数理解析研究所講究録
1106 巻 1999 年 149-160

Strategies for Rewrite Systems: Normalization and Optimality

Aart Middeldorp

Institute of Information Sciences and Electronics
University of Tsukuba, Tsukuba 305-8573, Japan
ami@is.tsukuba.ac.jp
http://www.score.is.tsukuba.ac.jp/~ami

Abstract

In this paper we present a gentle introduction to the important topic of rewrite strategies for term rewrite systems. We survey normalization results and introduce the recent framework of Durand and Middeldorp for the study of optimal rewrite strategies.

1 Introduction

Term rewriting is a general model of computation. One of the fundamental issues in term rewriting is the ability to compute normal forms. Given a rewrite system and a term, a rewrite strategy specifies which part(s) of the term to evaluate. The desirable property of rewrite strategies is *normalization*: repeated evaluation of the part(s) selected by the rewrite strategy leads to a normal form, if the term under consideration has a normal form.

Consider the term rewrite system (TRS for short) $\mathcal{R}_1$ consisting of the following rewrite rules, specifying addition and multiplication on natural numbers:

$$\begin{array}{rcl@{\qquad}rcl} 0+y & \to & y & 0\times y & \to & 0 \\ \mathsf{s}(x)+y & \to & \mathsf{s}(x+y) & \mathsf{s}(x)\times y & \to & x\times y+y \end{array}$$

Suppose we want to compute the term $(0+0\times\mathsf{s}(0))+0\times 0$. This term contains three instances of left-hand sides of rewrite rules (so-called *redexes*):

$$\overbrace{(0+\underbrace{0\times\mathsf{s}(0)}_{(2)})}^{(1)}+\overbrace{0\times 0}^{(3)}$$

Redexes (1) and (3) are not contained in larger redexes. We call them *outermost*. Redexes (2) and (3) are *innermost*; they do not contain smaller redexes. If we select the *leftmost-outermost* redex (1), we obtain the term $0 \times s(0) + 0 \times 0$ which contains two redexes. Selecting again the leftmost-outermost redex, we obtain the term $0 + 0 \times 0$. After two further contractions of the leftmost-outermost redex we arrive at the normal form 0. Instead of selecting a single redex in each step, we can also contract redexes in parallel. For instance, redexes (1) and (3) do not interfere, i.e., after contraction of redex (1), redex (3) is not affected, and vice-versa. If we contract all outermost redexes in parallel we also obtain the normal form 0: $(0 + 0 \times s(0)) + 0 \times 0 \rightarrow^* 0 \times s(0) + 0 \rightarrow 0 + 0 \rightarrow 0$.

Does it matter which redexes we select for contraction? For the above example the answer is no since $\mathcal{R}_1$ does not admit infinite rewrite sequences. Hence, no matter which redexes we select for contraction, we are guaranteed to find a normal form. In general, however, terms may have a normal form but also admit infinite rewrite sequences. Consider for example the TRS $\mathcal{R}_2$ consisting of the following rewrite rules:

$$\begin{array}{rcl@{\qquad}rcl} 0 + y & \rightarrow & y & \mathsf{head}(x : y) & \rightarrow & x \\ \mathsf{s}(x) + y & \rightarrow & \mathsf{s}(x + y) & \mathsf{tail}(x : y) & \rightarrow & y \\ \mathsf{fib} & \rightarrow & \mathsf{f}(\mathsf{s}(0), \mathsf{s}(0)) & \mathsf{f}(x, y) & \rightarrow & x : \mathsf{f}(y, x + y) \end{array}$$

This TRS computes the infinite list of Fibonacci numbers. The term $\mathsf{head}(\mathsf{tail}(\mathsf{tail}(\mathsf{fib})))$ can be reduced to its normal form $\mathsf{s}(\mathsf{s}(0))$, the third number in the list of Fibonacci numbers, e.g. by means of the rewrite strategy that always selects the leftmost-outermost redex, but repeatedly contracting an innermost redex will produce an infinite rewrite sequence.

If a term t has a normal form we can always compute a normal form of t by computing the reducts of t in a breadth-first manner until we encounter a normal form. However, in general this is a highly inefficient way to compute normal forms. In this paper we present more efficient ways to compute normal forms. The basic idea is to compute a single rewrite sequence rather than explore all rewrite sequences starting from a given term. The computation of this rewrite sequence is guided by a strategy.

Definition 1.1 A *rewrite strategy* for a TRS is a mapping $\mathcal{S}$ that assigns to every term t not in normal form a non-empty set of finite non-empty rewrite sequences starting from t. We write $t \rightarrow^{\mathcal{S}} t'$ if $t \rightarrow^+ t' \in \mathcal{S}(t)$. A rewrite strategy is called *deterministic* if $\mathcal{S}(t)$ contains a single rewrite sequence for every non-normal form t.

A strategy $\mathcal{S}$ is useless if for a term t that has a normal form a rewrite sequence computed by $\mathcal{S}$ misses the normal form.

Definition 1.2 A strategy $\mathcal{S}$ *normalizes* a term t if there are no infinite $\mathcal{S}$-rewrite sequences starting from t. We call $\mathcal{S}$ *normalizing* if it normalizes every term that has a normal form.

In the next section we present several rewrite strategies and summarize their normalization behaviour.

2 Normalizing Reduction Strategies

In the following we are mostly dealing with the class of left-linear TRSs without critical pairs, the so-called *orthogonal* TRSs. Orthogonality is a syntactic condition that ensures that redexes cannot eliminate each other in a harmful way. Orthogonal TRSs have the nice property that every term has at most one normal form. More information on orthogonal TRSs and term rewriting in general can be found in [1, 10].

Most rewrite strategies are defined by selecting the redexes which are to be contracted in each step. Below four such strategies are defined.

Definition 2.1 The *leftmost-outermost* rewrite strategy $\mathcal{S}_{\mathrm{lo}}$ always selects the leftmost of the outermost redexes, i.e., $s \rightarrow^{\mathcal{S}_{\mathrm{lo}}} t$ if t is obtained from s by contracting its leftmost-outermost redex. The *parallel-outermost* rewrite strategy $\mathcal{S}_{\mathrm{po}}$ contracts all outermost redexes in parallel. Likewise, the *leftmost-innermost* rewrite strategy $\mathcal{S}_{\mathrm{li}}$ contracts the leftmost of the innermost redexes and the *parallel-innermost* rewrite strategy $\mathcal{S}_{\mathrm{pi}}$ contracts all innermost redexes in parallel.

The above strategies are defined for arbitrary TRSs, but for non-orthogonal TRSs they need not be deterministic. The following strategy is only defined for orthogonal TRSs. Its well-definedness is a consequence of the complete developments theorem. A *complete development* of a set of redexes in a term t is a rewrite sequence starting from t in which all these redexes are contracted.

Definition 2.2 The *full substitution* rewrite strategy $\mathcal{S}_{\mathrm{fs}}$ assigns to every term t not in normal form a complete development of the set of all redexes in t.

Let us illustrate the different strategies on a small example. Consider the TRS $\mathcal{R}_1$ of Section 1 and the term $t = \mathsf{s}(0+0) \times (0+\mathsf{s}(0))$. We have

$$
\begin{aligned}
& t \rightarrow^{\mathcal{S}_{\mathrm{li}}} \mathsf{s}(0) \times (0+\mathsf{s}(0)) \rightarrow^{\mathcal{S}_{\mathrm{li}}} \mathsf{s}(0) \times \mathsf{s}(0) \rightarrow^{\mathcal{S}_{\mathrm{li}}} 0 \times \mathsf{s}(0) + \mathsf{s}(0) \rightarrow^{\mathcal{S}_{\mathrm{li}}} 0 + \mathsf{s}(0) \\
& \quad \rightarrow^{\mathcal{S}_{\mathrm{li}}} \mathsf{s}(0) \\
& t \rightarrow^{\mathcal{S}_{\mathrm{pi}}} \mathsf{s}(0) \times \mathsf{s}(0) \rightarrow^{\mathcal{S}_{\mathrm{pi}}} 0 \times \mathsf{s}(0) + \mathsf{s}(0) \rightarrow^{\mathcal{S}_{\mathrm{pi}}} 0 + \mathsf{s}(0) \rightarrow^{\mathcal{S}_{\mathrm{pi}}} \mathsf{s}(0) \\
& t \rightarrow^{\mathcal{S}_{\mathrm{lo}}} (0+0) \times (0+\mathsf{s}(0)) + (0+\mathsf{s}(0)) \rightarrow^{\mathcal{S}_{\mathrm{lo}}} 0 \times (0+\mathsf{s}(0)) + (0+\mathsf{s}(0)) \\
& \quad \rightarrow^{\mathcal{S}_{\mathrm{lo}}} 0 + (0+\mathsf{s}(0)) \rightarrow^{\mathcal{S}_{\mathrm{lo}}} 0 + \mathsf{s}(0) \rightarrow^{\mathcal{S}_{\mathrm{lo}}} \mathsf{s}(0) \\
& t \rightarrow^{\mathcal{S}_{\mathrm{po}}} (0+0) \times (0+\mathsf{s}(0)) + (0+\mathsf{s}(0)) \rightarrow^{\mathcal{S}_{\mathrm{po}}} 0 \times \mathsf{s}(0) + \mathsf{s}(0) \rightarrow^{\mathcal{S}_{\mathrm{po}}} 0 + \mathsf{s}(0) \\
& \quad \rightarrow^{\mathcal{S}_{\mathrm{po}}} \mathsf{s}(0) \\
& t \rightarrow^{\mathcal{S}_{\mathrm{fs}}} 0 \times \mathsf{s}(0) + \mathsf{s}(0) \rightarrow^{\mathcal{S}_{\mathrm{fs}}} 0 + \mathsf{s}(0) \rightarrow^{\mathcal{S}_{\mathrm{fs}}} \mathsf{s}(0)
\end{aligned}
$$

All five strategies succeed in computing the normal form $\mathsf{s}(0)$, which is not surprising as $\mathcal{R}_1$ is *terminating*. In the presence of infinite rewrite sequences, however, innermost strategies are best avoided because of the following result of O'Donnell [14].

Theorem 2.3 *A term t in an orthogonal TRS admits a rewrite sequence to normal form in which only innermost redexes are contracted if and only if t does not admit infinite rewrite sequences.* $\square$

Corollary 2.4 *An innermost strategy $\mathcal{S}$ is normalizing for an orthogonal TRS $\mathcal{R}$ if and only if $\mathcal{R}$ is terminating.* $\square$

Gramlich [6] showed that these results remain true for the larger class of locally confluent overlay systems.

The following theorem is intuitively clear: by repeatedly contracting all redexes it is impossible to miss the normal form.

Theorem 2.5 *The full substitution strategy is normalizing for orthogonal TRSs.*

O'Donnell [14] also showed that parallel-outermost is a normalizing strategy for orthogonal TRSs. As a matter of fact, he showed a stronger result: if a term in an orthogonal TRS has a normal form then it does not admit outermost-fair rewrite sequences.

Definition 2.6 An infinite rewrite sequence $t_1 \to t_2 \to t_3 \to \cdots$ is called *outermost-fair* if there do not exist a position p and an index $n \geqslant 1$ such that for all $i \geqslant n$, $t_{i|p}$ is an outermost redex in t_i which is not contracted in the step from t_i to t_{i+1}.

Let us illustrate the concept of outermost-fairness by means of two examples. The infinite rewrite sequence A: $\mathsf{f}(\mathsf{a}) \to \mathsf{g}(\mathsf{f}(\mathsf{a})) \to \mathsf{g}(\mathsf{g}(\mathsf{f}(\mathsf{a}))) \to \cdots$ with respect to the TRS $\{\mathsf{a} \to \mathsf{b}, \mathsf{f}(x) \to \mathsf{g}(\mathsf{f}(x))\}$ is outermost-fair since every term in A contains a single outermost redex which is immediately contracted. The infinite rewrite sequence B: $\mathsf{f}(\mathsf{a},\mathsf{c}) \to \mathsf{f}(\mathsf{a},\mathsf{d}) \to \mathsf{f}(\mathsf{a},\mathsf{c}) \to \mathsf{f}(\mathsf{a},\mathsf{d}) \to \cdots$ with respect to the TRS $\{\mathsf{a} \to \mathsf{b}, \mathsf{c} \to \mathsf{d}, \mathsf{f}(x,\mathsf{d}) \to \mathsf{f}(x,\mathsf{c})\}$ is also outermost-fair. Observe that redex a in B is only half of the time an outermost redex, even though it is never contracted.

Theorem 2.7 *If a term in an orthogonal TRS admits an outermost-fair rewrite sequence then it does not have a normal form.* $\square$

Since infinite rewrite sequences produced by the parallel-outermost strategy are trivially outermost-fair, we obtain the following result.

Corollary 2.8 *The parallel-outermost strategy is normalizing for orthogonal TRSs.* $\square$

Very recently van Oostrom [15] extended Theorem 2.7 to the class of weakly orthogonal higher-order rewrite systems.

The leftmost-outermost strategy is not normalizing for orthogonal TRSs. A simple counterexample is provided by the TRS $\mathcal{R} = \{\mathsf{a} \to \mathsf{b}, \mathsf{c} \to \mathsf{c}, \mathsf{f}(x,\mathsf{b}) \to \mathsf{b}\}$ and the term $\mathsf{f}(\mathsf{c},\mathsf{a})$.

The leftmost-outermost strategy selects redex c and hence will produce the infinite rewrite sequence $\mathsf{f}(\mathsf{c},\mathsf{a}) \to^{\mathcal{S}_{\mathrm{lo}}} \mathsf{f}(\mathsf{c},\mathsf{a}) \to^{\mathcal{S}_{\mathrm{lo}}} \cdots$ whereas the parallel-outermost strategy normalizes $\mathsf{f}(\mathsf{c},\mathsf{a})$ in two steps: $\mathsf{f}(\mathsf{c},\mathsf{a}) \to^{\mathcal{S}_{\mathrm{po}}} \mathsf{f}(\mathsf{c},\mathsf{b}) \to^{\mathcal{S}_{\mathrm{po}}} \mathsf{b}$. Nevertheless, there is an important subclass of the orthogonal TRSs for which $\mathcal{S}_{\mathrm{lo}}$ is normalizing.

Definition 2.9 A TRS is called *left-normal* if variables do not precede function symbols in the left-hand sides of the rewrite rules.

Note that the above $\mathcal{R}$ is not left-normal as the variable x in the left-hand side $\mathsf{f}(x,\mathsf{b})$ of the third rewrite rule precedes the constant b. The prime example of a left-normal orthogonal TRS is *combinatory logic.*

Theorem 2.10 (O'Donnell [14]) *The leftmost-outermost rewrite strategy is normalizing for left-normal orthogonal TRSs.* □

3 Optimal Rewrite Strategies

In the previous section we have seen that the parallel-outermost rewrite strategy is normalizing for orthogonal TRSs. However, it is not optimal. Consider the TRS $\mathcal{R}_1$ of Section 1 and the term $(0 \times \mathsf{s}(0)) \times (0 + \mathsf{s}(0))$. In the sequence $(\underline{0 \times \mathsf{s}(0)}) \times (\underline{0 + \mathsf{s}(0)}) \to^{\mathcal{S}_{\mathrm{po}}} \underline{0 \times \mathsf{s}(0)} \to^{\mathcal{S}_{\mathrm{po}}} 0$ the three underlined redexes are contracted. Without contracting the redex $0 + \mathsf{s}(0)$ the normal form 0 can still be reached: $(\underline{0 \times \mathsf{s}(0)}) \times (0 + \mathsf{s}(0)) \to \underline{0 \times (0 + \mathsf{s}(0))} \to 0$. So redex $0 + \mathsf{s}(0)$ is not needed to reach the normal form. An optimal rewrite strategy selects only needed redexes. The formal definition of neededness is given below.

Definition 3.1 A redex Δ in a term t is *needed* if in every rewrite sequence from t to normal form a descendant of Δ is contracted.

Let $\mathcal{R}$ be a TRS over a signature $\mathcal{F}$. We assume the existence of a constant $\bullet$ not appearing in $\mathcal{F}$ and we view $\mathcal{R}$ as a TRS over the extended signature $\mathcal{F}_\bullet = \mathcal{F} \cup \{\bullet\}$. So $\mathsf{NF}_{\mathcal{R}}$, the set of ground normal forms of $\mathcal{R}$, consists of all terms in $\mathcal{T}(\mathcal{F}_\bullet)$ that are in normal form. Let $\mathcal{R}_\bullet$ be the TRS $\mathcal{R} \cup \{\bullet \to \bullet\}$. Note that $\mathsf{NF}_{\mathcal{R}_\bullet}$ coincides with $\mathsf{NF}_{\mathcal{R}} \cap \mathcal{T}(\mathcal{F})$, the set of ground normal forms that do not contain the symbol $\bullet$. The following easy lemma gives an alternative definition of neededness, not depending on the notion of descendant.

Lemma 3.2 *Let $\mathcal{R}$ be an orthogonal TRS over a signature $\mathcal{F}$. Redex Δ in term $C[\Delta] \in \mathcal{T}(\mathcal{F})$ is needed if and only if there is no term $t \in \mathsf{NF}_{\mathcal{R}_\bullet}$ such that $C[\bullet] \to^*_{\mathcal{R}} t$.* □

Note that the above lemma implies that needed redexes are *uniform*, which means that only the position of a redex in a term is important for determining neededness. So if redex Δ in term $C[\Delta]$ is needed then so is redex Δ' in $C[\Delta']$.

The following theorem of Huet and Lévy [7] forms the basis of all results on optimal normalizing rewrite strategies for orthogonal TRSs.

Theorem 3.3 *Let $\mathcal{R}$ be an orthogonal TRS.*

1. *Every reducible term contains a needed redex.*
2. *Repeated contraction of needed redexes results in a normal form, whenever the term under consideration has a normal form.*

□

Unfortunately, needed redexes are not computable in general. Hence, in order to obtain a *computable* optimal rewrite strategy, we are left to find (1) decidable approximations of neededness and (2) decidable properties of TRSs which ensure that every reducible term has a needed redex identified by (1). Starting with the seminal work of Huet and Lévy [7] on strong sequentiality, these issues have been extensively investigated in the literature [2, 8, 9, 11, 12, 16, 18]. In all these works Huet and Lévy's notions of index, ω-reduction, and sequentiality figure prominently. Recently, Durand and Middeldorp [5] presented an approach to decidable call by need computations to normal form in which issues (1) and (2) above are addressed directly. Besides facilitating understanding, much larger classes of TRSs are shown to admit a computable optimal normalizing strategy. In the remainder of this paper we recall the framework of [5]. Due to lack of space, we omit most proofs.

4 Approximations

In the remaining part of the paper we are dealing with finite TRSs only. Moreover, we consider rewriting on ground terms only. So we assume that the set of ground terms is non-empty. This requirement entails no loss of generality. It is undecidable whether a redex in a term is needed with respect to a given (orthogonal) TRS. In this section we present decidable sufficient conditions for a redex to be needed.

Because the relation $\rightarrow_{\mathcal{R}}^*$ is in general not computable, neededness is undecidable. In order to obtain decidable sufficient conditions, the idea is now to approximate $\rightarrow_{\mathcal{R}}^*$ by $\rightarrow_{\mathcal{S}}^*$ for some suitable TRS $\mathcal{S}$ such that it is decidable whether a term admits an $\mathcal{S}$-rewrite sequence to a term in $\mathsf{NF}_{\mathcal{R}_\bullet}$.

Definition 4.1 Let $\mathcal{R}$ and $\mathcal{S}$ be TRSs over the same signature. We say that $\mathcal{S}$ *approximates* $\mathcal{R}$ if $\rightarrow_{\mathcal{R}}^* \subseteq \rightarrow_{\mathcal{S}}^*$ and $\mathsf{NF}_{\mathcal{R}} = \mathsf{NF}_{\mathcal{S}}$. We say that redex Δ in $C[\Delta] \in \mathcal{T}(\mathcal{F})$ is $\mathcal{R}$*-needed* if there is no term $t \in \mathsf{NF}_{\mathcal{R}_\bullet}$ such that $C[\bullet] \rightarrow_{\mathcal{R}}^* t$. The set of all such $C[\bullet]$ is denoted by $\mathcal{R}$-NEEDED.

So redex Δ in $C[\Delta] \in \mathcal{T}(\mathcal{F})$ is $\mathcal{R}$-needed if and only if $C[\bullet] \in \mathcal{R}$-$\mathsf{NEEDED}$. The following lemma is immediate from the definitions.

Lemma 4.2 *Let $\mathcal{R}$ be a TRS and let $\mathcal{S}$ be an approximation of $\mathcal{R}$. Every $\mathcal{S}$-needed redex is $\mathcal{R}$-needed.* □

Definition 4.3 An *approximation map* is a map α from TRSs to TRSs with the property that $\alpha(\mathcal{R})$ approximates $\mathcal{R}$, for every TRS $\mathcal{R}$. In the following we write $\mathcal{R}_\alpha$ instead of $\alpha(\mathcal{R})$. We define a partial order $\leqslant$ on approximation maps as follows: $\alpha \leqslant \beta$ if and only if $\mathcal{R}_\beta$ approximates $\mathcal{R}_\alpha$, for every TRS $\mathcal{R}$.

In the literature several approximations have been studied. These approximations have the same left-hand sides as the original TRS $\mathcal{R}$, hence the second requirement in the definition of approximation is trivially satisfied, but differ in the way they treat the right-hand sides of the rewrite rules of $\mathcal{R}$.

Definition 4.4 Let $\mathcal{R}$ be a TRS. The TRS $\mathcal{R}_{\mathrm{s}}$ is obtained from $\mathcal{R}$ by replacing the right-hand side of every rewrite rule by a variable that does not occur in the corresponding left-hand side.

The idea of approximating a TRS by ignoring the right-hand sides of its rewrite rules is due to Huet and Lévy [7]. Our $\mathcal{R}_{\mathrm{s}}$-needed redexes coincide with their *strongly needed* redexes. A better approximation is obtained by preserving the *n*on-*v*ariable parts of the right-hand sides of the rewrite rules.

Definition 4.5 Let $\mathcal{R}$ be a TRS. The TRS $\mathcal{R}_{\mathrm{nv}}$ is obtained from $\mathcal{R}$ by replacing the variables in the right-hand sides of the rewrite rules by pairwise distinct variables that do not occur in the corresponding left-hand sides.

The idea of approximating a TRS by ignoring the variables in the right-hand sides of the rewrite rules is due to Oyamaguchi [16]. Note that $\mathcal{R}_{\mathrm{nv}} = \mathcal{R}$ whenever $\mathcal{R}$ is right-ground. Hence for every orthogonal right-ground TRS $\mathcal{R}$, a redex is needed if and only if it is $\mathcal{R}_{\mathrm{nv}}$-needed.

Definition 4.6 A TRS $\mathcal{R}$ is called *growing* if for every rewrite rule $l \to r \in \mathcal{R}$ the common variables in l and r occur at depth 1 in l. (The depth of a subterm occurrence is the number of function symbols along the path to the root of the term.) We define $\mathcal{R}_{\mathrm{g}}$ as the growing TRS that is obtained from $\mathcal{R}$ by renaming those variables in the right-hand sides of the rewrite rules that violate the restrictions imposed on growing TRSs.

Growing TRSs, introduced by Jacquemard [8], are a proper extension of the shallow TRSs considered by Comon [2]. The growing approximation defined above stems from Nagaya and Toyama [13]. It extends the growing approximation in [8] in that the right-linearity requirement is dropped.

Let us illustrate the difference between the three approximations on a small example. Consider the TRS $\mathcal{R}$ consisting of the two rules

$$\begin{aligned} \mathsf{f}(\mathsf{a},\mathsf{b},x) &\to \mathsf{a} \\ \mathsf{f}(\mathsf{b},x,\mathsf{a}) &\to x \end{aligned}$$

and the term $t = \mathsf{f}(\Delta_1, \Delta_2, \Delta_3)$ with redexes $\Delta_1 = \mathsf{f}(\mathsf{a},\mathsf{b},\mathsf{a})$ and $\Delta_2 = \Delta_3 = \mathsf{f}(\mathsf{b},\mathsf{a},\mathsf{a})$. One easily verifies that all redexes are needed. In particular, as Δ_1 and Δ_2 rewrite only to a, Δ_3 cannot be erased by the first rule. Since $\mathcal{R}$ is orthogonal and $\mathcal{R}_{\mathrm{g}} = \mathcal{R}$, all redexes are $\mathcal{R}_{\mathrm{g}}$-needed. So using the growing approximation we are able to identify all needed redexes in t. The TRS $\mathcal{R}_{\mathrm{nv}}$ consists of the rules

$$\begin{aligned} \mathsf{f}(\mathsf{a},\mathsf{b},x) &\to \mathsf{a} \\ \mathsf{f}(\mathsf{b},x,\mathsf{a}) &\to y \end{aligned}$$

We have $\Delta_2 \to_{\mathcal{R}_{\mathrm{nv}}} \mathsf{b}$ and thus $\mathsf{f}(\Delta_1, \Delta_2, \bullet) \to^+_{\mathcal{R}_{\mathrm{nv}}} \mathsf{f}(\mathsf{a},\mathsf{b},\bullet) \to_{\mathcal{R}_{\mathrm{nv}}} \mathsf{a}$, showing that Δ_3 is not $\mathcal{R}_{\mathrm{nv}}$-needed. Redexes Δ_1 and Δ_2 are $\mathcal{R}_{\mathrm{nv}}$-needed. Finally, consider the TRS $\mathcal{R}_{\mathrm{s}}$ which consists of the rules

$$\begin{aligned} \mathsf{f}(\mathsf{a},\mathsf{b},x) &\to y \\ \mathsf{f}(\mathsf{b},x,\mathsf{a}) &\to y \end{aligned}$$

In $\mathcal{R}_{\mathrm{s}}$ a redex rewrites to every term. Hence $\mathsf{f}(\Delta_1, \bullet, \Delta_3) \to^+_{\mathcal{R}_{\mathrm{s}}} \mathsf{f}(\mathsf{b},\bullet,\mathsf{a}) \to_{\mathcal{R}_{\mathrm{s}}} \mathsf{a}$, showing that Δ_2 is not $\mathcal{R}_{\mathrm{s}}$-needed. Redex Δ_3 is also not $\mathcal{R}_{\mathrm{s}}$-needed (since it is not $\mathcal{R}_{\mathrm{nv}}$-needed), but Δ_1 is $\mathcal{R}_{\mathrm{s}}$-needed.

The above example confirms the intuition that with a better approximation, i.e., an approximation which is closer to the original rewrite system, more needed redexes can be identified.

A set of terms is *regular* if it is accepted by a finite tree automaton. We refer the reader to [3] for a comprehensive survey of tree automata techniques.

Theorem 4.7 *The set $\mathcal{R}_\alpha$-NEEDED is a regular tree language for left-linear $\mathcal{R}$ and $\alpha \in \{\mathrm{s}, \mathrm{nv}, \mathrm{g}\}$.* □

For $\alpha \in \{\mathrm{s}, \mathrm{nv}\}$ an easy proof using ground tree transducers is given in [5]. Nagaya and Toyama [13] obtained the regularity of $\mathcal{R}_{\mathrm{g}}$-NEEDED by modifying the construction given in Jacquemard [8] for right-linear $\mathcal{R}_{\mathrm{g}}$.

Since membership is decidable for regular tree languages, we obtain the following result.

Corollary 4.8 *Let $\mathcal{R}$ be a left-linear TRS and $\alpha \in \{\mathrm{s}, \mathrm{nv}, \mathrm{g}\}$. It is decidable whether a redex in a term is $\mathcal{R}_\alpha$-needed.* □

5 Decidable Call by Need Computations

A TRS $\mathcal{R}$ admits a computable call by need strategy if it has an approximation $\mathcal{S}$ such that (1) $\mathcal{S}$-needed redexes are computable and (2) every term not in normal form has an $\mathcal{S}$-needed redex. In the previous section we addressed the first issue. In this section we deal with the second issue.

Definition 5.1 Let α be an approximation map. The class of TRSs $\mathcal{R}$ such that every reducible ground term has an $\mathcal{R}_\alpha$-needed redex is denoted by CBN_α.

The next lemma is an easy consequence of Lemma 4.2.

Lemma 5.2 *Let α and β be approximation maps. If $\alpha \leqslant \beta$ then $\mathsf{CBN}_\beta \subseteq \mathsf{CBN}_\alpha$.* □

Lemma 5.3 *Let $\mathcal{R}$ be an orthogonal TRS. If $\mathcal{R}$ is right-ground then $\mathcal{R} \in \mathsf{CBN}_{\mathrm{nv}}$. If $\mathcal{R}$ is growing then $\mathcal{R} \in \mathsf{CBN}_{\mathrm{g}}$.* □

The proof of the following theorem relies on standard properties of regular tree languages and ground tree transducers.

Theorem 5.4 (Durand and Middeldorp [5]) *Let $\mathcal{R}$ be a left-linear TRS and α an approximation map such that $\mathcal{R}_\alpha$-NEEDED is regular. It is decidable whether $\mathcal{R} \in \mathsf{CBN}_\alpha$.* □

Corollary 5.5 *Let $\mathcal{R}$ be a left-linear TRS and $\alpha \in \{\mathrm{s}, \mathrm{nv}, \mathrm{sh}, \mathrm{g}\}$. It is decidable whether $\mathcal{R} \in \mathsf{CBN}_\alpha$.* □

It should not come as a surprise that a better approximation covers a larger class of TRSs. This is expressed formally in the next lemma.

Lemma 5.6 *We have $\mathsf{CBN}_{\mathrm{s}} \subset \mathsf{CBN}_{\mathrm{nv}} \subset \mathsf{CBN}_{\mathrm{g}}$, even when these classes are restricted to orthogonal TRSs.*

Proof. According to Lemma 5.2 $\mathsf{CBN}_{\mathrm{s}} \subseteq \mathsf{CBN}_{\mathrm{nv}} \subseteq \mathsf{CBN}_{\mathrm{g}}$. Consider the orthogonal TRSs

$$\begin{array}{rcl} & \mathcal{R}_1 & \\ \mathsf{f}(\mathsf{a},\mathsf{b},x) & \to & \mathsf{a} \\ \mathsf{f}(\mathsf{b},x,\mathsf{a}) & \to & \mathsf{b} \\ \mathsf{f}(x,\mathsf{a},\mathsf{b}) & \to & \mathsf{c} \end{array} \qquad \begin{array}{rcl} & \mathcal{R}_2 & \\ \mathsf{f}(\mathsf{a},\mathsf{b},x) & \to & \mathsf{a} \\ \mathsf{f}(\mathsf{b},x,\mathsf{a}) & \to & \mathsf{b} \\ \mathsf{f}(x,\mathsf{a},\mathsf{b}) & \to & x \end{array}$$

We have $\mathcal{R}_1 \in \mathsf{CBN}_{\mathrm{nv}}$ and $\mathcal{R}_2 \in \mathsf{CBN}_{\mathrm{g}}$ by Lemma 5.3. So it remains to show that $\mathcal{R}_1 \notin \mathsf{CBN}_{\mathrm{s}}$ and $\mathcal{R}_2 \notin \mathsf{CBN}_{\mathrm{nv}}$. We have

$$\begin{array}{rcl} & (\mathcal{R}_1)_{\mathrm{s}} & \\ \mathsf{f}(\mathsf{a},\mathsf{b},x) & \to & y \\ \mathsf{f}(\mathsf{b},x,\mathsf{a}) & \to & y \\ \mathsf{f}(x,\mathsf{a},\mathsf{b}) & \to & y \end{array} \qquad \begin{array}{rcl} & (\mathcal{R}_2)_{\mathrm{nv}} & \\ \mathsf{f}(\mathsf{a},\mathsf{b},x) & \to & \mathsf{a} \\ \mathsf{f}(\mathsf{b},x,\mathsf{a}) & \to & \mathsf{b} \\ \mathsf{f}(x,\mathsf{a},\mathsf{b}) & \to & y \end{array}$$

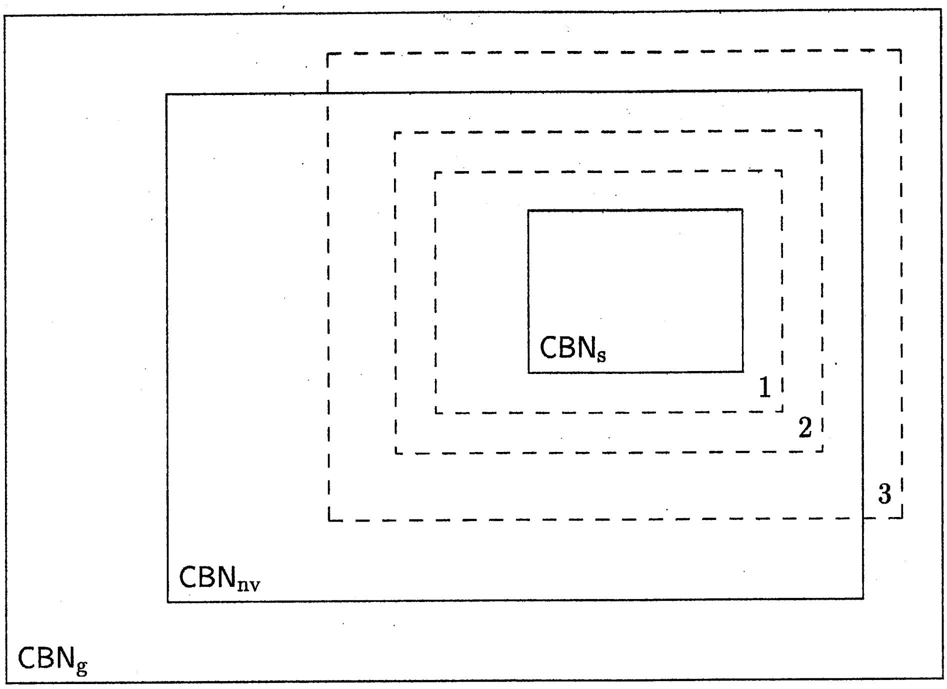

Figure 1: Comparison.

Let Δ be the redex $\mathsf{f(a,a,b)}$. The following rewrite sequences in $(\mathcal{R}_1)_s$ show that none of the redexes in $\mathsf{f}(\Delta,\Delta,\Delta)$ is $(\mathcal{R}_1)_s$-needed:

$$\begin{array}{lllllll} \mathsf{f}(\bullet,\Delta,\Delta) & \to & \mathsf{f}(\bullet,\mathsf{a},\Delta) & \to & \mathsf{f}(\bullet,\mathsf{a},\mathsf{b}) & \to & \mathsf{a} \\ \mathsf{f}(\Delta,\bullet,\Delta) & \to & \mathsf{f}(\mathsf{b},\bullet,\Delta) & \to & \mathsf{f}(\mathsf{b},\bullet,\mathsf{a}) & \to & \mathsf{a} \\ \mathsf{f}(\Delta,\Delta,\bullet) & \to & \mathsf{f}(\mathsf{a},\Delta,\bullet) & \to & \mathsf{f}(\mathsf{a},\mathsf{b},\bullet) & \to & \mathsf{a} \end{array}$$

Hence $\mathcal{R}_1 \notin \mathsf{CBN}_s$. In $(\mathcal{R}_2)_{nv}$ we have $\Delta \to t$ for every term t and thus

$$\begin{array}{lllllll} \mathsf{f}(\bullet,\Delta,\Delta) & \to & \mathsf{f}(\bullet,\mathsf{a},\Delta) & \to & \mathsf{f}(\bullet,\mathsf{a},\mathsf{b}) & \to & \mathsf{a} \\ \mathsf{f}(\Delta,\bullet,\Delta) & \to & \mathsf{f}(\mathsf{b},\bullet,\Delta) & \to & \mathsf{f}(\mathsf{b},\bullet,\mathsf{a}) & \to & \mathsf{b} \\ \mathsf{f}(\Delta,\Delta,\bullet) & \to & \mathsf{f}(\mathsf{a},\Delta,\bullet) & \to & \mathsf{f}(\mathsf{a},\mathsf{b},\bullet) & \to & \mathsf{a} \end{array}$$

Consequently, $\mathcal{R}_2 \notin \mathsf{CBN}_{nv}$. □

It can be shown that CBN_s coincides with the class of strongly sequential TRSs introduced by Huet and Lévy [7]. However, CBN_{nv} properly includes the class of NV-sequential TRSs introduced by Oyamaguchi [16] as well as the extension to NVNF-sequential TRSs considered by Nagaya *et al.* [12]. For instance, the TRS $\mathcal{R}_1$ defined in the above proof is neither NV-sequential nor NVNF-sequential (because the term $\mathsf{f}(\Omega,\Omega,\Omega)$ does not have an index).

Figure 1 shows the relationship between several classes of TRSs that admit decidable call by need computations to normal form. Areas (1), (2), and (3) consist of all NV, NVNF, and growing (Jacquemard [8]) sequential TRSs, respectively.

It is not difficult to show that the leftmost-outermost redex is always $\mathcal{R}_s$-needed for left-normal TRSs $\mathcal{R}$. Hence CBN_s includes the class of left-normal orthogonal TRSs and Theorem 2.10 is a special case of Theorem 3.3.

6 Conclusion

In this paper we addressed normalization and optimality results for rewrite strategies. We presented the framework of Durand and Middeldorp for the study of optimal rewrite strategies. Let us conclude with some remarks about complexity. Not much is known about the complexity of the problem of deciding membership in one of the classes that guarantees a computable optimal strategy. Comon [2] showed that strong sequentiality (i.e., membership in CBN_s) of a left-linear TRS can be decided in exponential time. Moreover, for left-linear TRS satisfying the additional syntactic condition that whenever two proper subterms of left-hand sides are unifiable one of them matches the other, strong sequentiality can be decided in polynomial time. The class of forward-branching systems (Strandh [17]), a proper subclass of the class of orthogonal strongly sequential systems, coincides with the class of transitive systems (Toyama *et al.* [19]) and can be decided in quadratic time (Durand [4]).

References

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

[2] H. Comon. Sequentiality, second-order monadic logic and tree automata. In *Proc. 10th LICS*, pages 508–517, 1995.

[3] H. Comon, M. Dauchet, R. Gilleron, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications, 1998. Draft, available from `http://www.grappa.univ-lille3.fr/tata/`.

[4] I. Durand. Bounded, strongly sequential and forward-branching term rewriting systems. *Journal of Symbolic Computation*, 18:319–352, 1994.

[5] I. Durand and A. Middeldorp. Decidable decidable call by need computations in term rewriting (extended abstract). In *Proc. 14th CADE*, volume 1249 of *LNAI*, pages 4–18, 1997.

[6] B. Gramlich. Abstract relations between restricted termination and confluence properties of rewrite systems. *Fundamenta Informaticae*, 24:3–23, 1995.

[7] G. Huet and J.-J. Lévy. Computations in orthogonal rewriting systems, I and II. In *Computational Logic, Essays in Honor of Alan Robinson*, pages 396–443. The MIT Press, 1991. Original version: Report 359, Inria, 1979.

[8] F. Jacquemard. Decidable approximations of term rewriting systems. In *Proc. 7th RTA*, volume 1103 of *LNCS*, pages 362–376, 1996.

[9] J.-P. Jouannaud and W. Sadfi. Strong sequentiality of left-linear overlapping rewrite systems. In *Proc. 4th CTRS*, volume 968 of *LNCS*, pages 235–246, 1995.

[10] J.W. Klop. Term rewriting systems. In *Handbook of Logic in Computer Science, Vol. 2*, pages 1–116. Oxford University Press, 1992.

[11] J.W. Klop and A. Middeldorp. Sequentiality in orthogonal term rewriting systems. *Journal of Symbolic Computation*, 12:161–195, 1991.

[12] T. Nagaya, M. Sakai, and Y. Toyama. NVNF-sequentiality of left-linear term rewriting systems. In *Proc. RIMS Workshop on Theory and Applications of Rewriting Systems*, pages 109–117, 1995.

[13] T. Nagaya and Y. Toyama. Decidability for left-linear growing term rewriting systems. In *Proc. 10th RTA*, LNCS, 1999. To appear.

[14] M.J. O'Donnell. *Computing in Systems Described by Equations*, volume 58 of *LNCS*. 1977.

[15] V. van Oostrom. Normalisation in weakly orthogonal rewriting. In *Proc. 10th RTA*, LNCS, 1999. To appear.

[16] M. Oyamaguchi. NV-sequentiality: A decidable condition for call-by-need computations in term rewriting systems. *SIAM Journal on Computation*, 22:114–135, 1993.

[17] R. Strandh. Classes of equational programs that compile into efficient machine code. In *Proc. 3rd RTA*, volume 355 of *LNCS*, pages 449–461, 1989.

[18] Y. Toyama. Strong sequentiality of left-linear overlapping term rewriting systems. In *Proc. 7th LICS*, pages 274–284, 1992.

[19] Y. Toyama, S. Smetsers, M. van Eekelen, and R. Plasmeijer. The functional strategy and transitive term rewriting systems. In *Term Graph Rewriting: Theory and Practice*, pages 61–75. Wiley, 1993.