS $\mathcal{R}_2$, a substitution σ and a reduction sequence A:

$$\mathcal{R}_2 = \{f(g(\lambda x.F(x))) \rhd F(g(\lambda x.h(F(x))))\},$$
$$\sigma = \{F \mapsto \lambda y.f(y)\},$$
$$A : f(g(\lambda x.f(x))) \equiv f(g(\lambda x.F(x)))\sigma\downarrow \to F(g(\lambda x.h(F(x))))\sigma\downarrow \equiv f(g(\lambda x.h(f(x)))).$$

The descendants $11\backslash A$ of position 11 by the reduction sequence A are following:

$$11\backslash A = PV(F(g(\lambda x.h(F(x)))), \sigma, F, \varepsilon) \quad \text{by Definition 3.}$$

From Example 2,

$$11\backslash A = \{111, \varepsilon\}.$$

In the following we are only interested in descendants of redex positions. For convenience we identify redex positions with redexes. We show the property that descendants of a redex are redexes.

Lemma 4. *Descendants are well-defined.*

Proof. We define a measurement of $PT(t, \sigma, p)$ and $PV(t, \sigma, F, p)$ by the term $t\sigma$ and the order $>_{def}$ defined as follows:

$$s >_{def} s' \Leftrightarrow \begin{array}{l} |s\downarrow| > |s'\downarrow|, \text{ or,} \\ |s\downarrow| \geq |s'\downarrow| \ \wedge \ min\beta(s) > min\beta(s'), \end{array}$$

where $min\beta(s)$ is the minimum steps m such that $s \xrightarrow{m}_\beta s\downarrow$.

We can prove that PV or PT recursively called in the right-hand side of their definition is smaller than that in the left-hand side by this measurement. Here, we show this only in the case (4) of Definition 1 of PV by comparing $t\sigma$ in $PV(t, \sigma, F, p)$ with $t'\sigma'$ in $PV(t', \sigma', y_i, q)$ by $>_{def}$. Since $t\sigma \equiv (G(t_1, \ldots, t_n)\sigma) \rightarrow_\beta^* ((\lambda y_1 \cdots y_n.t')(t_1\sigma\downarrow, \ldots, t_n\sigma\downarrow)) \rightarrow_\beta (t'\{y_1 \mapsto t_1\sigma\downarrow, \ldots, y_n \mapsto t_n\sigma\downarrow\}) \equiv t'\sigma'$, we have $t\sigma\downarrow \equiv t'\sigma'\downarrow$ and $min\beta(t\sigma) > min\beta(t'\sigma')$. Hence, we have $t\sigma >_{def} t'\sigma'$. Since we can show the claim in all other cases similarly to the above case, PV and PT are well-defined. Therefore, descendants are well-defined. □

Lemma 5 redex preservation on PV and PT. *Let t be a normalized term, σ be a normalized substitution and F be a variable.*

(a) *Let $F\sigma|_p$ be a redex for a position $p \in Pos(F\sigma)$. Then, for any $q \in PV(t, \sigma, F, p)$ there exists a substitution θ such that $(t\sigma\downarrow)|_q \equiv ((F\sigma)|_p \cdot \theta)\downarrow$. Hence, $(t\sigma\downarrow)|_q$ is a redex.*
(b) *Let $t|_p$ be a redex for a position $p \in Pos(t)$. Then, for any $q \in PT(t, \sigma, p)$ there exists a substitution θ such that $(t\sigma\downarrow)|_q \equiv (t|_p \cdot \theta)\downarrow$. Hence, $(t\sigma\downarrow)|_q$ is a redex.* □

The proof of Lemma 5 is found in the appendix.

Theorem 6 redex preservation of descendants. *Let $A: s \rightarrow t$. If $s|_v$ is a redex then $t|_p$ are also redexes for all $p \in v\backslash A$.*

Proof. This is trivial from Lemma 5. □

From the above, we can say that the definition of descendants is well-defined and descendants of redexes are also redexes.

4 Head normalization

We discuss an extension of head normalization on TRSs to that on HRSs.

4.1 Head-needed redex

Definition 7 head normal form. Let $\mathcal{R}$ be an HRS. A term that cannot be reduced to any redex is said to be in *head normal form.*

Lemma 8. *Let $\mathcal{R}$ be an HRS and a term s be in head normal form. If there exists a reduction sequence $s \rightarrow^* t$ then t is in head normal form.* □

This lemma trivially holds.

Lemma 9. *Let $\mathcal{R}$ be an orthogonal HRS and a term t be in head normal form. If there exists a reduction sequence $s \stackrel{>\varepsilon}{\rightarrow}{}^* t$, s is in head normal form.*

Proof. Assume that s is not in head normal form, then there exists a redex s' such that $s \overset{\geq\varepsilon}{\to}{}^* s'$. Since it is easy to see $\overset{\geq\varepsilon}{\to}$ is confluent, we have $s' \overset{\geq\varepsilon}{\to}{}^* t'$ and $t \overset{\geq\varepsilon}{\to}{}^* t'$. Since $s' \overset{\geq\varepsilon}{\to}{}^* t$ and s' is redex, t' is also redex from orthogonality. It is a contradiction to the fact that t' is in head normal form by Lemma 8. □

Definition 10 head-needed redex. Let $\mathcal{R}$ be an HRS. A redex r in a term t is head-needed, if in every reduction sequence from t to a head normal form a descendant of r is reduced.

Lemma 11. *Let $\mathcal{R}$ be an orthogonal HRS and t be not in head normal form. Then the pattern of the first redex, which appears in every reduction sequence from t to a redex, is unique.*

Proof. From orthogonality. □

Theorem 12. *Let $\mathcal{R}$ be an orthogonal HRS. Every term that is not in head normal form contains a head-needed redex.*

Proof. In similar to the proof of Theorem 4.3 in [7], it is proved by Lemma 11. □

4.2 Head normalizing strategy

The development $\twoheadrightarrow^D$ is defined by the following inference rules:

$$\frac{s_i \twoheadrightarrow^{D_i} t_i \ (i=1,\dots,n)}{a(\overline{s_n}) \twoheadrightarrow^{D} a(\overline{t_n})} \quad D = \{ip \mid p \in D_i\} \qquad (A)$$

$$\frac{s \twoheadrightarrow^{D} t}{\lambda x.s \twoheadrightarrow^{D} \lambda x.t} \qquad (L)$$

$$\frac{s_i \twoheadrightarrow^{D_i} t_i \ (i=1,\dots,n) \quad c(\overline{s_n})=l\theta' \quad c(\overline{t_n})=l\theta \quad (l \rhd r)\in R}{c(\overline{s_n}) \twoheadrightarrow^{D} r\theta} \quad D = \{\varepsilon\} \cup \{ip \mid p \in D_i\} \qquad (R)$$

where D is a set of redex positions.

Proposition 13 development sequence. *If $p_i \not< p_j$ for any $i < j$ then the following holds:* $\twoheadrightarrow^{\{p_1 \cdots p_n\}} = \twoheadrightarrow^{p_1} \cdot \twoheadrightarrow^{p_2} \cdots \twoheadrightarrow^{p_n}$.

Let A and B be developments such that $A : s \twoheadrightarrow t_1$ and $B : s \twoheadrightarrow t_2$. Let D be the set of positions of the redexes contracted in A. The development starting from t_2 in which all redexes at positions $D \backslash B$ are contracted is denoted by $A \backslash B$.

Let A and B be development sequences such that $A : s_1 \twoheadrightarrow^* s_2$ and $B : t_1 \twoheadrightarrow^* t_2$. We say A and B are *permutation equivalent* denoted by $A \simeq B$, if $s_1 \equiv t_1$, $s_2 \equiv t_2$ and $p \backslash A = p \backslash B$ for all redex positions p in s_1. $A; B$ denotes the concatenation of the development sequences A and B. The following lemma corresponds to Lemma 2.4 in [3].

Lemma 14. *Let $\mathcal{R}$ be an orthogonal HRS. If A and B are developments starting from the same term then $A; (B \backslash A) \simeq B; (A \backslash B)$.* □

Definition 15 ∇ and Δ. Let B and D be sets of positions. If $\forall p \in D, \exists q \in B, q \prec p$ then we denote $D_{\nabla B}$ for D. If $\forall p \in D, \forall q \in B, q \not\preceq p$ then we denote $D_{\Delta B}$ for D.

We sometimes write $\twoheadrightarrow^{\nabla}$ and $\twoheadrightarrow^{\Delta}$ for $\twoheadrightarrow^{D_{\nabla B}}$ and $\twoheadrightarrow^{D_{\Delta B}}$, respectively.

Lemma 16. *Let B, D and D' be sets of redex positions of term t such that $D_{\nabla B}$ and $D'_{\Delta B}$. Let $A_1 : t \twoheadrightarrow^{D_{\nabla B}} t_1$ and $A_2 : t_1 \twoheadrightarrow^{D'_{\Delta B}} t_2$ be developments. Then, there exists developments $A_3 : t \twoheadrightarrow^{D'_{\Delta B}} t_3$ and $A_4 : t_3 \twoheadrightarrow^{(D \backslash A_3)_{\nabla (B \backslash A_3)}} t_2$ for some t_3.* □

The proof of Lemma 16 is found in the appendix.

We define an order on development sequences with the same length.

Definition 17. Let $A = A_1; A_2; \cdots; A_n$ and $B = B_1; B_2; \cdots; B_n$ be development sequences with length n. We write $A > B$ if there exists an $i \in \{1, \ldots, n\}$ such that $|A_i| > |B_i|$ and $|A_j| = |B_j|$ for every $i < j \leq n$. We also write $A \geq B$ if $A > B$ or $|A_j| = |B_j|$ for every $1 \leq j \leq n$.

Definition 18. Let A be a development sequence and B be a development starting from the same term. We write $B \bot A$ if any descendant of redexes reduced in B is not reduced in A.

The following two lemmas actually correspond to Lemma 5.4 and 5.5 introduced in [7]. These two lemmas can be proved in the similar way as [7].

Lemma 19. *Let $A : s \twoheadrightarrow^{*} s_n$ and $B : s \twoheadrightarrow t$ be such that $B \bot A$. If s_n is in head normal form then there exists a $C : t \twoheadrightarrow^{*} t_n$ such that $A \geq C$ and t_n is in head normal form.*

Proof. This lemma is proved by using Lemma 14 and Lemma 16. □

Lemma 20. *Let $A : s \twoheadrightarrow^{*} s_n$ and $B : s \twoheadrightarrow t$ be such that $B \not\bot A$. If s_n is in head normal form then there exists a $C : t \twoheadrightarrow^{*} t_n$ such that $A > C$ and t_n is in head normal form.*

Proof. This lemma is proved by using Lemma 14 and Lemma 19. □

Theorem 21. *Let $\mathcal{R}$ be an orthogonal HRS. Let t be a term that has a head normalizing reduction. There is no development sequence starting from t that contains infinitely many head-needed rewriting steps.*

Proof. By using Lemma 19 and Lemma 20, this lemma is proved in the same way as [7]. □

From above, head-needed rewritings are head normalizing strategy in HRS. In other words, we can obtain a head normal form by rewriting head-needed redexes.

Acknowledgement

We thank Prof. Toshiki Sakabe for discussion. This work is partly supported by Grants from Ministry of Education, Science and Culture of Japan #11680352.

References

1. F. Baader and T. Nipkow. Term Rewriting and All That. Cambridge University Press (1998)
2. I. Durand and A. Middeldorp: Decidable Call by Need Computations in Term Rewriting (Extended Abstract). Proc. 14th CADE, LNAI **1249** (1997) 4–18
3. G. Huet and J.-J. Lévy: Computations in Orthogonal Rewriting Systems, I and II. Computational Logic, Essays in Honor of Alan Robinson, The MIT Press (1991) 396–443
4. M. Iwami and Y. Toyama: Simplification Ordering for Higher-order Rewrite systems. IPSJ Trans. on Programming **40** SIG 4 (PRO 3) (1999) 1–10
5. J.-P. Jouannaud and A. Rubio: Rewrite Orderings for Higher-order Terms in η-long β-normal Form and the Recursive Path Ordering. Theoretical Computer Science **208** (1998) 33–58
6. R. Mayr and T. Nipkow: Higher-order Rewrite Systems and their Confluence. Theoretical Computer Science **192** (1998) 3–29
7. A. Middeldorp: Call by Need Computations to Root-Stable Form. Proc. 24th POPL (1997) 94–105
8. D. Miller: A Logic Programming Language with Lambda-Abstraction, Function Variables, and Simple Unification. Journal of Symbolic Logic **1** 4 (1991) 497–536
9. T. Nipkow: Higher-Order Critical Pairs. Proc. 6th IEEE Symp. Logic in Computer Science (1991) 342–349
10. T. Nipkow: Orthogonal Higher-Order Rewrite Systems are Confluent. Proc. Int. Conf. Typed Lambda Calculi and Applications, LNCS **664** (1993) 306–317
11. V. van Oostrom: Confluence for Abstract and Higher-Order Rewriting. PhD thesis, Vrije Universiteit, Amsterdam, (1994)
12. C. Talcott: A Theory of Binding Structures and Applications to Rewriting. Theoretical Computer Science **112** (1993) 99–143

A Proof of Lemma 5

Lemma 5. *Let t be a normalized term, σ be a normalized substitution and F be a variable.*

(a) *Let $F\sigma|_p$ be a redex for a position $p \in Pos(F\sigma)$. Then, for any $q \in PV(t,\sigma,F,p)$ there exists a substitution θ such that $(t\sigma\!\downarrow)|_q \equiv ((F\sigma)|_p \cdot \theta)\!\downarrow$. Hence, $(t\sigma\!\downarrow)|_q$ is a redex.*

(b) *Let $t|_p$ be a redex for a position $p \in Pos(t)$. Then, for any $q \in PT(t,\sigma,p)$ there exists a substitution θ such that $(t\sigma\!\downarrow)|_q \equiv (t|_p \cdot \theta)\!\downarrow$. Hence, $(t\sigma\!\downarrow)|_q$ is a redex.*

Proof. We prove by mutual induction on the structure of $t\sigma$ with $>_{def}$.
(a) Let $F\sigma|_p$ be a redex and q be a position such that $q \in PV(t, \sigma, F, p)$. We show $(t\sigma{\downarrow})|_q \equiv ((F\sigma)|_p)\theta{\downarrow}$ for some substitution θ.

(1) If $t \equiv F$ then $q = p$. Since σ is a normalized substitution, $(t\sigma{\downarrow})|_q \equiv (F\sigma{\downarrow})|_p \equiv (F\sigma)|_p \equiv ((F\sigma)|_p){\downarrow}$ holds.
(2) If $t \equiv f(t_1, \ldots, t_n)$, there exists q' such that $q = iq'$ and $q' \in PV(t_i, \sigma, F, p)$. Since $t\sigma >_{def} t_i\sigma$, there exists θ' such that $(t_i\sigma{\downarrow})|_{q'} \equiv ((F\sigma)|_p)\theta'{\downarrow}$ by induction hypothesis. Then, $(t\sigma{\downarrow})|_q \equiv (f(t_1, \ldots, t_n)\sigma{\downarrow})|_{i\cdot q'} \equiv (t_i\sigma{\downarrow})|_{q'} \equiv ((F\sigma)|_p)\theta'{\downarrow}$ holds.
(3) If $t \equiv \lambda x_1 \cdots x_n.t'$ then $q \in PV(t', \sigma', F, p)$ where $\sigma' = \sigma|_{\overline{\{x_1,\ldots,x_n\}}}$. Since $t\sigma >_{def} t'\sigma'$, there exists θ' such that $(t'\sigma'{\downarrow})|_q \equiv ((F\sigma)|_p)\theta'{\downarrow}$ by induction hypothesis. Then, $(t\sigma{\downarrow})|_q \equiv ((F\sigma)|_p)\theta'{\downarrow}$ holds from $(t\sigma{\downarrow})|_q \equiv (t'\sigma'{\downarrow})|_q$.
(4) If $t \equiv G(t_1, \ldots, t_n)$ and $F \neq G$, there exist i and $q' \in PV(t_i, \sigma, F, p)$ such that $q \in PV(t', \sigma', y_i, q')$ where $\sigma' = \{y_1 \mapsto t_1\sigma{\downarrow}, \ldots, y_n \mapsto t_n\sigma{\downarrow}\}$ and $G\sigma \equiv \lambda y_1 \cdots y_n.t'$. Since $t\sigma >_{def} t'\sigma'$, there exists θ' such that $(t'\sigma'{\downarrow})|_q \equiv ((y_i\sigma')|_{q'})\theta'{\downarrow}$ by induction hypothesis. Then, $(t\sigma{\downarrow})|_q \equiv (t'\sigma'{\downarrow})|_q \equiv ((y_i\sigma')|_{q'})\theta'{\downarrow} \equiv ((t_i\sigma{\downarrow})|_{q'})\theta'{\downarrow}$ holds. Since $t\sigma >_{def} t_i\sigma$, there exists θ'' such that $(t_i\sigma{\downarrow})|_{q'} \equiv ((F\sigma)|_p)\theta''{\downarrow}$ by induction hypothesis. Then, $(t\sigma{\downarrow})|_q \equiv ((F\sigma)|_p)\theta''{\downarrow}\,\theta'{\downarrow} \equiv ((F\sigma)|_p)\theta''\theta'{\downarrow}$ holds.
(5) If $t \equiv F(t_1, \ldots, t_n)$, there exist i and $q' \in PV(t_i, \sigma, F, p)$ such that $q \in PV(t', \sigma', y_i, q')$ or $q \in PT(t', \sigma', p)$ where $\sigma' = \{y_1 \mapsto t_1\sigma{\downarrow}, \ldots, y_n \mapsto t_n\sigma{\downarrow}\}$ and $F\sigma \equiv \lambda y_1 \cdots y_n.t'$. The former case can be shown as (4). Considering the latter case, since $t\sigma >_{def} t'\sigma'$, there exists θ such that $(t'\sigma'{\downarrow})|_q \equiv (t'|_p)\theta{\downarrow}$. Then, $(t\sigma{\downarrow})|_q \equiv (t'\sigma'{\downarrow})|_q \equiv (t'|_p)\theta{\downarrow} \equiv ((\lambda y_1 \cdots y_n.t')|_p)\theta{\downarrow} \equiv ((F\sigma)|_p)\theta{\downarrow}$ holds.
(6) Otherwise, it is obvious from $PV(t, \sigma, F, p) = \emptyset$.

(b) Let $t|_p$ be a redex and q be a position such that $q \in PT(t, \sigma, p)$. We show $(t\sigma{\downarrow})|_q = (t|_p)\theta{\downarrow}$ for some substitution θ.

(1) If $p = \varepsilon$ then $q = \varepsilon$ follows from $PT(t, \sigma, \varepsilon) = \{\varepsilon\}$. Thus, we have $(t\sigma{\downarrow})|_q \equiv t\sigma{\downarrow} \equiv (t|_p)\sigma{\downarrow}$.
(2) If $t \equiv f(t_1, \ldots, t_n)$ and $p = ip'$, there exists q' such that $q = iq'$ and $q' \in PT(t_i, \sigma, p')$. Since $t|_p \equiv t_i|_{p'}$, $t_i|_{p'}$ is a redex. It follows from $t >_{def} t_i$ and induction hypothesis that $(t_i\sigma{\downarrow})|_{q'} \equiv (t_i|_{p'})\theta{\downarrow}$ for some θ. Thus, $(t\sigma{\downarrow})|_q \equiv f(t_1\sigma{\downarrow}, \ldots, t_n\sigma{\downarrow})|_{iq'} \equiv (t_i\sigma{\downarrow})|_{q'} \equiv (t_i|_{p'})\theta{\downarrow} \equiv (f(t_1, \ldots, t_n)|_{ip'})\theta{\downarrow} \equiv (t|_p)\theta{\downarrow}$.
(3) If $t \equiv G(t_1, \ldots, t_n)$ and $p = ip'$, there exists $q' \in PT(t_i, \sigma, p')$ such that $q \in PV(t', \sigma', y_i, q')$ where $\sigma' = \{y_1 \mapsto t_1\sigma{\downarrow}, \ldots, y_n \mapsto t_n\sigma{\downarrow}\}$ and $G\sigma \equiv \lambda y_1 \cdots y_n.t'$. Since $t\sigma >_{def} t'\sigma'$, there exists θ' such that $(t'\sigma'{\downarrow})|_q \equiv ((y_i\sigma')|_{q'})\theta'{\downarrow}$ by induction hypothesis. Thus, $(t\sigma)|_q \equiv (t'\sigma'{\downarrow})|_q \equiv ((y_i\sigma')|_{q'})\theta'{\downarrow} \equiv ((t_i\sigma{\downarrow})|_{q'})\theta'{\downarrow}$ holds. Moreover, since $t\sigma >_{def} t_i\sigma$, there exists a position θ'' such that $(t_i\sigma{\downarrow})|_{q'} \equiv ((t_i|_{p'})\theta''{\downarrow})$. Therefore, $(t\sigma)|_q \equiv ((t_i|_{p'})\theta''{\downarrow})\theta'{\downarrow} \equiv (G(t_1, \ldots, t_n)|_{ip'})\theta''\theta'{\downarrow} \equiv (t|_p)\theta''\theta'{\downarrow}$ holds.

□

B Proof of Lemma 16

Lemma 22. *Let F be a variable, σ be a substitution, and v and v' be positions such that $v' \prec v$. Then, the followings hold;*

(a) If $F\sigma|_{v'}$ and $F\sigma|_v$ are redexes, then $\forall p \in PV(t, \sigma, F, v)$, $\exists p' \in PV(t, \sigma, F, v')$, $p' \prec p$.
(b) If $t|_{v'}$ and $t|_v$ are redexes, then $\forall p \in PT(t, \sigma, v)$, $\exists p' \in PT(t, \sigma, v')$, $p' \prec p$.

Proof. We prove them by mutual induction on the structure of t with $>_{def}$. Firstly consider (a). Let $P = PV(t, \sigma, F, v)$ and $P' = PV(t, \sigma, F, v')$.

(1) In case of $t \equiv F$, we have $P = \{v\}$ and $P' = \{v'\}$. Hence, the claim holds.
(2) Consider the case that $t \equiv f(t_1, \ldots, t_n)$. Let $p \in P$. Then, we have $p = iq$ for some i and $q \in PV(t_i, \sigma, F, v)\}$. Since $t\sigma >_{def} t_i\sigma$, we have $q' \prec q$ for some $q' \in PV(t_i, \sigma, F, v')$ by induction hypothesis. Thus we have $iq' \prec iq = p$ and $iq' \in P'$.
(3) In case of $t \equiv \lambda x_1 \cdots x_n.t'$, we have $P = PV(t', \sigma', F, v)$ and $P' = PV(t', \sigma', F, v')$ where $\sigma' = \sigma|_{\overline{\{x_1,\ldots,x_n\}}}$. Since $t\sigma >_{def} t'\sigma'$, the claim follows from induction hypothesis.
(4) Consider the case that $t \equiv G(t_1, \ldots, t_n)$. Let $p \in P$, $Q = PV(t_i, \sigma, F, v)$ and $Q' = PV(t_i, \sigma, F, v')$. Then, $p \in PV(t', \sigma', y_i, q)$ for some $q \in Q$ and i, where $G\sigma \equiv \lambda y_1 \cdots y_n.t'$ and $\sigma' = \{y_1 \mapsto t_1\sigma\downarrow, \ldots, y_n \mapsto t_n\sigma\downarrow\}$. Since $t\sigma >_{def} t_i\sigma$, we have $q' \prec q$ for some $q' \in Q'$ by induction hypothesis. Since $t\sigma >_{def} t'\sigma'$, it follows from $q' \prec q$ by induction hypothesis that $p' \prec p$ for some $p' \in PV(t', \sigma', y_i, q') \subseteq P$.
(5) In case of $t \equiv F(t_1, \ldots, t_n)$, we only check the subcase that $p \in PT(t', \sigma', v)$ since the other case is proved in similar to the case (4). It follows from $t\sigma >_{def} t'\sigma'$ by induction hypothesis that $p' \prec p$ for some $p' \in PT(t', \sigma, v')$.
(6) In the other cases, that is $F \notin Var(t)$ or $P = P' = \emptyset$, it is obvious.

Next, consider (b). Let $P = PT(t, \sigma, v)$ and $P' = PT(t, \sigma, v')$.

(1) In case of $v' = \varepsilon$, we have $t \equiv f(t_1, \ldots, t_n)$ from the assumption that $t|_{v'}$ and $t|_v$ are redexes. Let $p \in P = PT(f(t_1, \ldots, t_n), \sigma, v)$. Then, we have $p \succ \varepsilon$ from the definition of PT. Thus, the claim follows from $P' = \{\varepsilon\}$.
(2) In case of $t \equiv f(t_1, \ldots, t_n)$ and $v' = iw'$, we have $v = iw$ and $w' \prec w$ for some w. Let $p \in P$. Then, we have $p = iq$ for some $q \in PT(t_i, \sigma, w)$. Since $t\sigma >_{def} t_i\sigma$, we have $q' \prec q$ for some $q' \in PT(t_i, \sigma, w')$ by induction hypothesis. Thus, we have $iq' \prec iq = p$ and $iq' \in P'$.
(3) In case of $t \equiv G(t_1, \ldots, t_n)$ and $v' = iw'$, we have $v = iw$ and $w' \prec w$ for some w. Let $p \in P$. Then, we have $p \in PV(t', \sigma', y_i, q)$ for some $q \in PT(t_i, \sigma, w)$ where $\sigma' = \{y_1 \mapsto t_1\sigma\downarrow, \ldots, y_n \mapsto t_n\sigma\downarrow\}$ and $G\sigma \equiv \lambda y_1 \cdots y_n.t'$. Since $t\sigma >_{def} t_i\sigma$, we have $q' \prec q$ for some $q' \in PT(t_i, \sigma, w')$ by induction hypothesis. Since $t\sigma >_{def} t'\sigma'$, it follows from $q' \prec q$ and induction hypothesis that $p' \prec p$ for some $p' \in PV(t', \sigma', y_i, q')$.

□

Lemma 23. *Let* $A : t \circ\!\!\!\rightarrow^{\{\varepsilon\}} t'$ *be a development,* B *and* D *be sets of redex positions of* t *such that* $D_{\nabla B}$. *Then,* $(D \backslash A)_{\nabla(B \backslash A)}$.

Proof. Let $l \rhd r$ be a rewrite rule, σ be a substitution such that $t \equiv l\sigma\downarrow$ and $t' \equiv r\sigma\downarrow$. Without loss of generality, we can assume that $Dom(\sigma) = Var(l)$ and σ is idempotent, that is $Dom(\sigma) \cap \bigcup_{X \in Dom(\sigma)} Var(X\sigma) = \emptyset$.

Now we prove that $\forall p \in D\backslash A, \exists p' \in B\backslash A, p' \prec p$. Let $p \in D\backslash A$. Then, there exists $q \in D$ such that $p \in q\backslash A$. Since $q \in D_{\nabla B}$, there exists $q' \in B$ such that $q' \prec q$. On the other hand, there exists $p_1 \in Pos_v(l)$ such that $q' = p_1p_2$ and $q = p_1p_2p'_2$ from orthogonality. From the definition of descendants, we have

$$q'\backslash A = PV(r, \sigma, top(l|_{p_1}), p_2), \text{ and}$$
$$q\backslash A = PV(r, \sigma, top(l|_{p_1}), p_2p'_2).$$

By Lemma 22 and the fact that $p \in q\backslash A$, we have $p' \prec p$ such that $p' \in q'\backslash A \subseteq B\backslash A$. Therefore, $(D\backslash A)_{\nabla(B\backslash A)}$ holds. □

Lemma 16. *Let* B, D *and* D' *be sets of redex positions of term* t *such that* $D_{\nabla B}$ *and* $D'_{\Delta B}$. *Let* $A_1 : t \circ\!\!\!\rightarrow^{D_{\nabla B}} t_1$ *and* $A_2 : t_1 \circ\!\!\!\rightarrow^{D'_{\Delta B}} t_2$ *be developments. Then, there exists developments* $A_3 : t \circ\!\!\!\rightarrow^{D'_{\Delta B}} t_3$ *and* $A_4 : t_3 \circ\!\!\!\rightarrow^{(D\backslash A_3)_{\nabla(B\backslash A_3)}} t_2$ *for some* t_3.

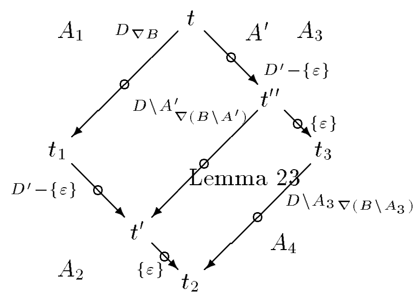

Fig. 2. The proof of Lemma 16

Proof. It is not difficult to show that $A_3 : t \circ\!\!\!\rightarrow^{D'_{\Delta B}} t_3$. We also have $A_4 : t_3 \circ\!\!\!\rightarrow^{(D\backslash A_3)} t_2$ is from Lemma 14. Now, we prove $(D\backslash A_3)_{\nabla(B\backslash A_3)}$ by induction on the structure of term t.

Since $D_{\nabla B}$, we have $\varepsilon \notin D$. In case of $\varepsilon \in B$, we have $D = \emptyset$. Hence, the claim obviously holds. In case of $\varepsilon \notin B \wedge \varepsilon \notin D'$, the claim can be shown by using induction hypothesis. Thus, we consider the most interesting case $\varepsilon \notin D \wedge \varepsilon \in D'$.

From Proposition 13, A_2 and A_3 can be represented as $A_2 : t_1 \circ\!\!\!\rightarrow^{D'-\{\varepsilon\}} t' \circ\!\!\!\rightarrow^{\{\varepsilon\}} t_2$ and $A_3 : t \circ\!\!\!\rightarrow^{D'-\{\varepsilon\}} t'' \circ\!\!\!\rightarrow^{\{\varepsilon\}} t_3$, respectively. Let $A' : t \circ\!\!\!\rightarrow^{D'-\{\varepsilon\}} t''$. We can show that $t'' \circ\!\!\!\rightarrow^{(D\setminus A')\triangledown(B\setminus A')} t'$ in the same way as the case that $\varepsilon \notin D \wedge \varepsilon \notin D'$. We can complete the Fig. 2 by Lemma 23. $\square$

This article was processed using the LaTeX macro package with LLNCS style