Head-Needed Strategy of Higher-Order Rewrite Systems and Its Decidable Classes

HIDETO KASUYA,† 1 MASAHIKO SAKAI‡ 2 and KIYOSHI AGUSA‡ 2

The present paper discusses a head-needed strategy and its decidable classes of higher-order rewrite systems (HRSs), which is an extension of the head-needed strategy of term rewriting systems (TRSs). We discuss strong sequential and NV-sequential classes having the following three properties, which are mandatory for practical use: (1) the strategy reducing a head-needed redex is head normalizing (2) whether a redex is head-needed is decidable, and (3) whether an HRS belongs to the class is decidable. The main difficulty in realizing (1) is caused by the $\beta$-reductions induced from the higher-order reductions. Since $\beta$-reduction changes the structure of higher-order terms, the definition of descendants for HRSs becomes complicated. In order to overcome this difficulty, we introduce a function, PV, to follow occurrences moved by $\beta$-reductions. We present a concrete definition of descendants for HRSs by using PV and then show property (1) for orthogonal systems. We also show properties (2) and (3) using tree automata techniques, a ground tree transducer (GTT), and recognizability of redexes.

1. Introduction

Higher-order rewrite systems (HRSs) 14), an extension of term rewriting systems (TRSs) obtained by introducing higher-order facility, are used in functional programming, logic programming, and theorem proving as a model that contains the notion of $\lambda$-calculus. Properties of HRSs such as termination and confluence have been investigated 7),8),11),14)–16). On the other hand, there have been several studies on reduction strategies of TRSs, which are related lazy evaluation or strict analysis of programs. Huet and Lévy presented the following theorem on the optimal normalizing strategy of an orthogonal TRS 6). They stated that 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 showed the normalization property, whereby the normal form of a given term can always be obtained by repeated reduction of the needed redexes. Middeldorp generalized this result for head-needed reduction, which computes the head normal forms of terms 12).

We discuss strong sequential 6) and NV-sequential classes 18) having the following three properties, which are mandatory for practical use: (1) the strategy reducing a head-needed redex is head normalizing (2) whether a redex is head-needed is decidable, and (3) whether an HRS belongs to the class is decidable.

Arguing the head normalization property requires the concept of descendants of redex in a given reduction sequence. Oostrom showed a definition of a descendant of a Pattern Rewriting System 16), but the definition was abstract. The main difficulty for (1) is caused by the $\beta$-reductions induced from the higher-order reductions. Since $\beta$-reduction changes the structure of the higher-order term, the definition of a descendant of an HRS becomes complicated. Origin tracking 3) has made it possible to represent descendants for HRSs, but this requires rather complicated steps. In the present paper, we introduce a function, PV, to follow occurrence moves caused by $\beta$-reduction sequences. More precisely, given a term $t$, a substitution $\sigma$, a variable $F$, and a position $p$ in $F\sigma$, PV computes positions in a $\beta$-normal form of $t\sigma$ corresponding to the position $p$. The PV function allows us to treat a $\beta$-reduction sequence rather easily. That is, we can use a concrete procedure to calculate descendants of higher-order reductions by using PV. We also define developments of HRS, corresponding to parallel reduction of TRS, and give a concrete proof of the diamond property for the developments. Based on the results of a previous study 10), we also show properties (2) and (3) using tree automata techniques, a ground tree transducer (GTT), and recognizability of redexes.

Oostrom shows the normalizing property of outer-most fair reduction 17). Although, its result strongly relates to the normalizing property of needed reduction, it is difficult to show the head normalizing property of head-needed reduction by Oostrom’s result.

Contributions of the present paper are following.

†1 Faculty of Information Science and Technology, Aichi Prefectural University
‡2 Graduate School of Information Science, Nagoya University
We propose a variant of position system that ignores $\lambda$-position. The system has a benefit that position movement caused by substitution into patterns can be treated first-order case.

We also introduced a recursive definition of patterns. This enables us to prove properties related patterns formally.

We give a simple definition of descendant relation by using function PV that characterizes position movement caused by $\beta$-reduction sequences directly. Unless using our position system and PV, we must compute reverse of $\beta$-reductions, replacement by rewrite rule, and $\beta$-reductions in sequence for descendants.

We show diamond property for HRSs, in which redexes that must be reduced are presented explicitly in each arrow in the diagram of the property.

We also introduced a recursive definition of patterns. This enables us to present the diamond property for HRSs directly. We show the diamond property for HRSs in which redexes that must be reduced are presented explicitly in each arrow in the diagram of the property.

We defined top-down decomposition of development sequences and define an order on them. This order is useful for various proofs dealing with development sequences.

Many of the results presented in the present paper were first reported in Reference. Since a number of proofs in FLOPS2002 turned out to be incomplete, we fixed bugs of proofs and redesigned the order of development. Moreover, we clarified the decidable classes.

2. Preliminaries

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

\[
\tau_s \supseteq S \\
\tau_s \supseteq \{ \alpha \to \alpha' | \alpha, \alpha' \in \tau_s \}
\]

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

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

If $t : \alpha$ is inferred from the rules, then $t$ is a simply typed $\lambda$-term of type $\alpha$. The set of all simply typed $\lambda$-terms is denoted as $\mathcal{T}$. A simply typed $\lambda$-term is called a $\textit{higher-order term}$ or a $\textit{term}$. We use the concepts of bound variables and free variables. The sets of bound and free variables occurring in a term $t$ are denoted as $BV(t)$ and $FV(t)$, respectively. The set $FV(t) \cup BV(t)$ is denoted as $\mathit{Var}(t)$. A higher-order term without free variables is called a $\textit{ground term}$. If a term $s$ is generated by renaming bound variables in a term $t$, then $s$ and $t$ are equivalent and are denoted as $s \equiv t$. We use $F, G, X, Y, Z$ and $M$ for free variables, and $x, y, z$ for bound variables unless it is known to be free or bound from other conditions. We sometimes write $\overline{x}$ for a sequence $x_1 x_2 \cdots x_m$ ($m \geq 0$). We also use $c, d, f, g, h$ and $a$ for function symbols and $a$ for a variable or a function symbol.

$\beta$-reduction is the operation that replaces $(\lambda x. s) t$ in a term by $s[x := t]$, where $(\lambda x. s)$ is called a $\beta$-redex. Let $s$ be a term of type $\alpha \to \alpha'$, and let $x \notin \mathit{Var}(s)$ be a variable of type $\alpha$. Then, $\eta$-expansion is the operation that replaces $s$ in a term by $\lambda x. (sx)$ if the term produces no new $\beta$-redex. A term is said to be $\eta$-long, if the term is in normal form with respect to $\eta$-expansion. In addition, a term is said to be normalized if the term is in $\beta\eta$-long normal form. A normalized term of $t$ is denoted as $\tilde{t}$. Each higher-order term has a unique normalized term.

A mapping $\sigma : X \mapsto T$ from variables to higher-order terms is called a substitution if $\sigma(X)$ has the same type as $X$ and the domain $\mathit{Dom}(\sigma) = \{ X \mid X \notin \mathit{Var}(\sigma) \}$ is finite. If $\mathit{Dom}(\sigma) = \{ X_1, \ldots, X_n \}$ and $\sigma(X_i) \equiv t_i$, we also write the mapping as $\sigma = \{ X_1 \mapsto t_1, \ldots, X_n \mapsto t_n \}$. Let $W$ be a set of variables, and let $\sigma$ be a substitution. We write $\sigma|_W$ for the substitution obtained by restricting the domain of $\sigma$ to $\mathit{Dom}(\sigma) \cap W$ and write $\sigma|_{\overline{X}W}$ for that obtained by restricting the domain of $\sigma$ to $\mathit{Dom}(\sigma) \cap (X - W)$. For a substitution $\sigma$, the set of free variables in the range of $\sigma$ is defined by $\mathit{V Ran}(\sigma) = \bigcup_{X \in \mathit{Dom}(\sigma)} \mathit{FV}(\sigma(X))$.

A substitution $\sigma = \{ X_1 \mapsto t_1, \ldots, X_n \mapsto t_n \}$ is extended to a mapping $\tilde{\sigma}$ from higher-order terms to higher-order terms as follows:

\[
\tilde{\sigma}(t) = ((\lambda X_1 \cdots X_n . t) t_1 \cdots t_n)|_\beta
\]

Generally, when we extend a substitution $\sigma$ to $\tilde{\sigma}$, we need the condition whereby the domain and range of $\sigma$ do not contain any bound variables of a term to which the substitution $\tilde{\sigma}$ is applied. Here, note that when we adopt the above definition of $\tilde{\sigma}$ obtained using $\beta$-reduction, we need not mention the condition explicitly. The above condition can always be satisfied by appropriately renaming.
the bound variables.

In the following, we will write simply \( \sigma \) for \( \tilde{\sigma} \) and \( t\sigma \) for \( \sigma(t) \). A substitution \( \sigma \) is said to be normalized if \( \sigma(X) \) is normalized for all \( X \in \text{Dom}(\sigma) \).

Each normalized term can be represented by the form \( \lambda x_1 \cdots x_m. \cdot \cdot \cdot (a_1) \cdots t_n \), where \( m, n \geq 0, a \in \mathcal{F} \cup \mathcal{X} \), and \( (a_1) \cdots t_n \) is a basic type. In the present section, we denote the term \( t \) by \( \lambda x_1 \cdots x_m.a(t_1, \ldots, t_n) \). The top symbol of \( t \) is defined as \( \text{top}(t) \equiv a \).

We define the notion of positions in normalized terms 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 given in the following section, we ignore lambda bindings in assigning positions to terms. A position of a normalized term is a sequence of natural numbers. The set of positions in \( t \equiv \lambda x.a(t_1, \ldots, t_n) \) is defined by \( \text{Pos}(t) = \{ \varepsilon \} \cup \{ ip \mid 1 \leq i \leq n, p \in \text{Pos}(t_i) \} \). Let \( p \) and \( r \) be positions. We write \( p \preceq r \) if \( pq = r \) for some position \( q \). Moreover, if \( q \neq \varepsilon \), that is, if \( p \neq r \), we write \( p < r \). If \( p \not\equiv r \) and \( p \not\equiv r \), we write \( p \downarrow r \). The subterm \( t\downarrow p \) of \( t \) at \( p \) is defined as follows:

\[
(t\downarrow p)|_p \equiv \begin{cases} a(t_1, \ldots, t_n) & \text{if } p = \varepsilon \\ t_i|_q & \text{if } p = iq \\ \end{cases}
\]

\( \text{Pos}_{\text{FV}}(t) \) indicates the set of all positions \( p \in \text{Pos}(t) \) such that \( \text{top}(t|_p) \) is a free variable in a normalized term \( t \). \( t|_p \) represents the term obtained by replacing \( t\downarrow p \) in a normalized term \( t \) by normalized term \( u \) having the same basic type as \( t\downarrow p \). This is defined as follows:

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

Let \( t \) be a normalized term such that \( \text{top}(t) \in \mathcal{F} \), and let \( u_i|_q \) denote the \( \eta \)-normal form of \( u \) in \( \lambda x. u \). Here, \( t \) is said to be a pattern if \( u_1|_q, \ldots, u_n|_q \) are different bound variables for the arguments \( u_i \) of each free variable \( F \) in \( t \equiv C[F(u_1, \ldots, u_n)] \). Moreover, \( t \) is said to be fully-extended if \( u_1|_q \cdots u_n|_q \) is a sequence of all bound variables in the scope of \( C \). The recursive definition of patterns is based on the concept of the \( B \)-pattern. Let \( B \) be a set of variables. Then, the set of \( B \)-patterns is defined as follows:

\[
(1) \ F(t_1, \ldots, t_n) \text{ is a } B \text{-pattern if } F \not\in B \text{ and } t_1, \ldots, t_n \text{ are } \eta \text{-normal forms of pairwise distinct variables in } B, \\
(2) \ a(t_1, \ldots, t_n) \text{ is a } B \text{-pattern if } a \in \mathcal{F} \cup B \text{ and } t_1, \ldots, t_n \text{ are } B \text{-patterns,} \\
(3) \ \lambda x_1 \cdots x_n.t \text{ is a } B \text{-pattern if } t \text{ is a } (B \cup \{ x_1, \ldots, x_n \}) \text{-pattern.}
\]

Patterns are defined using the concept of \( B \)-patterns as follows: \( t \) is a pattern if and only if \( t \) is a \( B \)-pattern and \( \text{top}(t) \in \mathcal{F} \). Furthermore, \( t \) is fully-extended, if \( t \) satisfies (1') rather than (1), where

\[
(1') \ F(t_1, \ldots, t_n) \text{ is a } B \text{-pattern if } F \not\in B \text{, } \{ t_1, \ldots, t_n \} = \{ x \mid x \in B \}, \text{ and } t_i \neq t_j \text{ for } i \neq j.
\]

Let \( \alpha \) be a basic type, let \( l : \alpha \) be a pattern, and let \( r : \alpha \) be a normalized term such that \( \text{FV}(l) \supseteq \text{FV}(r) \). Then, \( l \triangleright r : \alpha \) is called a higher-order rewrite rule of type \( \alpha \). A higher-order rewrite system (HRS) is a set of higher-order rewrite rules. Let \( \mathcal{R} \) be an HRS, let \( l \triangleright r \) be a rewrite rule of \( \mathcal{R} \), and let \( \sigma \) be a substitution. Then, \( l\sigma \downarrow r \) is said to be a redex. If \( p \) is a position such that \( s \equiv s|_{\sigma |_p} \) and \( t \equiv s|_{\sigma \upharpoonright_p} \), then \( s \) can be reduced to \( t \), which is denoted as \( s \xrightarrow{\mathcal{R}} t \), or simply \( s \xrightarrow{\mathcal{R}} t \). Since all rewrite rules are of basic type, \( t \) is normalized if \( s \) is so.

The reflexive transitive closure of the reduction relation \( \rightarrow \) is denoted as \( \rightarrow^* \).

If there exists an infinite reduction sequence \( t \equiv t_0 \rightarrow t_1 \rightarrow \cdots \) from \( t \), \( t \) is said to have an infinite reduction sequence. If there exists no term that has an infinite reduction sequence, \( \rightarrow \) is said to be terminating. If \( \rightarrow_{\mathcal{R}} \) is terminating, HRS \( \mathcal{R} \) is also said to be \( \mathcal{R} \) terminating. We sometimes refer to a reduction sequence \( A : t_0 \rightarrow t_1 \rightarrow \cdots \rightarrow t_n \) by the label \( A \). We sometimes denote the \( i \)-th reduction \( t_{i-1} \rightarrow t_i \) as \( A_i \). \( A \) is also denoted as \( A_1; A_2; \cdots; A_n \), where \( A_1; B \) denotes the concatenation of sequences \( A \) and \( B \).

Let \( BV_p(t) \) denote the set of variables that appears as lambda abstractions in the path from the position \( \varepsilon \) to \( p \) in normalized term \( t \). \( BV_p(t) \) is defined as follows:

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

Let \( l \triangleright r \) and \( l' \triangleright r' \) be rewrite rules. If there exist substitutions \( \sigma \) and \( \sigma' \) and
a position \( p \not\in Pos_{F}(t') \) such that \( l\sigma \equiv l'|_{\mathcal{B}(t')} \), then these rewrite rules are said to overlap\(^1\). If HRS \( \mathcal{R} \) has overlapping rules, \( \mathcal{R} \) is said to be overlapping. When \( \mathcal{R} \) is not overlapping and every rule of \( \mathcal{R} \) is left-linear, \( \mathcal{R} \) is said to be orthogonal.

3. Head Normalizing Strategy

3.1 Descendant

Considering a reduction \( s \to t \), the term \( t \) is obtained by replacing a redex in \( s \) by a term. Since the other redexes in \( s \) may appear in different positions in \( t \), we must take note of the redex positions in order to discuss the needed redex. Thus, the concept of descendants was proposed\(^5,6\). In the sequel, we extend the definition of descendants of TRSs to that of HRSs.

In TRSs, it is easy to track descendants of redexes. However, this is not easy in HRSs because the positions of redexes move considerably by \( \beta \)-reductions taken in a reduction.

**Example 1** Consider the following HRS \( \mathcal{R}_1 \),

\[
\mathcal{R}_1 = \left\{ \begin{array}{l}
apply(\lambda x. F(x), X) | F(X) \\

\end{array} \right\}
\]

and a reduction \( A_1 : s \equiv apply(\lambda x. f(g(x), x), c) \to f(g(c), c) \equiv t. \) Descendants of a redex \( c \) that occurs at position 2 of \( s \) are positions 2 and 11 of \( t \). As shown in Fig. 1, there is no descendant of redex position \( \varepsilon \) because it is reduced.

In order to follow the positions of redexes correctly, we present mutually recursive functions \( PV \) and \( PT \), each of which returns a set of positions. The function \( PV \) that takes a term \( t \), a substitution \( \sigma \), a variable \( F \), and a position \( p \) as arguments computes the set of positions in \( t\sigma \) that originate from \((F\sigma)|_p\). The function \( PT \) that takes a term \( t \), a substitution \( \sigma \), and a position \( p \) as arguments computes the set of positions in \( t\sigma \) that originate from \( t|_p \). In Example 1, we have \( PV(F(X), \{ F \mapsto \lambda x. f(g(x), x) \mapsto a \}, X, \epsilon) = \{ 11, 2 \} \).

**Definition 1** \( PV \) Let \( t \) be a normalized term, let \( \sigma \) be a normalized substitution, and let \( F \) be a variable. The function \( PV \) is defined as follows for a position \( p \in Pos(F\sigma) \).

\[
PV(F, \sigma, F, p) = \{ p \} \quad (PV1)
\]

\[
PV(a(t_1, \ldots, t_n), \sigma, F, p) = \bigcup \{ iq \mid q \in PV(t_i, \sigma, F, p) \} \quad (PV2)
\]

\[
PV(\lambda x_1 \cdots x_n. t', \sigma, F, p) = PV(t', \sigma|_{x_1 \cdots x_n}, F, p) \quad (PV3)
\]

\[
PV(G(\varepsilon) \equiv y_1 \cdots y_n. t' \equiv \bigcup \{ PV(t', \sigma', y_1, PV(t_i, \sigma, F, p)) \cup PT(t', \sigma', p) \} \quad (PV4)
\]

\[
PV(F(t_1, \ldots, t_n), \sigma, F, p) = \bigcup \{ PV(t_1, t', \sigma', y_1, PV(t_i, \sigma, F, p)) \} \quad (PV5)
\]

\[
PV(t, \sigma, F, p) = \emptyset \quad (PV6)
\]

Here, \( PV(t, \sigma, F, p) \) denotes \( \bigcup_{p \in F} PV(t, \sigma, F, p) \) for a set \( P \) of positions.

**Definition 2** \( PT \) Let \( t \) be a normalized term, and let \( \sigma \) be a normalized substitution. The function \( PT \) is defined as follows for a position \( p \in Pos(t) \).

For \( p = \varepsilon \),

\[
PT(t, \varepsilon, p) = \{ \} \quad (PT1)
\]

For \( p \not= \varepsilon \), (let \( p = \text{ip} \))

\[
PT(a(t_1, \ldots, t_n), \sigma, p') = \{ iq \mid q \in PT(t_i, \sigma, p') \} \quad (PT2)
\]

\[
PT(\lambda x_1 \cdots x_n. t', \sigma, p) = PT(t', \sigma|_{x_1 \cdots x_n}, p) \quad (PT3)
\]

\(^1\) The original definition of overlapping\(^4\) is formal but complicated because the concept of the lifter is used to prohibit the substitution to free variables in a subterm that is bound in the original term.
In the remainder of the present paper, the well-founded order
and
(9)
\( \sigma \)

For any substitution
(3)
For any substitution
\( \sigma \)
\( \sigma \)
by (PV1).

(4) For any substitution, \( \sigma, \sigma, \sigma, F, y, p \) = \( \emptyset \) by (PV6).

(6) \( PV(F(x), x, x, y) = \{ x \} \) by (PV5), (5), (4), and (1).

(7) \( PV(h(F(x)) \sigma, 1, F, \emptyset) = \{ 1 \} \) by (PV2) and (6).

(8) \( PV(\lambda x h(F(x)) \sigma, 1, F, \emptyset) = \{ 1 \} \) by (PV3) and (7).

(9) \( PV(g(x h(F(x)) \sigma, 1, F, \emptyset) = \{ 1 \} \) by (PV2) and (8).

(10) \( PV(F(y), 2, y, \emptyset) = \{ 1 \} \) by (PV2) and (2).

(11) \( PV(F(y), 3, y, \emptyset) = \{ 1 \} \) by (PV5), (10), (9), and (1).

Let us demonstrate that \( PV \) and \( PT \) are well-defined. For this purpose, we introduce a well-founded order \( \triangleright_{\beta} \) on pairs of a term and a substitution:

\[
(s, \theta) \triangleright_{\beta} (t, \sigma) \iff s \triangleright \theta \rightarrow s \sigma \text{ or } s \triangleright \sigma \triangleright t, \sigma,
\]

where \( \triangleright \) is the proper subterm relation of \( \triangleright \) defined as follows:

\[
s \triangleright t \iff \begin{cases} 
  s \equiv t, \\
  s \equiv \lambda x_1 \cdots x_n. s' \land s \triangleright t \lor s \equiv s_1 \land \exists s_2 \triangleright \sigma \end{cases}
\]

In the remainder of the present paper, the well-founded order \( \triangleright_{\beta} \) will play an important role in proving claims in the form of \( \forall t, \forall \sigma P(t, \sigma) \). These proofs will use Noetherian induction over the set of pairs \( (t, \sigma) \) with the order \( \triangleright_{\beta} \).

**Lemma 1** \( PV \) and \( PT \) are well-defined.

**Proof.** First, we show the termination of the recursive calls in the definitions of \( PV \) and \( PT \) by induction on \( (t, \sigma) \) with \( \triangleright_{\beta} \). This is trivial for the cases of Definition 1 and 2, except for (PV4), (PV5), and (PT4). Consider the case of (PV4), where we have two recursive calls of PV. Let \( t \equiv G(t_1, \ldots, t_n) \), \( \sigma \equiv \lambda y_1 \cdots y_n. t' \), and \( \sigma' = \{ y_1 \mapsto t_1 \sigma \downarrow, \ldots, y_n \mapsto t_n \sigma \downarrow \} \). Since \( \sigma \equiv (G(t_1, \ldots, t_n)) \sigma \equiv (\lambda y_1 \cdots y_n. t')(t_1, \ldots, t_n, \sigma) \rightarrow t', \sigma' \), we have \( (t, \sigma) \triangleright_{\beta} (t', \sigma') \). We also have \( \sigma \equiv (\lambda y_1 \cdots y_n. t')(t_1, \ldots, t_n, \sigma) \triangleright t_1 \sigma \). Thus, we know the termination of computing PV for the case of (PV4). The proofs for the cases of (PV5) and (PT4) can be performed in a manner similar to that described above.

Next, we show the following two claims:

\[
PV(t, \sigma, F, p) \subseteq Pos(\sigma) \text{ for } p \in Pos(F \sigma)
\]

\[
PT(t, \sigma, P) \subseteq Pos(\sigma) \text{ for } p \in Pos(t).
\]

This can be shown by simultaneous induction on the well-founded order \( \triangleright_{\beta} \) over the pairs \( (t, \sigma) \). Here, we give the proof only for the case (PV4). Let \( p \in Pos(F \sigma), t \equiv G(t_1, \ldots, t_n), \sigma \equiv \lambda y_1 \cdots y_n. t', \sigma' = \{ y_1 \mapsto t_1 \sigma \downarrow, \ldots, y_n \mapsto t_n \sigma \downarrow \} \). Since \( (t, \sigma) \triangleright_{\beta} (t_1, \sigma), \sigma \), we have PV \( PV(t_1, \sigma, F, p) \subseteq Pos(t \sigma) \downarrow \) by induction. Let \( q \in PV(t_1, \sigma, F, p) \). Then, PV \( (t', \sigma', y, q) \subseteq Pos(t' \sigma') \) follows from induction, because \( (t, \sigma) \triangleright_{\beta} (t', \sigma') \). The claim follows from \( \sigma \equiv t' \sigma' \). The claims for the cases (PV5) and (PT4) can be shown in a similar manner.

Thanks to the position system that ignores \( \lambda \)-positions, we have the following lemma on PV.

**Lemma 2** If \( l \) is a pattern, then \( PV(l, \sigma, F, p) = \{ p', p | t o p(l_{\text{prim}}) = F \} \) for every \( F \in FV(l) \).

The proof of Lemma 2 is found in Appendix A.1.

Next, we present a definition of descendants.

**Definition 3** (descendants of HRSs) Let \( A : s[\sigma] \downarrow u_\sigma \downarrow r \downarrow s[\sigma] \downarrow u \) be a reduction with a rewrite rule \( l \bullet r \in R \), a substitution \( \sigma \), a term \( s \), and a position \( u \) in \( s \). The set of descendants of a position \( v \) in \( Pos(s[\sigma] \downarrow u) \) by \( A \) is then defined as follows:

\[
v \setminus A = \begin{cases} 
  \{ v \} & \text{if } v \mid u \lor v \not< u \\
  \{ up | p_3 \in PV(r, \sigma, F, p_2) \} & \text{if } v = up \text{ and } p_3 \in PV(t, \sigma, F, p_2) \\
  \emptyset & \text{for some } F \in FV(l) \text{ and } p_2 \in Pos(F \sigma) \\
  \end{cases}
\]

This definition of descendants is rather general, because we have not considered
definition is simplified as follows by Lemma 2:

\[
v \setminus A = \begin{cases} 
  \{v\} & \text{if } v \mid u \text{ or } v \prec u \\
  \{ap : p \in PV(r, \sigma, \tau) \cap \text{Pos}(l)\} & \text{if } v = ap \text{ and } p \in \text{Pos}(l) \\
  \emptyset & \text{otherwise.}
\end{cases}
\]

For a set \( D \) of positions, \( D \setminus A \) denotes the set \( \bigcup_{v \in D} v \setminus A \). For a reduction sequence \( B : s \rightarrow^* t \), \( D \setminus B \) is naturally defined.

**Example 3** Consider the HRS \( \mathcal{R}_1 \) and the reduction sequence \( A_1 \) in Example 1. The descendants of a redex position 2 in \( s \) by the reduction sequence \( A_1 \) are as follows:

\[
2 \setminus A_1 = PV(F(X), \sigma, X, \varepsilon) = \{11, 2\}
\]

where \( \sigma = \{F \mapsto \lambda x.f(x), X \mapsto \epsilon\} \).

**Example 4** Consider the following HRS \( \mathcal{R}_2 \), a substitution \( \sigma \), and a reduction sequence \( A \):

\[
\begin{align*}
\mathcal{R}_2 &= \{ f(g(\lambda x.F(x))) \triangleright F(g(\lambda x.F(x))) \}, \\
\sigma &= \{ F \mapsto \lambda y.f(y) \}, \\
A &= \{ f(g(\lambda x.f(x))) \equiv f(g(\lambda x.F(x)))\sigma \}
\end{align*}
\]

The descendants of position 11 by the reduction sequence \( A \) are \( 11 \setminus A = \{111, \varepsilon\} \) because \( PV(F(g(\lambda x.F(x))), \sigma, F, \varepsilon) = \{111, \varepsilon\} \) from Example 2.

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

**Theorem 1 (redex preservation of descendants)** Let \( \mathcal{R} \) be an orthogonal HRS, and let \( A : s \rightarrow^* t \) be a reduction. If \( s|v \) is a redex, then \( t|p \) are also redexes for all \( p \in v \setminus A \).

Note that if \( u \prec v \), then \( t|p \) is an instance of \( s|v \). On the other hand, \( t|p = s|v \) for TRSs. The proof of Theorem 1 is found in Appendix A.2.

**3.2 Development and Its Properties**

Middendorp \(^{12}\) discussed the head normalization of TRS using the concept of parallel reduction \( \triangleright \). He used the property whereby descendants of redexes on parallel positions are redexes on parallel positions in order to show that the head-needed strategy is normalizing. However, the property does not hold in HRSs because of \( \beta \)-reduction. Thus, we cannot discuss the head normalization of HRS by directly following the discussions of Middendorp. Thus, we introduce the concept of development of HRS.

Now, we formalize the development with annotated redex positions and show the diamond property. The development \( \triangleright^D \) of normalized terms is defined by the following inference rules:

\[
\begin{align*}
A : s \triangleright^D t_1 \quad (i = 1, \ldots, n) & \quad D = \bigcup \{ip \in D_i\} \\
A' : s' \triangleright^D t' & \quad (L) \\
\lambda x_1 \cdots x_n s' \triangleright^D \lambda x_1 \cdots x_n t' & \quad (R)
\end{align*}
\]

where \( D \) is a set of redex positions. Sometimes it is convenient to use the following \( (R') \) for \( (R) \).

\[
\begin{align*}
A'' : l\theta_1 \triangleright^D l\theta_2 \quad (\bullet \tau) \in R & \quad (R') \\
\theta_1 \triangleright \theta_2 \quad \varepsilon \in D' \land D = D' \cup \{\varepsilon\}
\end{align*}
\]

The development \( \triangleright^D \) is also simply denoted as \( \triangleright \).

The descendants of developments are defined as follows.

**Definition 4 (descendants of developments)** Let \( p \) be a position in \( s \), and let \( A : s \triangleright^D t \) be a development. Then, the set of descendants of \( p \) by \( A \) is defined as follows:

\[
p \setminus A = \begin{cases} 
  \{\varepsilon\} & \text{in case of (A) and } p = \varepsilon \\
  \{ip | p' \in A_i\} & \text{in case of (A) and } p = ip' \\
  p \setminus A' & \text{in case of (L)} \\
  (p \setminus A') \setminus A'' & \text{in case of (R') where } A'' : l\theta_1 \triangleright \theta_1
\end{cases}
\]

where \( A_i, A', \) and \( A'' \) are reductions shown in the definition of developments.

Let \( A \) and \( B \) be developments such that \( A : s \triangleright^D t_1 \) and \( B : s \triangleright^D t_2 \). The
development starting from \( t_2 \), in which all redexes at positions \( D\backslash B = \{ p\backslash B \mid p \in D \} \) are contracted, is denoted as \( A\backslash B \). Let \( A \) and \( B \) be development sequences such that \( A : s_1 \Rightarrow^* s_2 \) and \( B : t_1 \Rightarrow^* t_2 \). Here, \( A \) and \( B \) are said to be permutation equivalent, which is denoted as \( A \simeq B \), if \( s_1 \equiv t_1 \), \( s_2 \equiv t_2 \) and \( p\backslash A = p\backslash B \) for every redex position \( p \) in \( s_1 \). The following lemma corresponds to Lemma 2.4 in Reference 6, which shows the diamond property (15)(16). Raamsdonk also reported the diamond property of development (19). However, our lemma shows not only the existence of reduction sequences, but also that the descendants of a redex in \( s \) with respect to two development sequences from \( s \) to \( u \) are the same.

**Lemma 3 (diamond property)** Let \( R \) be an orthogonal HRS. If \( A \) and \( B \) are developments starting from the same term, then \( A; (B\backslash A) \simeq B; (A\backslash B) \). □

The proof of Lemma 3 is found in Appendix A.3.

### 3.3 Head-Needed Redex

We extend the notion of head normalization of TRSs to that of HRSs. The remainder of this section assumes the orthogonality of HRSs \( R \).

**Definition 5 (head normal form)** Let \( R \) be an HRS. A term that cannot be reduced to any redex is said to be in head normal form.

**Lemma 4 (Reference 9)** Let \( t \) be in head normal form. If there exists a reduction sequence \( s \Rightarrow^* t \), \( s \) is in head normal form. □

**Definition 6 (head-needed redex)** A redex \( r \) in a term \( t \) is head-needed if a descendant of \( r \) is reduced in every reduction sequence from \( t \) to a head normal form.

**Lemma 5** Let \( t \) be in non-head-normal form. Then, the pattern of the first redex, which appears in every reduction sequence from \( t \) to a redex, is unique.

**Proof.** Similar to the proof of Lemma 4.2 in Ref.12), the lemma is proved by Theorem 1 and orthogonality. □

**Proposition 1** Let \( R \) be fully extended. Assume that we rewrite a term \( s \) at position \( p \) by some rewrite rule and obtain the term \( t \). If \( t \equiv \sigma \downarrow \) for a substitution \( \sigma \) and a fully-extended pattern \( l \) such that \( p \notin \text{Pos}_F(l) \), then there exists some substitution \( \sigma' \) such that \( s \equiv \lambda \sigma' \downarrow \).

**Proof.** We prove the more general claim.

**Claim.** Let \( B \) be a set of variables, and let \( l \) be a fully extended linear \( B \)-pattern. If \( s \not\equiv \sigma \downarrow \), \( \text{top}(l(p)) \not\in B \cup F \), and \( (\text{Dom}(\sigma) \cup \text{VRan}(\sigma)) \cap B = \emptyset \), then there exists a substitution \( \sigma' \) such that \( s \equiv \lambda \sigma' \downarrow \) and \( (\text{Dom}(\sigma') \cup \text{VRan}(\sigma')) \cap B = \emptyset \).

Our proposition is the claim for the case in which \( B = \emptyset \). The claim can be proved by induction on the structure of \( l \).

(1) Let \( l \equiv F(x_1 \downarrow, \ldots, x_n \downarrow) \), \( B = \{ x_1, \ldots, x_n \} \), and \( F \not\in B \). Consider the substitution \( \sigma' = \{ F \mapsto \lambda x_1 \cdots x_n . s \} \). Then, \( \lambda \sigma' \downarrow \equiv (\lambda x_1 \cdots x_n . s)(x_1 \downarrow, \ldots, x_n \downarrow) \equiv s \), i.e., \( \lambda \sigma' \downarrow = s \).

(2) Let \( l \equiv a(t_1, \ldots, t_n) \) for \( a \in F \cup B \) and \( n > 0 \). Since \( (\text{Dom}(\sigma) \cup \text{VRan}(\sigma)) \cap B = \emptyset \), the rewriting \( s \overset{\sigma}{\rightarrow} \lambda \sigma \downarrow \) means \( s \equiv a(s_1, \ldots, s_n) \) for some \( s_1, \ldots, s_n \), and \( \lambda \sigma \downarrow \equiv a(t_1 \sigma \downarrow, \ldots, t_n \sigma \downarrow) \), where \( s_j \overset{\sigma}{\rightarrow} t_j \sigma \downarrow \) for \( j \) such that \( p = j \), and for any \( i \neq j \), \( t_i \sigma \downarrow \equiv s_i \). By induction, there exists a substitution \( \sigma'' \) such that \( t_i \sigma'' \downarrow \equiv s_j \) and \( (\text{Dom}(\sigma'') \cup \text{VRan}(\sigma'')) \cap B = \emptyset \). Hence, \( s \equiv a(s_1, \ldots, t_j \sigma'' \downarrow, \ldots, s_n) \equiv a(t_1 \sigma'' \downarrow, \ldots, t_n \sigma'' \downarrow) \). Consider the substitution \( \sigma' \) such that \( \sigma' = \{ x \mapsto \sigma''(x) \mid x \in \text{Var}(t_j) \} \cup \{ x \mapsto \sigma(x) \mid x \notin \text{Var}(t_j) \wedge x \in \text{Var}(t_i) \) for some \( i \neq j \). From the linearity of \( l \), \( \sigma' \) holds \((\text{Dom}(\sigma') \cup \text{VRan}(\sigma')) \cap B = \emptyset \) and \( s \equiv a(t_1 \sigma' \downarrow, \ldots, t_n \sigma' \downarrow) \equiv \lambda \sigma' \downarrow \).

(3) Let \( l \equiv \lambda \sigma_1 \cdots x_n . t \), where \( t \) is a \((B \cup \{ x_1, \ldots, x_n \})\)-pattern. Let \( s \equiv \lambda \sigma_1 \cdots x_n . s' \), and let \( \sigma'' = \sigma | \{ x_1 \cdots x_n \} \). Then, \( (\text{Dom}(\sigma'') \cup \text{VRan}(\sigma'')) \cap (B \cup \{ x_1, \ldots, x_n \}) = \emptyset \) and \( s' \overset{\sigma''}{\rightarrow} \lambda \sigma'' \downarrow \). By induction hypothesis, there exists a substitution \( \sigma'' \) such that \( s' \equiv \lambda \sigma'' \downarrow \) and \( (\text{Dom}(\sigma'') \cup \text{VRan}(\sigma'')) \cap (B \cup \{ x_1, \ldots, x_n \}) = \emptyset \). Thus, \( s \equiv \lambda \sigma_1 \cdots x_n . s' \equiv \lambda \sigma_1 \cdots x_n . (\sigma'' \downarrow) \equiv \lambda \sigma' \downarrow \).

**□

**Theorem 2** Let \( R \) be a fully-extended orthogonal HRS. Every term that is not in head normal form contains a head-needed redex.

**Proof.** Similar to the proof of Theorem 4.3 in Ref.12), the theorem is proved by Lemma 4, Lemma 5, and Proposition 1. □

We cannot remove the fully-extended condition from this theorem because of the following counterexample.

**Counterexample 1** Consider the following orthogonal HRS:
\( \mathcal{R} = \left\{ \begin{array}{c}
\frac{f(\lambda x.z)}{c} \\
\frac{g(x)}{c},
\end{array} \right. \)

where \( f, \ g, \) and \( c \) are function symbols and \( z \) and \( x \) are variables. The term \( f(\lambda x.g(g(x))) \) is not in head normal form and contains no head-needed redex. The first rule cannot be applied to the term because the free variable \( z \) does not match \( g(g(x)) \), which contains a bound variable \( x \). The second rule can be applied to two redexes: \( g(x) \) and \( g(g(x)) \). However, neither redex is head-needed.

By a reduction sequence \( f(x) \rightarrow f(\lambda x.g(c)) \rightarrow c \), the redex \( g(g(x)) \) is not head-needed. By another reduction sequence \( f(\lambda x.g(g(x))) \rightarrow f(\lambda x.c) \rightarrow c \), the redex \( g(x) \) is not head-needed.

In the proof of Theorem 2, the properties shown in Proposition 1 are necessary, whereas in left-linear TRSs, the property holds trivially.

### 3.4 Top-down Decomposition of Development

To prove the main theorem of the present paper, we must introduce the cost of development. First, we define top-down decomposition.

**Definition 7 (top-down decomposition)** Let \( A : s \rightarrow t \) be a development. If there exists a sequence of positions \( p_1, p_2, \ldots, p_n \) such that \( s \equiv p_0 \overset{p_1}{\Rightarrow} p_1 \overset{p_2}{\Rightarrow} \cdots \overset{p_n}{\Rightarrow} p_n \equiv t \) and \( p_i \neq p_j \) for any \( i < j \) \((1 \leq i < j \leq n)\), then the rewrite sequence is called a top-down decomposition of the development \( A \). If the length of a top-down decomposition is minimal in top-down decompositions of the descendant \( A \), we call the decomposition a minimal top-down decomposition.

Here, we introduce the concept of the top-down property of a development, which is recursively defined as follows: Consider a development \( s \rightarrow t \), and let \( t \equiv \lambda x_1 \cdots x_n. \alpha(t_1, \ldots, t_m) \), where \( \alpha \in \mathcal{F} \cup \mathcal{X} \) and \( n \geq 0 \). The development \( s \rightarrow t \) has the top-down property, \( 1 \) if \( s \overset{\delta}{\Rightarrow} t \), i.e. \( s \equiv t \), or \( 2 \) if, for some \( k \geq 0 \), there exists a unique set of positions \( D(\varepsilon \notin D) \), and the term \( \equiv \lambda x_1 \cdots x_n. \alpha(u_1, \ldots, u_m) \), then \( s \overset{\varepsilon}{\Rightarrow} u \overset{D}{\Rightarrow} t \) and \( u_i \rightarrow t_i \) has the top-down property for all \( i = 1, \ldots, m \). We often write \( \overset{\varepsilon}{\Rightarrow} \) for \( \overset{D}{\Rightarrow} \) when \( \varepsilon \notin D \).

**Lemma 6** Any development has a minimal top-down decomposition, the length of which is uniquely determined. \( \square \)

The proof of Lemma 6 is found in Appendix A.4.

From Lemma 6, we can define the cost of a development \( A : s \rightarrow t \) by the length of the minimal top-down decomposition of \( A \), which is denoted as \( |s \rightarrow t| \).

### 3.5 Head Normalizing Strategy

Middeldorp introduced \( \vdash_{\nabla} \) and \( \vdash_{\Delta} \) in order to divide a parallel reduction \( \vdash_D \) into two parts with respect to a given set \( B \) of parallel redex positions \(^{12} \).

He used the property \( \vdash_{\nabla} \cdot \vdash_{\Delta} \subseteq \vdash_{\nabla} \cdot \vdash_{\Delta} \) to prove his main theorem. In this section, we prove the properties of development of an HRS corresponding to those of parallel reduction of a TRS. This allows us to follow Middeldorp’s example.

**Definition 8 (\( \nabla \) and \( \Delta \))** Let \( D \) be a set of positions, and let \( B \) be another set of positions. When \( D \) satisfies the condition \( \forall p \in D, \exists q \in B, q \prec p \), we write the set \( D \) by \( D_{\nabla B} \). In contrast, when \( \forall p \in D, \forall q \in B, q \not\prec p \), we write the set \( D \) by \( D_{\Delta B} \). We sometimes write \( D_{\nabla} \) for \( D_{\nabla B} \) and \( \nabla \) for \( D_{\nabla} \) and \( D_{\Delta} \) for \( D_{\Delta B} \) and \( \Delta \) for \( D_{\Delta} \) when \( D \) and \( B \) are interpreted as trivial.

Here, we prove the following Lemma 7, which means \( \vdash_{\nabla} \cdot \vdash_{\Delta} \subseteq \vdash_{\nabla} \cdot \vdash_{\Delta} \) \( \vdash_{\nabla} \cdot \vdash_{\Delta} \vdash_{\nabla} \cdot \vdash_{\Delta} \) in Ref. \( 12 \).

**Lemma 7** Let \( B \) be a set of redex positions of a term \( t \), and let \( D \) and \( D' \) be sets of the redex positions that can be written by \( D_{\nabla B} \) and \( D_{\Delta B} \), respectively. Let \( A_1 : t \overset{D}{\rightarrow} t_1 \) and \( A_2 : t_1 \overset{D'}{\rightarrow} t_2 \) be developments. Then, there exist developments \( A_3 : t \overset{D'}{\rightarrow} t_3 \) and \( A_4 : t_3 \overset{D''}{\rightarrow} t_2' \) such that \( D'' = D \setminus A_3 \) can be written as \( (D \setminus A_3)_{\nabla((B \setminus A_3)} \) for some \( t_3 \).

To prove Lemma 7, we must follow the moves of redexes, which complicates the proof. Thus, for the purpose of readability, we give the proof in Appendix A.5.

Here, note that \( |A_2| = |A_3| \) holds. In other words, the costs of developments \( A_2 \) and \( A_3 \) are equal in Lemma 7 because the reduced positions in \( A_1 \) are strictly below \( D' \) or are disjoint from \( D' \).

Now we are at the position to show the main result of this paper. The proof proceeds in a similar way to that of the main theorem in Ref. \( 12 \).

**Proposition 2** If a development \( s \rightarrow t \) is divided into \( s \overset{D}{\rightarrow} s' \) and \( s' \overset{D'}{\rightarrow} t \), where \( D_{\Delta} \) and \( D_{\nabla} \), then \( |s \rightarrow t| \geq |s \rightarrow s'| \).
Proof. The sequence obtained by concatenating the decomposition of \( s \Rightarrow^D s' \) and the sequence of \( s' \Rightarrow^{D'} t \) is a decomposition of \( s \Rightarrow t \). Thus, the proposition holds.

**Definition 9** Let \( A = A_1; A_2; \ldots; A_n \) and \( B = B_1; B_2; \ldots; B_n \) be development sequences of 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 10** Let \( A \) be a development sequence and \( B \) be a development starting from the same term. We write \( B \perp A \) if any descendant of redexes reduced in \( B \) is not reduced in \( A \).

The following two lemmas correspond to Lemma 5.4 and 5.5 in Ref. 12). These lemmas can be proved in the similar way to Ref. 12).

**Lemma 8** Let \( A : s \Rightarrow^* s_n \) and \( B : s \Rightarrow t \) be such that \( B \perp A \). If \( s_n \) is in head normal form then there exists a development sequence \( C : t \Rightarrow^* t_n \) such that \( A \geq C \) and \( t_n \) is in head normal form.

This lemma is proved using Lemma 3, Lemma 7, and Proposition 2.

**Lemma 9** Let \( A : s \Rightarrow^* s_n \) and \( B : s \Rightarrow t \), such that \( B \perp A \). If \( s_n \) is in head normal form, then there exists a development sequence \( C : t \Rightarrow^* t_n \) such that \( A > C \) and \( t_n \) is in head normal form.

This lemma is proved using Lemma 3 and Lemma 8.

**Theorem 3** 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.

Using Lemma 8 and Lemma 9, this theorem is proved in the same manner as Ref. 12).

From Theorem 2 and Theorem 3, the head-needed reduction is a head normalizing strategy in fully-extended orthogonal HRS. In other words, we obtain a head normal form by reducing head-needed redexes, if the head normal form exists.

### 4. Decidable Classes of Higher-Order Rewrite Systems

Since rewrite relations \( \Rightarrow^*_\mathcal{R} \) of HRSs are generally undecidable, in the same manner as TRSs, the neededness of a reduction is undecidable. Therefore, we give a sufficient condition with approximations of reductions by a manner similar to that described in Ref. 4). We have shown that \( \Rightarrow^*_\mathcal{R} [L] \) is recognizable by Theorem 14 of Ref. 10), which uses a Ground Tree Transducer (GTT)\(^2\).

#### 4.1 Approximations

Let \( \mathcal{R} \) be an HRS over the signature \( \mathcal{F} \). Let \( \bullet_\alpha \) be a fresh constant of basic type \( \alpha \). \( \mathcal{R} \) is extended to an HRS over \( \mathcal{F}_\bullet = \mathcal{F} \cup \{\bullet_\alpha \mid \alpha \in \mathcal{S}\} \). We denote the set of all normal forms with respect to \( \mathcal{R} \) over \( \mathcal{T}(\mathcal{F}_\bullet, \emptyset) \) as \( NF_{\mathcal{R}} \). Let \( \mathcal{R}_\bullet \) be \( \mathcal{R} \cup \{\bullet_\alpha \rightarrow \bullet_\alpha \mid \alpha \in \mathcal{S}\} \). Thus, \( NF_{\mathcal{R}_\bullet} \) is \( NF_{\mathcal{R}} \cap \mathcal{T}(\mathcal{F}_\bullet, \emptyset) \). The type of \( \bullet \) will be omitted if it is explicit from the context.

**Lemma 10** Let \( \mathcal{R} \) be an HRS in which both sides are patterns and share no variables. The set \( \{\Rightarrow^*_\mathcal{R}\}[NF_{\mathcal{R}_\bullet}] \) is recognizable.

**Proof.** This lemma follows from Theorem 14 of Ref. 10).

Here, we define approximations on HRS \( \mathcal{R}_a \) and \( \mathcal{R}_{nv} \), which correspond to strong sequential rewriting and NV-sequential rewriting on a TRS\(^4\), respectively.

**Definition 11 (approximation)** Let \( \mathcal{R} \) and \( \mathcal{S} \) be HRSs over the same signature. If \( \Rightarrow^*_\mathcal{R} \subseteq \Rightarrow^*_\mathcal{S} \) and \( NF_{\mathcal{R}} = NF_{\mathcal{S}} \), \( \mathcal{S} \) is said to approximate \( \mathcal{R} \).

**Definition 12 (\( \mathcal{R} \)-needed)** Let \( \mathcal{R} \) be an HRS over a signature \( \mathcal{F} \). Let \( \Delta \) be a redex of type \( \alpha \) in \( C[\Delta] \in \mathcal{T}(\mathcal{F}, \emptyset) \). \( \Delta \) is \( \mathcal{R} \)-needed if and only if there is no \( t \in NF_{\mathcal{R}_{\bullet}} \) such that \( C[t_{\alpha}] \Rightarrow^*_\mathcal{R} t \).

**Lemma 11** Let \( \mathcal{S} \) be an approximation of an HRS \( \mathcal{R} \). Each \( \mathcal{S} \)-needed redex is \( \mathcal{R} \)-needed.

**Proof.** Each redex of \( \mathcal{S} \) is also a redex of \( \mathcal{R} \) from Definition 11. Each reduction relation of \( \mathcal{R} \) is also a reduction relation of \( \mathcal{S} \). Thus, if a redex is \( \mathcal{S} \)-needed, then it is also \( \mathcal{R} \)-needed.

**Definition 13** An approximation \( \mathcal{R}_a \) is an HRS obtained from \( \mathcal{R} \) by replacing the right-hand side of each rewrite rule by new free variables.

**Definition 14** An approximation \( \mathcal{R}_{nv} \) is an HRS obtained from \( \mathcal{R} \) by replacing any subterms in the right-hand sides of rewrite rules in which the top is a free variable by new fully-extended free disjoint variables. \( \mathcal{R}_a \) and \( \mathcal{R}_{nv} \) satisfy conditions of approximations in Definition 11. Both sides of the rewrite rules of \( \mathcal{R}_a \) and \( \mathcal{R}_{nv} \) are patterns because the right-hand side of \( \mathcal{R}_a \) is a free variable and there are no nesting free variables in the right-hand side.
of $\mathcal{R}_{nv}$. If we can find an $\mathcal{R}_s (or \mathcal{R}_{nv})$-needed redex, the redex is also $\mathcal{R}$-needed from Lemma 11. Therefore, we can obtain head normal form on $\mathcal{R}$ by rewriting $\mathcal{R}_s (or \mathcal{R}_{nv})$-needed redexes repeatedly.

**Example 5** Approximations $\mathcal{R}_s$ and $\mathcal{R}_{nv}$ are as follows:

- $\mathcal{R}_s = \{ \text{map}(\lambda x. F(x), \text{cons}(X, Y)) \triangleright \text{cons}(F(X), \text{map}(\lambda x. F(x), Y)) \}$,
- $\mathcal{R}_{nv} = \{ \text{map}(\lambda x. F(x), \text{cons}(X, Y)) \triangleright \text{cons}(Z, \text{map}(\lambda x. G(x), W)) \}$.

**4.2 Needed Reductions**

Next, we discuss the properties of needed reductions on approximations of HRSs.

**Definition 15 ($R$-NEEDED)** Let $\mathcal{R}$ be an HRS, and let $M_s \in T(\mathcal{F}, \emptyset)$ be a set of all terms that contain exactly one occurrence of $\bullet$. If $C[\bullet] \in M_s$ such that there is no $t \in NF_{\mathcal{R}_s}$ with $C[\bullet] \rightarrow^*_R t$, then the set of all terms that satisfies $C[\bullet]$ is said to be $\mathcal{R}$-NEEDED.

The following theorem for HRS holds in a manner similar to The TRS version of this theorem (Theorem 15 in Ref. 4).

**Theorem 4** Let $\mathcal{R}$ be an HRS. If $\rightarrow^*_R [NF_{\mathcal{R}_s}]$ is recognizable, then $\mathcal{R}$-NEEDED is recognizable. □

**Lemma 12** Let $\mathcal{R}$ be an HRS. Relations $\rightarrow^*_R$ and $\rightarrow^*_R$ are recognizable.

**Proof.** For each approximation $S \in \{ \mathcal{R}_s, \mathcal{R}_{nv} \}$, there is a GTT that recognizes $\rightarrow^*_S$. Thus, the relation $\rightarrow^*_S$ is recognizable (Theorem 14 of Ref. 10). □

The following theorem follows from Lemma 10, Lemma 12, and Theorem 4.

**Theorem 5** Let $\mathcal{R}$ be a left-linear HRS over signature $\mathcal{F}$. For each approximation $S \in \{ \mathcal{R}_s, \mathcal{R}_{nv} \}$, whether a redex of a term in $T(\mathcal{F}, \emptyset)$ is $S$-needed is decidable. □

From this theorem, we can decide needed reductions of approximation of HRSs. Therefore, the theorem is a sufficient condition of the decision problem of HRSs.

For example, the following HRS $\mathcal{R}$ causes infinite reduction when using the inner-most strategy or the left-most outer-most strategy. However, needed reduction leads its result.

**Example 6**

\[
\mathcal{R} = \{ \text{if} (X, Y, \text{True}()) \triangleright X, \\
\text{if} (X, Y, \text{False}()) \triangleright Y, \\
\text{isZero}(\text{Zero}) \triangleright \text{True}(), \\
isZero(\text{Succ}(X)) \triangleright \text{False}(), \\
\text{apply}(\lambda x. F(x), X) \triangleright F(X), \\
\text{fromn}(X) \triangleright \text{cons}(X, \text{fromn}(\text{Succ}(X))) \}
\]

\[
\text{if} (\text{fromn}(\text{Succ}(\text{Zero})), \text{cons}(\text{Succ}(\text{Zero}), []), \text{apply}(\lambda x. \text{isZero}(x), \text{Succ}(\text{Zero}))) \\
\rightarrow \text{if} (\text{fromn}(\text{Succ}(\text{Zero})), \text{cons}(\text{Succ}(\text{Zero}), []), \text{isZero}(\text{Succ}(\text{Zero}))) \\
\rightarrow \text{if} (\text{fromn}(\text{Succ}(\text{Zero})), \text{cons}(\text{Succ}(\text{Zero}), []), \text{False}()) \\
\rightarrow \text{cons}(\text{Succ}(\text{Zero}), [])
\]

In this example, a function $\text{fromn}$ makes an infinite list in the inner-most strategy or the left-most outer-most strategy. However, the needed reduction strategy reduces the underlined parts, the functions $\text{apply}$ and $\text{isZero}$ of which are not left-most, but rather outer-most, and obtains its result.

The following are approximations $\mathcal{R}_s$ and $\mathcal{R}_{nv}$ of the example $\mathcal{R}$. Using a GTT made from the following rules, whether a redex of $\mathcal{R}$ is needed is decidable.

\[
\mathcal{R}_s = \{ \text{if} (X, Y, \text{True}()) \triangleright Z, \\
\text{if} (X, Y, \text{False}()) \triangleright Z, \\
isZero(\text{Zero}) \triangleright X, \\
isZero(\text{Succ}(X)) \triangleright Y, \\
\text{apply}(\lambda x. F(x), X) \triangleright Y, \\
\text{fromn}(X) \triangleright Y \}
\]
\[ \mathcal{R}_{nv} = \{ \text{if}(X, Y, \text{True}()) \triangleright Z, \text{if}(X, Y, \text{False}()) \triangleright Z, \text{isZero}(\text{Zero}) \triangleright \text{True}(), \text{isZero}(\text{Succ}(X)) \triangleright \text{False}(), \text{apply}(\lambda x. F(x), X) \triangleright Y, \text{fromn}(X) \triangleright \text{cons}(Y, \text{fromn}(\text{Succ}(Z))) \} \]

4.3 Head-Needed Reductions

In the remainder of the present paper, we generalize the above-mentioned research on needed reduction to relate to head-needed reduction. We can show the following theorem in the same manner as in the case of a TRS in Ref. 4).

**Definition 16** Let \( \mathcal{F} \) be a signature. Let \( \mathcal{F}_o = \mathcal{F} \cup \{ f_o \mid f \in \mathcal{F} \} \), where every \( f_o \) has the same arity as \( f \). Let \( \mathcal{R} \) be an orthogonal HRS over the signature \( \mathcal{F} \). Let \( \Delta \in \mathcal{T}(\mathcal{F}, \emptyset) \) be a redex. We write \( t^\Delta \) for the term that is obtained from \( t \) by marking its head symbol. \( \mathcal{R}_\Delta \) denotes the HRS \( \mathcal{R} \cup \{ l^\Delta \triangleright r \mid l \triangleright r \in \mathcal{R} \} \).

Let \( \mathcal{R} \) and \( \mathcal{S} \) be HRSs over the same signature \( \mathcal{F} \). Redex \( \Delta \) in \( C[\Delta] \in \mathcal{T}(\mathcal{F}, \emptyset) \) is said to be \( (\mathcal{R}, \mathcal{S}) \)-head-needed if there is no term \( t \in \mathcal{HNF}_{\mathcal{S}_\Delta} \) such that \( C[\Delta^o] \triangleright^*_\mathcal{R} t \). The set of all terms \( C[\Delta^o] \) such that there is no term \( t \in \mathcal{HNF}_{\mathcal{S}_\Delta} \) with \( C[\Delta^o] \triangleright^*_\mathcal{R} t \) is denoted as \( (\mathcal{R}, \mathcal{S}) \)-HEAD-NEEDED.

**Theorem 6** Let \( \mathcal{R} \) and \( \mathcal{S} \) be HRSs over the same signature \( \mathcal{F} \). If \( (\rightarrow^*_\mathcal{R}) \mathcal{HNF}_{\mathcal{S}_\Delta} \) is recognizable and \( \mathcal{R} \) is left-linear, then \( (\mathcal{R}, \mathcal{S}) \)-HEAD-NEEDED is recognizable.

**Proof.** Theorem 37 in Ref. 4), which holds on TRSs, also holds on HRSs. \( \square \)

**Lemma 13** Let \( \mathcal{R} \) be a left-linear HRS. The set \( (\rightarrow^*_\mathcal{R}) \mathcal{HNF}_{(\mathcal{R}, \mathcal{S})_\Delta} \) is recognizable for \( \alpha, \beta \in \{ s, nv \} \).

**Proof.** For \( \beta \in \{ s, nv \} \), the set of reducible terms on HRS \( \mathcal{R}_\beta \) is recognizable from Ref. 10). Therefore, \( \mathcal{HNF}_{(\mathcal{R}, \mathcal{S})_\Delta} \) is also recognizable. For \( \alpha \in \{ s, nv \} \), the relation \( (\rightarrow^*_\mathcal{R}) \mathcal{HNF}_{(\mathcal{R}, \mathcal{S})_\Delta} \) is recognizable by Lemma 12. Thus, \( (\rightarrow^*_\mathcal{R}) \mathcal{HNF}_{(\mathcal{R}, \mathcal{S})_\Delta} \) is recognizable for \( \alpha, \beta \in \{ s, nv \} \), i.e., \( (\mathcal{R}, \mathcal{S}) \)-HEAD-NEEDED is recognizable. \( \square \)

From Theorem 6 and Lemma 13, the head-neededness of a redex is decidable for \( s \) and \( nv \) approximations as follows.

**Corollary 1** Let \( \mathcal{R} \) be a left-linear HRS over a signature \( \mathcal{F} \). Whether a redex in a term is \( (\mathcal{R}_\alpha, \mathcal{R}_\beta) \)-head-needed for \( \alpha, \beta \in \{ s, nv \} \) is decidable.

4.4 Decidability of Membership Problem

The decidability of the membership problem on a TRS is discussed using an abstracted model \(^5\). Therefore, we can also discuss that on an HRS using the same method, i.e., the following theorems hold.

**Definition 17** Let \( \alpha \) and \( \beta \) be approximation mappings. The class of HRSs \( \mathcal{R} \) such that each term in non-\( \mathcal{R}_\beta \)-head normal form has an \( (\mathcal{R}_\alpha, \mathcal{R}_\beta) \)-head-needed redex is denoted as CBN-HNF\( \alpha, \beta \).

**Theorem 7** Let \( \mathcal{R} \) and \( \mathcal{S} \) be HRSs such that \( (\mathcal{R}, \mathcal{S}) \)-HEAD-NEEDED is recognizable. The set of terms that have an \( (\mathcal{R}, \mathcal{S}) \)-head-needed redex is recognizable.

**Proof.** Theorem 41 in Ref. 4) holds on HRSs. \( \square \)

**Theorem 8** Let \( \mathcal{R} \) be an HRS, and let \( \alpha \) and \( \beta \) be an approximation mapping such that \( \mathcal{HNF}_{\mathcal{R}_\beta} \) and \( (\mathcal{R}_\alpha, \mathcal{R}_\beta) \)-HEAD-NEEDED are recognizable. Whether \( \mathcal{R} \in \text{CBN-HNF}_{\alpha, \beta} \) holds is decidable.

**Proof.** Theorem 42 in Ref. 4) holds on HRSs. \( \square \)

5. Conclusions

We have introduced the function \( PV \) to follow the moves of an occurrence caused by \( \beta \)-reduction sequences and have given a concrete procedure to calculate the descendants for developments of an HRS using \( PV \). The proposed position system identifies occurrences of \( \lambda x.t \) and \( t \) in \( C[\lambda x.t] \), which has an advantage in that the behavior of the movement of positions is the same as that in the first-order case in applying a substitution to a pattern. We believe that \( PV \) is a useful tool to prove a number of properties related to HRSs. This function has helped us to prove the permutation equivalence of the diamond property. In addition, we have shown that the head-needed reduction is a normalizing strategy for orthogonal HRSs. Thus, we can derive a normal form of a term by repeated reduction of head-needed redexes. Using a GTT and recognizability of redexes, we have shown that we can find head-needed redex and execute head-needed reduction. In addition, we have shown that whether the HRS belongs to a class of head-needed reduction for a given HRS is decidable.

Oostrom showed the (head-)normalizing property of outer-most fair reduc-
tion\textsuperscript{17)}\textsuperscript{*1}. since the result on this reduction is strongly related to the (head-)normalizing property of (head-)needed reduction, one may think that the latter result are derived from the former or vice versa. However, these are difficult. Based on results for both head-needed reduction and outer-most fair reduction, the following relationship between head-needed reduction and outer-most fair reduction is obtained.

(1) \textit{Outer-most fair reduction starting from a term having a head normal form}
\textit{is hyper head-needed reduction.}
Since outer-most fair reduction from a term is normalizing\textsuperscript{17)}, it is also hyper head-needed reduction\textsuperscript{2}.

(2) \textit{Head-needed reduction starting from a term having a head normal form is outer-most fair reduction.}
Since head-needed reduction is head normalizing, if a term has a head normal form, head-needed reduction from the term is also outer-most fair reduction.

\textbf{Acknowledgments} We thank anonymous referees for giving useful comments. The present study was supported in part by MEXT. KAKENHI \#18500011, \#20300010, and \#20500008.

\textbf{References}
Appendix

A.1 Proof of Lemma 2
In order to prove Lemma 2, we must first describe a number of properties.

Proposition 3 Let \( t \) be a term, let \( \sigma \) be a substitution, let \( F \) be a variable, and let \( p \) be a position.

(a) If \( F \not\in \text{FV}(t) \), then \( PV(t, \sigma, F, p) = \emptyset \).

(b) If \( t = G(X_1, \ldots, X_n) \), \( p \in \text{Pos}(X_i, \sigma) \), and \( G \not\in \text{Dom}(\sigma) \), then
\[
PV(t, \sigma, X_i, p) = \{ip\}.
\]
Proof. From the definition of \( PV \), this proposition is trivial. \( \Box \)

Lemma 14 Let \( t \) be a term, and let \( \sigma \) be a substitution for \( \forall X \in \text{Dom}(\sigma) \), \( X_0 = Y \) for some variable \( Y \). If \( p \in \text{Pos}(t) \), then \( PT(t, \sigma, p) = \{p\} \).

Proof. We prove the claim by induction on \( t \). We have four cases from the definition of \( PT \).

(PT1) Since \( p = \epsilon \), we have \( PT(t, \sigma, p) = \{p\} \).

(PT2) Let \( t \equiv a(t_1, \ldots, t_n) \) and \( p = ip' \). Then, by induction, \( PT(t, \sigma, p') = \{p\} \).

Hence, \( PT(t, \sigma, p) = \{iq \mid q \in PT(t_i, \sigma, p')\} = \{ip'\} = \{p\} \).

(PT3) Let \( t \equiv \lambda x_1 \cdots x_n. t' \). Then, by induction, we have \( PT(t, \sigma, p) = PT(t', \sigma |_{x_1 = \ldots = x_n}) = \{p\} \).

(PT4) Let \( t \equiv G(t_1, \ldots, t_n), p = ip' \), \( G = Y = \lambda y_1 \cdots y_n. Y(y_1, \ldots, y_n) \), and \( \sigma' = \{y_1 \mapsto t_1 \sigma \downarrow, \ldots, y_n \mapsto t_n \sigma \downarrow\} \). Then, \( PT(t, \sigma, p') = \{p'\} \subseteq \text{Pos}(t\sigma\downarrow) \) by induction and Lemma 1. We have \( PT(t, \sigma, p) = PV(Y(y_1, \ldots, y_n), \sigma', y_i, p') = \{ip\} \) by Proposition 3(b), because \( p' \in \text{Pos}(t_i \sigma) \) is \( \text{Pos}(y_i \sigma') \).

\( \Box \)

Lemma 15 Let \( t \) be a term such that \( t = F(X_1, \ldots, X_n) \). If \( p \in \text{Pos}(F, \sigma) \), \( F \in \text{Dom}(\sigma) \), and \( X_i \not\in \text{Dom}(\sigma) \) for all \( i \), then \( PV(t, \sigma, F, p) = \{p\} \).

Proof. From Proposition 3(a), \( PV(X_1, \ldots, X_n, \sigma, F) = \emptyset \). Let \( F = \lambda y_1 \cdots y_n. t' \). Thus, \( PV(V(X_1, \ldots, X_n), \sigma, F, p) = PT(t', \sigma', p) \), where \( \sigma' = \{y_1 \mapsto X_1 \downarrow \sigma \downarrow, \ldots, y_n \mapsto X_n \downarrow \sigma \downarrow\} = \{y_1 \mapsto X_1 \downarrow, \ldots, y_n \mapsto X_n \downarrow\} \). Since \( p \in \text{Pos}(t') \), we have \( PT(t', \sigma', p) = \{p\} \) by Lemma 14. \( \Box \)

Based on the above lemmas, we obtain the following lemma.

Lemma 2 If \( l \) is a pattern, then \( PV(l, \sigma, F, p) = \{p'p \mid \text{top}(l|_p') = F\} \) for every \( F \in \text{FV}(l) \).

Proof. We show by induction on the structure of \( l \) that \( PV(l, \sigma, F, p) = \{p'p \mid \text{top}(l|_{p'}) = F\} \) for every \( F \in \text{FV}(l) \).

(Proof of Lemma 2)

Appendix

A.2 Proof of Theorem 1
In order to prove this theorem, the following lemma must be prepared.

Lemma 16 Let \( t \) be a normalized term, let \( \sigma \) be a normalized substitution, and let \( F \) be a variable.

(a) Let \( p \in \text{Pos}(F, \sigma) \) be a position. Then, for any \( q \in PV(t, \sigma, F, p) \) there exists a substitution \( \theta \) such that \( (t\sigma)|_q \equiv (F\sigma)|_p \theta \).

(b) Let \( p \in \text{Pos}(t) \) be a position. Then, for any \( q \in PT(t, \sigma, p) \) there exists a substitution \( \theta \) such that \( (t\sigma)|_q \equiv (t|_p\sigma)|_p \theta \).

Proof. We prove (a) and (b) simultaneously by induction on \( (t, \sigma) \) with \( \succ \cdot \beta \).

First, we show the proof of (a). Let \( p \in \text{Pos}(F, \sigma) \) and \( q \in PV(t, \sigma, F, p) \). We have six cases from the definition of \( PV \).

(Proof of Theorem 1)

PV1) We have \( PV(F, \sigma, F, p) = \{p\} = \{p'p \mid \text{top}(F|_{p'}) = F\} \).

PV2) Let \( l \equiv a(t_1, \ldots, t_n) \). We have \( PV(t, \sigma, F, p) = \{p'p \mid \text{top}(t|_{p'}) = F\} \) by induction. Thus, \( PV(l, \sigma, F, p) = \bigcup_i \{ip'p \mid \text{top}(t_i|_{p'}) = F\} \). Hence, \( PV(l, \sigma, F, p) = \{p'p \mid \text{top}(l|_{p'}) = F\} \).

PV3) Let \( l \equiv \lambda x_1 \cdots x_n. t \). Then, by induction, \( PV(t, \sigma, F, p) = \{p'p \mid \text{top}(t|_{p'}) = F\} \), because \( \text{Dom}(\sigma |_{x_1 \cdots x_n}) \cap (B \cup \{x_1, \ldots, x_n\}) = \emptyset \).

PV4) Let \( l \equiv G(t_1, \ldots, t_n) \). Then, \( G \not\in B \) and \( t_i = x_i \downarrow \), where \( x_i \) is a pairwise distinct variable in \( B \). Hence, the claim holds by Lemma 15.

PV5) Same as for (PV4).

PV6) Trivial. \( \Box \)

This lemma means that a position \( p \in \text{Pos}(F, \sigma) \) moves to \( p'p \) for the position \( p' \) of \( F \) in pattern \( l \), which is the same behavior as that observed in the case of the first order.
induction. Then, \((s|\tau)|_q \equiv ((s|\tau)|_p)^{\theta'}}\) holds from \((t|\sigma)|_q \equiv (t'|\sigma')|_q\) and \(F \notin \{x_1, \ldots, x_n\}\).

**(PV4)** Let \(t \equiv G(t_1, \ldots, t_n)\), \(F \neq G\), and \(G\sigma \equiv \lambda y_1 \ldots y_n t'\). Then, there exist \(q'\) and \(i\) such that \(q' \in PV(t_1, \sigma, F, p)\) and \(q \in PV(t'|\sigma'|, y, q')\), where \(\sigma' = \{y_1 \mapsto t_1 \sigma_1, \ldots, y_n \mapsto t_n \sigma_1\}\). Thus, \((t|\sigma)|_q \equiv (t'|\sigma')|_q\), which is \(\theta'|\sigma'|\) by induction.

**(PV5)** Let \(t \equiv F(t_1, \ldots, t_n)\) and \(F\sigma \equiv \lambda y_1 \ldots y_n t'\). Then, there exist \(i\) such that \(q \in PV(t'|\sigma'|, y, (t_1, \sigma, F, p))\) or \(q \in PT(t'|\sigma'|, p)\), where \(\sigma' = \{y_1 \mapsto t_1 \sigma_1, \ldots, y_n \mapsto t_n \sigma_1\}\). The former case can be shown in the same manner as (PV4). In the latter case, since \((t, \sigma) >_{\gamma, \beta} (t', \sigma')\), there exists \(\theta\) such that \((t'|\sigma')|_q \equiv (t'|\sigma')|_p\) by induction. Thus, \((t|\sigma)|_q \equiv (t'|\sigma')|_q \equiv (t'|\sigma')|_q \equiv (t|\sigma)|_q\) holds.

**(PV6)** (PV6) is obvious because \(PV(t, \sigma, F, p) = \emptyset\).

Next, we show the proof of (b). Let \(t|p|\) be a redex, and let \(q \in PT(t, \sigma, p)\). We have four cases from the definition of PT.

**Proof**

**(PT1)** We have \(q = \epsilon\) from \(p = \epsilon\). Thus, we have \((t|\sigma)|_q \equiv s|\sigma| \equiv (t|p)|\sigma|\).

**(PT2)** Let \(p = iq'\) and \(t \equiv a(t_1, \ldots, t_n)\). There exists \(q'\) such that \(q = iq'\) and \(q' \in PT(t_i, \sigma, p')\). Since \((t, \sigma) >_{\gamma, \beta} (t_i, \sigma)\) and \(t|p| = t|p'|\), there exists \(\theta\) such that \((t|\sigma)|_q \equiv (t_i|\sigma)|_q\) by induction. From \(a \in F \cup \text{Dom}(\sigma)\), we have \((t|\sigma)|_q \equiv a(t_1, \ldots, t_n |\sigma)|_q\), where \(\sigma = \{t|\sigma|_q \equiv (t_i|\sigma)|_q \equiv (t_i|\sigma)|_q \equiv (t|\sigma)|_q\} \equiv (t|\sigma)|_q\) holds.

**(PT3)** Let \(t \equiv \lambda y_1 \ldots x_i t'\) and let \(\sigma' = \sigma|_{\{x_1, \ldots, x_n\}}\). Then, \(q \in PT(t', \sigma', p)\). Since \((t, \sigma) >_{\gamma, \beta} (t', \sigma')\), there exists \(\theta'\) such that \((t'|\sigma')|_q \equiv (t'|\sigma')|_q\) by induction. Then, \((t|\sigma)|_q \equiv (t|\sigma)|_q\) holds from \((t|\sigma)|_q \equiv (t|\sigma)|_q\).

**(PT4)** Let \(p = iq'\), \(t \equiv G(t_1, \ldots, t_n)\), and \(G\sigma \equiv \lambda y_1 \ldots y_n t'\). Then, there exists \(q'\) such that \(q' \in PT(t_i, \sigma, p')\) and \(q \in PV(t'|\sigma'|, y, q')\), where \(\sigma' = \{y_1 \mapsto t_1 \sigma_1, \ldots, y_n \mapsto t_n \sigma_1\}\). Since \((t, \sigma) >_{\gamma, \beta} (t', \sigma')\), there exists \(\theta'\) such that \((t'|\sigma')|_q \equiv (t'|\sigma')|_q\) by induction. Thus, \((t|\sigma)|_q \equiv (t|\sigma)|_q \equiv (t|\sigma)|_q \equiv (t|\sigma)|_q\). Moreover, since \(t, \sigma) >_{\gamma, \beta} (t, \sigma)\), there exists a position \(\theta''\) such that \((t|\sigma)|_q \equiv (t_i|\sigma)|_q \equiv (t_i|\sigma)|_q \equiv (t|\sigma)|_q\). Therefore, \((t|\sigma)|_q \equiv (t_i|\sigma)|_q \equiv (t_i|\sigma)|_q \equiv (t|\sigma)|_q\) holds.

**Theorem 1** Let \(R\) be an orthogonal HRS, and let \(A: s \rightarrow t\) be a reduction. If \(s|\tau|_u\) is a redex, then \(t|p|\) are also redexes for all \(p \in v/A\).

**Proof** Let \(s \equiv s|\sigma|_u \rightarrow s|\sigma|_u \equiv t\). In case of \(v/u < u\), the theorem follows from orthogonality. In case of \(v = up_j p_k\), (PV1) of Theorem 1 follows from Lemma 16 and the fact that instances of redex are redexes.

**A.3 Proof of Lemma 3**

In this section, we assume orthogonality.

**Proposition 4** Let \(s\) and \(t\) be normalized terms. Let \(s \rightarrow_D t\) be a development, and let \(\theta\) and \(\sigma\) be normalized substitutions. If for each \(F \in \text{Dom}(\theta) \cap \text{PV}(s)\), we have some \(D_F\) such that \(F\theta \rightarrow_D F\sigma\), then \(s|\sigma| \rightarrow_D t|\sigma|\), where \(D' = \bigcup_F PV(s, \theta, F, D_F) \cup PT(s, \theta, D)\).

**Proof** By Noetherian induction on \((s, \theta)\) with \(\rightarrow_D\), we prove the claim \(P(s, \theta)\) defined as follows:

- If \(s\rightarrow_D t\) and \(F\theta \rightarrow_D F\sigma\) for any \(F \in \text{Dom}(\theta) \cap \text{PV}(s)\), there exists a development \(s|\theta| \rightarrow_D t|\theta|\), where \(D' = \bigcup_F PV(s, \theta, F, D_F) \cup PT(s, \theta, D)\).

From the definition of developments, we have several cases.

(A) First, consider the case in which \(s \rightarrow_D t\) is derived by the inference rule (A) of the definition of developments. We have two subcases:

1. If \(\text{top}(s) \in F \cup \text{Dom}(\theta)\), then we have \(s \equiv a(s_1, \ldots, s_n)\), \(s_i \rightarrow_D t_i\), and \(D = \bigcup_i \{iq | q \in D_i\}\). Since \((s, \theta) >_{\gamma, \beta} (s, \theta)\), the induction hypothesis asserts that \(s|\sigma| \rightarrow_D t|\sigma|\) for \(D' = \bigcup_F PV(s, \theta, F, D_F) \cup PT(s, \theta, D)\). Thus, by the definition of developments, we have \(s|\theta| \rightarrow_D t|\theta|\), where \(D' = \bigcup_i \{iq | q \in D_i\}\).

2. **Furthermore**, we can calculate \(D'\) from \(D';

- \(D' = \bigcup_i \{iq | q \in PV(s, \theta, F, D_F) \cup PT(s, \theta, D)\}

\]
can assert $s_iθ|\Rightarrow^D D_i' = \bigcup_F PV(s_i, θ, F, D_F) \cup PT(s_i, θ, D_i)$ by induction. Let $Gθ ≡ λy_1 \cdots y_n.u$, and let $Gσ ≡ λy_1 \cdots y_n.u'$. Then, we have $sθ|\equiv (Gθ)(s_iθ, \ldots, s_nθ)|\equiv sθ'$, where $θ = \{ y_1 | s_iθ, \ldots, y_n | s_nθ \}$, and $tσ|\equiv (Gσ)(tσ, \ldots, tσ)|\equiv uθ'$, where $σ = \{ y_1 | tσ, \ldots, y_n | tσ \}$. Thus, we have $sθ'|\equiv s_iθ|\Rightarrow^D D_i' tσ|\equiv yθ'$, where $D_i' = \bigcup_F PV(s_i, θ, F, D_F) \cup PT(s_i, θ, D_i)$. Moreover, note that $Gθ \Rightarrow^D Gσ$, and $u \Rightarrow^D u$ follows from $λy_1 \cdots y_n.u \equiv Gθ \Rightarrow^D Gσ \equiv λy_1 \cdots y_n.u'$. Since $sθ|\equiv uθ'$, i.e., $sθ|\equiv^D uθ'$, we have $(s, θ) \Rightarrow^β (u, θ')$. Thus, we have $uθ'|\equiv^D θ'uθ'|$ for $D' = \bigcup_F PV(u, θ', y, D_i') \cup PT(u, θ', D_G)$ by induction. Hence, $sθ|\equiv uθ'|\Rightarrow^D θ'uθ'|\equiv tσ|$. Here, we can calculate $D'$ as follows: Moreover, we have

$$D' = \bigcup_F PV(u, θ', y, (\bigcup_F PV(s_i, θ, F, D_F) \cup PT(s_i, θ, D_i)))$$
$$\cup PT(u, θ', D_G)$$

$$= \bigcup_F PV(u, θ', y, \bigcup_F PV(s_i, θ, F, D_F))$$
$$\cup \bigcup_F PV(u, θ', y, PT(s_i, θ, D_i)) \cup PT(u, θ', D_G)$$

$$= \bigcup_F PV(u, θ, y, PV(s_i, θ, F, D_F))$$
$$\cup PT(u, θ', D_G) \cup \bigcup_F PV(u, θ, PT(s_i, θ, D_i))$$

$$= \bigcup_F PV(G(s_1, \ldots, s_n), θ, F, D_F) \cup PT(G(s_1, \ldots, s_n), θ, D).$$

Thus, we have $D' = \bigcup_F PV(s, θ, F, D_F) \cup PT(s, θ, D)$. (L) Next, consider the case of applying rule (L), that is, $s \equiv λx_1 \cdots x_n.s' \Rightarrow^D \lambda x_1 \cdots x_n.t' \equiv \top$. Here, since $sθ|\equiv sθ'$, we have $(s, θ) \Rightarrow^β (s', θ')$. Thus, by induction, we have $sθ'|\equiv^D t'θ'|$ for $D' = \bigcup_F PV(s', θ, F, D_F) \cup PT(s', θ, D)$. Applying rule (L) to development, we have $sθ|\equiv λx_1 \cdots x_n.(sθ')|\Rightarrow^D λx_1 \cdots x_n.(t'θ)|\equiv tσ|$. Note that the condition $Dom(θ) \cap \{ x_1, \ldots, x_n \} = \emptyset$ is required, but this problem can be solved trivially. (R) Let $s \equiv f(s_1, \ldots, s_n) = lθ', t = rσ', f(t_1, \ldots, t_n) = lσ', l \Rightarrow r \in R$, $s_i \Rightarrow^D t_i$, and $D = \{ ε \} \cup \bigcup_i \{ iq | q \in D_i \}$. Since $(s, θ) \Rightarrow^β (s_i, θ)$, we have $sθ'|\equiv^D t_iθ'|$ by induction, where $D_i' = \bigcup_F PV(s_i, θ, F, D_F) \cup PT(s_i, θ, D_i)$. We have $sθ|\equiv f(s_1θ', \ldots, s_nθ') \equiv (lθ')|\equiv lθ'$ and $f(t_1σ', \ldots, t_nσ') \equiv (lσ')|\equiv lσ'$ for some $θ'$ and $σ'$. By rule (R) to development, $f(s_iθ, \ldots, s_nθ) \Rightarrow^σ θ'σ'$. Note that $(lθ')|\equiv lθ'$ implies that $tσ|\equiv (rσ')|\equiv rσ'$ because $PV(l) \supseteq PV(r)$. Therefore, we have $sθ|\Rightarrow^D tσ|$, where $D' = \{ ε \} \cup \bigcup_i \{ iq | q \in D_i \}$. Here, we have

$$D' = \bigcup_F PV(s, θ, F, D_F)\cup PT(s, θ, D).$$

As a special case of Proposition 4, the following corollary holds. **Corollary 2** Let $s$ be a term, and let $θ$ and $σ$ be normalized substitutions. Let $Fθ \Rightarrow^D_F Fσ$ for each $F \in Dom(θ) \cap PV(θ)$. Then, $sθ|\Rightarrow^D sσ|$, where (1)

$$D' = \bigcup_F PV(s, θ, F, D_F).$$

In particular, (2) if $s$ is a pattern, then $D' = \bigcup_F \{ p'p | top(p') = F, p \in D_F \}$. □

From the definition of descendants, the following proposition is trivial. **Proposition 5** Let $s_1$ and $t_1$ be terms, and let $A$ and $A'$ be developments such that $A : x.A.s_1 \Rightarrow^D_A χ.x.A_1$ and $A' : s_1 \Rightarrow^D_A t_1$. Their descendants hold $(q \setminus A) = \{ q \setminus A \}$ for any position $q$. □

Next, we present the following lemma in order to prove the main lemma of this section.

**Lemma 17** Let $A_F : Fθ \Rightarrow^D_F θ_F$ be a development, for every $F \in Dom(θ)$. In addition, let $A : s \Rightarrow^D t$ and $A_1 : s_1 \Rightarrow^D t_1$ be developments.

(a) $PV(s, θ, F, p) \setminus \{ A' \} = PV(t, θ, F, p) \setminus \{ A_F \}$ for any $p, F \in Occ(Fθ)$ is a redex on $s$, and $F \in Dom(θ)$. (b) $PT(s, F, p) \setminus \{ A' \} = PT(t, θ, p, \emptyset) \setminus \{ A_F \}$ for any $p, F \in Occ(θ)$, which is a redex on $s$. Proof. We prove (a) and (b) simultaneously by induction on $s \Rightarrow^D t$. First, we show (a). (A) Consider the case $s \equiv a(s_1, \ldots, s_n) \Rightarrow^D a(t_1, \ldots, t_n) \equiv t$ and $a \in F \cup X$. Let $D_i = \{ p'ip | p' \in D \}$ for each $i$, and let $A_i : s_i \Rightarrow^D t_i$. Let $A_i' : s_iθ|\Rightarrow^D D_i' t_iσ|$. Let $D_i' = \bigcup_F PV(s_i, θ, F, D_F) \cup PT(s_i, θ, D_i)$.
\[
PV(s, \theta, F, p')\setminus A'_1 = PV(t, \sigma, F, p'\setminus A_F) \quad \text{if } \exists F, p' \in Occ(F) \theta
\]
\[
PT(s, \theta, p')\setminus A'_1 = PT(t, \sigma, p'\setminus A_1) \quad \text{if } \exists p' \in Occ(s) \theta
\]
holds by induction.

**A-PV2** Consider the case in which \( p' \in Occ(F) \theta \) and \( a \in F \cup \overline{Dom(\theta)} \).
\[
PV(s, \theta, F, p')\setminus A' = \bigcup \{ q \in PV(s, \theta, F, p') \} \setminus (a(1, n, s_n) \theta \vdash a(t_1, n, t_n) \sigma_{1})
\]
by PV2
\[
\text{by Def. } 4(A)
\]
\[
= \bigcup \{ q \in PV(s, \theta, F, p') \} \setminus (a(1, n, s_n) \theta \vdash a(t_1, n, t_n) \sigma_{1})
\]
from I.H.
\[
= PV(t, \sigma, F, p'\setminus A_F)
\]
by PV2

**A-PV4** Consider the case in which \( a = G \in Dom(\theta) \). Let \( G \theta \equiv \lambda y_1 \ldots y_n. s', s \equiv G(s_1, \ldots, s_n) \theta_{1}, t \equiv G(t_1, \ldots, t_n) \sigma_{1}, \) and \( \theta = \sigma' \).
\[
PV(s, \theta, F, p') \setminus A' = \bigcup \{ PV(s', \theta', y_i, PV(s, \theta, F, p')) \} \setminus (G(1, n, s_n) \theta \vdash G(t_1, n, t_n) \sigma_{1})
\]
by def. of PV4
\[
= \bigcup \{ PV(s', \theta', y_i, PV(s, \theta, F, p')) \} \setminus (G(1, n, s_n) \theta \vdash G(t_1, n, t_n) \sigma_{1})
\]
from I.H.
\[
= PV(t, \sigma, F, p'\setminus A_F)
\]
by def. of PV4.

**A-PV5** Consider the case in which \( a = F \in Dom(\theta) \). Let \( F \theta \equiv \lambda y_1 \ldots y_n. s', s \equiv F(s_1, \ldots, s_n) \theta_{1}, F \equiv \lambda y_1 \ldots y_n. t', t \equiv F(t_1, \ldots, t_n) \sigma_{1}. \) Let \( \theta' = \{ y_1 \rightarrow s_1 \theta_{1}, \ldots, y_n \rightarrow s_n \theta_{1} \}, \sigma' = \{ y_1 \rightarrow t_1 \theta, \ldots, y_n \rightarrow t_n \theta \}, \) and \( s\theta' = s\theta \).
\[
PV(s, \theta, F, p') \setminus A' = \bigcup \{ PV(s', \theta', y_i, PV(s, \theta, F, p')) \cup PT(s', \theta', p') \} \setminus (s\theta' \vdash t'\sigma')
\]
by Def. of PV5
\[
= \bigcup \{ PV(s', \theta', y_i, PV(s, \theta, F, p')) \cup PT(s', \theta', p') \} \setminus (s\theta' \vdash t'\sigma')
\]
by I.H. of (a) and (b)
\[
= PV(t, \sigma, F, p'\setminus A_F)
\]
by I.H. of (a) and (b)
\[
= PV(t, \sigma, F, p'\setminus A_F)
\]
by I.H. of (a) and (b)

**L** Consider the case in which \( s \equiv \lambda x_1 \ldots x_n. s' \equiv \lambda x_1 \ldots x_n. t' \equiv t \) and \( A_1 : s' \equiv D_{1} t'. \) Let \( A'_1 : s' \theta \vdash D_{1} t' \sigma \). For developments \( A_F \) and \( A_1 \),
\[
PV(s', \theta, F, p')\setminus A'_1 = PV(t', \sigma, F, p'\setminus A_F)
\]
if \( \exists F, p' \in Occ(F) \theta \)
\[
PT(s', \theta, p')\setminus A'_1 = PT(t', \sigma, p'\setminus A_1)
\]
if \( \exists p' \in Occ(s') \)
holds by induction.

**L-PV3** Consider the case in which \( p' \in Occ(F) \theta \).
\[
PV(s, \theta, F, p') \setminus A' = PV(t', \sigma, F, p'\setminus A_F)
\]
by def. of PV3
\[
PT(s, \theta, p') \setminus A' = PT(t', \sigma, p'\setminus A_1)
\]
from I.H.

**R** Consider the case in which \( s \equiv f(s_1, \ldots, s_n) \equiv f(t_1, \ldots, t_n) \equiv r\theta'' \equiv t \) and \( A_1 : s_1 \theta \vdash D_{1} t \).
For developments \( A_F \) and \( A_1 \),
\[
PV(s, \theta, F, p') \setminus A'_1 = PV(t, \sigma, F, p'\setminus A_F)
\]
if \( \exists F, p' \in Occ(F) \theta \)
\[
PT(s, \theta, p') \setminus A'_1 = PT(t, \sigma, p'\setminus A_1)
\]
if \( \exists p' \in Occ(s) \theta \)
holds by induction.

**R-PV1** Consider the case in which \( p' \in Occ(F) \theta \), we have
\[
PV(s, \theta, F, p') \setminus \{ s_1 \theta \vdash f(s_1, \ldots, s_n) \theta \vdash f(t_1, \ldots, t_n) \sigma \vdash t \sigma \} = \{ q \in PV(t, \sigma, F, p'\setminus A_F) \} \setminus \{ f(t_1, \ldots, t_n) \sigma \vdash t \sigma \} \setminus \{ \} \quad \text{from the case (A). Since we can denote,} \quad f(t_1, \ldots, t_n) \sigma \vdash r\theta'' \sigma, \quad \text{letting} \quad \theta'' = \theta, \quad \text{we have} \quad PV(t, \sigma, F, p') \setminus \{ f(t_1, \ldots, t_n) \sigma \vdash r\theta'' \sigma \} \quad \text{by def. of PV5.}
\]
Consider the case in which \( p' \in \text{Occ}(F\theta) \), \( p \in PV(s,\theta,F,p') \).
\[
p\{s|\theta\} \ni f(s_1,\ldots,s_n)\theta ] \Rightarrow t\theta \Leftrightarrow t\sigma \]
\[
= p' \{f(s_1,\ldots,s_n)\theta ] \Rightarrow f(t_1,\ldots,t_n)\theta ] \Leftrightarrow \quad t\theta ] \Rightarrow t\sigma \}
\[
= PV(t,\theta,F,p')\{t\theta ] \Rightarrow t\sigma \}
\]
Next, we show (b).

(A) Consider the case in which \( s \equiv a(s_1,\ldots,s_n) \Rightarrow D t_1,\ldots,t_n \equiv \top t_1,\ldots,t_n \equiv \top a(t_1,\ldots,t_n) \equiv t \) and \( a \in F \cup A \). Let \( D_1 = \{ p''\mid ip'' \in D \} \) for each \( i \), and let \( A_i : s_i \Rightarrow D_1, t_1 \). Let
\[
A_i : s_i\theta ] \Rightarrow D_1 t_1 \sigma \]
holds from induction.

(A-PT2) Consider the case in which \( p'' \in \text{Occ}(s) \) and \( a \in F \cup \text{Dom}(\theta) \),
\[
PT(s,\theta,p')\{A' \}
= \{ iq \in PT(s,\theta,p') \mid p'' = ip' \} \}
= \{ iq \in PT(s,\theta,p') \mid A' \}= \{ iq \in PT(t,\theta,p') \mid \}
= PT(t,\sigma,p')\{A \}
\]
from PT2

(A-PT4) Consider the case in which \( p'' \in \text{Occ}(s) \) and \( a \in \text{Dom}(\theta) \),
\[
PT(s,\sigma,p')\{A' \}
= PV(s,\theta',\sigma,t_1,PT(s,\theta,p')\{A' \}) \}
= PV(t,\sigma,p')\{A' \} \}
= PT(t,\sigma,p')\{A \}
\]
holds from induction.

(L) Consider the case in which \( s \equiv \lambda x_1 \ldots x_n s' \Rightarrow \lambda x_1 \ldots x_n t' \equiv \top t_1,\ldots,t_n \equiv \top A_1 : s' \Rightarrow D_1 t' \sigma \). For developments \( A_1 \) and \( A_1 \),
\[
PV(s',\theta,F,p')\{A' \} = PV(t',\theta,F,p'\{A \}) \]
\[
PT(s,\sigma,p')\{A' \} = PT(t,\sigma,p'\{A \})
\]
holds from induction.

(L-PT3) Consider the case in which \( p' \in \text{Occ}(s) \),
\[
PT(s,\sigma,p')\{A' \}
= PT'(s',\{p''\mid \theta ] \Rightarrow \top t_1,\ldots,t_n \equiv \top A' \}
= PT'(s',\{p''\mid \theta ] \Rightarrow \top t_1,\ldots,t_n \equiv \top A' \}
= PT'(t',\sigma,p')\{A' \}
= PT(t,\sigma,p')\{A \}
\]
from PT3

(R) Consider the case in which \( s \equiv f(s_1,\ldots,s_n) \Rightarrow f(t_1,\ldots,t_n) \Rightarrow r_0'' \equiv \top r_1'' \) and \( A_i : s_i \Rightarrow D_1, t_i \). Let \( A_i : s_i\theta ] \Rightarrow D_1 t_i \sigma \). For developments \( A_F \) and \( A_i \),
\[
PV(s,\theta,F,p')\{A' \} = PV(t,\sigma,F,p'\{A \}) \]
\[
PT(s,\sigma,p')\{A' \} = PT(t,\sigma,p'\{A \})
\]
holds by induction.

(R-PT-1) Consider the case in which \( p' \in \text{Occ}(s) \),
\[
PT(s,\theta,p')\{s\theta ] \Rightarrow f(s_1,\ldots,s_n)\theta ] \Rightarrow \top t_1,\ldots,t_n \equiv \top A' \}
= PT'(s',\{p''\mid \theta ] \Rightarrow \top t_1,\ldots,t_n \equiv \top A' \}
= PT'(s',\{p''\mid \theta ] \Rightarrow \top t_1,\ldots,t_n \equiv \top A' \}
= PT(t,\sigma,F,p')\{A \}
\]
from the case (A)

(R-PT-2) Consider the case in which \( p' \in \text{Occ}(s), p \in PT(s,\theta,p') \),
\[
p\{s\theta ] \Rightarrow f(s_1,\ldots,s_n)\theta ] \Rightarrow \top t_1,\ldots,t_n \equiv \top A' \}
= PT'(s',\{p''\mid \theta ] \Rightarrow \top t_1,\ldots,t_n \equiv \top A' \}
= PT'(s',\{p''\mid \theta ] \Rightarrow \top t_1,\ldots,t_n \equiv \top A' \}
= PT(t,\sigma,p')\{A \}
\]
from def. of PT

Lemma 18 Let \( A : s \Rightarrow D t \) and \( B : s \Rightarrow t' \) such that \( \varepsilon \not\subseteq D \). Then, \( A ; B \Rightarrow s \varepsilon \Rightarrow B ; A \Rightarrow D \\>

Proof. Let \( s \equiv \theta ] \equiv \top t_1,\ldots,t_n \equiv \top A_1 \), and let \( t' \equiv r_1 ] \). First, we show that \( r_1 ] \Rightarrow D \ v_1, \because t_1 \equiv r_1 ] \)
holds trivially. Let \( F \in \text{Dom}(\theta) \cap \text{Fin}(r) \). Then, from orthogonality, \( F\theta ] \Rightarrow D\ v_1, \text{for some set } D_F \text{ of positions.} \) We have \( r_1 ] \Rightarrow D' \ v_1, \text{by Corollary 2 (1), where } D' = \bigcup_F PV(r,\theta,F,D_F) \). We also have \( D = \bigcup_F \{ p' \mid \top t_1,\ldots,t_n \equiv F, p \in D_F \} \) by Corollary 2 (2). On the other hand, \( D ; B = \bigcup_{v_1 \in D} \{ p_3 \mid
we assume Proof. A from Lemma 17 (a). Therefore, the claim holds. □

Second, we show \( q(A; B \backslash A) = q(B; A \backslash B) \) for each redex position \( q \) in \( s \). In the case of \( q = \varepsilon \), this is trivial. Consider the case of \( q \neq \varepsilon \). From orthogonality, we assume \( q = p_1 p_2 \) and \( \text{top}(l_{p_1}) = F \). Let \( A_F : F \theta \Rightarrow^D \varepsilon F \sigma \). Since \( q(A) = \{ p_1 p_3 \mid p_3 \in p_2 \setminus A_F \} \), we have

\[
q(A; B \backslash A) = \bigcup_{p_3 \in p_2 \setminus A_F} p_1 p_3 (B \backslash A) = p_1 (B \backslash A) p_3 \varepsilon F \sigma = p_1 p_3 (A) \varepsilon F \sigma = p_1 p_3 (A) \varepsilon F \sigma = p_1 p_3 (A) \varepsilon F \sigma = p \varepsilon F \sigma.
\]

Since \( q(B; A \backslash B) = PV(r, \theta, F, p_2, A) \), we have

\[
q(B; A \backslash B) = PV(r, \theta, F, p_2, A) \backslash (A \backslash B) = PV(r, \theta, F, p_2, A) \varepsilon F \sigma = p \varepsilon F \sigma.
\]

From Lemma 17 (a). Therefore, the claim holds. □

Lemma 3. Let \( R \) be an orthogonal HRS. If \( A \) and \( B \) are developments starting from the same term, then \( A; (B \backslash A) \simeq B; (A \backslash B) \).

Proof. Let \( A \) and \( B \) be developments such that \( A : s \Rightarrow^D t \) and \( B : s \Rightarrow^D t' \), respectively. Then, we show the following:

1. There exists \( u \) such that \( t \Rightarrow^D u \) and \( t' \Rightarrow^D u \), and
2. For every redex \( s \), \( p(A; B \backslash A) = p(B; A \backslash B) \)

by induction on the structure of \( s \). We have several cases, according to the inference rules of development applied for \( A \) and \( B \).

1. Consider the case in which inference rule (L) is used for both \( A \) and \( B \). Let \( s \equiv \lambda x_1 \cdots x_n u_1 \). Since \( A_1 : s_1 \Rightarrow^D t_1 \) and \( B_1 : s_1 \Rightarrow^D t'_1 \), where \( t \equiv \lambda x_1 \cdots x_n t_1 \) and \( t' \equiv \lambda x_1 \cdots x_n t'_1 \), it follows from induction that \( t_1 \Rightarrow^D \lambda x_1 \cdots x_n u_1 \) and \( t'_1 \Rightarrow^D \lambda x_1 \cdots x_n u_1 \) for some \( u_1 \), and \( p(A_1; (B_1 \backslash A_1)) = p(B_1; (A_1 \backslash B_1)) \).

2. Consider the case in which inference rule (A) is used for both \( A \) and \( B \). Let \( s \equiv \lambda x_1 \cdots x_n u_1 \). We have \( A_1 : s_1 \Rightarrow^D t_1 \) and \( B_1 : s_1 \Rightarrow^D t'_1 \), where \( t \equiv \lambda x_1 \cdots x_n t_1 \) and \( t' \equiv \lambda x_1 \cdots x_n t'_1 \), from which induction that \( t_1 \Rightarrow^D \lambda x_1 \cdots x_n u_1 \) and \( t'_1 \Rightarrow^D \lambda x_1 \cdots x_n u_1 \) for some \( u_1 \), and \( p(A_1; (B_1 \backslash A_1)) = p(B_1; (A_1 \backslash B_1)) \).

3. Consider the case in which (A) is used for \( A \) and (R) is used for \( B \). Let \( s \equiv \lambda x_1 \cdots x_n u_1 \). We have \( A_1 : s_1 \Rightarrow^D t_1 \) and \( B_1 : s_1 \Rightarrow^D t'_1 \), where \( t \equiv \lambda x_1 \cdots x_n t_1 \) and \( t' \equiv \lambda x_1 \cdots x_n t'_1 \), from which induction that \( t_1 \Rightarrow^D \lambda x_1 \cdots x_n u_1 \) and \( t'_1 \Rightarrow^D \lambda x_1 \cdots x_n u_1 \) for some \( u_1 \), and \( p(A_1; (B_1 \backslash A_1)) = p(B_1; (A_1 \backslash B_1)) \).

4. Consider the case in which (R) is used for both of \( A \) and \( B \). Let \( s \equiv \lambda x_1 \cdots x_n u_1 \). We have \( A_1 : s_1 \Rightarrow^D t_1 \) and \( B_1 : s_1 \Rightarrow^D t'_1 \), where \( t \equiv \lambda x_1 \cdots x_n t_1 \) and \( t' \equiv \lambda x_1 \cdots x_n t'_1 \), from which induction that \( t_1 \Rightarrow^D \lambda x_1 \cdots x_n u_1 \) and \( t'_1 \Rightarrow^D \lambda x_1 \cdots x_n u_1 \) for some \( u_1 \), and \( p(A_1; (B_1 \backslash A_1)) = p(B_1; (A_1 \backslash B_1)) \).
We prove this proposition by induction on the structure of $s$.

**Proposition 6**

In order to prove Lemma 6, we need to describe a number of properties. From case 2, there exist developments $B' : t'' \to D_B u''$ and $A' : t'' \to D_A u''$, where $\varepsilon \notin D_A \cup D_B$. Therefore, there exist developments $t'' \to u'' \to u'$ and $t'' \Rightarrow t \Rightarrow u$ from Lemma 18. There also exist developments $t'' \to u'' \to u$ and $t'' \Rightarrow t \Rightarrow u$ from Lemma 18. Since developments $u'' \Rightarrow u$ and $u'' \Rightarrow u'$ are contracted at the same position $\varepsilon$, $u$ and $u'$ are the same. We also have $p\langle (A; (B \setminus A)) = p\langle (B; (A \setminus B))$ from Case 2 and Lemma 18. (See Fig. 3)

**A.4 Proof of Lemma 6**

In order to prove Lemma 6, we need to describe a number of properties.

**Proposition 6** For any terms $s$ and $t$, if $s \Rightarrow t$ satisfies the top-down property then if $s \Rightarrow t$ has minimal top-down decomposition, and vice versa.

**Proof.** We prove this proposition by induction on the structure of $t$. Since $s \Rightarrow t$ has the top-down property, there exists a development $s \Rightarrow u \equiv \lambda x_1 \cdots x_n.a(u_1, \ldots, u_m) \Rightarrow D \lambda x_1 \cdots x_n.a(t_1, \ldots, t_m) \equiv t$ for some $k$, where $u$ and $D$ are uniquely determined and $u_i \Rightarrow t_i$ has the top-down property for all $i = 1, \ldots, m$. By induction, $u_i \Rightarrow t_i$ has minimal top-down decomposition, the length of which is $k_i$. Let $k_0$ be the minimal $k$ such that $s \Rightarrow_k u$. Thus, we can make a minimal top-down decomposition such that $s \Rightarrow_{k_0} \lambda x_1 \cdots x_n.a(u_1, \ldots, u_m) \Rightarrow_k \cdots \Rightarrow_k = \lambda x_1 \cdots x_n.a(t_1, \ldots, t_m)$ the length of which is $k_0 + k_1 + \cdots + k_m$. The reverse is trivial. □

**Lemma 19** Let $s$ and $t$ be normalized terms, and let $\theta$ and $\sigma$ be normalized substitutions. If $s \Rightarrow t$ satisfies the top-down property and $F\theta \Rightarrow F\sigma$ satisfies the top-down property for any $F \in FV(t) \cap \text{Dom}(\sigma)$, then $s\theta \Rightarrow t\sigma$ satisfies the top-down property.

**Proof.** Proposition 4 asserts that there exists the development $s\theta \Rightarrow t\sigma$. This development can be proved to have the top-down property as follows.

Since $s \Rightarrow t$ has the top-down property, we know that there exists a term $u$ such that $s \Rightarrow u \Rightarrow \Rightarrow t$ and $u \Rightarrow \Rightarrow t$ has the top-down property.

First, based on the fact that $s \Rightarrow u$, we can easily prove that

1. $s\theta \Rightarrow u\theta$.

Second, the fact that $u \Rightarrow \Rightarrow t$ has the top-down property allows us to prove that

2. $u\theta \Rightarrow u\theta \theta$ has top-down property.

Combining (1) and (2), we know that $s\theta \Rightarrow t\sigma \Rightarrow \Rightarrow \theta$ has the top-down property.

The above fact (2) can be proved by induction on $(t, \sigma)$ with $\Rightarrow_{\beta_{\beta}}$:

(A) Consider the case in which the development $u \Rightarrow t$ is generated from definition rule (A). Let $u \equiv a(u_1, \ldots, u_n)$ and $t \equiv a(t_1, \ldots, t_n)$.

1. If $a \in F \cup \text{Dom}(\sigma)$, then $u_i \Rightarrow t_i$ satisfies the top-down property for each $i$, because $u \Rightarrow t$ satisfies the top-down property. It follows from $(t, \sigma) \Rightarrow_{\beta_{\beta}} (t_i, \sigma)$ that $u_i\theta \Rightarrow t_i\sigma \Rightarrow \Rightarrow \theta$ satisfies the top-down property by induction. Thus, $u\theta \Rightarrow a(u_i\theta \downarrow, \ldots, u_n\theta \downarrow) \Rightarrow \Rightarrow a(t_i\sigma \downarrow, \ldots, t_n\sigma \downarrow) \equiv t\sigma \downarrow$ satisfies the top-down property.

2. Consider the subcase in which $a \in \text{Dom}(\sigma)$. We write $G$ for $a$.

Let $G\theta \equiv \lambda x_1 \cdots x_n.a$, $G\sigma \equiv \lambda x_1 \cdots x_n.a$, $\theta' = \{x_1 \Rightarrow u_1\theta \downarrow,$
... $x_n \mapsto u_n \downarrow \},$ and $\sigma' = \{ x_1 \mapsto t_1 \downarrow, \ldots, x_n \mapsto t_n \sigma \downarrow \}.$ Then, $u \theta \Downarrow \equiv (\lambda x_1 \cdots x_n. u') (u_\theta, \ldots, u_n \theta) \Downarrow \equiv u' \theta' \downarrow,$ and $t \sigma \Downarrow \equiv (\lambda x_1 \cdots x_n. t') (t_\sigma, \ldots, t_n \sigma) \Downarrow t' \sigma' \downarrow.$ Since $G \theta \Downarrow \equiv G \sigma \Downarrow$ satisfies the top-down property, $u' \Downarrow \Downarrow t' \Downarrow$ satisfies the top-down property. Therefore, $x, t \theta \Downarrow \equiv u, t \sigma \Downarrow \equiv x, t \sigma'$ satisfies the top-down property by induction. Since $(t, \sigma) \succ_{\beta} (t', \sigma')$, the development $u' \theta' \Downarrow \equiv t' \sigma' \downarrow$ satisfies the top-down property by induction.

Let $s \Downarrow t$ has the top-down property, the claim is easily shown by induction.

Hence, $u \theta \Downarrow \equiv v \Downarrow \Downarrow \Downarrow v \sigma \Downarrow v'$ satisfies the top-down property for any $i \in \text{Pos}(u')$. Therefore, $s \theta \Downarrow \equiv t \sigma \Downarrow$ satisfies the top-down property. □

Lemma 6 Any development has minimal top-down decomposition, the length of which is uniquely determined.

Proof. We show that any $s \Downarrow t$ has the top-down property. This can be proved inductively with respect to the definition of developments. Thus, Proposition 6 asserts that there exists a minimal top-down decomposition of the development for which the length is uniquely determined.

(A) Case in which $s \equiv a(s_1, \ldots, s_n) \Downarrow a(t_1, \ldots, t_n) \Downarrow t$ and $s_i \Downarrow t_i$: Developments $s_i \Downarrow t_i$ have the top-down property by induction. Since we can write the development $s \Downarrow t$ as $s \Downarrow^0 s \Downarrow^1 s \Downarrow t$, the development $s \Downarrow t$ has the top-down property.

(L) The case in which $s \equiv \lambda x_1 \cdots x_n. s' \Downarrow \lambda x_1 \cdots x_n. t' \Downarrow t$: By induction, development $s' \Downarrow t'$ has the top-down property. Thus, development $s \Downarrow t$ also has the top-down property.

(R') The case in which $s \equiv l \theta' \Downarrow \Downarrow D : l \theta \Downarrow \Downarrow r \theta \Downarrow \Downarrow t$ where $\varepsilon \not\in D'$: For any $F \in FV(l)$, the orthogonality of the HRS under consideration asserts that there exist developments $F \theta' \Downarrow F \theta,$ which have the top-down property, by induction. A development $s \Downarrow r$ also has the top-down property because we can write $s \Downarrow \varepsilon.$ Thus, by Lemma 19, the development $l \theta' \Downarrow \Downarrow r \theta \Downarrow$ has the top-down property. Therefore, the development has top-down decomposition by Proposition 6.

A.5 Proof of Lemma 7

In order to prove Lemma 7, we must prepare three lemmas.

Lemma 20 Let $F$ be a variable, let $t$ be a term, let $\sigma$ be a substitution, and let $v$ and $v'$ be positions such that $v' \prec v$. Then, the following hold:

(a) If $F \sigma |_{v'}$ is a redex, $\forall p \in PV(t, \sigma, F, v), \exists q \in PV(t, \sigma, F, v'), p' \prec p$.

(b) If $t |_{v'}$ is a redex, $\forall p \in PT(t, \sigma, v), \exists q \in PT(t, \sigma, v'), p' \prec p$.

Proof. We prove (a) and (b) simultaneously by induction on $(t, \sigma)$ with $\succ_{\beta}$. First, we consider (a). Let $P = PV(t, \sigma, F, v)$, and let $P' = PV(t, \sigma, F, v')$. We have six cases according to Definition 1.

(PV1) We have $P = \{ v \}$ and $P' = \{ v' \}$. Hence, the claim holds.

(PV2) Let $t \equiv a(t_1, \ldots, t_n)$ and $p \in P$. Then, we have $p = iq$ for some $i$ and $q \in PV(t_1, \sigma, F, v)$. Since $(t, \sigma) \succ_{\beta} (t_1, \sigma)$, we have $q' \prec q$ for some $q' \in PV(t_1, \sigma, F, v')$ by induction. Thus, we have $iq' \prec iq = p$ and $iq' \in P'$.

(PV3) Let $t \equiv \lambda x_1 \cdots x_n. t', F \sigma \equiv \lambda y_1 \cdots y_n. t'$, and $P = PV(t', \sigma', F, v)$ and $P' = PV(t', \sigma', F, v')$. Then, we have $(t, \sigma) \succ_{\beta} (t', \sigma')$. The claim follows from induction.

(PV4) Let $t \equiv G(t_1, \ldots, t_n)$ and $G \sigma \equiv \lambda y_1 \cdots y_n. t'$. Let $p \in P$, $Q = PV(t_1, \sigma, F, v)$, and $Q' = PV(t_1, \sigma, F, v')$. Then, we have $p \equiv PV(t', \sigma', y_1, q)$ for some $q \in Q$ and $i, s_1 \equiv s_1 \Downarrow t_1$, $s_2 \equiv s_2 \Downarrow t_2$, and so on. Since $s \equiv s_1 \Downarrow t_1$, $s_2 \Downarrow t_2$, and so on, we have $s' \sim q$ for some $q' \in Q'$ by induction. Thus, we have $s' \sim q$ by induction that $p' \prec p$ for some $p' \in PV(t', \sigma', y_1, q') \subseteq P$.

(PV5) Let $t \equiv F(t_1, \ldots, t_n)$ and $F \sigma \equiv \lambda y_1 \cdots y_n. t'$. We only check the subcase in which $p \in PV(t', \sigma', v)$, because other subcases are similar to (PV4). $t' |_{v'}$ is a redex because $(F \sigma) |_{v'} \equiv (\lambda y_1 \cdots y_n. t') |_{v'} \equiv t' |_{v'}$. It follows from $(t, \sigma) \succ_{\beta} (t', \sigma')$ by induction that $p' \prec p$ for some $p' \in PV(t', \sigma', v')$.

(PV6) It is obvious from $P = P' = \{ \}$. Next, we consider (b). Let $P = PT(t, \sigma, v)$ and $P' = PT(t, \sigma, v')$. We have four cases according to Definition 2.

(PT1) $v' = \varepsilon$. Based on the assumption that $t |_{v'}$ is a redex, we have $t \equiv t$. Let $p \in P = PT(t_1, \sigma, v)$. Then, from the definition of $PT$, we have $p \approx t$. Thus, the claim follows from $P' = \{ \}$.

(PT2) Let $v' = iw'$ and $t \equiv f(t_1, \ldots, t_n)$. We have $v = iu$ and $u' \prec w$ for some $w$. Let $p \in P$. Then, we have $p = iu$ for some $q \in PT(t_1, \sigma, w)$. Since...
(t, σ) ↛_β (t_i, σ), and t_i|w' is a redex, we have q' ↛ q for some q' ∈ PT(t_i, σ, w') by induction. Thus, we have iq' ↛ iq = p and iq' ∈ P'.

\textbf{(PT3) Let } t ≡ λx_1⋯x_n t'. \text{ We have } P = PT(t', σ', v) \text{ and } P' = PT(t', σ', v'), \text{ where } σ' = σ|_{\{x_1,...,x_n\}}. \text{ Since } (t, σ) ↛_β (t', σ') \text{ and } t_i|w' ≡ t_i'|w', \text{ the claim follows from induction.}

\textbf{(PT4) Let } v' = iv', t ≡ G(t_1,...,t_n), \text{ and } Gσ ≡ λy_1⋯y_n t'. \text{ We have } v = iw \text{ and } w' ≪ w \text{ for some } w. \text{ Let } p ∈ P. \text{ Then, we have } p ∈ PV(t', σ', y_i, q) \text{ for some } q ∈ PT(t_1, σ, w), \text{ where } σ' = \{y_1 ≡ t_1σ \downarrow, ..., y_n ≡ t_nσ \downarrow\}. \text{ Since } (t, σ) ↛_β (t_1, σ), \text{ and } t_i|w' \text{ is a redex, we have } q' ↛ q \text{ for some } q' ∈ PT(t_1, σ, w') \text{ by induction. } (y_iσ')|w' ≡ (t_1σ)|w' \text{ is a redex because } q' ∈ PT(t_1, σ, w') \text{ and } t_i|w' \text{ is a redex from Lemma 16(a). Since } (t, σ) ↛_β (t', σ'), \text{ it follows from } q' ↛ q \text{ and induction that } p' ↛ p \text{ for some } p' ∈ PV(t', σ', y_i, q'). \text{□}

\textbf{Lemma 21} \text{ Let } t ↛ r \text{ be a rewrite rule, let } σ \text{ be a substitution, and let } A \text{ be a development such that } A : σ\downarrow \Rightarrow rσ\downarrow. \text{ Let } p \text{ and } p' \text{ be redexes in } Pos(σ\downarrow) \text{ such that } \varepsilon < p' < p. \text{ Then, } ∀q, \forall\delta p, 3q' ∈ p\backslash A, q' < q.

\textbf{Proof.} \text{ There exists } p_1 ∈ Pos_{FV}(t) \text{ such that } p' = p_1p_2 \text{ and } p = p_1p_2p_2' \text{ for some } p_2 \text{ and } p_2' \text{ from orthogonality. From the definition of descendants, we have } p\backslash A = PV(r, σ, top(p_1), p_2) \text{ and } p\backslash A = PV(r, σ, top(p_1), p_2p_2'). \text{□}

\text{Here, } (top(p_1), σ)|_p \text{ is a redex. By Lemma 20, this claim holds.}\text{□}

\textbf{Lemma 22} \text{ Let } A : t ↛(ε) t' \text{ be a development, let } B \text{ and } D \text{ be sets of redex positions of } t \text{ such that } D_{\nabla B}, \text{ and } ε \notin B. \text{ Then, } (D\backslash A)_{\nabla (B \backslash A)}. \text{□}

\textbf{Proof.} \forall p ∈ D, \text{ there exists } p' ∈ B \text{ such that } ε < p' < p \text{ from } D_{\nabla B}. \text{ Therefore, this lemma holds from Lemma 21.} \text{□}

\textbf{Lemma 7} \text{ Let } B \text{ be a set of redex positions of a term } t, \text{ and let } D \text{ and } D' \text{ be sets of redex positions such that we can write them by } D_{\nabla B} \text{ and } D'_{\nabla B}, \text{ respectively. Let } A_1 : t ↛ D t_1 \text{ and } A_2 : t_1 ↛ D' t_2 \text{ be developments. Then, there exist developments } A_3 : t ↛ D' t_3 \text{ and } A_4 : t_3 ↛ D'' t_2 \text{ such that } D'' = D\backslash A_3 \text{ can be written by } (D\backslash A_3)_{\nabla (B \backslash A_3)} \text{ for some } t_3.

\textbf{Proof.} \text{ If } ε \notin B, \text{ the claim holds because } D' = \emptyset. \text{ Consider the case in which } ε \notin B. \text{ There exists a development } A_3 \text{ corresponding to } A_2. \text{ There exists development } A_4 : t_3 ↛ D'' t_2 \text{ from Lemma 3. We show that } (D\backslash A_3)_{\nabla (B \backslash A_3)} \text{ by induction on the definition of the development of } A_3 \text{ (Fig. 4). The cases of } (L) \text{ and } (A) \text{ hold from induction. In the case of } (R'), \text{ } A_3 \text{ is divided into } A'; A'', \text{ where } A' : t ↛ D' \Downarrow t''' \text{ and } A'' : t'' \Leftarrow t_3. \text{ Then, } (D\backslash A')_{\nabla (B \backslash A')} \text{ by induction. } (D\backslash A')_{\nabla (B \backslash A')} \text{ by Lemma 22. Therefore, from the definition of developments, } (D\backslash A'; A'')_{\nabla (B \backslash A'; A'')} \text{□}

\text{(Received September 28, 2008)}
\text{(Accepted December 27, 2008)}
\text{(Released April 2, 2009)}

\text{(Original version of this article can be found in the IPSJ Transactions on Programming, Vol.2, No.2, pp.144–165.)}

\textbf{Hideto Kasuya} completed graduate course of Nagoya University in 1997. He is a Research Associate of the Faculty of Information Science and Technology, Aichi Prefectural University. He is interested in term rewriting system and rewriting strategy. He is a member of IPSJ, IEICE and JSSST.
**Masahiko Sakai** completed graduate course of Nagoya University in 1989 and became Assistant Professor, where he obtained a D.E. degree in 1992. From April 1993 to March 1997, he was Associate Professor in JAIST, Hokuriku. In 1996 he stayed at SUNY at Stony Broook for six months as Visiting Research Professor. From April 1997, he was Associate Professor in Nagoya University. Since December 2002, he has been Professor. He is interested in term rewriting system, verification of specification and software generation. He received the Best Paper Award from IEICE in 1992. He is a member of IEICE and JSSST.

**Kiyoshi Agusa** is a professor of Department of Information Systems, Graduate School of Information Science, Nagoya University. He received Ph.D. degree in computer science from Kyoto University in 1982. His research area is software engineering, especially dependable software, programming environment and software reusing. He is a member of ACM, IEEE, IPSJ, IEICE and JSSST.