Innermost Reductions Find All Normal Forms on Right-Linear Terminating Overlay TRSs

Masahiko Sakai,Kouji Okamoto,Toshiki Sakabe

Graduate School of Information Science,
Nagoya University,
Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan

Abstract

A strategy of TRSs is said to be complete if all normal forms of a given term are reachable from the term. We show the innermost strategy is complete for terminating, right-linear and overlay TRSs. This strategy is fairly efficient to calculate all normal forms of a given term by searching reduction trees. We also discuss the possibilities for weakening the conditions.

1 Introduction

Most of properties of non-CR (Church Rosser) or non-UN (uniquely normalizing) TRSs are still hidden in mist. Concerning normalizing strategies that guarantee a safe evaluation in order to obtain a normal form, a lot of studies are effective on orthogonal TRSs[7,9,13,14,5,6]. There are several studies on normalizing strategies for non-orthogonal TRSs[11,18,3], which are effective on UN TRSs.

On the other hand, the authors have been studying on a transformational approach of inverse computation of functions given by a TRS[15,16]. The conversion itself can be done for constructor TRSs, which contain no defined symbol in any proper subterms of the left-hand sides. For instance, consider the following TRS:

$$R_1 \;=\; \begin{cases} d(0) \to 0, \\ d(s(0)) \to 0, \\ d(s(s(x)) \to s(d(x)) \end{cases}$$

where the function d is for division by two. The transformation[15,16] gives the following TRS from R_1:

$$R_2 = \begin{cases} D(0) \to 0, \\ D(0) \to s(0), \\ D(s(y)) \to U(D(y)), \\ U(x) \to s(s(x)) \end{cases}.$$

This TRS R_2 has a function D that calculates inverse image with respect to d; since $d(s(s(s(0)))) \xrightarrow{*}_{R_1} s(0)$, we have $D(s(0)) \xrightarrow{*}_{R_2} s(s(s(0)))$. On the other hand, $D(s(0))$ is also reachable to $s(s(0))$, because $d(s(s(0))) \xrightarrow{*}_{R_1} s(0)$. As this example shows, the output TRSs are non-UN in general. Thus, strategies that can find all normal forms are important.

In this paper, we first give the notion of the complete strategy, which can find all normal forms of a given term. Then, we show that innermost reductions is a complete strategy of terminating, right-linear and overlay TRSs. On UN TRSs, we know that any strategy is complete, of course. However, it is not true in non-UN setting.

As a result, it appears that the innermost reductions contribute in efficiency to find all normal forms shown as follows.

Example 1.1 Consider the following TRS:

$$R_3 = \begin{cases} f(x) \to x, \\ g(x) \to i(x), \\ h(x, y) \to y, \\ h(x, i(y)) \to x \end{cases}$$

R_3 is terminating but not UN (thus not CR). Hence, we have to search all spaces in order to obtain all normal forms 0 and $i(0)$ of a term $h(f(0), g(0))$. We have 14 reductions required to calculate its all normal forms without memorizing terms encountered through the search. The search space is reduced to only 4 reductions by the leftmost innermost reduction as shown in Figure 1. Of course, we can use dag search which takes 11 reductions. However constructing the dag that shares the same terms is considerably heavy.

Consider the transformation[15,16] again. In general, the output TRSs are non-terminating and overlay, and contain extra variables, where extra variables are variables that appear in the right-hand side of a rule and not in the left-hand side of the rule. However, we know that the output TRSs have no extra variables if the inputs are non-erasing, and that they are right-linear if the inputs are left-linear. Therefore, innermost reductions contribute to an efficient computation of the inverse image of functions, if the input constructor TRSs are left-linear and non-erasing and the output TRSs are terminating.

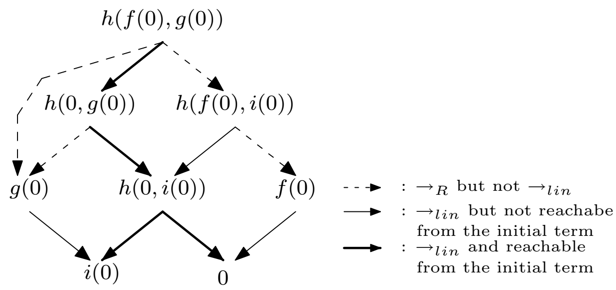

Fig. 1. The search space from $h(f(0), g(0))$ on R_3

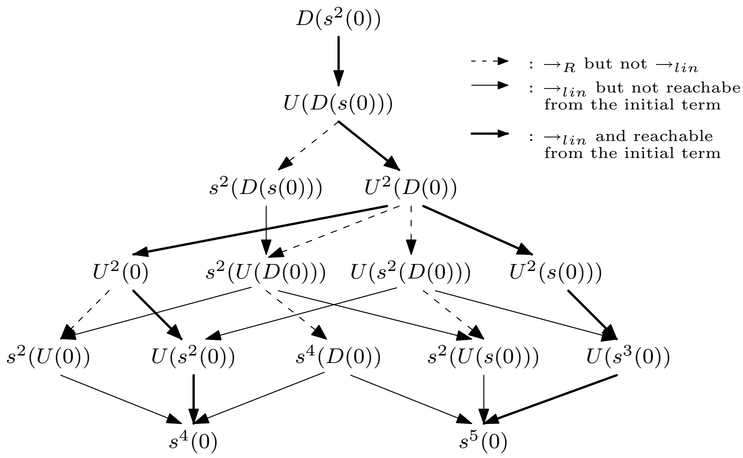

Fig. 2. The search space from $D(s(s(0)))$ on R_2

Example 1.2 Consider the TRS R_2 above. While 34 reductions are required to calculate all normal forms of $D(s(s(0)))$ by the depth-first tree search on $\rightarrow_{R_2}$, the search space is reduced to only 8 reductions by the leftmost innermost reduction as shown in Figure 2.

2 Preliminary Concepts

In this paper, we mainly follow the notation of [4,12]. An *abstraction reduction system* is a structure $A = (S, \rightarrow)$ where S is a set and $\rightarrow$ is a binary relation over S, which is called a *reduction relation*. A *reduction sequence* is a finite sequence $a_0 \rightarrow a_1 \rightarrow \cdots \rightarrow a_n$ $(n \geq 0)$ or an infinite sequence $a_0 \rightarrow a_1 \rightarrow \cdots$ of reductions. The identity of elements a and b in S is denoted by $a \equiv b$. The reflexive and transitive closure of $\rightarrow$ is denoted by $\xrightarrow{*}$. The transitive closure of $\rightarrow$ is denoted by $\xrightarrow{+}$. If there is no element b such that $a \rightarrow b$, then we say a is a *normal form* (with respect to A). We use NF_A or $NF_{\rightarrow}$ for the set of all normal forms. If $a \xrightarrow{*} b \in NF_A$, we say b is a normal form of a or a has a normal form b. We say b is *reachable* from a (by $\rightarrow$) if $a \xrightarrow{*} b$. We say A is *confluent* (or equivalently Church-Rosser, CR) if $a \xrightarrow{*} b \wedge a \xrightarrow{*} b'$ implies $\exists c \in S$, $b \xrightarrow{*} c \wedge b' \xrightarrow{*} c$ for all a, b and b' in S. We say A is *uniquely normalizing* (UN) if every element a has at most one normal forms. We say

A is *terminating* if there exists no infinite reduction sequence.

Let F be a set of function symbols accompanied with a mapping *arity* from F to the set of natural numbers, which is called a *signature*. Let X be a set of variables. The set of all terms over F and X is denoted by $T(F, X)$ (or simply T). We often use f and g for function symbols, x, y and z for variables, and s, t and u for terms. The set of variables which appear in a term t is denoted by $Var(t)$. For any term t and function symbol f with $arity(f) = 1$, we use $f^n(t)$ to represent $\overbrace{f(\cdots f(}^{n} t) \cdots)$. A term t is called *linear* if every variable in t occurs only once.

We use an extra constant $\Box$ as *hole* in terms. A term C in $T(F \cup \{\Box\}, X)$ is called a *context*. For a context C with n $\Box$s and for $t_1, \ldots, t_n \in T(F, X)$, $C[t_1, \ldots, t_n]$ denotes the term obtained by replacing $\Box$s with $t_1, \ldots, t_n$ from left to right order. In the sequel, we use contexts that possess exactly one $\Box$. We say a reduction $\rightarrow$ is monotonic if $t \rightarrow s$ implies $C[t] \rightarrow C[s]$ for any context C. If $t \equiv C[s]$, we say that s is a *subterm* of t written as $t \trianglerighteq s$. Moreover, if $C \not\equiv \Box$, s is a *proper subterm* of t, denoted by $t \rhd s$. The reduction $(\rightarrow \cup \rhd)$ is terminating if $\rightarrow$ is monotonic and terminating.

The notation $\{x_1 \mapsto u_1, \ldots, x_n \mapsto u_n\}$ denotes a *substitution* σ such that $x_i\sigma \equiv u_i$.

A *rewrite rule* is a pair (l, r), denoted by $l \rightarrow r$, where the left-hand side l ($\notin X$) and the right-hand side r are terms such that $Var(l) \supseteq Var(r)$. Letting R be a set of rewrite rules, $(T(F, X), \rightarrow_R)$ is called a *term rewriting system* (*TRS*), where $s \rightarrow_R t$ if and only if $s \equiv C[l\sigma]$ and $t \equiv C[r\sigma]$ for some $l \rightarrow r$ in R, context C and substitution σ. We say $l\sigma$ a redex. If $C \equiv \Box$ we say it is a top reduction, denoted by $s \xrightarrow[\varepsilon]{} t$; Otherwise, $s \xrightarrow[\varepsilon<]{} t$. In the sequel, we identify R with $(T(F, X), \rightarrow_R)$ if that causes no confusion. An another definition of $\rightarrow_R$ is as follows:

(a) $l\sigma \rightarrow_R r\sigma$ for a substitution σ and a rewrite rule $l \rightarrow r$.

(b) $f(s_1, \ldots, s_{i-1}, s, s_{i+1}, \ldots, s_n) \rightarrow_R f(s_1, \ldots, s_{i-1}, t, s_{i+1}, \ldots, s_n)$, if $s \rightarrow_R t$.

We say that a rewrite rule $l \rightarrow r$ is *right-linear* if r is linear, and say that a rewrite rule $l \rightarrow r$ is *left-linear* if l is linear. A TRS is called *right-linear* (*left-linear*) if every rule is right-linear (left-linear). The rules $l \rightarrow r$ and $l' \rightarrow r'$ *overlap* if there exist a subterm s of l' and substitutions σ and σ' such that $s \notin X$ and $l\sigma \equiv s\sigma'$. Especially, we say that they overlap at non-root position if $s \not\equiv l'$. A TRS is *overlay* if it has no overlapping rules at non-root position. A TRS is *non-erasing* if $Var(l) = Var(r)$ for every rule $l \rightarrow r$.

3 Complete strategy of TRSs

Firstly, we give a notion of complete strategy.

Definition 3.1 Let R be a TRS. A relation $\rightarrow_c$ is called a *strategy* of R, if

(a) $\rightarrow_c \subseteq \rightarrow_R$, and

(b) $NF_R = NF_{\rightarrow_c}$.

A strategy $\rightarrow_c$ of R is *complete*, if

(c) $t \xrightarrow{*}_R u \in NF_R$ implies $t \xrightarrow{*}_c u$.

Obviously $\rightarrow_R$ itself is a trivial complete strategy of R, although it is nonsense.

We can regard the sequence $t \xrightarrow{*}_c u$ as a standard sequence of $t \xrightarrow{*}_R u$, if the sequence $t \xrightarrow{*}_c u$ is unique. Hence, the notion of complete strategies are related to the notion of standardizations.

If we can find a terminating complete strategy $\rightarrow_c$, it can find all normal forms of a given term with respect to R. Even if R is terminating, non-trivial complete strategies are worthful, because they contribute to the efficiency. Note that the condition (b) is not necessary for finding all normal forms. Assume we want to find all normal forms of t by using a terminating strategy $\xrightarrow{*}_c$ that satisfies (a) and (c). Since $NF_R \subseteq NF_{\rightarrow_c}$ by (a), the only difference is that it may find a normal form of t in $NF_{\rightarrow_c}$ but not in NF_R. However, we can simply dispose the term since (c) ensures that all normal forms of t with respect R are reachable from t by $\rightarrow_c$.

Next, we give a formal definition of innermost reductions.

Definition 3.2 Let R be a TRS. The *innermost* reduction relation of R, denoted by $\rightarrow_{in}$, is defined as follows:

(a) $l\sigma \rightarrow_{in} r\sigma$ for any substitution σ and rewrite rule $l \rightarrow r$ if all proper subterms of $l\sigma$ are normal forms.

(b) $f(s_1, \ldots, s_{i-1}, s, s_{i+1}, \ldots, s_n) \rightarrow_{in} f(s_1, \ldots, s_{i-1}, t, s_{i+1}, \ldots, s_n)$ if $s \rightarrow_{in} t$.

The *leftmost innermost* reduction $\rightarrow_{lin}$ (*rightmost innermost* reduction $\rightarrow_{rin}$) is defined in similar to above by adding to (b) an extra condition that $s_1, \ldots, s_{i-1}$ ($s_{i+1}, \ldots, s_n$) are normal forms.

Obviously, $\rightarrow_{lin} \subseteq \rightarrow_{in} \subseteq \rightarrow_R$ and $\rightarrow_{rin} \subseteq \rightarrow_{in} \subseteq \rightarrow_R$.

The following two lemmas show general properties of innermost reductions.

Lemma 3.3 *Let R be a TRS. Let $\rightarrow_c$ be either $\rightarrow_{lin}$, $\rightarrow_{rin}$, or $\rightarrow_{in}$. If $t \xrightarrow{*}_c s \in NF_R$, then there exists a term t' such that $t \underset{\varepsilon< \ c}{\xrightarrow{*}} t' \xrightarrow{*}_c s$ and every proper subterm of t' is a normal form.*

Proof. If the reduction sequence contains no top reduction, it is trivial by taking t as t'. Otherwise, we can represent the reduction sequence as $t \underset{\varepsilon< \ c}{\xrightarrow{*}} t' \underset{\varepsilon \ c}{\rightarrow} t'' \xrightarrow{*}_c s$. Since $t' \underset{\varepsilon \ c}{\rightarrow} t''$ is innermost, all proper subterms of t' are normal forms. □

Lemma 3.4 *Let R be a TRS, t be a linear term, s be a normal form, and σ be a substitution. Let $\rightarrow_c$ be either $\rightarrow_{lin}$, $\rightarrow_{rin}$, or $\rightarrow_{in}$. If $t\sigma \xrightarrow{*}_c s$, then there exists a normalized substitution σ' such that $t\sigma \xrightarrow{*}_{in} t\sigma' \xrightarrow{*}_c s$.*

Proof. We prove by structural induction on t. In the case that t is a variable x, it is enough by taking $\{x \mapsto s\}$ as σ'. In the case that $t \equiv f(t_1, \ldots, t_n)$, we

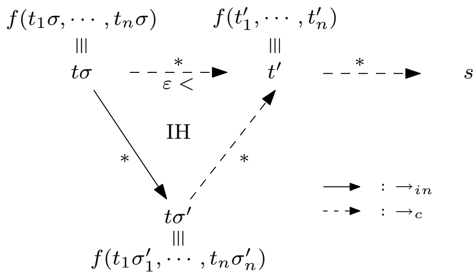

Fig. 3. The proof of Lemma 3.4

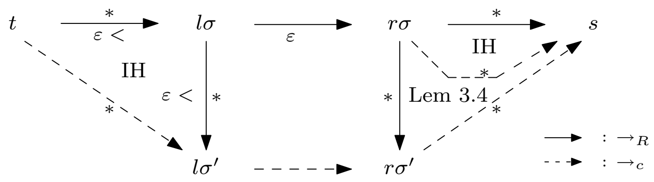

Fig. 4. The proof of Theorem 3.5 (the former case)

have $t\sigma \equiv f(t_1\sigma, \ldots, t_n\sigma)$. From Lemma 3.3, there exists a term t' such that $t\sigma \xrightarrow[\varepsilon<]{*}_c t' \xrightarrow{*}_c s$ and every proper subterm of t' is a normal form (See Figure 3). Since there is no top reduction in $t\sigma \xrightarrow[\varepsilon<]{*}_c t'$, the term t' can be represented as $f(t'_1, \ldots, t'_n)$. Since we have $t_i\sigma \xrightarrow{*}_R t'_i$, it follows from induction hypothesis that there exists normalized substitution σ'_i such that $t_i\sigma \xrightarrow{*}_{in} t_i\sigma'_i \xrightarrow{*}_c t'_i$ for each i. Hence, $t\sigma \equiv f(t_1\sigma, \ldots, t_n\sigma) \xrightarrow{*}_{in} f(t_1\sigma'_1, \ldots, t_n\sigma'_n) \xrightarrow{*}_c f(t'_i, \ldots, t'_n) \equiv t' \xrightarrow{*}_c s$. Here, we can compose σ'_is into one substitution σ' from the linearity of t. Therefore, we conclude this case since $f(t_1\sigma'_1, \ldots, t_n\sigma'_n) \equiv t\sigma'$. $\square$

Theorem 3.5 *Let R be a right-linear terminating overlay TRS. Then, the innermost strategy, the leftmost innermost strategy and the rightmost innermost strategy are complete strategies of R.*

Proof. Let $\to_c$ be either $\to_{lin}$, $\to_{rin}$, or $\to_{in}$. We prove that $t \xrightarrow{*}_R s \in NF_R$ implies $t \xrightarrow{*}_c s$ by Noetherian Induction on t with respect to $(\to_R \cup \rhd)$. We have two cases whether $t \xrightarrow{*}_R s$ contains top reductions or not.

Firstly, we consider the latter case that is easier. Let $t \equiv f(t_1, \ldots, t_n)$. Then, s can be represented as $f(s_1, \ldots, s_n)$. Since $t_i \xrightarrow{*}_R s_i \in NF_R$ for all i, we have $t_i \xrightarrow{*}_c s_i$ by induction hypothesis. Thus, $t \xrightarrow{*}_c s$ follows.

Secondly, we consider the former case. By focusing the first top reduction, it can be represented as $t \xrightarrow[\varepsilon<]{*}_R l\sigma \xrightarrow[\varepsilon]{}_R r\sigma \xrightarrow{*}_R s \in NF_R$ (See Figure 4). Since $t \xrightarrow{+}_R r\sigma$ we can apply the induction hypothesis to $r\sigma \xrightarrow{*}_R s$, which results $r\sigma \xrightarrow{*}_c s$. It follows from the right-linearity of r and Lemma 3.4 that $r\sigma \xrightarrow{*}_R r\sigma' \xrightarrow{*}_c s$ for some normalized substitution σ'. Then, we have $l\sigma' \to_c r\sigma'$ since σ' is normalized and R is overlay. We also have $t \xrightarrow[\varepsilon<]{*} l\sigma \xrightarrow[\varepsilon<]{*} l\sigma'$ since l is not a variable. Let $t \equiv f(t_1, \ldots, t_n)$. Then, $l\sigma'$ can be represented as $f(s_1, \ldots, s_n)$ and $t_i \xrightarrow{*}_R s_i \in NF_R$. we have $t_i \xrightarrow{*}_c s_i$ by induction hypothesis. Whatever $\to_c$ is either of $\to_{lin}$, $\to_{rin}$ or $\to_{in}$, we can show $t \xrightarrow{*}_c l\sigma'$. Therefore, $t \xrightarrow{*}_c s$. $\square$

Followings are counter examples showing that all of the conditions in Theorem 3.5 are essential.

Example 3.6 Consider the following TRS that is overlay and right linear but not terminating:

$$R_4 \;=\; \begin{cases} f(x) \to b, \\ a \to a \end{cases} .$$

We have $f(a) \to_{R_4} b \in NF_{R_4}$ but not $f(a) \xrightarrow{*}_{in} b$.

Example 3.7 Consider the following TRS that is terminating and right linear but not overlay:

$$R_5 \;=\; \begin{cases} f(a) \to b, \\ a \to c \end{cases} .$$

We have $f(a) \to_{R_5} b \in NF_{R_5}$ but not $f(a) \xrightarrow{*}_{in} b$.

Example 3.8 Consider the following TRS that is terminating and overlay but not right linear:

$$R_6 \;=\; \begin{cases} f(x) \to g(x,x), \\ a \to b, \\ a \to c \end{cases} .$$

We have $f(a) \xrightarrow{*}_{R_6} g(b,c) \in NF_{R_6}$ but not $f(a) \xrightarrow{*}_{in} g(b,c)$.

4 Discussion

We discuss the possibilities for weakening the conditions of Theorem 3.5.

Let's consider the condition "terminating". Even if we replace it by the condition "innermost terminating", the well-founded ordering used in Theorem 3.5 works no more. Nevertheless, we think the conjecture obtained from the theorem by completely removing the condition may hold if we add extra conditions "left-linear" and "non-erasing", because the number of reductions may work as a measurement for the proof. Moreover, if so, the condition "left-linear" will be removed by using the parallel reduction[8]. Hence, we give the following conjecture.

Conjecture 4.1 *Let R be a right-linear non-erasing overlay TRS. Then, the innermost strategy, the leftmost innermost strategy and the rightmost innermost strategy are complete strategies of R.*

Let's consider the following overlay TRS that is not innermost terminating.

$$R_7 \;=\; \begin{cases} f(b,c) \to f(a,a), \\ a \to b, \\ a \to c \end{cases}.$$

We have leftmost innermost reduction sequences from $f(b,c)$ to normal forms $f(b,b)$, $f(c,b)$ and $f(c,c)$ of a term $f(b,c)$. For instance, $f(b,c) \to_{R_7} f(a,a) \to_{R_7} f(c,a) \to_{R_7} f(c,c)$. Hence, the leftmost innermost strategy is complete for R_7.

Even if Conjecture 4.1 holds, we require "innermost terminating" in order to find all normal forms as long as we use innermost strategies. The dependency pair method[1,2] is usable to show the innermost terminating property of TRSs.

If we remove the condition "non-erasing" from Conjecture 4.1, it does not hold any more as shown by Example 3.6. In this case, the authors think that the basic strategy used in basic narrowing[10] is a candidate for a replacement of innermost strategies. Intuitively, a reduction sequence is basic, if every redex, ever substituted into a variable of the rule in a previous reduction, is not reduced. Now, Example 3.6 is not a counter example, because $f(a) \to_{R_4} b$ is a basic reduction sequence. Thus, we have another conjecture.

Conjecture 4.2 *Let R be a right-linear overlay TRS. Then, the basic strategy is a complete strategy of R.*

How about the condition "overlay"? In case that we have non-overlay critical pairs like Example 3.7, complete strategies cannot ignore either reducts of the critical pairs. Hence, we need to introduce weakly innermost strategy, where a redex is weakly innermost if it is innermost or it overlaps some innermost redex. The conjecture is as follows:

Conjecture 4.3 *Let R be a right-linear terminating TRS. Then, the weakly innermost strategy is a complete strategy of R.*

On the other hand, removing the condition "right-linear" seems to be impossible as long as we use strategies based on innermost reduction.

As for outermost based reductions, the authors think that there is a possibility to work as complete strategies of left-linear overlay TRSs. For example, consider the following TRS:

$$R_8 \;=\; \begin{cases} f(a,x) \to c, \\ f(x,a) \to d, \\ b \to a \end{cases}.$$

For a sequence $f(b,b) \xrightarrow{*}_{R_8} d \in NF_{R_8}$, we have no leftmost outermost reduction sequence, which shows that the leftmost outermost strategy is not complete for R_8. However, the outermost strategy is complete, although it is not efficient. The authors prospect that the left-linearity is needed at least, since we have the following counter example:

$$R_9 \ = \ \begin{cases} f(x,x) \to c, \\ f(x,y) \to d, \\ a \to b \end{cases} .$$

Although $f(a,b) \xrightarrow{*}_{R_9} c$, the only outermost sequence is $f(a,b) \to_{R_9} d$.

Developing normalizing strategies, we mean terminating complete strategy of TRSs, supposed to be difficult but should be explored. As for TRSs with extra variables, which we call EV-TRS, the authors have proposed a simulation method[17]. Hence, complete strategies for the simulation are also to be explored.

Acknowledgement

We thank anonymous referees, Dr. Mizuhito Ogawa, and members of TRS Meeting Japan for suggestions. This work is partially supported by Grants from Ministry of Education, Science and Culture of Japan #15500007, by the Intelligent Media Integration Project of Nagoya University as one of the 21st Century COE of JAPAN and by the Nitto foundation.

References

[1] T. Arts and J. Gisel, Proving Innermost Termination Automatically, Proc. of the 8th Conf. on Rewriting Techniques and Applications (RTA'97), LNCS, *1232*, 1997, pp.157–171.

[2] T. Arts, Automatically Proving Termination and Innermost Normalization of Term Rewriting Systems, PhD thesis, Universiteit Utrecht, 1997.

[3] S. Antoy and A. Middeldorp, A Sequential Reduction Strategy, Theoretical Computer Science, *165(1)*, 1996, pp.75–95.

[4] F. Baader and T. Nipkow, Term Rewriting and All That, Cambridge University Press, 1998.

[5] H. Comon, Sequentiality, Monadic Second Order Logic and Tree Automata, Information and Computation, *157*, 2000, pp.25–51.

[6] I. Durand and A. Middeldorp, Decidable Call by Need Computations in Term Rewriting (Extended Abstract), Proc. of the 14th Int. Conf. on Automated Deduction (CADE'97), LNAI, *1249*, 1997, pp.4–18.

[7] G. Huet and J.-J. Lévy, Call-by-Need Computations in Non-Ambiguous Linear Term Rewriting Systems, Rapport INRIA, *359*, 1979.

[8] G. Huet, Confluent Reductions: Abstract Properties and Applications to Term Rewriting Systems, Journal of the ACM, *27*, 1980, pp.797–821.

[9] G. Huet and J.-J. Lévy, Computations in Orthogonal Rewriting Systems, I and II, in Computational Logic, Essays in Honor of Alan Robinson, The MIT Press, 1991, pp.396–443.

[10] J.M. Hullot, Canonical Forms and Unification, Proc. of the 5th Conf. on Automated Deduction (CADE'80), LNCS, *87*, 1980, pp.318–334.

[11] J.R. Kennaway, Sequential Evaluation Strategies for Parallel-Or and Related Reduction Systems, Annals of Pure and Applied Logic, *43*, 1989, pp.31–56.

[12] J. W. Klop, Term Rewriting Systems, in *Handbook of Logic in Computer Science*, ed.Abramsky et al., Oxford University Press, 1992.

[13] M. Oyamaguchi, NV-Sequentiality: A Decidable Condition for Call-by-Need Computations in Term Rewriting Systems, SIAM Journal on Computing, *22(1)*, 1993, pp.114-135.

[14] T. Nagaya, M. Sakai and Y. Toyama, NVNF-Sequentiality of Left-Linear Term Rewriting Systems, RIMS Technical Report, *918*, 1995, pp.109–117.

[15] N. Nishida, M. Sakai and T. Sakabe, Generation of Inverse Term Rewriting Systems for Pure Treeless Functions, The International Workshop on Rewriting in Proof and Computation (RPC'01), Sendai, Japan, October 25-27, 2001, pp.188–198.

[16] N. Nishida, M. Sakai and T. Sakabe, Generation of a TRS Implementing the Inverses of the Functions with Specified Arguments Fixed, Technical Report of IEICE, *COMP2001-67*, 2001, pp.33–40.

[17] N. Nishida, M. Sakai and T. Sakabe, Narrowing-based Simulation of Term Rewriting Systems with Extra variables and its Termination Proof, Proc. of 12th Int'l Workshop on Functional and (Constraint) Logic Programming (WFLP'03), Valencia, Spain, June 12–13, 2003(to appear).

[18] Y. Toyama, Strong Sequentiality of Left-Linear Overlapping Term Rewriting Systems, The 7th annual IEEE Symposium on Logic in Computer Science, 1992, pp.274–284,