Optimal Normalization in Orthogonal Term Rewriting Systems

Zurab Khasidashvili

INRIA-Rocquencourt, 78153 Le Chesnay, France
Zurab.Khasidashvili@inria.fr

Abstract. We design a normalizing strategy for orthogonal term rewriting systems (OTRSs), which is a generalization of the call-by-need strategy of Huet-Lévy [4]. The redexes contracted in our strategy are essential in the sense that they have "descendants" under any reduction of a given term. There is an essential redex in any term not in normal form. We further show that contraction of the innermost essential redexes gives an optimal reduction to normal form, if it exists. We classify OTRSs depending on possible kinds of redex creation as non-creating, persistent, inside-creating, non-left-absorbing, etc. All these classes are decidable. TRSs in these classes are sequential, but they do not need to be strongly sequential. For non-creating and persistent OTRSs, we show that our optimal strategy is efficient as well.

1 Introduction

In this paper, we study correct and optimal computations in Orthogonal Term Rewriting Systems (OTRSs). We only consider one-step rewriting strategies, which select a single redex to contract at each rewriting step. A strategy is correct when it succeeds in computing the normal form of any term, whenever such a normal form exists. A correct strategy is optimal when the computation of normal form is performed for a minimal cost. The choice of one-step rewriting provides us with a very simple and natural cost measure: the cost of a reduction is just the number of elementary steps it comprises, which is also the number of redexes it contracts. Clearly, such a cost notion assumes a term implementation of term rewriting. By contrast, the more sophisticated notions of cost of [1, 12, 13] take the contraction of one set of similar redexes, a so-called redex family, as the unitary cost. Thus, in a realistic implementation, redex families must be implemented as a single shared object, so that optimal reduction are indeed optimal, when real (i.e., implemented) rewriting steps are considered.

The seminal paper on strategies in Orthogonal Term Rewriting Systems is [4] by Huet and Lévy. They designed the call-by-need strategy and proved its correctness for computation of normal forms. Following Huet and Lévy's definition, a redex u in a term t is needed when at least one residual of u is contracted by all normalizing reductions starting from t. Call-by-need is then defined as any strategy that contracts needed redexes only. Huet and Lévy did not limit themselves to the existence of correct strategies; they were also concerned with the feasibility of such strategies. They thus introduced the class of strongly sequential systems. In a strongly sequential system, it is possible to give an algorithm to find a needed redex in any term not in normal form. They also proved that strong sequentiality is a decidable property.

This fundamental work was later extended by many authors. Kennaway and Sleep [6] gave a proof of correctness of the call-by-need strategy for all orthogonal combinatory reduction systems using labeling. Maranget [13] studied the optimality of the graph implementation of call-by-need. Kennaway [5] showed that every OTRS admits a computable correct strategy, and not only strongly sequential systems. In Klop and Middeldorp [10, 11], simpler proofs of decidability of strong sequentiality are given. In Thatte [18] the notion of left-sequentiality is introduced. In Oyamaguchi [15] a class of sufficiently sequential OTRSs, which properly contains the class of strongly sequential OTRSs, is shown to be decidable. In Kesner [7] sequentiality is studied in OTRSs with order-sorted alphabets. Toyama [19] generalized some results from [4] to left-linear overlapping TRSs.

The essential strategy, as defined in this paper, is based on a notion of descendant, which allows the tracing of all subterms, including contracted redexes, during reductions. Essential subterms of a term t have descendants under any reduction starting from t. So they contribute to the finite or infinite result of a computation. For the case of redexes, essentiality coincides with Maranget's improved notion of neededness (developed independently) [13]. By contrast with the notion of neededness of Huet and Lévy, the latter notion also makes sense in the case when the term does not have a normal form.

The existence of an essential redex in any term t not in normal form follows from two observations. First, one can easily construct an "erasing reduction" starting from t in which inessential redexes have no descendants. Second, for any reduction from t to some term s there is at least one redex in t that has a descendant in s. Applying this property to the erasing reduction for t, we obtain a proof that there is at least one redex in t that is essential. (In the body of paper we prove a somewhat stronger theorem.)

The correctness proof of call-by-need in [4] is not very direct, as pointed out by Klop in [10]. Our proof is simple and short. The only thing we need is that essential redexes have essential residuals under any reduction that does not contract them. Therefore, if a term t has a reduction P that contracts infinitely many essential redexes and Q is a reduction from t to a term s, then s possesses a reduction P' contracting infinitely many essential redexes. Essential redexes contracted in P' are residuals of essential redexes contracted in P that are not contracted in Q; thus t is not normalizable. Hence, essential reductions always succeed in finding a normal form, whenever it exists.

Further, we show that reductions are optimal if and only if they contract non-duplicating essential redexes, i.e., redexes that do not duplicate other essential redexes. In particular, innermost essential reductions are optimal. This optimality result relies on the crucial fact that the essential residuals of non-duplicating essential redexes remain non-duplicating. Our strategy is also optimal in the sense of [1, 12, 13], because non-duplicating essential redexes are alone in their families.

We note that complexity of an OTRS depends crucially on the possible kinds of redex creation. We define non-creating, persistent, inside-creating, and outside-creating OTRSs. In non-creating systems, there is no creation of redexes. In persistent systems, creation depends on rewrite rules only; the actual arguments and the context in which reduction takes place do not matter. In particular, Recursive (Applicative) Program Schemes [2] are persistent. In inside-creating systems, the arguments, but not the context, can also take part in creation; any created redex is inside the contractum of a creating redex. In outside-creating systems, the context always takes part in redex creation.

Since before erasure a redex must be absorbed in an argument of a redex above it, it is also relevant to consider a classification of outside creation depending on what kind of absorption is possible. For example, the leftmost outermost redex in a term is essential in left-normal TRSs (where function symbols precede variables in the left-hand sides of rules [14]), because these TRSs are non-left-absorbing. So we distinguish non-absorbing, non-left-absorbing, and non-right-absorbing systems.

All these classes have different normalization behavior. For example, in outside-creating systems, all reductions are terminating [17]. In persistent systems, normalization and termination are decidable. In non-absorbing systems, all outermost redexes are essential, etc. Furthermore, these classes are easily decidable. In fact, the above classification is a classification depending on critical pairs between left-hand sides and right-hand sides of rewrite rules. In our opinion, a similar classification of TRSs will be useful for further development of TRS theory. Here we study the Persistent TRSs in more depth. In particular, we show that in Persistent TRSs the innermost essential strategy is computable and that weak normalization is decidable. Note that some of the Persistent TRSs may not be strongly sequential. Such a system can be obtained from any non-strongly-sequential system by replacing in its rules right-hand sides with fresh constants.

The rest of the paper is organized as follows. In section 2 we introduce the notion of strict equivalence of reductions, based on the descendant notion, and establish the "strict" Church-Rosser theorem. In section 3 we prove the correctness of the essential strategy. Section 4 is devoted to the classification of OTRSs. In section 5 we obtain the optimality result. In section 6 we study properties of inessential and independent subterms. In section 7 we establish the relation between neededness and essentiality, and in section 8 we study optimal reductions in Persistent TRSs.

2 Strict Confluence for OTRSs

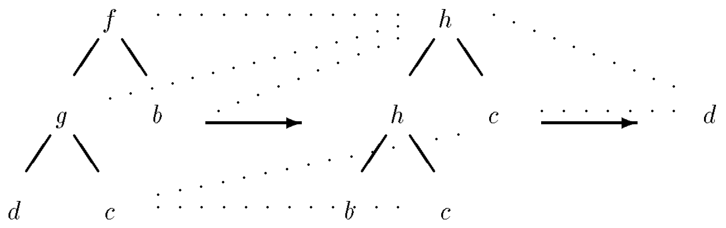
We refer to [3] and [10] for introductions in the TRS theory. Below we consider only OTRSs.

Notation We use R for OTRSs, t, s, e, o for terms, r for rewrite rules, u, v, w for redexes, and P, Q for reductions. We write $s \in t$ if s is a subterm of t. A one-step reduction in which a redex u in a term t is contracted is written as $t \xrightarrow{u} s$ or $t \rightarrow s$. We write $P : t \twoheadrightarrow s$ if P denotes a reduction of t to s comprising 0 or more steps. $\emptyset_t$, or simply $\emptyset$, denotes the empty reduction of a term t; the symbol $\emptyset$ is also used to denote the empty set. A normal form (nf for short) of t is denoted by $!t$. If the last term of P coincides with the initial term of Q, then $P + Q$ denotes the concatenation of P and Q. $|P|$ denotes the length, i.e., the number of steps, of P. WN stands for (weak) normalization (i.e., existence of a nf) and SN stands for strong normalization (i.e., termination of all reductions).

Definition 2.1 (1) Let $r : t \rightarrow s$ be a rewrite rule and v be an r-redex. Subterms of v that correspond to variables of t are the *arguments* of v and the rest is the *pattern* of v. Subterms of v rooted at the pattern are called the *pattern-subterms* of v. The *arguments*, *pattern*, and *pattern-subterms* are defined analogously in the contractum of v.

(2) Let $t' \xrightarrow{u} s'$ and let e be the contractum of u in s'. For each argument o of u there are 0 or more arguments of e. We call them (u-)*descendants* of o. Correspondingly, subterms of o have 0 or more *descendants*. By definition, the *descendant* of each pattern-subterm of u is e. *Descendants* of all redexes of t' except u are also called *residuals*. By definition, u does not have *residuals* in s. A redex of s' is said to be *created* by contracting u or to be an (u)-*new* redex if it is not a residual of a redex of t'. It is clear what is to be meant by *descendants* of subterms that are not in u. The notions of *descendant* and *residual* extend naturally to arbitrary reductions.

Example 2.1 The notion of descendant is illustrated by the picture below of a reduction in the OTRS $R = \{f(g(x, y), b) \rightarrow h(h(b, y), y), h(x, c) \rightarrow d\}$.



Notation If F is a set of redexes in t and $P : t \twoheadrightarrow s$, then F/P denotes the set of all residuals of redexes from F in s. If $F = \{u\}$, then we write u/P for $\{u\}/P$. In the following, F will also denote a complete F-development, where the residuals of redexes from F are contracted as long as possible. Similarly, if $u \in t$, then u will also denote the reduction $t \xrightarrow{u} s$.

Definition 2.2 Let $Q : t \twoheadrightarrow s$ and $t \xrightarrow{u} e$. Then the *residual* Q/u *of* Q *by* u is defined by induction on $|Q|$ as follows. If $Q = \emptyset_t$, then $Q/u = \emptyset_e$. If $Q = Q_1 + v$, then $Q/u = Q_1/u + v/(u/Q_1)$.

Definition 2.3 Let $P : t \twoheadrightarrow s$ and $Q : t \twoheadrightarrow e$. Then the *residual* P/Q *of* P *by* Q and the *residual* Q/P *of* Q *by* P are defined by induction on $|P|$ as follows.

(1) If $P = \emptyset_t$, then $P/Q = \emptyset_e$ and $Q/P = Q$.

(2) If $P = P_1 + u$, then $P/Q = P_1/Q + u/(Q/P_1)$ and $Q/P = (Q/P_1)/u$.
We write $P \sqcup Q$ for $P + Q/P$.

Definition 2.4 We call the reductions $P : t \twoheadrightarrow t'$ and $Q : s \twoheadrightarrow s'$ *strictly equivalent* (written $P \approx Q$) if $t = s$, $t' = s'$, and P-descendants and Q-descendants of any subterm of t are the same occurrences in t' and s'.

Strict equivalence is weaker then strong equivalence [4], but it is enough for our purposes.

Theorem 2.1 (Strict Church-Rosser Theorem) Let P and Q be any co-initial reductions in an OTRS R. Then $P \sqcup Q \approx Q \sqcup P$.

Proof. Similar to the ordinary case. One has only to check that reduction steps u and v in a term "strictly commute": $u + v/u \approx v + u/v$.

3 Correctness of the Essential Strategy

Definition 3.1 We call a subterm s in t *essential* (written $ES(s,t)$) if s has at least one descendant under any reduction starting from t and *inessential* (written $IE(s,t)$) otherwise.

Lemma 3.1 Let $s_0, \ldots, s_k \in t$ be such that $IE(s_i,t)$ for all $i = 0, \ldots, k$. Then there exists a reduction P starting from t such that none of the subterms $s_0, \ldots, s_k$ have P-descendants.

Proof. Let P_i be a reduction starting from t such that s_i does not have P_i-descendants (P_i exists since $IE(s_i,t)$). Then, by Theorem 2.1, one can take $P = (\ldots(P_1 \sqcup P_2) \sqcup \ldots \sqcup P_n)$.

Lemma 3.2 Let $t \xrightarrow{u} t'$ and $s \in t$. Then $IE(s,t)$ iff, for every u-descendant s' of s, $IE(s',t')$.

Proof. ($\Rightarrow$) Let $IE(s,t)$. Then there is some reduction P starting from t such that s does not have P-descendants. By Theorem 2.1, $P + u/P \approx u + P/u$. Hence, s' does not have P/u-descendants, i.e., $IE(s',t')$. ($\Leftarrow$) If all u-descendants of s are inessential in t', then, by Lemma 3.1, there is some reduction P starting from t' under which none of them have descendants. Thus s does not have $u + P$-descendants, i.e., $IE(s,t)$.

Corollary 3.1 Let $P : t \twoheadrightarrow t'$ and $s \in t$. Then $IE(s,t)$ iff all P-descendants of s are inessential in t'. In particular, if t' is a normal form, then $ES(s,t)$ iff s has a P-descendant.

Lemma 3.3 (1) Let $t \xrightarrow{u} t'$ and $e \in s \in t$, and let e' be a u-descendant of e. Then s has a u-descendant that contains e'.

(2) Let $e \in s \in t$ and $ES(e,t)$. Then $ES(s,t)$.

(3) Let s be a pattern-subterm of a redex $u \in t$. Then $ES(u,t)$ iff $ES(s,t)$.

Proof. (1) By considering all relative positions of u and e, using Definition 2.1.

(2) By (1).

(3) Let $t \xrightarrow{u} t'$. The contractum of u is the unique descendant of u, as well as of s. Hence the lemma follows from Corollary 3.1.

Definition 3.2 We call a subterm s of a term t *unabsorbed in a reduction* $P : t \twoheadrightarrow e$ if none of the descendants of s appear in redex-arguments of terms in P, and call s *absorbed in* P otherwise. We call s *unabsorbed in* t if it is unabsorbed in any reduction starting from t and *absorbed in* t otherwise.

Remark 3.1 It is easy to see that a redex $u \in t$ is unabsorbed iff u is *external* [4] in t. Note that if $s \in t$ is unabsorbed in $P : t \twoheadrightarrow e$, then s has exactly one descendant in each term of P.

Lemma 3.4 Let $P : t_0 \xrightarrow{u_0} t_1 \xrightarrow{u_1} \ldots \rightarrow t_n$. Then for any outermost redex $v \in t_n$ there is an unabsorbed in P redex $w \in t_0$ that has a descendant $s \in t_n$ containing a pattern-subterm e of v.

Proof. By induction on $|P|$. If $|P| = 0$, then the result is obvious. So suppose that $|P| > 0$.

(a) Let v be a residual of a redex v' of t_{n-1}. Further, let $v^* = v'$ if $v' \notin u_{n-1}$ and $v^* = u_{n-1}$ otherwise. Since v is outermost, v^* is outermost as well. By the induction assumption, the descendant $s' \in t_{n-1}$ of $w \in t_0$ contains a pattern-subterm of v^*. Therefore, by Lemma 3.3.(1), s', and hence also w, have a descendant that contains a pattern-subterm of v, and w is unabsorbed in P.

(b) Assume that v is a new redex. Then u_{n-1} is outermost and, by the induction assumption, a pattern-subterm e' of u_{n-1} is inside a descendant s' of w. Let e be the maximal common subterm of v and the contractum of u_{n-1}. Then e is a pattern-subterm of v. Since e' is a pattern-subterm of u_{n-1}, its descendant contains e. By Lemma 3.3.(1), s', and hence also w, have a descendant s that contains the pattern-subterm e of v. So w is unabsorbed in P as well.

Lemma 3.5 For any reduction $P : t \twoheadrightarrow s$ starting from a term t not in nf there is an outermost redex u in t that is unabsorbed in P.

Proof. If $|P| = 0$ or $|P| > 0$ and s is not in nf, then the lemma follows immediately from Lemma 3.4. Otherwise, let $P : t \twoheadrightarrow s' \xrightarrow{v} s$. By Lemma 3.4 there is an unabsorbed in $t \twoheadrightarrow s'$ redex $u \in t$ the only descendant o of which contains a pattern-subterm of v. Thus u has a descendant in s that is not in a redex-argument, since s is in nf. Hence, u is unabsorbed in P.

Lemma 3.6 Let u be absorbed in $P : t \twoheadrightarrow s$ and let $Q : t \twoheadrightarrow e$. Then u is absorbed in $Q \sqcup P$.

Proof. It suffices to prove the lemma for $|Q| = 1$, i.e., $Q = w$ for a redex w in t. The rest follows by induction on $|Q|$. Furthermore, we can assume that u is absorbed in the last step of P, i.e., there is a redex v created in the last step of P

and containing the only descendant o of u in its argument. If o does not have a w/P-descendant, then, by Theorem 2.1, u does not have $(w \sqcup P)-$descendants, so u is absorbed in $w \sqcup P$ (otherwise its descendants cannot be erased). If o has a w/P-descendant o', then, by Lemma 3.3.(1), v also has a w/P-descendant that contains o'. Since in w/P only residuals of w are contracted and v is a new redex, v has a residual v' that contains o'. But, by Theorem 2.1, o' also is a $(w \sqcup P)-$descendant of u. Hence, u is absorbed in $w \sqcup P$.

Theorem 3.1 Let t be a term not in normal form in an OTRS. Then there is an unabsorbed outermost — hence essential — redex in t.

Proof. If, for any outermost redex u_i of t, there is a reduction P_i that absorbs u_i $(i = 1, \ldots, k)$, then, by Lemma 3.6, $P_1 \sqcup \ldots \sqcup P_k$ absorbs all redexes u_i, contrary to Lemma 3.5.

Lemma 3.7 Let $t \xrightarrow{u} t'$, $IE(u, t)$, and let u' be a u-new redex. Then $IE(u', t')$.

Proof. Let e' be the contractum of u. By Corollary 3.1, $IE(e', t')$. So if $u' \in e'$, then by Lemma 3.3.(2), $IE(u', t')$, and if $e' \in u'$, then e' is a pattern-subterm of u' and, by Lemma 3.3.(3), $IE(u', t')$.

Lemma 3.8 Let $P : t_0 \xrightarrow{u_0} t_1 \xrightarrow{u_1} \ldots$ contain infinitely many essential steps and let $t_0 \xrightarrow{u} s_0$. Then s_0 also has a reduction with infinitely many essential steps.

Proof. Let $F_i = u/(u_0 + \ldots + u_{i-1})$ for $i = 0, 1, \ldots$ Using Theorem 2.1, we can construct the diagram

$$\begin{tikzcd}
t_0 \arrow[r, "u_0"] \arrow[d, "u = P_0"'] & t_1 \arrow[r, "u_1"] \arrow[d, "P_1"] & t_2 \arrow[r, dashed] \arrow[d, "P_2"] & {} \\
s_0 \arrow[r, "Q_0"'] & s_1 \arrow[r, "Q_1"'] & s_2 \arrow[r, dashed] & {}
\end{tikzcd}$$

where $P_i : t_i \twoheadrightarrow s_i$ is a complete F_i-development and $Q_i : s_i \twoheadrightarrow s_{i+1}$ is a complete u_i/P_i-development. It is easy to see that there are infinitely many k such that $ES(u_k, t_k)$ and $u_k \notin F_k$, since otherwise there would be some m such that in $t_m \xrightarrow{u_m} t_{m+1} \xrightarrow{u_{m+1}} \ldots$ all contracted essential redexes were residuals of u. But this is impossible, since by Lemmas 3.3.(2) and 3.7 inessential steps cannot duplicate or create essential redexes, and residuals of u are disjoint. By Corollary 3.1, $u_k \notin F_k$ and $ES(u_k, t_k)$ imply that u_k has at least one essential P_k-residual in s_k, i.e., Q_k contains at least one essential step. Hence $Q_0 + Q_1 + \ldots$ contains infinitely many essential steps.

Theorem 3.2 Let t be a term in an OTRS R. Then t has a normal form iff it does not possess a reduction contracting infinitely many essential redexes.

Proof. From Lemma 3.8 and Theorem 3.1.

4 A Classification of OTRSs

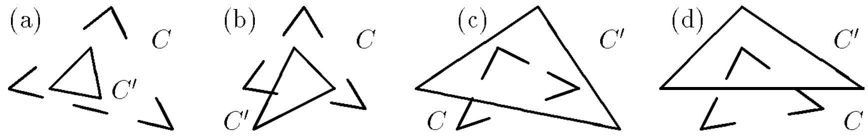
Definition 4.1 (1) Let $t \xrightarrow{u} s$; let v be a new redex in s; let o be the contractum of u, $C[\,]$ the pattern of o, and $C'[\,]$ the pattern of v.

(a) We call v a *generated* redex if C' is inside C.

(b) We call v an *inside-created* redex if the top of C' is in C, but C' may contain a symbol outside C. In the latter case, we call v an *argument-dependent* inside-created redex.

(c) We call v an *argument-dependent outside-created* redex if the top of C is in C' and is strictly below the top of C', and if there is at least one edge symbol of C that is strictly inside C' (i.e. that is not an edge symbol of C').

(d) We call v an *argument-independent outside-created* redex if the top of C is in C' and is strictly below the top of C', and if there is no edge symbol of C strictly inside C'.



(e) We call v a *collapsing outside-created* redex if C is empty and call v *non-collapsing* otherwise.

(f) We call v *non-absorbing* if all its arguments are in o. We call v *left-absorbing* (resp. *right-absorbing*) if v has an argument to the left (resp. to the right) of o.

(2) We call an OTRS R respectively *non-creating, persistent, inside-creating, outside-creating, non-absorbing, non-left-absorbing*, or *non-right-absorbing*, if no creation, only generation, only inside-creation, only outside-creation, only non-absorbing creation, no left-absorbing creation, or no right-absorbing-creation is possible in R.

Definition 4.2 We call a non-trivial subterm $C'[\,]$ of a redex-pattern $C[\,]$ a *sub-redex-pattern*. $C'[\,]$ can be represented as $C''[C_1[\,], \ldots, C_n[\,]]$ for some non-trivial context C'' (in general, in various ways). We call C'' a *pre-sub-redex-pattern*. C'' is *proper* if at least one C_i is non-trivial. *Sub-contractum-pattern* and *pre-sub-contractum-pattern* are defined analogously. We call the pre-sub-redex-pattern C'' *non-left* (resp. *non-right*) if there is a hole in the redex-pattern $C[\,]$ to the left (resp. to the right) of the sub-redex-pattern C'.

Lemma 4.1 Generation of a redex is possible in an OTRS R iff a redex-pattern is a pre-sub-contractum pattern.

Proof. From Definitions 4.1 and 4.2 (see the picture).

Lemma 4.2 (1) Inside-creation is possible in an OTRS R iff a pre-redex-pattern C^* coincides with a pre-sub-contractum-pattern C'' and, for any edge symbol of C^* that is not an edge symbol in the redex-pattern, the corresponding symbol in the contractum-pattern is an edge symbol.

(2) Moreover, the inside-creation is argument-dependent iff the pre-redex-pattern C^* is proper.

Proof. (1) ($\Rightarrow$) By Definition 4.1. ($\Leftarrow$) If $r : t \to s$ is the rule corresponding to the contractum-pattern, then, to create a redex inside contractum, it suffices to contract an appropriate r-redex. The variables of t that correspond to holes of C'' must be instances of the sub-redex-patterns, obtained from the redex-pattern by removing the pre-redex-pattern C^*.

(2) By (1) and Definition 4.2.

Remark 4.1 Consider the OTRS $R = \{f(g(x)) \to f(h(x))$. The pre-redex-pattern $f([\ ])$ is also a pre-(sub)-contractum-pattern, but there is no inside-creation in R, since f is an edge symbol of $f([\ ])$ but is not an edge symbol of the contractum-pattern $f(h[\ ])$.

Lemma 4.3 A collapsing outside-creation is possible in an OTRS R iff R contains a collapsing rule (whose right-hand side is a variable) and a rule with at least two function symbols in the redex-pattern.

Proof. ($\Rightarrow$) By Definition 4.1. ($\Leftarrow$) A redex-pattern that has at least two function symbols can be represented as $C_1[C_2[\ ]]$ with non-trivial contexts C_1 and C_2. Now, if $r : t \to x$ is a collapsing rule, then there is a collapsing outside-creation after contraction, in the context $C_1[\ ]$, of any r-redex in which x is instantiated by a term $C_2[e]$.

Lemma 4.4 A non-collapsing outside-creation is possible in an OTRS R iff a pre-proper-sub-redex-pattern C^* coincides with a pre-contractum-pattern C'' and, for any edge symbol of C'' that is not an edge symbol of the contractum-pattern, the corresponding symbol in the redex-pattern is an edge symbol. The non-collapsing outside creation is argument-dependent iff there is at least one edge symbol in C'' whose corresponding symbol in C^* is not an edge symbol in the redex-pattern.

Proof. ($\Rightarrow$) By Definition 4.1. ($\Leftarrow$) The redex-pattern can be represented as $C_1[C_2[\ ]]$ where the top of C_2 coincides with the top of C''. If the contractum-pattern corresponds to the rule $r : t \to s$, then there is an argument-dependent outside-creation after contraction of an appropriate r-redex u in the context C_1. The arguments of u that correspond to edge symbols of the contractum-pattern that are strictly inside the redex-pattern must be instances of the sub-redex-patterns rooted at these edge symbols.

Obviously, in (b)-(d) all possible cases of redex creation are considered, and (e)-(f) is a further classification of outside-creation. Note, that the collapsing outside-creation in Lemma 4.3 is left-absorbing (resp. right-absorbing) iff C_2 is non-left (resp. non-right). The non-collapsing outside creation in Lemma 4.4 is left-absorbing (resp. right-absorbing) iff C^* is non-left (resp. non-right). Hence, we have the following proposition.

Proposition 4.1 It is decidable whether an OTRS is non-creating, persistent, inside-creating, outside-creating, non-absorbing, non-left-absorbing, or non-right-absorbing.

Theorem 4.1 (1) A term t in a non-left-absorbing OTRS has a normal form iff it does not have a reduction contracting infinitely many leftmost outermost redexes.

(2) A term t in a non-absorbing OTRS has a normal form iff it does not have a reduction contracting infinitely many outermost redexes.

(3) A term t in a non-right-absorbing OTRS has a normal form iff it does not have a reduction contracting infinitely many rightmost outermost redexes.

Proof. (1) For any reduction P starting from t, one can show by induction on $|P|$ that the leftmost outermost redex u of t is unabsorbed in P. Hence u is essential and the theorem follows from Theorem 3.2. Parts (2) and (3) are proved analogously.

5 Optimal Normalization

Definition 5.1 We call a reduction P *quasi-innermost-essential* if each essential redex contracted in P is innermost among essential redexes. $|P|_I$ denotes the number of innermost essential steps in P. We call a redex $v \in t$ *non-duplicating essential* if it is essential and any essential redex u of t other than v has exactly one essential v-residual (u may have inessential residuals as well).

Lemma 5.1 Let $t \xrightarrow{u} s$, where u is an innermost essential redex or an inessential redex in t. Further, let v be any redex in t different from u and let $t \xrightarrow{v} e$.

(1) If v is essential, then it has exactly one u-residual v', which is essential.

(2) If u is innermost essential, then u/v is quasi-innermost-essential, $|u/v|_I \geq 1$, and if v is non-duplicating essential, then $|u/v|_I = 1$. If u is inessential, then each step in u/v is inessential.

(3) If v is non-duplicating essential, then v' is non-duplicating essential.

Proof. (1) By Corollary 3.1 and Lemma 3.3.(2).

(2) Again by Corollary 3.1 and Lemma 3.3.(2), since the residuals of u are disjoint.

(3) Let v be non-duplicating essential. Further, let w' be an essential redex in s, different from v'. If w' is a residual of a redex $w \in t$, then $w \neq v$ and by Corollary 3.1, w is essential. Since, by (2), u/v is quasi-innermost-essential, it follows from (1) that w' is the only essential residual of w, and the only essential v-residual of w has exactly one essential u/v-residual. Now it follows from Theorem 2.1 that w' has exactly one essential v'-residual. If w' is a new redex, then the contractum s' of u is a pattern-subterm of w' or $w' \in s'$. By Lemma 3.7, u is essential. It follows from (2) that u has exactly one essential $(v \sqcup u)$-descendant. Hence, by Theorem 2.1, s' has exactly one essential v'-descendent. So if s' is

a pattern-subterm of w', then it follows from Lemma 3.3.(3) that the essential v'-residual of w' is the one for which the essential v'-descendant of s' is a pattern-subterm. If $w' \in s'$, then each residual of w' that is inside an inessential v'-descendent of s' is inessential by Lemma 3.3.(2). Thus in this case w' has at most one essential residual. Hence, v' is non-duplicating essential.

Lemma 5.2 Let $P : t_0 \twoheadrightarrow t_n$ be a normalizing quasi-innermost-essential reduction and let $t_0 \xrightarrow{v_0} s_0$. Then P/v_0 also is a normalizing quasi-innermost-essential reduction and $|P|_I \leq |P/v_0|_I + 1$. Moreover, $|P|_I = |P/v_0|_I + 1$ iff v_0 is non-duplicating essential.

Proof. Let $P : t_0 \xrightarrow{u_0} t_1 \xrightarrow{u_1} \ldots \to t_n$. Using Theorem 2.1, we can construct the diagram

$$\begin{tikzcd}
t_0 \arrow[r, "u_0"] \arrow[d, "v_0"'] & t_1 \arrow[r, dashed, -] \arrow[d, "P_1"'] & t_{k-1} \arrow[r, "u_{k-1}"] \arrow[d, "P_{k-1}"'] & t_k \arrow[r, "u_k"] \arrow[d, "P_k"'] & t_{k+1} \arrow[r, dashed, -] \arrow[d, "P_{k+1}"'] & t_n \arrow[d, "P_n"', "\emptyset"] \\
s_0 \arrow[r, "Q_0"'] & s_1 \arrow[r, dashed, -] & s_{k-1} \arrow[r, "Q_{k-1}"'] & s_k \arrow[r, "Q_k"'] & s_{k+1} \arrow[r, dashed, -] & s_n
\end{tikzcd}$$

where $P_i : t_i \twoheadrightarrow s_i$ is a complete $v_0/(u_0 + \ldots + u_{i-1})$-development and $Q_j : s_j \twoheadrightarrow s_{j+1}$ is a complete u_j/P_j-development, $i = 1, \ldots, n; j = 0, \ldots, n-1$. If u_j is inessential, then, by Corollary 3.1, Q_j is also inessential. If v_0 is inessential, then, by Corollary 3.1, the reductions P_i are inessential. Hence, it follows from Lemma 5.1 that P/v_0 is quasi-innermost-essential as well, and $|P/v_0|_I = |P|_I$. Now let us consider the case $ES(v_0, t_0)$. Since t_n is a nf, we have $P_n = \emptyset$. Let k be the minimal number such that $P_k = \emptyset$. Then, for any $j \geq k$, $P_j = \emptyset$ and $Q_j = u_j$. By Lemma 5.1.(1), for all $l = 1, \ldots, k-1$, P_l contracts only one redex — say, v_l — which is the only essential residual of v_0. Hence, $P_k = \emptyset$ implies that $v_{k-1} = u_{k-1}$ and $Q_{k-1} = \emptyset$. By Lemma 5.1.(2), for all m such that $1 \leq m \leq k-2$ and u_m is innermost essential, Q_m is quasi-innermost-essential and $|Q_m|_I \geq 1$. It remains to show that $|Q_m|_I = 1$ iff v_0 is non-duplicating essential. Indeed, if v_0 is non-duplicating essential, then, by Lemma 5.1.(3), $v_1, \ldots, v_{k-1}$ are non-duplicating essential and, by Lemma 5.1.(2), $|Q_m|_I = 1$. If v_0 is not non-duplicating essential, then there is an essential redex $w \in t_0$ that has at least two essential v_0-residuals. By Lemma 5.1.(1), w has exactly one essential residual in each term of P until it is contracted, say in $t_{m'}$. Since the residuals of w are not contracted in $Q_0, \ldots, Q_{m-1}$, it follows from Corollary 3.1 that there are at least two essential residuals of w in $s_{m'}$. So, by Theorem 2.1, $u_{m'}$ has at least two essential residuals in $s_{m'}$. Therefore, $m' < k-1$ and, by Lemma 5.1.(2), $|Q_{m'}|_I > 1$.

Theorem 5.1 Let t be a normalizable term in an OTRS. A normalizing reduction P starting from t is optimal iff in each term it contracts a non-duplicating essential redex. In particular, innermost essential normalizing reductions are optimal.

Proof. Let $P : t \twoheadrightarrow s$ be an optimal normalizing reduction and Q be an innermost essential normalizing reduction (Q exists by Theorem 3.2). It follows from Lemma 5.2 that $|Q|_I \leq |Q/P|_I + |P| = |P|$ and $|Q|_I = |P|$ iff each step in P is non-duplicating essential. But P is optimal, so $|P| \leq |Q| = |Q|_I \leq |P|$. Hence each step in P is non-duplicating essential and Q is optimal.

6 Properties of Inessential and Independent Subterms

Notation We write $t = (t_1//s_1, \ldots, t_n//s_n)s$, if $s_1, \ldots, s_n$ are disjoint subterms in s and t is obtained from s by replacing them with $t_1, \ldots, t_n$, respectively.

Definition 6.1 We call a subterm s in t *free* (written $F(s,t)$) if s is not a proper pattern-subterm of a redex in t. We call $s \in t$ *independent* (written $I(s,t)$) if, for any reduction P starting from t, all P-descendants of s are free.

Definition 6.2 Let $t = (t_1//s_1, \ldots, t_k//s_n)s$ and let $P : s = e_0 \xrightarrow{v_0} e_1 \xrightarrow{v_1} \ldots$ Let us construct the reduction $P||(t) : t = o_0 \xrightarrow{u_0} o_1 \xrightarrow{u_1} \ldots$ as follows. If the pattern of v_0 does not overlap with $s_1, \ldots, s_n$, then u_0 is the corresponding subterm of v_0 in $t = o_0$. Otherwise, $u_0 = \emptyset$. In o_1 we can choose the redex u_1 analogously, since o_1 is obtained from e_1 by replacing the descendants of $s_1, \ldots, s_n$ with $t_1, \ldots, t_n$ respectively, and so on.

Remark 6.1 The reduction $P||(t)$ depends not only on P and t, but also on the choice of $s_1, \ldots, s_k$, but the notation does not give rise to ambiguity.

Lemma 6.1 Let $t = (t_1//s_1, \ldots, t_n//s_n)s$, let the subterms $s_1, \ldots, s_k$ be independent and essential in s, and let $s_{k+1}, \ldots, s_n$ be inessential in s. Further, let $P : s \twoheadrightarrow s'$ and $Q = P||(t) : t \twoheadrightarrow t'$. Then

(1) The term t' can be obtained from s' by replacing the descendants of $s_1, \ldots, s_k$ with $t_1, \ldots, t_k$ and replacing the descendants of $s_{k+1}, \ldots, s_n$ with some terms. The replaced occurrences of $t_1, \ldots, t_k$ in t' are the only descendants of $t_1, \ldots, t_k \in t$.

(2) If e and o are corresponding subterms in s and t, then e has P-descendants iff o has Q-descendants; each Q-descendant of o is in the subterm that corresponds to some P-descendant of e.

Proof. By induction on $|P|$.

Corollary 6.1 Let $t = (t_1//s_1, \ldots, t_n//s_n)s$, where $s_1, \ldots, s_k$ are essential and independent in s, subterms $s_{k+1}, \ldots, s_n$ are inessential in s, and $t_1, \ldots, t_k$ are independent in t. If s' and t' are any corresponding subterms in s and t, then $ES(s', s)$ iff $ES(t', t)$.

Lemma 6.2 Let $I(s,t)$, $P : t \twoheadrightarrow t'$, and let s have a P-descendant $s' \in t'$. Then there exists a reduction $Q : s \twoheadrightarrow s'$ such that any subterm $e \in s$ has a P-descendant in $s' \in t'$ iff e has a Q-descendant in s'.

Proof. Easy induction on $|P|$.

Corollary 6.2 Let $ES(e,s)$, $ES(s,t)$, and $I(s,t)$. Then $ES(e,t)$.

7 Neededness and Essentiality

Definition 7.1 A redex u in t is called *needed* [4] if in each reduction of t to normal form (if any) at least one residual of u is contracted; u is called *needed** [13] if u has at least one residual under any reduction starting from t that does not contract residuals of u.

Theorem 7.1 (1) Let t be a normalizable term in an OTRS R and let u be a redex in t. Then u is essential in t iff it is needed in t.

(2) Let t be a term in an OTRS R and let u be a redex in t. Then u is essential in t iff it is needed* in t.

Proof. (1) ($\Rightarrow$) Let $IE(u,t)$. By Theorem 3.2, there is a normalizing essential reduction $P : t \twoheadrightarrow !t$. By Corollary 3.1, the residuals of u are not contracted in P. Hence u is not needed in t. ($\Leftarrow$) From Definitions 3.1 and 7.1.

(2) ($\Rightarrow$) Let $IE(u,t)$. Further, let x be a fresh variable and let $s = (x//u)t$. By Corollary 6.1, $IE(x,s)$, i.e., there is some reduction P starting from s such that x does not have P-descendants. Hence, by Lemma 6.1.(2), s does not have descendants under $P||(t)$, which does not contract residuals of u. ($\Leftarrow$) From Definitions 3.1 and 7.1.

8 Optimal Reductions in Persistent TRSs

Lemma 8.1 Let t be a term in a PTRS. Then $F(e,t)$ iff $I(e,t)$.

Proof. By considering all relative positions of e and u.

Definition 8.1 Let r be a rule in a PTRS. We call the sequence of numbers of essential arguments of the left-hand side of r the *essentiality indicator* of r. By definition, all r-redexes have the same *essentiality indicator*.

Lemma 8.2 The i-th argument of a redex u is essential iff i belongs to the essentiality indicator of u.

Proof. From Corollary 6.1.

Lemma 8.3 Let t be a term in a PTRS R.

(1) Each outermost redex in t is essential.

(2) If $F(s,t)$, $ES(s,t)$, and $ES(e,s)$, then $ES(e,t)$.

(3) $IE(e,t)$ iff e is in an inessential argument of an essential redex of t.

Proof. (1) By Definition 4.1.

(2) By Lemma 8.1 and Corollary 6.2.

(3) ($\Rightarrow$) Since $IE(e,t)$, it follows from the persistency of R that e is inside an outermost, hence essential, redex. Let u be the innermost essential redex containing e. By Lemma 3.3.(3), e is in an argument s of u. It follows from (2) that there is no redex in s that contains e. Thus $ES(e,s)$. But $F(u,t)$ and $F(s,u)$. Hence, by (2), $IE(e,u)$ and $IE(s,u)$. ($\Leftarrow$) By Lemma 3.3.(2) and (2).

Corollary 8.1 Let t be a term in a PTRS and let essentiality indicators of redexes in t be known. Then one can find all essential redexes in t as follows: Mark outermost redexes in t; in essential arguments of marked redexes mark outermost redexes; and so on, as long as possible. The marked redexes are all the essential redexes in t.

Definition 8.2 Let R be a PTRS and $r, r' \in R$. We call r' an *essential successor* of r if an r'-redex has an essential occurrence in the right-hand side of r. We call a sequence of rules $r_0, r_1, \ldots$, where r_{i+1} is an essential successor of r_i, an *essential chain* or an *essential r_0-chain*. For any r_0-redex u, we also call an essential r_0-chain an *essential u-chain*.

Lemma 8.4 Let r' be an essential successor of r. If u is an essential r-redex in t and $t \xrightarrow{u} s$, then s contains an essential r'-redex created by u.

Proof. By Definition 8.2, the contractum e of u contains an essential r'-redex v created by u. By Corollary 3.1, $ES(e,s)$. Thus, by Lemmas 8.1 and 8.3.(2), $ES(v,s)$.

Lemma 8.5 Let t be a term in a PTRS. Then t is WN iff any essential chain of an essential redex in t is finite.

Proof. ($\Rightarrow$) Consider an infinite essential chain $r_0, r_1, \ldots$ of an essential redex in t. We show by induction on i that $(\alpha)_i$: for any reduction $P : t = t_0 \xrightarrow{u_0} t_1 \xrightarrow{u_1} \ldots$ and for each $i < |P|$ there is a k_i such that t_i contains an essential r_{k_i}-redex. $(\alpha)_0$ is obvious. If u_{k_i} is an essential r_{k_i}-redex, then, by Lemma 8.4, t_{i+1} contains an essential r_{k_i+1}-redex. Otherwise, by Corollary 3.1, any essential r_{k_i}-redex has an essential residual in t_{i+1}. Thus $(\alpha)_i$ holds for all i, and t is not normalizable. ($\Leftarrow$) For any r-redex u, let the *weight* of u be the length of a maximal essential r-chain. Since the contraction of any essential redex generates essential redexes only with smaller weights, a reduction that in each step contracts an innermost redex among essential ones with maximal weight terminates in a nf.

Lemma 8.6 Let $R = \{r_1, \ldots, r_m\}$ be a PTRS. Then the essentiality indicators of R-rules can be found using the following algorithm:

1^{st} *step:* Mark in the left-hand side of r_1 all variables that do not occur in the right-hand side of r_1.

n^{th} *step* ($n > 1$)*:* Let r_i be the rule used in the $(n-1)^{th}$ step, and let $r^n : t^n \to s^n$ be r_{i+1} if $i < m$ and r_1 if $i = m$. Mark in s^n the unmarked redex-arguments that correspond to marked variables in left-hand sides of corresponding rules; mark in t^n all unmarked variables that occur in s^n only in marked subterms.

If no occurrences are marked during the last m steps, then the algorithm stops — the unmarked variables in left-hand sides of rules are all the essential arguments. Otherwise proceed by performing the next step.

Proof. Let us show by induction on n that $(\alpha)_n$: the marked occurrences in r^n are inessential. The case $n = 1$ is clear ($r^1 = r_1$). Suppose that $(\alpha)_k$ is valid. By Lemma 8.3, all marked occurrences in s^{k+1} are inessential. Hence, by Corollary 3.1, all marked variables in t^{k+1} are inessential. Thus marked occurrences in R-rules are inessential. Now let $r_i : t' \to s'$, $IE(x, t')$, and let $P : t' \to s' \twoheadrightarrow o'$ be a minimal-length reduction such that x does not have P-descendants. Let us show by induction on $|P|$ that x is marked in t' after termination of the marking procedure. The case $|P| = 1$ is clear — x is an erased argument of t'. Let x' be a descendant of x in s' and let u^* be the minimal redex that contains x' and has a descendant in o'. Then the argument o^* of u^* that contains x' does not have $P\|u^*$-descendants. Hence, by the induction assumption, if u^* is say an r_j-redex, then o^* corresponds to a marked argument of r_j. Thus each occurrence of x in s' is in a marked subterm. Hence, x is marked in t'.

Theorem 8.1 Let t be a term in a PTRS with a finite number of rules. It is decidable whether t has a nf, and the nf can be found using the innermost essential strategy in the minimal number of steps.

Proof. By Lemmas 8.2, 8.5, and 8.6 and Theorem 5.1.

Acknowledgements

The correctness and optimality results were first obtained under the supervision of Sh. Pkhakadze and Kh. Rukhaia for some reductions in the "Notation Theory" of Sh. Pkhakadze [16]. The results were generalized for OTRSs and reported in [8] after I became acquainted with the TRS theory by J. W. Klop's introductory paper [9], pointed to me out by G. Mints. I am grateful to them, as well as to H. Barendregt and J.-J. Lévy, for constant support. I thank H. Barendregt, P.-L. Curien, H. Ganzinger, D. Kesner, J. W. Klop, G. Kucherov, and D. Rösner for organizing my talks at Nijmegen, Paris, Saarbrücken, Orsay, Amsterdam, Nancy, and Ulm. I would like to thank L. Maranget for useful discussions and many helpful comments, and Sh. Aditua, D. Doligez, G. Gonthier, I. Jacobs, B. Pierce, G. Tagviashvili, and K. Urbaitis for their help in the preparation of this paper.

References

1. Berry G., Lévy J.-J. Minimal and optimal computations of recursive programs. JACM 26, 1979, p.148-175.
2. Courcelle B. Recursive Applicative Program Schemes. In: J.van Leeuwen ed. Handbook of Theoretical Computer Science, Chapter 9, vol.B, 1990, p. 459-492.

3. Dershowitz N., Jouannaud J.-P. Rewrite Systems. In: J.van Leeuwen ed. Handbook of Theoretical Computer Science, Chapter 6, vol.B, 1990, p. 243-320.
4. Huet G., Lévy J.-J. Computations in Orthogonal Rewriting Systems. In: Computational Logic, Essays in Honor of Alan Robinson, ed. by J.-L. Lassez and G. Plotkin, MIT Press, 1991. (First appeared in 1979, as IRIA Laboria report number 359).
5. Kennaway J.R. Sequential evaluation strategy for parallel-or and related reduction systems. Annals of Pure and Applied Logic 43, 1989, p.31-56.
6. Kennaway J.R., Sleep M.R. Neededness is hypernormalizing in regular combinatory reduction systems. Preprint, School of Information Systems, University of East Anglia, Norwich, 1989.
7. Kesner D. Free sequentiality in orthogonal order-sorted rewriting systems with constructors, in Proc. 11th Int. Conf. on Automated Deduction, Saratoga Springs, NY, LNAI 607, 1992.
8. Khasidashvili Z. Minimal Normalizing Strategy for Orthogonal and Persistent Term Rewriting Systems (short version). Proceedings of I.Vekua Institute of Applied Mathematics of Tbilisi State University, vol.36, 1990. p.189-199.
9. Klop J.W. Term Rewriting Systems: a tutorial. Bulletin of the EATCS 32, 1987, p. 143-182.
10. Klop J.W. Term Rewriting Systems. In: S.Abramsky, D.Gabby, and T.Maibaum eds. Handbook of Logic in Computer Science, vol. II, Oxford University Press, 1992.
11. Klop J.W., Middeldorp A. Sequentiality in orthogonal term rewriting systems. Report CS-R8932, CWI Amsterdam, 1989.
12. Lévy J.-J. Optimal Reduction in the Lambda-Calculus. In: To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism, J. P. Seldin and J. R. Hindley editors, Academic Press, 1980.
13. Maranget L. "La stratégie paresseuse", These de l'Université' de PARIS VII, 1992.
14. O'Donnell M.J. Computing in systems described by equations. Springer LNCS 58, 1977.
15. Oyamaguchi M. Sufficient Sequentiality: A decidability Condition for call-by-need computations in term rewriting systems, Report, Mie University, 1987.
16. Pkhakadze Sh. Some problems of the Notation Theory (in Russian). Proceedings of I.Vekua Institute of Applied Mathematics, Tbilisi 1977.
17. Van Raamsdonk F. A simple proof of confluence for weakly orthogonal combinatory reduction systems. Report CS-R9234, CWI Amsterdam, 1992.
18. Thatte S. A refinement of strong sequentiality for term rewriting with constructors. Information and computation 72, 1987, p.46-65.
19. Toyama Y. Strong sequentiality of left-linear overlapping term rewriting systems. LICS, 1992.

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