Context-Sensitive Rewriting Strategies

Salvador Lucas

DSIC, Universidad Politécnica de Valencia, Camino de Vera s/n, E-46022 Valencia, Spain
E-mail: slucas@dsic.upv.es

Received April 4, 2000; revised May 23, 2001

Context-sensitive rewriting is a simple rewriting restriction which is formalized by imposing fixed restrictions on replacements. Such a restriction is given on a purely syntactic basis: it is given on the arguments of symbols of the signature and inductively extended to arbitrary positions of terms built from those symbols. The termination behavior is not only preserved but usually improved and several methods have been developed to formally prove it. In this paper, we investigate the definition, properties, and use of context-sensitive rewriting strategies, i.e., particular, fixed sequences of context-sensitive rewriting steps. We study how to define them in order to obtain efficient computations and to ensure that context-sensitive computations terminate whenever possible. We give conditions enabling the use of these strategies for root-normalization, normalization, and infinitary normalization. We show that this theory is suitable for formalizing the definition and analysis of real computational strategies which are used in programming languages such as OBJ or ELAN.

Key Words: infinitary normalization; normalization; replacement restrictions; root normalization; sequentiality; strategies; term rewriting.

1. INTRODUCTION

Context-sensitive rewriting (CSR) [Luc98a] is a rewriting restriction which can be associated to every term rewriting system (TRS). Given a signature \( \mathcal{F} \), a mapping \( \mu : \mathcal{F} \to \mathcal{P}(\mathbb{N}) \), called the replacement map, discriminates some argument positions \( \mu(f) \subseteq \{1, \ldots, k\} \) for each \( k \)-ary symbol \( f \). Given a function call \( f(t_1, \ldots, t_k) \), the replacements are allowed on arguments \( t_i \) such that \( i \in \mu(f) \) and are forbidden for the other argument positions. These restrictions are raised to arbitrary positions of terms in the obvious way.

**Example 1.** Consider the TRS \( \mathcal{R} \):

\[
\begin{align*}
\text{sel}(0, x:y) & \to x \\
\text{first}(0, x) & \to [] \\
\text{sel}(s(x), y:z) & \to \text{sel}(x,z) \\
\text{first}(s(x), y:z) & \to y: \text{first}(x,z) \\
\text{from}(x) & \to x: \text{from}(s(x))
\end{align*}
\]

Together with the replacement map

\[
\mu(s) = \mu(\cdot) = \mu(\text{from}) = \{1\} \quad \text{and} \quad \mu(\text{sel}) = \mu(\text{first}) = \{1, 2\}.
\]

The following derivation is allowed with CSR under \( \mu \) (we underline the redex which is contracted in each \( \mu \)-rewriting step):

\[
\begin{align*}
\text{sel}(s(0), \text{from}(s(0))) & \to \text{sel}(s(0), s(0): \text{from}(s(s(0)))) \\
& \to \text{sel}(0, \text{from}(s(0))) \\
& \to \text{sel}(0, s(s(0)): \text{from}(s(s(0)))) \\
& \to s(s(0))
\end{align*}
\]

However, the infinite (meaningless) derivation

---


294
Context-sensitive rewrites under a replacement map $\mu$ obtain (at most) $\mu$-normal forms, i.e., terms which cannot be further $\mu$-rewritten. In general, the $\mu$-normal forms of a TRS $R$ strictly include its normal forms (e.g., the subterm $s(0) : \text{from}(s(s(0)))$ which appears in Example 1 is a $\mu$-normal form which is not a normal form).

Remark 1. A rewriting strategy (roughly, a rule for appropriately choosing rewriting steps to be issued in a computation) is a restriction (i.e., a subset) of the rewriting relation. However, an important feature of strategies is that they remain “active” as long as possible, i.e., the normal forms of a strategy are normal forms. Thus, with regard to normalization, they can still achieve full computational power. In contrast, the normal forms of CSR are not (in general) normal forms. In this sense, CSR could better be thought of as a computational restriction of term rewriting.

Sufficient conditions to ensure that CSR is still able to compute root-stable terms (also called head-normal forms) and values have been established in [Luc98a]. In fact, given a TRS $R$ we are able to automatically provide replacement maps supporting such computations. In this setting, the canonical replacement map (denoted by $\mu^\text{can}_R$) is specially important, as it specifies the most restrictive replacement map which can be (automatically) associated to a TRS $R$ in order to achieve completeness of context-sensitive computations (see [Luc98a] for more details). Roughly speaking, the canonical replacement map is the most restrictive replacement map which ensures that the (positions of) nonvariable subterms of the left-hand sides of the rules of $R$ are replacing.

For instance, the replacement map $\mu$ in Example 1 is, in fact, less restrictive than the canonical replacement map $\mu^\text{can}_R$ of TRS $R$ in the example (see Example 12 below). With the replacement map $\mu$ in Example 1, we are able to ensure that every term $t$ having a value $s^n(0)$ for some $n \geq 0$ can be evaluated using CSR (see [Luc98a] for a formal justification of this claim).

1.1. Related Work

The search for complete implementations of eager languages such as Lisp [McC60, McC78] originated the first proposals for the use of syntactic replacement restrictions in programming [FW76, HM76]. Since list processing is prominent in the design of Lisp, the authors of these works studied implementations where the list constructor operator $\text{cons}$ (":" in this paper) did not evaluate its arguments during certain stages of the computation (lazy cons). As a motivating example, Friedman and Wise propose the following Lisp definition for computing (by evaluating (terms 1)) an infinite sequence of fractions

$$\frac{1}{1}, \frac{1}{4}, \frac{1}{9}, \ldots, \frac{1}{n^2}, \ldots$$

whose partial sums converge to $\pi^2/6$ (see [FW76, p. 265]):

$$(\text{terms } n) \equiv (\text{cons} (\text{reciprocal} (\text{square } n)) (\text{terms} (\text{add1 } n))).$$

Their idea is to “violate the data type of Lisp 1.0” in such a way that “the evaluation (of, e.g., (terms 1)) does not immediately diverge; it results in a node referencing two suspensions” [FW76].

---

2 See, e.g., [OV02] for a very recent survey on the topic where this requirement is part of the definition of strategy (Definition 8.1.1); similar requirements can be found in [AM96, Klo92, Mid97]. In contrast, [BEGK 87] admits strategies which are not forced to reduce terms containing redexes.

3 Here, a value is a term which contains no defined symbol, i.e., symbols occurring at the outermost position of the left-hand sides of any rule of the TRS.
Example 2. With the TRS:

\[
\begin{align*}
\text{sqr}(0) & \rightarrow 0 \\
\text{sqr}(s(x)) & \rightarrow s(\text{sqr}(x) + \text{dbl}(x)) \\
\text{dbl}(0) & \rightarrow 0 \\
\text{dbl}(s(x)) & \rightarrow s(\text{dbl}(x)) \\
\text{terms}(n) & \rightarrow \text{recip}(\text{sqr}(n)) : \text{terms}(s(n))
\end{align*}
\]

together with \(\mu(\cdot) = \emptyset\) (or even \(\mu(\cdot) = \{1\}\)) and \(\mu(f) = \{1, \ldots, k\}\) for any other \(k\)-ary symbol \(f\), we are able to obtain the desired restriction.

Moreover, by adding the two rules for function \(\text{first}\) of Example 1:

\[
\begin{align*}
\text{first}(0, x) & \rightarrow \emptyset \\
\text{first}(s(x), y : z) & \rightarrow y : \text{first}(x, z)
\end{align*}
\]

we can obtain (as \(\text{first}(n, \text{terms}(1))\)) the first \(n\) terms of the series that approximates \(\pi^2/6\), thus obtaining an arbitrary precision for the approximation (see Section 9 and Example 38 below). Here, as usual, \(n\) abbreviates \(s^n(0)\). In both cases, we can formally prove the termination of CSR.

Example 3. Consider the previous rules as a TRS \(\mathcal{R}\) and the replacement map \(\mu\) (with \(\mu(\cdot) = \{1\}\)). Then, \(\mathcal{R}_L^\mu\):

\[
\begin{align*}
\text{sqr}(0) & \rightarrow 0 \\
\text{sqr}(s(x)) & \rightarrow s(\text{sqr}(x) + \text{dbl}(x)) \\
\text{dbl}(0) & \rightarrow 0 \\
\text{dbl}(s(x)) & \rightarrow s(\text{dbl}(x)) \\
\text{terms}(n) & \rightarrow : (\text{recip}(\text{sqr}(n)))
\end{align*}
\]

is obtained by removing the non-\(\mu\)-replacing arguments of \(\text{terms}\) that integrate the rules of \(\mathcal{R}\) (and by appropriately decreasing the arities of symbols). Termination of \(\mathcal{R}_L^\mu\) ensures termination of CSR under \(\mu\) for \(\mathcal{R}\) (see [Luc96]). Here, \(\mathcal{R}_L^\mu\) is terminating: use a recursive path ordering \(\text{rpo}, [\text{Der87, Zan02}]\) with precedence

\[
\text{terms} > : ; \text{recip}, \text{sqr}; \text{sqr} > \text{dbl}, + > s; \text{ and first} > [ ]
\]

Friedman and Wise also use replacement restrictions to provide alternative (more efficient) definitions to logical connectives \(\text{and}\), \(\text{or}\). In fact, they implement their “short-cut” definitions of these boolean operators using lazy cons in such a way that the evaluation is suspended after the first argument which evaluates to \(\text{false}\) (resp. \(\text{true}\)) if connective and (resp. \(\text{or}\)) is considered (see [FW76, p. 277]). This is implemented (without using lists) with the TRS:

\[
\begin{align*}
\text{and}(&\text{(false, x)} \rightarrow \text{false} \quad \text{or(true, x)} \rightarrow \text{true} \\
\text{and(true, x)} & \rightarrow x \quad \text{or(false, x)} \rightarrow x
\end{align*}
\]

together with \(\mu(\text{and}) = \mu(\text{or}) = \{1\}\).

Syntactic replacement restrictions have been explicitly included in the design of several (eager) programming languages. For instance, the so-called strategy annotations have been used in the OBJ family of languages\(^4\) to introduce replacement restrictions aimed at improving the efficiency of computations (by reducing the number of attempted matchings). Their usefulness has been demonstrated in practice: in [FGJM85] the authors remark that, due to their use in OBJ2 programs, “the ratio between attempted matches and successful matches is usually around 2/3, which is really impressive.” For instance, OBJ’s built-in conditional operator has the following (implicit) strategy annotation\(^5\) ([FGJM85, Sect. 4.4], [GWMFJ00, Sect. 2.4.4])

\(^4\) As in [GWMFJ00], by OBJ we mean OBJ2, OBJ3, CafeOBJ, or Maude.

\(^5\) A more precise and general definition can be found in OBJ3’s standard prelude, see Appendix D.3 of [GWMFJ00].
\[
\text{op if\_then\_else\_fi : Bool Int Int} \rightarrow \text{Int [strat (1 0)]}
\]

which says to evaluate the first argument until it is reduced, and then apply rules at the top (indicated by ‘0’). Reductions on the second or third arguments of calls to \text{if\_then\_else\_fi} are never attempted.

Other eager programming languages such as ELAN [BKKMR98] incorporate the specification of syntactic replacement restrictions as an ingredient of the definition of more complex rewriting strategies which can be used to guide the evaluation of expressions.

In \textit{(lazy)} functional programming, different kinds of syntactic annotations on the program (such as strictness annotations [Pey87], or global and local annotations [PE93]) have been introduced in order to drive local changes in the basic underlying lazy evaluation strategy and obtain more efficient executions [Bur91, MN92, Myc80, PE93, Pey87]. In these languages, constructor symbols are lazy, i.e., their arguments are not evaluated until needed. This permits structures that contain elements which, if evaluated, would lead to an error or fail to terminate [HPP99]. Since there are a number of overheads in the implementation of this feature (see [Pey87]), lazy functional languages such as Gofer [Jon92] and Haskell [HPW92] allow for syntactic annotations on the arguments of datatype constructors, thus allowing an immediate evaluation.

\textbf{Example 4.} The following definition in Haskell

\[
data \text{List a} = \text{Nil} | \text{Cons} \! \text{a (List a)}
\]

declares a (polymorphic) type \text{List a} whose binary data constructor \text{Cons} evaluates the first argument (of type \text{a}). This is specified by using the symbol “!” in the first argument of \text{Cons}.

Other lazy functional languages, such as Clean [ENPS92, NSEP92, PE93], allow for more general annotations.

\textbf{Example 5.} The following specification

\[
\text{if} :: \! \text{Bool a a} \rightarrow \text{a}
\]

\[
\text{if True} \quad \text{x y} = \text{x}
\]

\[
\text{if False} \quad \text{x y} = \text{y}
\]

is an annotated definition of the function \text{if} which forces the evaluation of the first argument of each \text{if} call (see the mark “!” in the type declaration of \text{if}; this is called a \textit{global annotation} [PE93]).

The use of annotations of this kind can be understood as follows [PE93]: “A given lazy strategy indicates whether an argument \(t_i\) of a function call \(f(t_1, \ldots, t_k)\) must be evaluated. However, we overcome this rule by evaluating \(t_i\) (up to a head-normal form) if the \(i\)th argument is annotated in the profile of \(f\)”.

Thus, annotations play a \textit{secondary} role in the global execution mechanism: an underlying strategy is assumed. Program annotations are usually obtained from some kind of strictness\textsuperscript{6} analysis. Strictness analyses are usually costly as they involve fixpoint computations [CP85, Myc80]. In this case, the safety of this deviation of the main strategy is ensured because strictness analyses are derived from the semantics of the functional language. Sometimes, the programmer is allowed (but discouraged) to annotate the program by himself or herself. In this case, however, there is no way to determine what kind of modification of the semantics or computational behavior is introduced by the annotations.

Context-sensitive rewriting takes the symmetric approach; it can be thought of as a mechanization of the syntactic annotations themselves. We do not assume any extra sophisticated evaluation mechanism. In [Luc96, Luc98a] (and also in this paper), we have analyzed computational properties of CSR–confluence, termination, and completeness in computations leading to (infinite) normal forms, (infinite) values, (infinite) values,

\textsuperscript{6}Let \(D_1, \ldots, D_k, D\) be ordered sets with least elements \(\perp_1, \ldots, \perp_k, \perp\) respectively, expressing undenedness. A mapping \(f : D_1 \times \cdots \times D_k \rightarrow D\) is said to be strict in its \(i\)th argument if \(f(d_1, \ldots, \perp_i, \ldots, d_k) = \perp\) for all \(d_1 \in D_1, \ldots, d_k \in D_k\). See [Luc98a, Sect. 3.3.3] for connections between strictness and CSR.
head-normal forms, and constructor head-normal forms—, thus giving a solid theory for computing. Our methods do not depend on the source of the annotations: strictness analyses, programmer’s notes, or whatever else. Indeed, as the use of strictness information is often claimed to be a suitable way for specifying the replacing arguments of functions, we want to note that (in contrast to $\mu_{\text{str}}$) the (exclusive) use of strictness information for defining a replacement map (for instance, by letting $i \in \mu_{\text{str}}^i(f)$ if and only if the $i$th argument of $f$ is strict) does not ensure the good computational properties of CSR. Moreover, there is no clear correspondence between $\mu_{\text{str}}^i$ and $\mu_{\text{str}}$. 

Example 6. Consider the function $f$ defined by

$$f(x) \rightarrow x.$$ 

Thus, $f$ is strict and $1 \in \mu_{\text{str}}(f)$. However, $1 \notin \mu_{\text{str}}^i(f)$. On the other hand, consider the rules defining first in Example 1. Then, $2 \in \mu_{\text{str}}^i(\text{first})$ but $2 \notin \mu_{\text{str}}^i(\text{first})$, since first$(0,\bot) = [] \neq \bot$, i.e., first is not strict in its second argument. Note that first$(s(0),\text{from}(0))$ does not $\mu_{\text{str}}^i$-rewrite to $0:\text{first}(0,\text{from}(s(0)))$ because the second argument of first$(s(0),\text{from}(0))$ is not reducible using $\mu_{\text{str}}^i$. Thus, the use of strictness information in CSR does not even ensure root-normalization of terms. This is in contrast with $\mu_{\text{str}}$. 

In general, strictness information is not adequate for defining a replacement map (for CSR) as there is no underlying (lazy) strategy which can be altered according to strictness annotations, i.e., these annotations play a secondary role in the computation. In contrast, syntactic annotations are the only way to activate reductions in CSR; no underlying computational mechanism is assumed.

1.2. Contributions of the Paper

As CSR is a rewriting restriction, it always preserves termination. Thus, if a TRS is terminating (this means that no term initiates an infinite rewrite sequence), then it is also $\mu$-terminating (no term initiates an infinite $\mu$-rewrite sequence). However, it is more interesting to use CSR to achieve termination, i.e., to avoid infinite rewrite sequences even if they exist. In term rewriting, there are two main approaches for addressing the problem of ensuring finiteness of rewriting computations issued from a TRS $\mathcal{R}$:

1. studying termination of $\mathcal{R}$ to prove that no term starts an infinite rewrite sequence (see [Zan02] for a recent survey on this topic),
2. defining normalizing strategies for $\mathcal{R}$ (i.e., rules to define specific rewrite sequences that avoid infinite rewritings starting from terms that have a normal form and finally obtain such a normal form).

The first approach excludes the second one: the definition of normalizing strategies for terminating TRSs becomes trivial (except for efficiency issues). Unfortunately, termination is, in general, undecidable. Moreover, requiring termination of a TRS is considered to be quite a strong restriction for many applications. Thus, many researchers have investigated how to define normalizing rewriting strategies. In most cases, such strategies only work for TRSs that satisfy strong syntactic requirements (typically orthogonality or almost orthogonality together with left-normality, inductive sequentaility, strong sequentiality, etc.) on the shape of rules of the TRSs [Ant92, AM96, DM97, HL91, Ken89, O’Do77, O’Do85, SR93, Toy92]. Formal techniques for proving termination are much more general since they usually apply to arbitrary TRSs [AG00, BFR00, Der87]. On the other hand, in contrast to termination analysis, checking whether a TRS satisfies the syntactic requirements for applying a given normalizing strategy is usually easy (e.g., with (almost) orthogonality, left-normality, inductive sequentiality).

In order to formalize the claim that “CSR can be used to improve the termination behavior,” we are going to show that $\mu$-normalization (i.e., the computation of $\mu$-normal forms) can be used (for instance) to compute normal forms whenever they exist. As in unrestricted rewriting, we also have two possibilities for ensuring finiteness of context-sensitive computations. First we can try to show that there is no infinite $\mu$-rewrite sequence by proving $\mu$-termination of the TRS. Fortunately, several methods
have been developed for addressing this task (see Example 3) [BLR02, FR99, GL02, GM99, GM02, Luc96, Luc02c, SX98, Zan97]. There is also a software tool which helps to apply most of these methods [Luc02b].

On the other hand, we can also look for some computation rule which avoids infinite \( \mu \)-rewrite sequences starting from terms having a \( \mu \)-normal form. In contrast to unrestricted rewriting, we show that these two approaches are useful (and complement each other) for defining normalizing strategies since there are \( \mu \)-terminating TRSs that are not terminating (for instance, the TRSs in Examples 1 and 2). This is expressed by the following result which we demonstrate in this paper: Left-linear, confluent, and \( \mu_{\text{can}} \)-terminating TRSs admit a computable (one-step) normalizing strategy.

In this paper, we investigate the definition, properties, and use of context-sensitive rewriting strategies (or just \( \mu \)-strategies for a given replacement map \( \mu \)), i.e., concrete, fixed sequences of context-sensitive rewriting steps. To our knowledge, these strategies have not been studied before. In particular, we pay attention to the definition of \( \mu \)-normalizing \( \mu \)-strategies, i.e., \( \mu \)-strategies that obtain a \( \mu \)-normal form of a term whenever it exists (of course, without issuing infinite \( \mu \)-rewrite sequences, even though they exist). We prove that, for every left-linear, confluent TRS \( \mathcal{R} \), every \( \mu_{\text{can}} \)-normalizing \( \mu_{\text{can}} \)-strategy for \( \mathcal{R} \) induces a normalizing strategy for \( \mathcal{R} \).

We also show that \( \mu \)-normalizing strategies can be used for infinitary normalization, i.e., for obtaining existing infinite normal forms of a term.

Every \( \mu \)-strategy is trivially \( \mu \)-normalizing for \( \mu \)-terminating TRSs. However, whenever a replacement map \( \mu \) cannot ensure \( \mu \)-termination (or we fail to prove that it can), we need to provide \( \mu \)-normalizing \( \mu \)-strategies. A well-known theory for defining (efficient) normalizing strategies is Huet and Lévy’s theory of needed reductions\(^7\) [HL79, HL91]. However, reduction of needed redexes is not adequate for defining \( \mu \)-normalizing \( \mu \)-strategies. This is because there are terms that have a \( \mu \)-normal form but have no normal form. Since each redex in a term which does not have a normal form is needed, needness is not useful for discriminating the redexes which should be contracted to \( \mu \)-normalize (such) a term. Instead, we use Middeldorp’s root-normalizing and root-needed computations [Mid97]. A redex in a term is root-needed if the redex (itself or some of its descendants) is reduced in every root-normalizing derivation issued from this term. Every term which is not root-stable contains a root-needed redex and reduction of root-needed redexes is root-normalizing [Mid97]. Since we have proven that, under certain conditions, \( \mu \)-normal forms are root-stable [Luc98a], Middeldorp’s theory provides an appropriate framework for the definition of \( \mu \)-normalizing \( \mu \)-strategies which we can summarize as follows: Orthogonal TRSs \( \mathcal{R} \) admit a (one-step) \( \mu_{\text{can}} \)-normalizing \( \mu_{\text{can}} \)-strategy.

This result does not provide an immediate “operative” definition of \( \mu_{\text{can}} \)-normalizing \( \mu_{\text{can}} \)-strategies. This is because both root-stability and root-neededness are undecidable. By using the decidable notion of \( \mu_{\text{can}} \)-normal form (instead of that root-stable term) and existing decidable approximations to root-neededness (see [Luc98b]), we are able to finally give a suitable notion of \( \mu \)-normalizing \( \mu \)-strategy which can be implemented in the corresponding class of TRSs. Summarizing: Almost orthogonal, strongly (or NV-) sequential TRSs \( \mathcal{R} \) admit a computable (one-step) \( \mu_{\text{can}} \)-normalizing \( \mu_{\text{can}} \)-strategy.

After some preliminary definitions in Section 2 and a brief introduction to context-sensitive rewriting (Section 3), this paper addresses four main topics:

1. Characterization of \( \mu \)-normal forms and the \( \mu \)-normalization process with respect to unrestricted rewriting (Section 4).
2. Definition of the notion of context-sensitive rewriting strategy and analysis of its general properties (Section 5).
3. Effective definition of \( \mu \)-normalizing context-sensitive rewriting strategies (Sections 6, 7, and 8).
4. Use of context-sensitive strategies for defining root-normalizing, normalizing, and infinitary normalizing rewriting strategies (Sections 9 and 10).

\(^7\) A needed redex of a term \( t \) is a redex which is contracted (either itself or its descendants) in every rewrite sequence which normalizes \( t \).
We conclude the paper by describing the application of our techniques to the definition of normalizing strategies for TRSs which do not admit a normalizing strategy based on the usual techniques for doing so. In addition, we apply our results to the analysis of computational properties of the strategies used in rewriting-based programming languages such as OBJ [GWMFJ00] or ELAN [BKKMR98] (Section 11). Some final concluding remarks and directions for future work are given in Section 12.

2. PRELIMINARIES

Let us first introduce the main notations used in the paper. For full definitions refer to [AM96, BN98, DJ90, Klo92].

Binary Relations. Let $R \subseteq A \times A$ be a binary relation on a set $A$. We denote the transitive closure of $R$ by $R^+$ and its reflexive and transitive closure by $R^*$. A finite $R$-sequence is a sequence $a_1, a_2, \ldots, a_n$ of elements taken from $A$ such that $a_i Ra_{i+1}$ for $1 \leq i < n$; we say that such a sequence begins in $a_1$ and ends in $a_n$. We say that $R$ is confluent if, for every $a, b, c \in A$, whenever $a R^b$ and $a R^c$, there exists $d \in A$ such that $b R^d$ and $c R^d$. An element $a \in A$ is said to be an $R$-normal form if there exists no $b$ such that $a R b$; otherwise, $a$ is called $R$-reducible. We say that $b$ is an $R$-normal form of $a$ (written $a R^b$) if $b$ is an $R$-normal form and $a R^a b$; in this case, we also say that $a$ is $R$-normalizing. We say that $R$ is normalizing if every $a \in A$ has an $R$-normal form, i.e., for all $a \in A$, there is $b \in A$ such that $a R^b$.

In a normalizing relation, each element $a \in A$ has (at least) one normal form. In a confluent and normalizing relation, the normal form exists and is unique. We say that $R$ is terminating iff there is no infinite sequence $a_1 R a_2 R a_3 \ldots$. Obviously, terminating relations are normalizing.

Terms and Positions. Throughout the paper, $\mathcal{X}$ denotes a countable set of variables and $\mathcal{F}$ a set of function symbols $\{f, g, \ldots\}$, each having a fixed arity given by a function $ar : \mathcal{F} \to \mathbb{N}$. We denote the set of terms by $T(\mathcal{F}, \mathcal{X})$. A $k$-tuple $t_1, \ldots, t_k$ of terms is written $\vec{t}$. The number $k$ of elements of the tuple $\vec{t}$ will be clarified by the context. $\text{Var}(t)$ is the set of variables in $t$.

Terms are viewed as labeled trees in the usual way. Positions $p, q, \ldots$ are represented by chains of positive natural numbers which are used to address subterms of $t$. We denote the empty chain by $\Lambda$. We denote the length of a chain $p$ as $|p|$. If $p$ is a position, and $Q$ is a set of positions, $p Q$ is the set $\{p, q \mid q \in Q\}$. Positions are ordered by the standard prefix ordering: $p \leq q$ iff $\exists q'$ such that $q = p q'$; $p \parallel q$ means $p \not\leq q$ and $q \not\leq p$. The subterm at position $p$ of $t$ is denoted as $t_p$ and $t[s]_p$ is the term $t$ with the subterm at position $p$ replaced with $s$. We denote the set of positions of a term $t$ by $\text{Pos}(t)$. Given terms $t$ and $s$, $\text{Pos}(t)$ denotes the set of positions of $s$ in $t$, i.e., $p \in \text{Pos}(t)$ iff $t|_p = s$. Positions of nonvariable symbols in $t$ are denoted as $\text{Pos}_s(t)$ and $\text{Pos}_f(t)$ are the positions of variable occurrences. A term is said to be linear if it has no multiple occurrences of a single variable. The symbol labeling the root of a $t$ is denoted as $\text{root}(t)$. The chain of symbols lying on positions above or on $p \in \text{Pos}(t)$ is $\text{prefix}(\Lambda) = \text{root}(t), \text{prefix}(i, p) = \text{root}(t) \text{prefix}_{i_p}(p)$. The strict prefix $\text{prefix}(\Lambda) = \Lambda, \text{prefix}(i, p) = \text{prefix}(p)$. A context is a term $C \in T(\mathcal{F} \cup \{\square\}, \mathcal{X})$ with zero or more holes $\square$ (a fresh constant symbol). We write $C[\ ]_p$ to denote that there is a (usually single) hole $\square$ at position $p$ of $C$. Generally, we write $C[\ ]$ to denote an arbitrary context (where the number and location of the holes is clarified ‘in situ’) and $C[t_1, \ldots, t_n]$ to denote the term obtained by filling the holes of a context $C[\ ]$ with terms $t_1, \ldots, t_n$. $C[\ ] = \square$ is called the empty context.

Term Rewriting Systems. A rewrite rule is an ordered pair $(l, r)$, written $l \rightarrow r$, with $l, r \in T(\mathcal{F}, \mathcal{X})$, $l \notin \mathcal{X}$, and $\text{Var}(r) \subseteq \text{Var}(l)$. The left-hand side (lhs) of the rule is $l$ and the right-hand side (rhs) is $r$. A TRS is a set $\mathcal{R} = (\mathcal{F}, \mathcal{R})$ where $\mathcal{R}$ is a set of rewrite rules. $\text{L}(\mathcal{R})$ denotes the lhs’s of $\mathcal{R}$. Given a substitution $\sigma$, an instance $\sigma(l)$ of a lhs $l$ of a rule is a redex. The set of redex positions in $t$ is $\text{Pos}_R(t) = \{p \in \text{Pos}(t) \mid \exists l \in \text{L}(\mathcal{R}) : t|_p = \sigma(l)\}$.

A TRS $\mathcal{R}$ is left-linear if for all $l \in \text{L}(\mathcal{R})$, $l$ is a linear term. Two rules $l \rightarrow r$ and $l' \rightarrow r'$ (whose variables have been possibly renamed to satisfy that $\text{Var}(l) \cap \text{Var}(l') = \emptyset$) overlap if there is a nonvariable position $p \in \text{Pos}_R(l)$ and a most-general unifier $\sigma$ such that $\sigma(l|_p) = l'(t)$ at $p$. The pair $(\sigma(l)[\sigma(r')|_p], \sigma(r))$ is called a critical pair and is also called an overlay if $p = \Lambda$. A critical pair $(t, s)$ is trivial if $t = s$. The critical pairs of a TRS $\mathcal{R}$ are the critical pairs between any two of its (renamed)
Then, we have \( t \rightarrow \) the one-step rewrite relation for \( f \). Since \( \rightarrow \) normalizing terms are said to be normalizing. A TRS is con- nected to a redex. A term is said to be root-normalizing if it has a root-stable reduct. In this paper, \( t \) are replaced by \( t \) if \( t \) is not a redex of \( t \). The inner reduction relation is \( \Rightarrow \) if \( \Rightarrow \) and \( \Rightarrow \) are \( \Rightarrow \)-normal forms (resp. \( \Rightarrow \)-reducible terms) are called normal forms (resp. reducible terms); \( \Rightarrow \) normalizing terms are said to be normalizing. A TRS is confluent (resp. normalizing, terminating) if \( \Rightarrow \) is confluent (resp. normalizing, terminating).

2.1. \( \Omega \)-Reduction

A new constant symbol \( \Omega \) is introduced to represent arbitrary terms. Terms in \( \mathcal{T}(\mathcal{F} \cup \{\Omega\}, \mathcal{X}) \) (which we denote as \( \mathcal{T}_\Omega(\mathcal{F}, \mathcal{X}) \)) are said to be \( \Omega \)-terms and they are used to denote prefixes of terms. We denote by \( t_0 \) the term \( t \) where all variables are replaced by \( \Omega \). Positions \( \in \mathcal{Pos}_\Omega(t) \) are said to be \( \Omega \)-positions of \( t \). An ordering \( \leq \) on \( \Omega \)-terms is given: \( \Omega \leq t \) for all \( t, s \leq x \) if \( x \in \mathcal{X} \), and \( f(i) \leq f(j) \) if \( t_i \leq s_i \) for all \( 1 \leq i \leq \text{ar}(f) \). Thus, \( t \leq s \) means that \( t \) is a prefix of \( s \). We write \( t \uparrow s \) if \( t \) and \( s \) are compatible, i.e., if there exists \( u \) such that \( t \leq u \) and \( u \leq s \). We note that, if \( t \uparrow s \), there is a maximal context \( C[.] \) such that \( t = C[t_1, \ldots, t_n] \), \( s = C[s_1, \ldots, s_n] \), and for all \( j, 1 \leq j \leq n \), either \( t_j \leq s_j \) (and thus \( t_j = \Omega \)) or \( t_j \geq s_j \) (and thus \( s_j = \Omega \)). We denote by \( t^\Omega \) the term \( t \) where all outermost redexes are replaced by \( \Omega \). An \( \Omega \)-normal form is an \( \Omega \)-term \( t \) such that \( \mathcal{Pos}_\Omega(t) = \emptyset \) and \( \mathcal{Pos}_\Omega(t) \neq \emptyset \). The following fact is used later.

**Lemma 1.** Let \( \mathcal{R} \) be a left-linear TRS. Let \( t \) be a term and \( l \in \mathcal{L}(\mathcal{R}) \). Then, \( t \geq l^\Omega \) iff \( t \) is a redex.

Left-linearity is required for the only if part of this lemma. For instance, \( f(a, b) \geq f(\Omega, \Omega) \), but \( f(a, b) \) is not a redex of \( f(x, x) \).

In the following, we will use the reduction relation \( \rightarrow_\Omega \) (called \( \Omega \)-reduction, [KM91]) on \( \Omega \)-terms: \( t \rightarrow_\Omega s \) (or just \( t \rightarrow s \)) if \( p \in \mathcal{Pos}(t) - \mathcal{Pos}_\Omega(t), t|_p \uparrow l^\Omega \) for some \( l \in \mathcal{L}(\mathcal{R}) \), and \( s = t[\Omega]|_p \).

**Example 7.** Consider the TRS \( \mathcal{R} \):

\[
\begin{align*}
f(a) & \rightarrow a \\
b & \rightarrow c.
\end{align*}
\]

Then, we have

\[
\begin{align*}
f(b) & \rightarrow_\Omega f(\Omega).
\end{align*}
\]

Note that redex \( b \) has simply been replaced by \( \Omega \), without considering the \( rhs \) of the rule \( b \rightarrow c \). Now, since \( f(\Omega) \uparrow f(a) \), we also have

\[
\begin{align*}
f(\Omega) & \rightarrow_\Omega \Omega.
\end{align*}
\]

The reduction relation \( \rightarrow_\Omega \) is confluent and terminating for arbitrary TRSs [HL91, KM91]. Let \( \omega(t) \) be the \( \rightarrow_\Omega \)-normal form of \( t \). Clearly, \( \omega(\Omega) = \Omega \) and, whenever \( t \) is a redex, \( \omega(t) = \Omega \).

**Proposition 1** [KM91]. Let \( \mathcal{R} = (\mathcal{F}, \mathcal{R}) \) be a TRS. Let \( t, s \in \mathcal{T}_\Omega(\mathcal{F}, \mathcal{X}) \) and let \( p \in \mathcal{Pos}(t) \). Then,

1. \( \omega(t) \leq t \)
2. \( \omega(t) = \omega(t[\omega(t)|_p])_p \)
3. \( t \leq s \Rightarrow \omega(t) \leq \omega(s) \)
4. \( t \rightarrow s \Rightarrow \omega(t) \leq \omega(s) \).

A term \( t \in \mathcal{T}_\Omega(\mathcal{F}, \mathcal{X}) \) is strongly root-stable (or a strong head-normal form) if \( \omega(t) \neq \Omega \).
PROPOSITION 2 [Ken94]. Let $\mathcal{R}$ be a TRS. If $t$ is strongly root-stable, then $t$ is root-stable.

Proof. If $t$ is not root-stable, then $t \rightarrow^* \sigma(l)$ for some $l \in L(\mathcal{R})$. By Proposition 1, $\omega(t) \leq \omega(\sigma(l)) = \Omega$. Hence $\omega(t) = \Omega$, a contradiction. ■

In general, the converse statement is not true [Ken94].

EXAMPLE 8. Consider the TRS of Example 7. The term $f(b)$ is root-stable, but, as shown in Example 7, it is not strongly root-stable.

A term $t$ is rigid if $\omega(t) = t$ and soft if $\omega(t) = \Omega$ [KM91, Klo92]. We also say that the context $C[\ ]$ is rigid if $C[\Omega]$ is a rigid term.

PROPOSITION 3 [KM91, Klo92]. Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS. Every term $t \in T_\Omega(\mathcal{F}, \mathcal{X})$ can be uniquely written as $t = C[t_1, \ldots, t_n]$ such that $C[\Omega, \ldots, \Omega]$ is rigid and the $t_i, 1 \leq i \leq n$ are soft.

Rigidity is compositional. This fact will be used later.

Lemma 2. Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $C[\ ], C_1[\ ], \ldots, C_n[\ ]$ be rigid contexts. Then, $C[C_1[\ ], \ldots, C_n[\ ]]$ is a rigid context.

Proof. By contradiction. If $t = C[C_1[\ ], \ldots, C_n[\ ]]$ is not rigid, since $C_1[\ ], \ldots, C_n[\ ]$ are rigid, it follows that there is $p \in Pos_f(C[\Omega])$ such that $t_p$ is compatible with $l_{\Omega}$ for some $l \in L(R)$. Therefore, since $C[\Omega]|_p \leq t_p$, we also get $C[\Omega]|_p \uparrow l_{\Omega}$. Hence, $\omega(C[\Omega]) < C[\Omega]$ and $C[\Omega]$ is not rigid. ■

3. CONTEXT-SENSITIVE REWRITING

Given a signature $\mathcal{F}$, a mapping $\mu : \mathcal{F} \rightarrow \mathcal{P}(\mathbb{N})$ is a replacement map (or $\mathcal{F}$-map) if for all $f \in \mathcal{F}$, $\mu(f) \subseteq \{1, \ldots, ar(f)\}$. The replacement map $\mu$ determines the argument positions which can be reduced for each symbol in $\mathcal{F}$ [Luc98a].

Example 9. The conditional operator $\text{if-then-else}$ can be given the following replacement map

$$
\mu(\text{if-then-else}) = \{1\}
$$

which is intended (as expected) to permit reductions only on the first argument.

The set of all $\mathcal{F}$-maps is $M_{\mathcal{F}}$. If the signature $\mathcal{F}$ corresponds to a TRS $\mathcal{R} = (\mathcal{F}, R)$, we also write $M_{\mathcal{R}}$ rather than $M_{\mathcal{F}}$.

The inclusion ordering $\subseteq$ on $\mathcal{P}(\mathbb{N})$ extends to an ordering $\sqsubseteq$ on $M_{\mathcal{F}}$, the set of all $\mathcal{F}$-maps: $\mu \sqsubseteq \mu'$ if for all $f \in \mathcal{F}$, $\mu(f) \subseteq \mu'(f)$; accordingly, the lattice $(\mathcal{P}(\mathbb{N}), \subseteq, \emptyset, \mathbb{N}, \cup, \cap)$ induces a lattice $(M_{\mathcal{F}}, \subseteq, \mu_\perp, \mu_\top, \cup, \cap)$: The minimum element is $\mu_\perp$, given by $\mu_\perp(f) = \emptyset$ for all $f \in \mathcal{F}$; the maximum element is $\mu_\top$, given by $\mu_\top(f) = \{1, \ldots, ar(f)\}$ for all $f \in \mathcal{F}$; and the least upper bound $(\cup b)$, $\cup$, and greatest lower bound $(\cap b)$, $\cap$, are given by $(\mu \cup \mu')(f) = \mu(f) \cup \mu'(f)$ and $(\mu \cap \mu')(f) = \mu(f) \cap \mu'(f)$ for all $f \in \mathcal{F}$. Thus, $\mu \sqsubseteq \mu'$ means that $\mu$ considers less positions than $\mu'$ for reduction. We also say that $\mu$ is more restrictive than (or equally restrictive to) $\mu'$.

Example 10. The following figure depicts the lattice of replacing arguments that can be associated to symbol $\text{if-then-else}$:

```
{1, 2, 3}  {1, 2}  {1, 3}  {2, 3}  {1}  {2}  {3}  \emptyset
```

Given $t \in T(\mathcal{F}, \mathcal{X})$ and $\mu \in M_\mathcal{F}$, the set of $\mu$-replacing positions $\mathcal{P}os^\mu(t)$ is

$$\mathcal{P}os^\mu(t) = \{\Lambda\},$$

if $t \in \mathcal{X}$ and

$$\mathcal{P}os^\mu(t) = \{\Lambda\} \cup \bigcup_{i \in \mu(\text{mot}(t))} i.\mathcal{P}os^\mu(t|_i)$$

if $t \not\in \mathcal{X}$. The non-$\mu$-replacing positions are $\overline{\mathcal{P}os^\mu}(t) = \mathcal{P}os(t) - \mathcal{P}os^\mu(t)$. By abuse, the occurrence of subterm $t|_p$ at position $p$ is called replacing (resp. nonreplacing) if $p \in \mathcal{P}os^\mu(t)$ (resp. $p \in \overline{\mathcal{P}os^\mu}(t)$).

**Proposition 4 [Luc98a].** Let $t \in T(\mathcal{F}, \mathcal{X})$ and $p = q.q' \in \mathcal{P}os(t)$. Then $p \in \mathcal{P}os^\mu(t)$ iff $q \in \mathcal{P}os^\mu(t) \land q' \in \mathcal{P}os^\mu(t|_q)$.

The non-$\mu$-replacing positions never lie above replacing ones.

**Proposition 5.** Let $t \in T(\mathcal{F}, \mathcal{X})$, $p \in \mathcal{P}os^\mu(t)$, and $q \in \overline{\mathcal{P}os^\mu}(t)$. Then $q \not\leq p$.

Hence, the (ordered) set of replacing positions $(\mathcal{P}os(t), \leq)$ is downward closed: for all $p \in \mathcal{P}os^\mu(t)$, $q \leq p \Rightarrow q \in \mathcal{P}os^\mu(t)$. This fact will be used later.

The following proposition establishes that the replacing nature of a position within a term does not depend on the context surrounding that position.

**Proposition 6 [Luc98a].** If $p \in \mathcal{P}os(t) \cap \mathcal{P}os(s)$ and $\text{spre}x_{\mu}(p) = \text{spre}x_{\mu}(p)$, then $p \in \mathcal{P}os^\mu(t) \iff p \in \mathcal{P}os^\mu(s)$.

In context-sensitive rewriting (CSR), we rewrite subterms at replacing positions: $t \xrightarrow{\mu} s$,

$$t \xrightarrow{\mu} \mathcal{R}(\mu) s$$

(or simply $t \xleftarrow{\mu} \mathcal{R}(\mu) s$ or $t \xrightarrow{} s$) if $t \xrightarrow{\mu} \mathcal{R} s$ and $p \in \mathcal{P}os^\mu(t)$. The $\xrightarrow{} \mu$-normal forms (or $\mu$-normal terms) are called $\mu$-normal forms ($\mu$-reducible terms). The set of replacing redexes is $\mathcal{P}os^\mu_{\mathcal{R}}(t) = \mathcal{P}os(t) \cap \mathcal{P}os^\mu(t)$. Obviously, a term $t$ is a $\mu$-normal form if and only if $\mathcal{P}os^\mu(t) = \emptyset$. Let $\text{NF}^\mu_{\mathcal{R}}$ be the set of $\mu$-normal forms of $\mathcal{R}$.

**Example 11.** Consider $\mathcal{R}$ and $\mu$ as in Example 1. Now, we write

$$\text{sel}(s(0), \text{from}(s(0))) \xrightarrow{} \text{sel}(s(0), s(0): \text{from}(s(s(0))))$$

However,

$$\text{sel}(s(0), s(0): \text{from}(s(s(0)))) \not\xrightarrow{} \text{sel}(s(0), s(0): s(s(0)): \text{from}(s(s(s(0))))$$

since the restriction $\mu(\cdot) = \{1\}$ avoids the replacement of redex $\text{from}(s(s(0)))$ in term $\text{sel}(s(0), s(s(0)): \text{from}(s(s(s(0))))$.

A finite $\xrightarrow{} \mu$-sequence is called a $\mu$-rewrite sequence. Given $\mu \in M_\mathcal{R}$, we say that a TRS $\mathcal{R}$ is $\mu$-confluent (resp. $\mu$-normalizing, $\mu$-terminating) if $\xrightarrow{} \mathcal{R}(\mu)$ is confluent (resp. normalizing, terminating).

### 3.1. Canonical Context-Sensitive Rewriting

The **canonical replacement map** for a TRS $\mathcal{R}$ is $\mu^{\text{can}}_\mathcal{R}$ [Luc98a]: The most restrictive replacement map which ensures that the (positions of) nonvariable subterms of the left-hand sides of the rules of $\mathcal{R}$ are replacing.

---

8 In [Zan97], Zantema uses “active and forbidden” instead of $\mu$-replacing and non-$\mu$-replacing. However, this terminology has also been used elsewhere for formalizing different (but related) rewriting restrictions [FKW00, Luc02a].

9 In [Luc98a], we used $\mu^{\text{can}}_\mathcal{R}$ instead of $\mu^{\text{can}}_\mathcal{R}$ to denote the canonical replacement map. We believe that the new notation is clearer.
Note that \( \mu^\text{can}_R \) can be automatically associated to \( R \) by means of a very simple calculus: \( \forall f \in \mathcal{F}, i \in \{1, \ldots, ar(f)\}, \)

\[
i \in \mu^\text{can}_R(f) \iff \exists l \in L(R), \ p \in \mathcal{P} \sigma(l), \ \text{(root}(l | \mu) = f \wedge p.i \in \mathcal{P} \sigma(l)).
\]

**Example 12.** Consider the TRS \( R \) of Example 1. Since we have the rules

\[
\text{first}(0, x) \to [] \\
\text{first}(s(x), y : z) \to y \text{:first}(x, z)
\]

we have \( 1 \in \mu^\text{can}_R(\text{first}) \) because, e.g., \( \text{first}(0, x)|_1 = 0 \not\in \mathcal{X} \); and \( 2 \in \mu^\text{can}_R(\text{first}) \) because \( \text{first}(s(x), y : z)|_2 = y : z \not\in \mathcal{X} \). On the other hand, since we have the rules

\[
\text{sel}(0, x : y) \to x \\
\text{sel}(s(x), y : z) \to \text{sel}(x, z)
\]

we have \( \text{sel}(0, x : y)|_1 = 0 \not\in \mathcal{X} \) and \( \text{sel}(0, x : y)|_2 = x : y \not\in \mathcal{X} \); i.e., \( \mu^\text{can}_R(\text{sel}) = \{1, 2\} \). Finally, \( \mu^\text{can}_R(\text{from}) = \varnothing \) for \( f \in \{\text{from}, :\} \). Therefore,

\[
\mu^\text{can}_R(\text{first}) = \mu^\text{can}_R(\text{sel}) = \{1, 2\} \quad \text{and} \quad \mu^\text{can}_R(\text{from}) = \varnothing.
\]

**Example 13.** Consider the TRS \( R \):

\[
\text{half}(0) \to 0 \\
\text{half}(s(s(x))) \to s(\text{half}(x)) \\
\text{half}(s(0)) \to 0.
\]

Then, we have \( \mu^\text{can}_R(\text{half}) = \{1\} \), because, e.g., \( \text{half}(0)|_1 = 0 \not\in \mathcal{X} \). On the other hand, \( \mu^\text{can}_R(\text{from}) = \{1\} \) because, e.g., \( \text{root}(\text{half}(s(0)))|_1 = s \) and \( \text{half}(s(0))|_{1,1} = 0 \not\in \mathcal{X} \).

Given a TRS \( R \), \( CM_R = \{ \mu \in M \ | \ \mu^\text{can}_R \subseteq \mu \} \) is the set of replacement maps which are less restrictive than or equally restrictive to \( \mu^\text{can}_R \). By abuse, if \( \mu \in CM_R \), we say that \( \mu \) is a canonical replacement map of \( R \). We say that \( \leadsto_R \) is a *canonical* context-sensitive rewrite relation if \( \mu \in CM_R \). It is not difficult to see that \( (CM_R, \subset, \mu^\text{can}_R, \mu_\|, \mu \lor, \mu \land) \) is a complete sublattice of \( (M_R, \subset, \mu_\|, \mu_\lor, \mu \land) \).

**Example 14.** Consider the (usual) rules defining the *if-then-else* operator:

\[
\text{if true then } x \ \text{else } y \to x \\
\text{if false then } x \ \text{else } y \to y.
\]

The following figure depicts the (sub)lattice of indices of replacing arguments that can be associated to symbol *if-then-else* by *canonical* replacement maps:

\[
\begin{array}{ccc}
{1, 2, 3} & & \\
| & \text{——} & |
\end{array}
\]

\[
\begin{array}{ccc}
{1, 2} & & {1, 3} \\
| & \text{——} & |
\end{array}
\]

\[
\begin{array}{ccc}
{1} & & \\
| & \text{——} & |
\end{array}
\]

Canonical replacement maps make CSR complete for root-normalization.

**Proposition 7** [Luc98a]. Let \( R \) be a left-linear TRS, \( l \in L(R) \), and \( \mu \in CM_R \). If \( t \to^* \sigma(l) \) for some substitution \( \sigma \), then there is a substitution \( \theta \) such that \( t \to^* \theta(l) \) and \( \theta(x) \to^* \sigma(x) \) for all \( x \in \text{Var}(l) \).
**Theorem 1** [Luc98a]. Let \( R = (F, R) \) be a left-linear TRS and \( \mu \in CM_R \). Let \( t \in T(F, \mathcal{X}) \) and \( s \) be a root-stable term. If \( t \xrightarrow{*} s \), then there exists \( s' \) such that \( t \xrightarrow{*} s' \), root\( (s) = \text{root}(s') \), and \( s' \xrightarrow{\Lambda} s \).

Given \( R = (F, R) \), we take \( F \) as the disjoint union \( F = C \cup D \) of symbols \( c \in C \), called constructors, and symbols \( f \in D \), called defined functions, where \( D = \{ \text{root}(l) \mid l \rightarrow r \in R \} \) and \( C = F - D \).

**Theorem 2** [Luc98a]. Let \( R = (C \cup D, R) \) be a left-linear TRS and \( \mu \in CM_R \). Let \( t \in T(F, \mathcal{X}) \), \( x \in \mathcal{X} \), and \( s = c(\bar{v}) \) for some \( c \in C \). If \( t \xrightarrow{*} x \), then \( t \xrightarrow{*} x \). If \( t \xrightarrow{*} s \), then there exists \( s' = c(\bar{v}) \) such that \( t \xrightarrow{*} s' \) and \( s' \xrightarrow{\Lambda} s \).

**Theorem 3** [Luc98a]. Let \( R = (F, R) \) be an almost orthogonal TRS and \( \mu \in CM_R \). Let \( t \in T(F, \mathcal{X}) \) and \( s \) be a root-stable term. If \( t \xrightarrow{*} s \), then there exists a root-stable term \( s' \) such that \( t \xrightarrow{*} s' \) and \( s' \xrightarrow{\Lambda} s \).

### 4. Characterization of \( \mu \)-Normal Forms

Proposition 5 motivates the definition of maximal replacing context of a term \( t \) which is the maximal prefix of \( t \) whose positions are \( \mu \)-replacing in \( t \).

**Definition 1** (Maximal replacing context of a term). Let \( F \) be a signature, \( t \in T(F, \mathcal{X}) \), and \( \mu \in M_F \). The maximal replacing context of \( t \) is the context \( MRC^\mu(t) \) where:

\[
MRC^\mu(t) = \begin{cases} 
    x & \text{if } t = x \\
    f(C_1, \ldots, C_k) & \text{if } t = f(t_1, \ldots, t_k) \text{ and } C_i = \begin{cases} 
        \text{MRC}^\mu(t_i) & \text{if } i \in \mu(f) \\
        \square & \text{if } i \notin \mu(f)
    \end{cases} \\
      & \text{for } 1 \leq i \leq k.
\end{cases}
\]

If \( C[] = MRC^\mu(t) \), then \( t = C[t_1, \ldots, t_n] \) is such that

\[\{p_1, \ldots, p_n\} = \text{minimal}(\text{Pos}^\mu(t))\]

are the positions of \( t_1, \ldots, t_n \) in \( t \); i.e., \( t_i = t|_{p_i} \) for \( 1 \leq i \leq n \). Downward closedness of \( \text{Pos}^\mu(t) \) in \( (\text{Pos}(s), \preceq) \), together with Proposition 6, implies that every subterm of \( MRC^\mu(t) \) is \( \mu \)-replacing, except for the holes in the context. This justifies the name “maximal replacing context.”

**Proposition 8**. Let \( F \) be a signature, \( t \in T(F, \mathcal{X}) \), and \( \mu \in M_F \). Then, we have \( \overline{\text{Pos}}^\mu(MRC^\mu(t)) = \text{Pos}_\mu(MRC^\mu(t)) \).

**Proof**. By induction on the structure of \( MRC^\mu(t) \).

The maximal replacing context of a term \( t \) is never empty (i.e., \( MRC^\mu(t) \neq \square \)). We prove that maximal replacing contexts of \( \mu \)-normal forms are rigid contexts.

**Proposition 9**. Let \( R = (F, R) \) be a left-linear TRS and \( \mu \in CM_R \). Let \( t \in T_\Omega(F, \mathcal{X}) \) be a \( \mu \)-normal form such that \( \text{Pos}_\Omega^\mu(t) = \emptyset \), and \( C[] = MRC^\mu(t) \). Then, \( C[\Omega] \) is rigid and strongly root-stable.

**Proof**. First, we prove that \( s = C[\Omega] \) is rigid, i.e., \( \omega(C[\Omega]) = C[\Omega] \). We proceed by induction on \( s = C[\Omega] \). If \( C[\Omega] = x \in \mathcal{X} \) (the case \( C[\Omega] = \Omega \) is not possible as \( C[] \neq \square \)), the proof is immediate. If \( C[\Omega] = f(C_1[\Omega], \ldots, C_{ar(f)}[\Omega]) \), then, by I.H., we assume that each \( C_i[\Omega] \) for \( 1 \leq i \leq ar(f) \) is rigid. Therefore, if \( C[\Omega] \) is not rigid, there is \( l \in L(R) \) such that \( s \xrightarrow{l} \). Let \( C[] \) be a maximal context such that \( s = C[s_1, \ldots, s_n] \) and \( l_\Omega = C[l_1, \ldots, l_n] \); i.e., \( C[] \) is the common part of both \( s \) and \( l_\Omega \). Since \( s \xrightarrow{l} \), for all \( 1 \leq i \leq m \) either \( s_i \leq l_j \) (and thus \( s_j = \Omega \)) or \( s_j \geq l_j \) (and thus \( l_j = \Omega \)). There must be at least one \( i \), \( 1 \leq i \leq m \) such that \( s_i < l_i \), otherwise, \( s_j \geq l_j \) for all \( 1 \leq j \leq n \) and \( s \geq l_\Omega \). In this case, since \( t \geq s \geq l_\Omega \) and \( R \) is left-linear, by Lemma 1 \( t \) is also a redex. However, since \( \Lambda \in \text{Pos}^\mu(t) \), \( t \) is a \( \mu \)-redex and it is not a \( \mu \)-normal form. Therefore, since \( s_i < l_i \), we have \( s_i = \Omega = s|_p \), and \( l_i \neq \Omega \).
Hence, \( p \in \mathcal{P}os_{\mathcal{F}}(\Omega) \). Since \( t \) has no replacing \( \Omega \)-positions, by definition of \( s \), and by Proposition 8, we have \( p \in \mathcal{P}os^{\mu}(s) \). However, since \( \mu_{R}^{can} \subseteq \mu \) and \( p \in \mathcal{P}os_{\mathcal{F}}(\Omega) \), we have \( p \in \mathcal{P}os^{\mu}(\Omega) \). Since \( s\text{pre}(p) = s\text{pre}(\mu) \), by Proposition 6, we get \( p \in \mathcal{P}os^{\mu}(s) \) thus leading to a contradiction.

Now, we prove that \( \omega(C[\Omega]) > \Omega \); i.e., \( C[\Omega] \) is strongly root-stable. Since \( C[\ ] \) is not empty, \( C[\Omega] > \Omega \). Since \( C[\Omega] \) is rigid, \( \Omega < C[\Omega] = \omega(C[\Omega]) \), and the conclusion follows.

Left-linearity is required for this proposition (see also Example 16 of [Luc98a] which is quite a related one).

**Example 15.** Consider the non-left-linear TRS \( \mathcal{R} \),

\[
\begin{align*}
  f(x, x) &\rightarrow c(x) \\
  a &\rightarrow b,
\end{align*}
\]

and let \( \mu \) be given by \( \mu(f) = \{1, 2\} \) and \( \mu(c) = \varnothing \). Note that \( \mu_{R}^{can}(f) = \mu_{R}^{can}(c) = \varnothing \); thus, \( \mu \in CM_{R} \). Hence, \( f(c(\square), c(\square)) \) is the maximal \( \mu \)-replacing context of \( f(c(a), c(b)) \). However, despite the fact that \( f(c(a), c(b)) \) is a \( \mu \)-normal form, the term \( f(c(\Omega), c(\Omega)) \) is neither rigid, nor strongly root-stable, since \( f(c(\Omega), c(\Omega)) \rightarrow_{\Omega} \Omega \). Note that variables in the left-hand side of the non-left-linear rule are \( \mu \)-replacing.

The use of a canonical replacement map is required for Proposition 9.

**Example 16.** Consider the TRS \( \mathcal{R} \) of Example 1. If we let \( \mu(\text{first}) = \varnothing \), we have \( \mu \notin CM_{R} \). Then, \( \text{first}(t, x) \) is a \( \mu \)-normal form for any redex \( t \). The maximal replacing context is \( \text{first}(\square, \square) \). However, \( \text{first}(\Omega, \Omega) \) is not rigid, since \( \text{first}(\Omega, \Omega) \uparrow \text{first}(0, \Omega) \). Hence, \( \omega(\text{first}(\Omega, \Omega)) = \Omega \).

Finally, we note that, if \( t \) is not a \( \mu \)-normal form (or it contains \( \mu \)-replacing occurrences of \( \Omega \), \( \text{MRC}^{\mu}(t) \) does not need to be rigid.

**Example 17.** Consider the TRS \( \mathcal{R} \) of Example 7. Term \( t = f(b) \) is not a \( \mu_{R}^{can} \)-normal form (since \( \mu_{R}^{can}(f) = \{1\} \) and \( b \) is a redex). Since \( C[\ ] = \text{MRC}^{\mu_{R}^{can}}(t) = t = C[\Omega] \) and \( \omega(t) = \Omega < t \), we conclude that \( \text{MRC}^{\mu_{R}^{can}}(t) \) is not rigid. On the other hand, \( f(\Omega) \) is a \( \mu_{R}^{can} \)-normal form (but \( \mathcal{P}os_{\Omega}^{\mu_{R}^{can}}(f(\Omega)) = \{1\} \neq \varnothing \) and we also have \( \omega(f(\Omega)) = \Omega < t \).

Maximal replacing contexts of \( \mu \)-normal forms are stable parts of further reductions from those \( \mu \)-normal forms (using unrestricted rewriting).

**Proposition 10.** Let \( \mathcal{R} \) be a left-linear TRS and \( \mu \in CM_{R} \). Let \( t \in \text{NF}_{R} \) and let \( C[\ ] = \text{MRC}^{\mu}(t) \). If \( t \rightarrow^{\ast} s \), then \( C[\Omega] \leq s \), and \( s \) is strong root-stable.

**Proof.** By Proposition 9, \( \omega(C[\Omega]) = C[\Omega] \). Since \( t \rightarrow^{\ast} s \), by Proposition 1, we have \( \omega(t) \leq \omega(s) \leq s \). Since \( C[\Omega] \leq t \), again by Proposition 1, \( \Omega < C[\Omega] = \omega(C[\Omega]) \leq \omega(t) \leq \omega(s) \leq s \) and the conclusion follows.

**Corollary 1.** Let \( \mathcal{R} \) be a left-linear TRS and \( \mu \in CM_{R} \). Every \( \mu \)-normal form is strongly root-stable.

Note that Examples 15 and 16 also show the need for the conditions imposed in Corollary 1 (e.g., \( f(c(a), c(b)) \) in Example 15 and \( \text{first}(t, x) \) in Example 16 are \( \mu \)-normal forms which are not root-stable).

Using Corollary 1 and Proposition 2, we easily conclude the following result.

**Theorem 4 [Luc98a].** Let \( \mathcal{R} \) be a left-linear TRS and \( \mu \in CM_{R} \). Every \( \mu \)-normal form is root-stable.

See [Luc98a] for motivation about the conditions imposed in Theorem 4 and corollary below.

**Corollary 2.** Let \( \mathcal{R} \) be a left-linear TRS and \( \mu \in CM_{R} \). Every \( \mu \)-normalizing term is root-normalizing.

4.1. \( \mu \)-Normalization

The following proposition establishes conditions to ensure that the set of \( \mu \)-normal forms is closed under (unrestricted) rewriting.
Proposition 11. Let $\mathcal{R}$ be a left-linear TRS, $\mu \in \text{CM}_R$, and $t \in \text{NF}_R^\mu$. If $t \rightarrow^* s$, then $s \in \text{NF}_R^\mu$ and $\text{MRC}^{\mu}(t) = \text{MRC}^{\mu}(s)$.

Proof. If $s$ is not a $\mu$-normal form, then $\text{Pos}_R^\mu(s) \neq \emptyset$; thus, assume $p \in \text{Pos}_R^\mu(s) \neq \emptyset$. Let $C[\ ] = \text{MRC}^{\mu}(t)$ and $C'[\ ] = \text{MRC}^{\mu}(s)$. By Proposition 10, $C[\Omega] \leq s$; i.e., $s = C[s_1, \ldots, s_n]$.

If there is $1 \leq i \leq n$ such that $s_i \rightarrow s_i$ and $p = p_i$, then, by Proposition 8, $p_i \in \text{Pos}_R^\mu(t)$. By Proposition 6, $p_i \in \text{Pos}_R^\mu(s)$ and by Proposition 4, $p \in \text{Pos}_R^\mu(s)$ contradicting that $p \notin \text{Pos}_R^\mu(s)$. Therefore, $p \in \text{Pos}_R^\mu(C[\ ])$. Since $p \in \text{Pos}_R^\mu(s)$, $s_i$ is a redex, thus contradicting rigidness of $C[\Omega]$ (Proposition 9).

On the other hand, since $C[\Omega] \leq s$, it follows that $C[\Omega] \leq C'[\Omega]$. If $C[\Omega] < C'[\Omega]$, then there exists $p \in \text{Pos}_R^\mu(C[\Omega])$ such that $C'[\Omega] \neq \Omega$. Nevertheless, by Propositions 8 and 6, $p \in \text{Pos}_R^\mu(C[\Omega])$. By Proposition 6, $p \in \text{Pos}_R^\mu(C'[\Omega])$, thus contradicting Proposition 8.

Left-linearity cannot be dropped in Proposition 11.

Example 18. Consider the TRS $\mathcal{R}$ in Example 15. Note that $f(a, b)$ is a $\mu^\text{cm}_R$-normal form and that $f(g, b) \rightarrow f(b, b)$, which is not a $\mu$-normal form.

If a TRS is confluent, the unique normal form property holds: if a term has a normal form, it is unique.

Confluence does not imply the unique $\mu$-normal form property:

Example 19. Consider the confluent TRS $\mathcal{R}$,

$$f(x, y) \rightarrow c(x)$$

$$a \rightarrow b,$$

and the replacement map $\mu$ such that $\mu(f) = \{1\}$ and $\mu(c) = \emptyset$. Note that $\mu \in \text{CM}_R$. However, we have the following $\mu$-normalizing derivations

$$f(a, b) \leftrightarrow c(a) \quad \text{and} \quad f(a, b) \leftrightarrow f(b, b) \leftrightarrow c(b)$$

which yield different $\mu$-normal forms $c(a)$ and $c(b)$.

Nevertheless, we have the following result.

Proposition 12. Let $\mathcal{R} = (\mathcal{F}, R)$ be a left-linear, confluent TRS, $\mu \in \text{CM}_R$, $t \in T(\mathcal{F}, \mathcal{X})$, and $s', s'' \in \text{NF}_R^\mu$. If $t \rightarrow^* s'$ and $t \rightarrow^* s''$, then $\text{MRC}^{\mu}(s') = \text{MRC}^{\mu}(s'')$.

Proof. Let $C'[] = \text{MRC}^{\mu}(s')$ and $C''[] = \text{MRC}^{\mu}(s'')$. By confluence of $\mathcal{R}$, there is $s$ such that $s' \rightarrow^* s$ and $s'' \rightarrow^* s$. By Proposition 11, $\text{MRC}^{\mu}(s') = \text{MRC}^{\mu}(s') = \text{MRC}^{\mu}(s'')$. 

Unfortunately, confluence alone does not suffice to ensure Proposition 12.

Example 20. Consider the TRS $\mathcal{R}$ of Example 15. Since $\mathcal{R}$ is terminating and has no critical pairs, it is confluent. However, if we let $\mu(f) = \{1\}$ (note that $\mu \in \text{CM}_R$), we have the following ($\mu$-) derivations

$$f(a, a) \leftrightarrow b \quad \text{and} \quad f(a, a) \leftrightarrow f(b, a)$$

producing two different $\mu$-normal forms (namely $b$ and $f(b, a)$) of $f(a, a)$ whose maximal replacing contexts $b$ and $f(b, \square)$ do not coincide.

An immediate consequence of Proposition 12 is the unicity of maximal replacing contexts of $\mu$-normal forms of terms (in presence of confluence and left-linearity).

Theorem 5. Let $\mathcal{R}$ be a left-linear, confluent TRS and $\mu \in \text{CM}_R$. If $s'$ and $s''$ are $\mu$-normal forms of a term $t$, then $\text{MRC}^{\mu}(s') = \text{MRC}^{\mu}(s'')$.

The unique $\mu$-normal form property is ensured by $\mu$-confluence, i.e., confluence of $\leftrightarrow_{\mathcal{R}^{(\mu)}}$. The following proposition establishes that every (unrestricted) reduct of a $\mu$-normalizing term is also $\mu$-normalizing.
To guarantee that Existence: strategy, we need to address the following four main issues:

**Proposition 13.** Let \( \mathcal{R} \) be a left-linear, confluent TRS and \( \mu \in \text{CM}_\mathcal{R} \). If \( t \) is \( \mu \)-normalizing and \( t \rightarrow^* s \), then \( s \) is \( \mu \)-normalizing.

**Proof.** Let \( u \) be a \( \mu \)-normal form of \( t \); i.e., \( t \leftarrow^* u \). Since \( t \rightarrow^* s \), by confluence, there exists a term \( s' \) and derivations \( u \rightarrow^* s' \) and \( s \rightarrow^* s' \). By Proposition 11, \( s' \) is a \( \mu \)-normal form. Let \( C[ ] = \text{MRC}^\mu(s') \).

We prove, by induction on the structure of \( C[ ] \), that there exists a \( \mu \)-normal form \( v \) of \( s \). If \( C[ ] \) is a constant or a variable, then \( s' \) is the normal form of \( s \) and the unique root-stable reduct of \( s \). By Theorem 1, \( s \leftarrow^* s' \) and we let \( v = s' \). If \( C[ ] = f(C_1[ ], \ldots, C_n[ ]) \), since \( s' = f(s'_1, \ldots, s'_k) \) is root-stable (Theorem 4), by Theorem 1 \( s \leftarrow^* f(s'_1, \ldots, s'_k) \) and \( s'' \rightarrow^* s' \) for \( 1 \leq i \leq k \). Since each \( s'_i \) for \( i \in \mu(f) \) is a \( \mu \)-normal form and \( \text{MRC}^\mu(s'_i) = C_i[ ] \), by the induction hypothesis there are \( \mu \)-normal forms \( v_i \) such that \( s'' \rightarrow^* v_i \) for \( i \in \mu(f) \). Hence, since \( \text{MRC}^\mu (s') \) is rigid (Proposition 9), \( v = f(v_1, \ldots, v_k) \) (where \( v_j = s'_j \) if \( j \notin \mu(f) \)) is a \( \mu \)-normal form and \( s \leftarrow^* v \). □

Confluence alone is not sufficient to ensure Proposition 13.

**Example 21.** Consider the confluent TRS \( \mathcal{R} \):

\[
\begin{align*}
f(x, x) & \rightarrow a \\
b & \rightarrow b \\
c & \rightarrow b
\end{align*}
\]

and \( \mu(f) = \{1\} \). Note that \( \mu \in \text{CM}_\mathcal{R} \). Term \( f(c, c) \) has a \( \mu \)-normal form:

\[
f(c, c) \leftarrow a
\]

but \( f(b, c) \), which is a (\( \mu \))-reduct of \( f(c, c) \), has no \( \mu \)-normal form.

### 5. CONTEXT-SENSITIVE REWRITING STRATEGIES

A (nondeterministic) rewriting strategy for a TRS \( \mathcal{R} \) is a function \( S \) that assigns a nonempty set of nonempty finite rewrite sequences each beginning with \( t \) to every term \( t \) which is not a normal form [BEGK’87, Mid97]. As a specialization of the previous notion, by a one-step (nondeterministic) rewriting strategy for a TRS \( \mathcal{R} \) we mean a function \( S \) that assigns a nonempty set \( S(t) \subseteq \text{Pos}_\mathcal{R}(t) \) of redex positions of \( t \) to every reducible term \( t \) [Mid97]. For TRSs that are not weakly orthogonal, we also need to supply the rewrite rule according to which the selected redex is to be contracted since a redex may have more than one contractum (see [AM96] for details).

**Remark 2.** Requiring that reducible terms have nonempty sets of (nonempty) rewrite sequences (or redex positions) is intended to keep the strategy active as long as the term contains redexes. Even though Definition 6.2 of [BEGK’87] admits strategies that assign an empty set (of reductions) to a reducible term, we rather follow [AM96, Klo92, Mid97, OV02] in this respect.

We write \( t \rightarrow_S s \) if \( S(t) \) contains a reduction sequence ending with \( s \) (or \( t \rightarrow_{S_p} s \) and \( p \in S(t) \) for one-step strategies). An \( S \)-sequence is a reduction sequence of the form \( t_1 \rightarrow_S t_2 \rightarrow_S \cdots \). If \( t \rightarrow_{S_p} s \), we say that \( s \) is an \( S \)-reduct of \( t \).

A strategy \( S \) is root-normalizing if for all root-normalizing terms \( t \), every possible infinite \( S \)-sequence starting from \( t \) contains a root-stable term; \( S \) is \( \mu \)-normalizing if for all \( \mu \)-normalizing terms \( t \), every infinite \( S \)-sequence starting from \( t \) contains a \( \mu \)-normal form whose \( S \)-reductions are always \( \mu \)-normal forms; \( S \) is normalizing if, for all normalizing terms \( t \), there is no infinite \( S \)-sequence starting from \( t \).

**Proposition 14.** Let \( \mathcal{R} \) be a left-linear TRS, \( \mu \in \text{CM}_\mathcal{R} \), and \( S \) be a rewriting strategy such that every infinite \( S \)-sequence contains a \( \mu \)-normal form. Then, \( S \) is \( \mu \)-normalizing.

**Proof.** Proposition 11. □

In order to formally (and practically) use a given intuitive principle for the definition of a rewriting strategy, we need to address the following four main issues:

**Existence:** To guarantee that \( S(t) \neq \emptyset \) for every reducible term \( t \).

\(^{10}\) This is required to ensure that reaching a \( \mu \)-normal form implies that further reductions using the strategy do not drastically change the current status of computation. We believe this to be a natural assumption.
Computability: To provide an effective\textsuperscript{11} method for computing the strategy.

Good behavior: To provide evidence of some good computational property for the strategy, typically that $S$ is root-normalizing, normalizing, etc.

Efficiency: To ensure that computations achieved by using the strategy satisfy some criterion for efficiency; for instance, minimality of normalizing derivations.

For orthogonal TRSs, Huet and Levy’s notion of needed reduction provides a framework for defining normalizing strategies [HL79, HL91]. A needed redex in a term $t$ is a redex which must be reduced (either itself or some descendant) in any normalizing derivation starting from $t$ [HL91]. Reduction sequences which only contract needed redexes are called needed reductions. Neededness has two main theoretical aspects:

1. It formalizes a notion of efficiency for rewriting computations. If a normalizing derivation only contracts needed redexes, it can be considered to be the most efficient since no useless reductions are performed.\textsuperscript{12}

2. It allows the definition of normalizing strategies: needed reduction is normalizing.

For orthogonal TRSs, every reducible term contains a needed redex [HL91]. Thus, rewriting strategies that only contract needed redexes actually exist.

As for unrestricted rewriting, we can similarly consider the notion of a context-sensitive rewriting strategy.

**Definition 2 (Context-sensitive rewriting strategy).** Let $\mathcal{R}$ be a TRS and $\mu \in M_{\mathcal{R}}$. A (nondeterministic) context-sensitive rewriting strategy (or just $\mu$-strategy) for $\mathcal{R}$ is a function $H$ that assigns a nonempty set of nonempty finite $\mu$-rewrite sequences each beginning with $t$ to every $\mu$-reducible term $t$. A one-step $\mu$-strategy is a function $H$ that assigns a nonempty set $H(t) \subseteq \mathcal{P}o_{\mathcal{R}}(t)$ to every $\mu$-reducible term $t$.

We write $t \rightarrow_{H} s$ if $H(t)$ contains a $\mu$-reduction sequence ending with $s$ (or $t \rightarrow^p s$ and $p \in H(t)$ for one-step $\mu$-strategies). For a given $\mu$-strategy $H$, an $H$-sequence is a $\mu$-reduction sequence of the form $t_1 \rightarrow_{H} t_2 \rightarrow_{H} \cdots$. A finite $H$-sequence $t_1 \rightarrow_{H} t_2 \rightarrow_{H} \cdots \rightarrow_{H} t_n$ is maximal if $t_n$ is a $\mu$-normal form. Note that, by using $\mu$-strategies, $\mu$-normal forms cannot be further reduced. Thus, whenever $\mu \neq \mu_{\top}$, a $\mu$-strategy is not necessarily a rewriting strategy.

A $\mu$-strategy $H$ is root-normalizing if, for all root-normalizing terms $t$, every possible maximal or infinite $H$-sequence starting from $t$ contains a root-stable term; $H$ is $\mu$-normalizing if, for all $\mu$-normalizing terms $t$, there is no infinite $H$-sequence starting from $t$.

**Remark 3.** Dealing with $\mu$-strategies $H$, we need to clarify that maximal $H$-sequences should contain root-stable terms in order to keep the natural assumption that root-normalizing $\mu$-strategies generate rewriting sequences that always compute a root-stable term of the initial (root-normalizing) term. The reason is that, since a $\mu$-strategy $H$ is not forced to reduce beyond a $\mu$-normal form, it is possible that finite maximal $H$-sequences contain no root-stable term. In principle, only if $\mu \in CM_{\mathcal{R}}$ $\mu$-normal forms are guaranteed to be root-stable (for left-linear TRSs, see Theorem 4).

The advantage of computing $\mu$-normal forms is that, in contrast to root-stable terms, they are decidable, i.e., it is decidable whether a term is a $\mu$-normal form or not (at least for finite TRSs). We have an immediate property of $\mu$-strategies.

**Theorem 6 (Root-normalization via $\mu$-normalization).** Let $\mathcal{R}$ be a left-linear TRS and $\mu \in CM_{\mathcal{R}}$. If $\mathcal{R}$ is $\mu$-normalizing, then every $\mu$-normalizing $\mu$-strategy is root-normalizing.

**Proof.** Let $H$ be a $\mu$-normalizing $\mu$-strategy for $\mathcal{R}$. Since $\mathcal{R}$ is $\mu$-normalizing, every term has a $\mu$-normal form. Since $H$ is $\mu$-normalizing, no term initiates an infinite $H$-sequence. Maximal (finite) $H$-sequences end in a $\mu$-normal form which, by Theorem 4, is root-stable. \hfill $\blacksquare$

\textsuperscript{11} Here, “effective” is used in the sense of [KM91]: a reduction strategy $S$ is effective if $S(t)$ can be computed from every term $t$.

\textsuperscript{12} Actually, this is only true if we consider a graph-based implementation which stores different occurrences of the same redex in a single, shared location (see [O’Do95, OV02]).
Every $\mu$-strategy is $\mu$-normalizing for $\mu$-terminating TRSs. Then, we have:

**Corollary 3.** Let $R$ be a left-linear TRS and $\mu \in \text{CM}_R$. If $R$ is $\mu$-terminating, then every $\mu$-strategy is root-normalizing.

In general, we cannot extend Theorem 6 to non-$\mu$-normalizing TRSs.

**Example 22.** Consider the following (orthogonal) TRS $R$:

$$f(x) \rightarrow g(b)$$
$$b \rightarrow b.$$

Let $\mu(f) = \mu(g) = \{1\}$. Note that terms $f(t)$ for arbitrary terms $t$ have no $\mu$-normal form. Thus, a $\mu$-strategy $H$ that always reduces inner redexes in terms $f(t)$ is $\mu$-normalizing but it is not root-normalizing: we have

$$f(b) \xrightarrow{H} f(b) \xrightarrow{H} \cdots$$

while the reduction step

$$f(b) \rightarrow g(b)$$

root-normalizes $f(b)$.

Example 22 also shows that confluence (or orthogonality) does not help (in general) to improve the previous results.

Every one-step $\mu$-strategy $H$ extends to a one-step strategy $S_H$ as follows:

$$S_H(t) = \begin{cases} 
H(t) & \text{if } t \notin \text{NF}_R^\mu \\
\bigcup_{1 \leq i \leq n} S_H(t_i) & \text{otherwise, where:} \\
C[ ] = MRC^\mu(t), t = C[t_1, \ldots, t_n], \\
\text{and } t_i = t|_{p_i} \text{ for } 1 \leq i \leq n. 
\end{cases}$$

The generalization to arbitrary $\mu$-strategies $H$ is immediate:

$$S_H(t) = \begin{cases} 
H(t) & \text{if } t \notin \text{NF}_R^\mu \\
C[S_H(t_1), \ldots, S_H(t_n)] & \text{if } t \in \text{NF}_R^\mu \setminus \text{NF}_R, \text{ where} \\
\emptyset & \text{otherwise} 
\end{cases}$$

where $C[ ]$ is the rewrite context and $S_1, \ldots, S_n$ are the sets of rewrite sequences from $C[t_1, \ldots, t_n]$ to $C[s_1, \ldots, s_n]$, where, for $1 \leq i \leq n$, either $s_i = t_i$ (during the whole sequence) or $s_i$ is the end point of a sequence in $S_i$ (and at least one of the $s_i$ must be taken in this way). Moreover, if $C[u_1, \ldots, u_n] \rightarrow C[u'_1, \ldots, u'_n]$ is a single rewriting step of one of such sequences, then there exists $1 \leq i \leq n$ such that $u_i \rightarrow u'_i$ is a rewriting step (issued inside) of a sequence in $S_i$ and $u_j = u'_j$ for $j \neq i$.

We have the following property.

**Proposition 15.** Let $R$ be a left-linear TRS and $\mu \in \text{CM}_R$. If $H$ is a $\mu$-normalizing (resp. root-normalizing) $\mu$-strategy, then $S_H$ is $\mu$-normalizing (resp. root-normalizing).

**Proof:** By Proposition 14, if $S_H$ is not $\mu$-normalizing (root-normalizing), then there exists an infinite $S_H$-sequence

$$t_1 \rightarrow s_n t_2 \rightarrow s_n \cdots$$
issued from a $\mu$-normalizing (root-normalizing) term $t_1$ which does not contain a $\mu$-normal form (root-stable term). By definition of $S_H$ (and using the fact that whenever the $S_H$-sequence does not contain root-stable terms then it does not contain $\mu$-normal forms, see Theorem 4), this corresponds to an infinite $H$-sequence

$$t_1 \hookrightarrow_H t_2 \hookrightarrow_H \cdots$$

which contradicts that $H$ is $\mu$-normalizing (root-normalizing). $\blacksquare$

6. DEFINITION OF $\mu$-NORMALIZING $\mu$-STRATEGIES

Since $\mu$-normal forms do not need to be normal forms, needed reduction is not useful for defining $\mu$-normalizing $\mu$-strategies. This is because, by definition, every redex in a term having no (finite) normal form is a needed redex.

**Example 23.** Consider the following TRS $R$ [Mid97]:

$$f(x) \rightarrow g(f(x))$$
$$b \rightarrow b.$$ 

If $\mu(f) = \{1\}$ and $\mu(g) = \emptyset$, then $g(f(b))$ is a $\mu$-normal form of $f(b)$:

$$f(b) \hookrightarrow g(f(b)).$$

However, $f(b)$ has no normal form. Thus, redex $b$ of $f(b)$ is needed but repeated $\mu$-reductions of this redex

$$f(b) \hookrightarrow f(b) \hookrightarrow \cdots$$

do not $\mu$-normalize $f(b)$.

Theorem 4 shows that $\mu$-normal forms of a left-linear TRS $R$ are root-stable (if $\mu \in CM_R$). As it is possible to normalize a term $t$ by successively root-normalizing maximal non-root-stable subterms of (reducts of) $t$, we can think of root-normalization [Ken94, Mid97] as a basis for defining $\mu$-normalizing computations, as every derivation leading to a $\mu$-normal form yields a root-stable term in some step of the derivation. In fact, according to Corollary 2, whenever $\mu \in CM_R$, the $\mu$-normalization of a term $t$ can be thought of as a preliminary root-normalization of $t$ that obtains a root-stable term $s = f(\bar{s})$ followed by the $\mu$-normalization of the replacing arguments $s_i$, for $i \in \mu(f)$, of $s$. Root-normalizing strategies do not need to be normalizing.

**Example 24.** Consider the TRS $R$ [Mid97]

$$f(x) \rightarrow g(a)$$
$$b \rightarrow b$$

and the strategy $S$ that always selects the (unique) outermost redex, except when it faces the term $g(f(b))$ in which case the redex $b$ is selected. Clearly, $S$ is root-normalizing for $R$. However, it is not normalizing, because we have the infinite $S$-reduction sequence

$$g(f(b)) \rightarrow g(f(b)) \rightarrow \cdots.$$ 

However, $g(f(b))$ has a normal form which can be computed by

$$g(f(b)) \rightarrow g(g(a)).$$

Middeldorp pointed out a solution to this problem based on using special root-normalizing strategies, namely context-free strategies.
Definition 3 (Context-free strategy) [Mid97]. A one-step strategy \( S \) is context-free if, for all root-stable terms \( t = f(t_1, \ldots, t_k) \) and \( i \in \{1, \ldots, k\} \), such that \( t \rightarrow_S f(t_1, \ldots, t'_i, \ldots, t_k) \), we have \( t_i \rightarrow_S t'_i \).

For arbitrary strategies \( S \), context-freeness is defined as follows [Mid97]: \( S \) is context-free if for all root-stable terms \( t = f(t_1, \ldots, t_k) \) and \( i \in \{1, \ldots, k\} \) such that \( t \rightarrow_S f(t'_1, \ldots, t'_i, \ldots, t'_k) \) and the subsequence from \( t_i \) to \( t'_i \) is nonempty, we have \( t_i \rightarrow_S t'_i \).

Theorem 7 [Mid97]. Let \( \mathcal{R} \) be a confluent TRS. Every context-free root-normalizing reduction strategy for \( \mathcal{R} \) is normalizing.

On the basis of a similar result by Middeldorp (see [Mid99]), we can even improve Theorem 7. In the following result, we say that the strategy \( S' \) extends the strategy \( S \) if for all terms \( t, s, t \rightarrow_S s \) implies \( t \rightarrow_{S'} s \).

Corollary 4. Let \( \mathcal{R} \) be a confluent TRS. Every reduction strategy for \( \mathcal{R} \) that can be extended to a context-free root-normalizing reduction strategy for \( \mathcal{R} \) is normalizing.

Proof. Let \( S \) be a reduction strategy for \( \mathcal{R} \) which can be extended to a context-free root-normalizing strategy \( S' \) for \( \mathcal{R} \). If \( S \) is not normalizing, then there exists an infinite reduction \( S \)-sequence \( t \rightarrow_S t' \rightarrow_S \cdots \) for a normalizing term \( t \). Since \( S' \) extends \( S \), there is an infinite reduction \( S' \)-sequence, thus contradicting Theorem 7.

Nevertheless, root-normalizing, context-free rewriting strategies do not need to be \( \mu \)-normalizing (even for confluent TRSs).

Example 25. Consider the (orthogonal, hence confluent) TRS \( \mathcal{R} \)

\[
f(x) \rightarrow c(x, a) \quad b \rightarrow b \\
a \rightarrow d
\]

Together with \( \mu(f) = \emptyset \) and \( \mu(c) = \{2\} \). Note that \( \mu \in CM_\mathcal{R} \). Consider the rewriting strategy \( S \) that contracts the leftmost-outermost redex of the leftmost maximal non-root-stable subterm of a term. The \( S \)-sequence

\[
\\overline{f(b)} \rightarrow_S c(\overline{b}, a) \rightarrow_S c(\overline{b}, a) \rightarrow_S \cdots
\]

does not compute the \( \mu \)-normal form \( c(b, d) \) which can be obtained by the \( \mu \)-reduction sequence

\[
\\overline{f(b)} \leftarrow_c c(b, a) \leftarrow_c c(b, d).
\]

If we restrict ourselves to reduction sequences that contract replacing redexes, i.e., to context-sensitive strategies, we can prove that root-normalization is an adequate basis for \( \mu \)-normalization. Note that, in the realm of CSR, the notion of one-step context-free \( \mu \)-strategy could be equivalently formulated by imposing the condition that for all root-stable terms \( t = f(t_1, \ldots, t_k) \) and \( i \in \mu(f) \), such that \( t \leftarrow_H f(t_1, \ldots, t'_i, \ldots, t_k) \), we have \( t_i \leftarrow_H t'_i \) (with the analogous generalization for arbitrary \( \mu \)-strategies).

Theorem 8. Let \( \mathcal{R} \) be a left-linear, confluent TRS and \( \mu \in CM_\mathcal{R} \). Every context-free root-normalizing reduction \( \mu \)-strategy for \( \mathcal{R} \) is \( \mu \)-normalizing.

Proof. Let \( \mathcal{H} \) be a context-free root-normalizing reduction \( \mu \)-strategy and let \( t \) be a \( \mu \)-normalizing term having a \( \mu \)-normal form \( s \). We proceed by induction on the structure of \( C[\ ] = MRC^\mu(s) \), the maximal replacing context of \( s \) (by Theorem 5, \( C[\ ] \) does not depend on the selected \( \mu \)-normal form \( s \)).

If \( s \) is a constant or a variable, then \( s \) is the (unique) normal form of \( t \). Thus, \( s = C[\ ] \) is the unique root-stable reduct of \( t \). Since \( \mathcal{H} \) is root-normalizing, there is no infinite \( H \)-sequence starting from \( t \).

If \( s = f(s_1, \ldots, s_k) \), assume that \( A : t = t_1 \leftarrow_H t_2 \leftarrow_H \cdots \) is infinite. Since \( \mathcal{H} \) is root-normalizing, \( A \) contains a root-stable term \( t' = f(t'_1, \ldots, t'_j) \); i.e., \( t' = t_j \) for some \( j \geq 1 \). By Proposition 13, \( t' \) also has a \( \mu \)-normal form \( s' \) which, by Theorem 5, has the same maximal replacing context,
CONCEPTUAL REWRITING STRATEGIES

Let \( C[s] = MRC^\mu(s) = MRC^\mu(s') \). Since \( t' \) is root-stable, it must be \( s' = f(s'_1, \ldots, s'_i) \); hence, we can write \( C[s] = f(C[s'_1], \ldots, C[s'_i]) \). Moreover, \( t'_i \leftarrow s'_i \), i.e., each \( t'_i \mu \)-normalizes in \( s'_i \), and \( C[s'_i] = MRC^\mu(s'_i) \) for each \( i \in \mu(f) \). Since \( A \) is infinite and context-free, it must be \( i \in \mu(f) \) such that \( t'_i \) initiates an infinite \( H \)-sequence. This contradicts the (induction) hypothesis that \( H \) is \( \mu \)-normalizing on \( t'_i \). 

**THEOREM 9** [Mid97]. Let \( t \) be a root-normalizing term. There are no parallel rewrite sequences starting from \( t \) that contain \( \mu \)-normalizing \( \mu \)-strategies.

**EXAMPLE 26.** Consider the TRS \( \mathcal{R} \)

\[
g(a, x, y) \rightarrow x \quad b \rightarrow a \\
g(d, x, a) \rightarrow a \quad c \rightarrow d
\]

and the set of all root-normalizing derivations for \( t = g(a, b, c) \):

1. \( g(a, b, c) \rightarrow g(a, a, c) \rightarrow a \)
2. \( g(a, b, c) \rightarrow b \rightarrow a \)
3. \( g(a, b, c) \rightarrow g(a, a, c) \rightarrow g(a, a, d) \rightarrow a \)
4. \( g(a, b, c) \rightarrow g(a, b, d) \rightarrow g(a, a, d) \rightarrow a \)
5. \( g(a, b, c) \rightarrow g(a, b, d) \rightarrow b \rightarrow a. \)

Note that redex \( b \) at position 2 of \( t \) is root-needed. On the other hand, the redex \( c \) in \( t \) is not root-needed as the first derivation does not contract it.

For orthogonal TRSs, every non-root-stable term contains a root-needed redex. Root-stable terms have no root-needed redex, and redexes in terms having no root-stable reduct are trivially root-needed. Root-needed redexes in maximal non-root-stable subterms of a term are needed.

A **root-necessary** set of redexes is a set of redexes such that at least one of the redexes in the set, or one of its descendants, is reduced in each root-normalizing derivation. The repeated contraction of root-necessary sets of redexes is called **root-necessary reduction** [Mid97].

**THEOREM 9** [Mid97]. Let \( t \) be a root-normalizing term. There are no parallel rewrite sequences starting from \( t \) that contain infinitely many root-necessary steps.

Root-necessary reduction is root-normalizing for almost orthogonal TRSs. In particular, repeated contraction of root-needed redexes (called **root-needed reduction**) is root-normalizing for orthogonal TRSs. Our aim is to use root-neededness for defining \( \mu \)-normalizing \( \mu \)-strategies. According to the four points enumerated in Section 5, we first address the problem of proving the existence of such strategies.

Our first result corresponds to Theorem 4.3 in [Mid97]: “For orthogonal TRSs, every non-root-stable term has a root-needed redex.” We prove that, more precisely, every non-root-stable term has a replacing root-needed redex. First, we need some previous results.

**LEMMA 3** [Mid97]. Let \( \mathcal{R} \) be an orthogonal TRS. If a term \( t \) rewrites to a redex, then the pattern of the first such redex is unique.

**LEMMA 4.** Let \( \mathcal{R} \) be an orthogonal TRS. Let \( t \) be a term which is neither root-stable nor a redex, and such that \( t \xrightarrow{\Lambda^+} \sigma(l) \) for some \( l \in L(\mathcal{R}) \). Let \( P \) be the set of positions of non-root-stable proper
subterms of \( t \), \( p \in \text{minimal}(P \cap \mathcal{Pos}_\mathcal{F}(l)) \), and \( q \in \mathcal{Pos}_\mathcal{R}(t|_p) \) be a position of a root-needed redex of \( t|_p \). Then \( t|_{p,q} \) is a root-needed redex of \( t \).

**Proof.** Implicit in Middeldorp’s proof of Theorem 4.3 in [Mid97]. ■

**Lemma 5.** Let \( \mathcal{R} \) be an orthogonal TRS and \( \mu \in \text{CM}_\mathcal{R} \). Let \( t \) be a non-root-stable term such that \( t \xrightarrow{A}^+ \sigma(l) \) for some \( l \in L(\mathcal{R}) \). Let \( P \) be the set of positions of proper non-root-stable subterms of \( t \). Then, \( \text{minimal}(P \cap \mathcal{Pos}_\mathcal{F}(l)) \subseteq \mathcal{Pos}_\mathcal{R}(t) \).

**Proof.** Assume \( p \in \text{minimal}(P \cap \mathcal{Pos}_\mathcal{F}(l)) \). Since \( t \xrightarrow{A}^+ \sigma(l) \), we have \( \text{spre}(p) = \text{spre}(p) \). Otherwise, some subterm \( t|_{q} \) with \( q < p \) should be reduced to allow the matching with \( l|_{q} \); thus \( t|_{q} \) would be non-root-stable and \( q \in P \). Moreover, since \( p \in \mathcal{Pos}_\mathcal{F}(l) \), \( q \in P \cap \mathcal{Pos}_\mathcal{F}(l) \), but, since \( q < p \), \( p \) is not minimal in \( P \cap \mathcal{Pos}_\mathcal{F}(l) \) and this leads to a contradiction. Since \( \text{spre}(p) = \text{spre}(p) \) and \( \mu \in \text{CM}_\mathcal{R} \), we have \( p \in \mathcal{Pos}_\mathcal{R}(l) \subseteq \mathcal{Pos}_\mathcal{R}(t) \). By Proposition 6, \( p \in \mathcal{Pos}_\mathcal{R}(t) \). ■

**Theorem 10.** Let \( \mathcal{R} \) be an orthogonal TRS and \( \mu \in \text{CM}_\mathcal{R} \). Every non-root-stable term has a \( \mu \)-replacing root-needed redex.

**Proof.** We follow the proof of Theorem 4.3 in [Mid97]. Therefore, we just outline the proof and comment on the particulars of context-sensitive rewriting.

Middeldorp proceeds by induction (on the depth of redexes) and distinguishes between \( t \) being a redex and \( t \) being a nonredex. The first case is the same for us, since we always have \( \Delta \in \mathcal{Pos}_\mathcal{R}(t) \) and thus \( \Delta \in \mathcal{Pos}_\mathcal{R}(t) \) which, for (almost) orthogonal TRSs, corresponds to a root-needed redex. For the second one, any rewrite sequence \( A \) leading from \( t \) to a root-stable form \( t' \) splits into \( t \xrightarrow{\Delta} \sigma(l) \xrightarrow{\Delta} t'' \rightarrow^+ t' \). The redex pattern \( l \) for the reduction \( \sigma(l) \rightarrow t'' \) does not depend on the particular derivation (Lemma 3). Now let us consider the set \( P \) of occurrences of non-root-stable proper subterms of \( t \). The candidates to root-needed redexes are root-needed redexes of subterms \( t|_{p} \) with \( p \in \text{minimal}(P \cap \mathcal{Pos}_\mathcal{F}(l)) \), where \( P \) is as in Lemma 5. We must first prove that \( p \in \mathcal{Pos}_\mathcal{R}(t) \). This follows immediately from Lemma 5.

By I.H., \( t|_{p} \) has a replacing root-needed redex \( s \). By Lemma 4, \( s \) is a root-needed redex of \( t \). Since \( s \) is \( \mu \)-replacing in \( t|_{p} \) and \( p \in \mathcal{Pos}_\mathcal{R}(t) \), by Proposition 4, \( s \) is a \( \mu \)-replacing root-needed redex of \( t \). ■

Note that \( \mu^{\text{can}} \) does not completely capture root-neededness.

**Example 27.** Consider again the TRS \( \mathcal{R} \) and the derivations of Example 26. As shown in the example, redex \( b \) at the occurrence 2 of \( t = g(a, b, c) \) is root-needed. Since \( 2 \notin \mathcal{Pos}_\mathcal{R}^{\text{can}}(g(a, b, c)) = \{ \Lambda, 1, 3 \} \), this redex is not \( \mu^{\text{can}} \)-replacing. Thus, there are root-needed redexes that are not \( \mu^{\text{can}} \)-replacing.

Example 26 also shows that the replacing redex \( c \) in \( t \) is not root-needed. Therefore, there are \( \mu^{\text{can}} \)-replacing redexes that are not root-needed.

Theorem 10 establishes the possibility of defining the following one-step \( \mu \)-strategy for a given orthogonal TRS \( \mathcal{R} \) and whenever \( \mu \in \text{CM}_\mathcal{R} \):

\[
H_{\text{need}}(t) = \begin{cases}
\{ p \in \mathcal{Pos}_\mathcal{R}^{\sigma}(t) | t|_{p} \text{ is root-needed in } t \} & \text{if } t \text{ is not root-stable} \\
\bigcup_{i \in \mu(f)} i \cdot H_{\text{need}}(t_i) & \text{if } t = f(\bar{t}) \text{ is root-stable} \\
\emptyset & \text{if } t \text{ is a variable.}
\end{cases}
\]

By construction, \( H_{\text{need}} \) is context-free and Theorem 8 can be invoked to justify its \( \mu \)-normalizing character. Therefore, every orthogonal TRS admits a one-step \( \mu \)-normalizing \( \mu \)-strategy.

Unfortunately, the definition of \( H_{\text{need}} \) is not effective since root-neededness and root-stability are undecidable. Thus, we are interested in establishing additional conditions which enable the effective selection of a (replacing) root-needed redex or a subset of replacing redexes \( I \subseteq \mathcal{Pos}_\mathcal{R}^{\sigma}(t) \) which is root-necessary. We address the first problem in Section 7. With regard to the second problem, we can refine Middeldorp’s result: “outermost redexes are a root-necessary set of redexes.” Again, we can restrict to outermost replacing redexes.

**Theorem 11.** Let \( \mathcal{R} \) be an orthogonal TRS and \( \mu \in \text{CM}_\mathcal{R} \). If \( t \) is not root-stable, then minimal \( (\mathcal{Pos}_\mathcal{R}^{\sigma}(t)) \) is a root-necessary set of redexes.
The following section is devoted to the départicular case of NV-sequentiality, which is the most general approximation to root-neededness. We have demonstrated that, among them, NV-sequentiality \cite{Oya93} (hence strong sequentiality \cite{HL91}, Jac96, JS94, KM91, NST95, NT99, Oya93, TKS00, Toy92). Recently, we investigated the class of TRSs ensuring that every reducible term has a redex for which the previous method succeeds. Necessary it is necessary (1) to provide a method to decide whether a redex is needed and (2) to identify the necessary set of redexes for all terms. Hence, \mu\text{-strategy} \(H_{po}\) is:

\[
H_{po}(t) = \{ t = t_1 \overset{p_1}{\rightarrow} t_2 \leftarrow \cdots \leftarrow t_n \overset{p_n}{\rightarrow} t_{n+1} \mid \{p_1, \ldots, p_n\} = \text{minimal}(\mathcal{Pos}_{\mu}(t)) \}.
\]

Note that, since \(p_i \parallel p_j\) for every \(1 \leq i < j \leq n\), every possible \(H_{po}\)-step \(t \leftarrow_{n_{po}} s\) can be thought of as a single step of parallel (context-sensitive) rewriting (see \cite{Luc98a}). For almost orthogonal TRSs, since \(p_i \parallel p_j\) for every \(1 \leq i < j \leq n\), every possible \(H_{po}\)-sequence issued from \(t\) leads to the same term \(s\). \(H_{po}\) is clearly context-free. Then, we have:

\textbf{Theorem 12.} Let \(\mathcal{R}\) be an orthogonal TRS and \(\mu \in \mathcal{CM}_{\mathcal{R}}\). Then, \(H_{po}\) is \(\mu\)-normalizing.

\textbf{Proof.} By Theorems 11 and 9, \(H_{po}\) is root-normalizing. Since \(H_{po}\) is context-free, by Theorem 8 the conclusion follows. \(\blacksquare\)

Theorem 12 provides a first effective example of a computable (parallel) \(\mu\)-normalizing \(\mu\)-strategy. The following section is devoted to the definition of one-step \(\mu\)-normalizing \(\mu\)-strategies.

\section*{7. EFFECTIVE DEFINITION OF ONE-STEP \(\mu\)-NORMALIZING \(\mu\)-STRATEGIES}

Both neededness and root-neededness are undecidable and they must be approximated. This means that it is necessary (1) to provide a method to decide whether a redex is needed and (2) to identify the class of TRSs ensuring that every reducible term has a redex for which the previous method succeeds \cite{DM97}. Decidable approximations to neededness have been extensively explored \cite{Com00, DM97, HL91, Jac96, JS94, KM91, NST95, NT99, Oya93, TKS00, Toy92}. Recently, we investigated the use of these approximations to capture root-neededness for \textit{almost orthogonal} TRSs \cite{Luc98b}. We have demonstrated that, among them, NV-sequentiality \cite{Oya93} (hence strong sequentiality \cite{HL91}, a particular case of NV-sequentiality) is the most general approximation to root-neededness.

\subsection*{7.1. Sequentiality}

Sequentiality is based on the notion of index. An \(\Omega\)-position \(p \in \mathcal{Pos}_{\Omega}(t)\) of an \(\Omega\)-term \(t \in \mathcal{T}_{\Omega}(\mathcal{F}, \mathcal{X})\) is an \textit{index} with respect to a predicate \(P\) on \(\Omega\)-terms if, for every \(\Omega\)-term \(s\) with \(s \geq t\), \(P(s)\) implies \(s|_p \neq \Omega\) \cite{KM91}. The set of indices of \(t\) with respect to \(P\) is denoted by \(\mathcal{I}_P(t)\). Given a term \(t \in \mathcal{T}(\mathcal{F}, \mathcal{X})\) (without \(\Omega\)-occurrences), we can test whether a redex position \(p \in \mathcal{Pos}_{\Omega}(t)\) “is an index” by applying the previous definition to \(r|_{\Omega}p\). A monotone predicate \(P\) is \textit{sequential} if, for all \(t \in \mathcal{T}_{\Omega}(\mathcal{F}, \mathcal{X})\), whenever \(P(t)\) does not hold and there exists \(s\) such that \(s \geq t\) and \(P(s)\) holds, it follows that \(\mathcal{I}_p(t) \neq \emptyset\).

A TRS \(\mathcal{R}\) is \textit{sequential} if the predicate \(\mathcal{n}_R^f\) on \(\Omega\)-terms (where \(\mathcal{n}_R^f(t)\) holds if and only if \(t\) has a normal form in \(\mathcal{T}(\mathcal{F}, \mathcal{X})\)) is sequential.

\textbf{Example 28.} Consider the TRS \(\mathcal{R}\) \cite{HL91}:

\[
\begin{align*}
f(a, b, x) & \rightarrow k \quad c \rightarrow a \quad f(b, x, a) & \rightarrow k \quad c \rightarrow b.
\end{align*}
\]

\(\mathcal{R}\) is \textit{not} sequential because the term \(t = f(c, \Omega, \Omega)\) has no index: Note that \(s = f(c, \Omega, a) > f(c, \Omega, \Omega)\) and \(f(c, \Omega, a)\) has a normal form (without \(\Omega\)'s).
Given a fresh symbol $\bullet$.

Note that $\text{NV-Sequentiality}$ 7.1.2. has been proven decidable for left-linear TRSs in [JS94]. The following properties are used later.

For orthogonal TRSs, sequential indices with respect to predicate $nf_R$ serve to approximate needed redexes of $t \in T(\mathcal{F}, \mathcal{X})$: if $p \in Pos_R(t)$ is an index of $t[\Omega]_p$ w.r.t. $nf_R$, then $t[p]$ is a needed redex of $t$ [HL91].

Both sequentiality of indices and that of TRSs are undecidable and several decidable approximations have been investigated. According to our discussion at the beginning of Section 7, we consider the strongly sequential and NV-sequential approximations.

### 7.1.1. Strong Sequentiality

Given a TRS $R$, the reduction relation $\rightarrow_\gamma$ (arbitrary reduction) on $T(\mathcal{F}, \mathcal{X})$ is defined as follows: $t \rightarrow_\gamma s$ if there are $p \in Pos_R(t)$ and $s'$ such that $s = t[s']_p$ [KM91]. Clearly, the $\rightarrow_\gamma$-normal forms and the normal forms coincide. A TRS $R$ is strongly sequential [HL91, KM91] if predicate $nf_s$ is sequential (where $nf_s(t)$ holds if there exists an arbitrary reduction sequence $t \rightarrow_\gamma s$ to some normal form $s \in T(\mathcal{F}, \mathcal{X})$). Indices with respect to predicate $nf_s$ are said to be strong indices (and they are sequential indices, i.e., indices of $nf_R$). The set of strong indices of a term $t$ is denoted by $I_s(t)$ (rather than $I_{nf_s}(t)$). Strong indices can also be effectively computed by using $\Omega$-reduction in Section 2.1: Given a fresh symbol $\bullet$ and $p \in Pos_R(t)$, we have that $p \in I_s(t)$ iff $\omega(t[\bullet]_p)_p = \bullet$ [KM91, Toy92]. In fact, we take this result as a (re-)definition of strong index [JS94, Toy92].

**Example 29.** Consider the TRS $R$:

$$
\begin{align*}
  f(x, a) &\rightarrow c \\
g(a, x) &\rightarrow c.
\end{align*}
$$

Note that $t = f(g(\Omega, x), g(\Omega, x))$ is an $\Omega$-normal form. Position 2.1 corresponds to a strong index, since the $\rightarrow_\Omega$-reduction step

$$f(g(\Omega, x), g(\bullet, x)) \rightarrow_\Omega f(\Omega, g(\bullet, x))$$

computes the $\rightarrow_\Omega$-normal form $\omega(t[\bullet]_{2.1})$ of $t[\bullet]_{2.1}$ (remember that $\rightarrow_\Omega$ is confluent) which does contain $\bullet$. Thus, 2.1 $\in I_s(t)$. However,

$$f(g(\bullet, x), g(\Omega, x)) \rightarrow_\Omega f(g(\bullet, x), \Omega) \rightarrow_\Omega \Omega;$$

that is, $1.1 \not\in I_s(t)$.

A TRS $R$ is strongly sequential if $I_s(t) \neq \emptyset$ for every $\Omega$-normal form $t$. Strong sequentiality has been proven decidable for left-linear TRSs in [JS94]. The following properties are used later.

**Proposition 16** [JS94, KM91]. Let $R = (\mathcal{F}, R)$ be a TRS. Let $t \in I_s(t)$. If $p, q \in I_s(t)$, then $q \in I_s(t[p])$. If $p \in I_s(t)$ and $t \leq s$, then $p \in I(s[\Omega]_p)$.

**Proposition 17** [Klo92, KM91]. Let $R = (\mathcal{F}, R)$ be a TRS. Let $t = C[t_1, \ldots, t_n]$ be such that $C[\Omega, \ldots, \Omega]$ is rigid and the $t_i$, $1 \leq i \leq n$ are soft. Let $t_i = C'[\Omega]_q = t'_p$. If $q \in I_s(t_i)$, then $p, q \in I_s(t)$.

### 7.1.2. NV-Sequentiality

Given a TRS $R = (\mathcal{F}, R)$, we write $t \rightarrow_m s$ if and only if there exist $p \in Pos(t)$ and $l \rightarrow r \in R$ such that $t[p] \geq l[q]$ and $s = t[r]_p$ for some $r'$ such that $r' \geq r_\Omega$. Note that $\rightarrow_m$ is $\rightarrow_\omega$-confluent. The predicate $\text{term}$ is defined as follows: $\text{term}(t)$ holds if and only if there exists $s \in T(\mathcal{F}, \mathcal{X})$ such that $t \rightarrow_m^* s$ (i.e., $\rightarrow_m$
succeeds in removing the \( \Omega \)-occurrences from \( t \)). Predicate term is clearly monotone. The set of indices in a term with respect to term is written \( \mathcal{I}_m(t) \) (rather than \( \mathcal{I}_{term}(t) \)). It is decidable (in polynomial time) whether or not a position \( p \in \mathcal{P}os_\Omega(t) \) is an \( n \)-index \[Oya93\].

A TRS \( \mathcal{R} \) is \( NV \)-sequential if \( \mathcal{I}_m(t) \neq \emptyset \) for every \( \Omega \)-normal form \( t \) \[Oya93\]; \( NV \)-sequentiality is decidable for left-linear TRSs. Strong and \( NV \)-sequentiality are related.

**Proposition 18** [Oya93]. Strong indices are \( n \)-indices.

**Proposition 19** [Oya93]. Strongly sequential TRSs are \( NV \)-sequential.

We write \( t \xrightarrow{\omega} s \) (or just \( t \xrightarrow{} s \)) if \( \exists p \in \mathcal{P}os(t) \) such that \( t|_p \neq \Omega, t|_p \uparrow l_\Omega \) for some rule \( l \rightarrow r \), and \( s = t[r_\Omega]_p \).

**Example 30.** Consider \( \mathcal{R} \) and \( t \) as in Example 29. We have the \( \xrightarrow{\omega} \)-reduction sequence

\[
\begin{align*}
f(g(\Omega, x), g(\Omega, x)) & \xrightarrow{\omega} f(g(\Omega, x), c) \xrightarrow{\omega} f(c, c)
\end{align*}
\]

that can be compared with the following \( \xrightarrow{\Omega} \)-reduction sequence:

\[
\begin{align*}
f(g(\Omega, x), g(\Omega, x)) & \xrightarrow{\Omega} f(g(\Omega, x), c) \xrightarrow{\Omega} f(c, c)
\end{align*}
\]

The following result connects \( \xrightarrow{\omega} \)-reduction (used for defining the notion of \( n \)-index) and \( \xrightarrow{\omega} \)-reduction (used for finding \( n \)-indices, see Lemma 7 below).

**Lemma 6** [Oya93]. Let \( t \xrightarrow{m} t' \) and \( s \leq t \) where \( t, t', s \in \mathcal{T}_{\Omega}(\mathcal{F}, \mathcal{X}) \). Then either \( s \leq t' \) or there exists \( s' \in \mathcal{T}_{\Omega}(\mathcal{F}, \mathcal{X}) \) such that \( s \xrightarrow{\omega} s' \) and \( s' \leq t' \). (This implies that \( \exists s' \) such that \( s \xrightarrow{\omega} s' \) and \( s' \leq t' \).)

Oyamaguchi characterizes \( n \)-indices as follows.

**Lemma 7** [Oya93]. Let \( t \in \mathcal{T}_{\Omega}(\mathcal{F}, \mathcal{X}) \) and \( p \in \mathcal{P}os_\Omega(t) \). Then \( p \notin \mathcal{I}_m(t) \) if and only if there exist \( q \in \mathcal{P}os(t), where q < p \), and \( s \in \mathcal{T}_{\Omega}(\mathcal{F}, \mathcal{X}) \) such that \( t[\bullet]_q \xrightarrow{\omega} s \) and \( s \uparrow l_\Omega \) for some \( l \in L(\mathcal{R}) \).

**Example 31.** Consider \( \mathcal{R} \) and \( t \) as in Example 29. Position 1.1 corresponds to an \( n \)-index, since \( t|_1 = g(\bullet, x) \) is not compatible with \( g(a, \Omega) \) and the only \( \xrightarrow{\omega} \)-reduction step which can be given on \( t = t|_\Lambda \) is

\[
\begin{align*}
f(g(\bullet, x), g(\Omega, x)) & \xrightarrow{\omega} f(g(\bullet, x), c)
\end{align*}
\]

where \( s = f(g(\bullet, x), c) \) is not compatible with \( f(\Omega, a) \). Thus, 1.1 \( \in \mathcal{I}_m(t) \). However, as shown in Example 29, 1.1 \( \notin \mathcal{I}_m(t) \).

The following properties are used below. They correspond to Proposition 16 for strong indices.

**Lemma 8** [Oya93]. If \( p, q \in \mathcal{I}_m(t) \), then \( q \in \mathcal{I}_m(t|_p) \).

**Lemma 9** [Oya93]. Let \( p, q \in \mathcal{P}os(t) \) be such that \( p \parallel q \). If \( p \in \mathcal{I}_m(t[\Omega]_q) \), then \( p \in \mathcal{I}_m(t) \).

The following lemma establishes that redexes placed on \( n \)-indices are always outermost when dealing with almost orthogonal TRSs.

**Lemma 10** [Luc98b]. Let \( \mathcal{R} \) be an almost orthogonal TRS and \( t \) be a term. Let \( p, q \in \mathcal{P}os_\mathcal{R}(t) \) such that \( p \neq q \). If \( p \in \mathcal{I}_m(t[\Omega]_p) \), then \( q \notin p \).

The following lemma proves that \( n \)-indices are preserved by \( \xrightarrow{\omega} \)-reductions on disjoint positions.

**Lemma 11.** Let \( \mathcal{R} \) be a TRS, \( t \in \mathcal{T}_{\Omega}(\mathcal{F}, \mathcal{X}), \) and \( p, q \in \mathcal{P}os(t) \), where \( p \parallel q \) and \( p \in \mathcal{I}_m(t) \). If \( t \xrightarrow{q} t' \xrightarrow{\omega} t'' \), then \( p \in \mathcal{I}_m(t') \).

**Proof.** Note that, since \( t|_p = \Omega \) and \( p \parallel q \), we have that \( t'|_p = \Omega \). If \( p \notin \mathcal{I}_m(t') \), by Lemma 7 there exist \( p' < p \) and \( s \neq \Omega \) such that \( t'[\bullet]_{p'} \xrightarrow{\omega} s \) and \( s \uparrow l_\Omega \) for some \( l \in L(\mathcal{R}) \). Note that, since \( p \parallel q \),
we have that \( q \not\leq p' \). If \( p' \parallel q \), then \( t'[\bullet]_p[p'] = t[\bullet]_p[p'] \). If \( p' < q \), then \( t[\bullet]_p[p'] \rightarrow_\omega t'[\bullet]_p[p'] \). In both cases, we have that \( t[\bullet]_p[p'] \rightarrow_\omega \), thus contradicting that \( p \in \text{I}_m(t) \). \( \blacksquare \)

Lemma 11 does not hold if \( p \) and \( q \) are comparable.

Example 32. Consider the TRS \( \mathcal{R} \):

\[
\begin{align*}
\text{f}(x,a) & \rightarrow c & \text{h}(x) & \rightarrow x \\
\text{g}(a,x) & \rightarrow c & b & \rightarrow c.
\end{align*}
\]

Note that 1.1 is an \( m \)-index of \( t = \text{f}(\text{g}(\Omega,x),b) \) (proceed as in Example 31). However, we have that

\[
\text{f}(\text{g}(\Omega,x),b) \rightarrow_\omega \text{f}(c,x)
\]

and 1.1 is not an \( m \)-index of \( \text{f}(c,x) \).

The following result ensures that \( m \)-indices of an \( \Omega \)-term are preserved under arbitrary reductions.

Lemma 12. Let \( \mathcal{R} \) be a TRS, \( t \in \text{T}_\omega(\mathcal{F}, \mathcal{X}) \), and \( p \in \text{I}_m(t) \). If \( t \not\rightarrow \), then \( p \in \text{I}_m(t) \).

Proof. Since \( t[p] = \Omega \), we have that \( q \not\leq p \). Moreover, \( q \not\leq p \); otherwise, since there is no occurrence of \( \Omega \) in rules of \( \mathcal{R} \), if \( t[q] \) is a redex then \( t[\bullet]_p[q] \) is also, i.e., there is \( l \in \text{L}(\mathcal{R}) \) and substitution \( \sigma \) such that \( t[\bullet]_p[q] = \sigma(l) \). Therefore, \( t[\bullet]_p[q] \uparrow \text{L} \). By Lemma 7, this contradicts that \( p \in \text{I}_m(t) \).

Therefore, \( p \parallel q \) holds. Note that \( t[\bullet]_p = \Omega \). Since \( \rightarrow \subseteq \rightarrow_\omega \), by Lemma 6, either \( t \leq t' \) or \( t \rightarrow_\omega t'' \) and \( t'' \leq t' \). In the first case, by Lemma 9, \( p \in \text{I}_m(t') \). In the second case, by Lemma 11, \( p \in \text{I}_m(t'') \) and by Lemma 9, \( p \in \text{I}_m(t') \). Thus, the conclusion follows. \( \blacksquare \)

### 7.2. NV-Indices and Replacement Restrictions

According to our previous discussion in Section 6, our interest in NV-sequentiability stems from the following result.

Theorem 13 [Luc98b]. Let \( \mathcal{R} = (\mathcal{F}, \mathcal{X}) \) be an almost orthogonal TRS, \( t \in \text{T}(\mathcal{F}, \mathcal{X}) \) be non-root-stable, and \( p \in \text{Pos}_R(t) \). If \( p \in \text{I}_m(t[\Omega]_p) \), then \( t[p] \) is root-needed.

In order to distinguish between replacing and nonreplacing indices of a term \( t \), we denote replacing indices by \( \text{I}_m^p(t) = \text{I}_m(t) \cap \text{Pos}_p^\omega(t) \) for \( \alpha \in \{s, m\} \). Our first result establishes that the canonical replacement map captures all \( m \)-indices in non-root-stable terms.

Theorem 14. Let \( \mathcal{R} = (\mathcal{F}, \mathcal{X}) \) be an almost orthogonal TRS and \( \mu \in \text{CM}_R \). If \( t \in \text{T}_\omega(\mathcal{F}, \mathcal{X}) \) is not root-stable and \( p \in \text{Pos}_2(t) \), then \( p \in \text{I}_m(t[\Omega]_p) \iff p \in \text{I}_m^p(t[\Omega]_p) \).

Proof. Since \( \text{I}_m^p(t[\Omega]_p) \subseteq \text{I}_m(t[\Omega]_p) \), we only need to prove that \( p \in \text{I}_m(t[\Omega]_p) \) implies \( p \in \text{Pos}_p^\omega(t[\Omega]_p) \). If \( p \in \text{Pos}_p^\omega(t[\Omega]_p) \), we note that \( p \neq \Lambda \). Since \( t \) is not root-stable, it rewrites to a redex of \( l \in \text{L}(\mathcal{R}) \). By Proposition 7 there is a \( \mu \)-derivation \( A \)

\[
t = t_1 \overset{p_1}{\rightarrow} t_2 \leftrightarrow \cdots \leftrightarrow t_{n-1} \overset{p_{n-1}}{\rightarrow} t_n = \sigma(l)
\]

for some substitution \( \sigma \). We can assume that, for all \( p_i, 1 \leq i \leq n - 1 \), \( p_i \neq \Lambda \); i.e., \( \mu \)-rewriting steps from \( t \) to \( \sigma(l) \) only contract inner positions. We prove, by induction on the length of \( A \), that for all \( 1 \leq i \leq n \), \( p_i \in \text{I}_m(t_1[\Omega]_p) \) and \( p_i \notin \text{Pos}_p(t_1) \). For the base case, consider that if \( t = \sigma(l) \), since \( p \neq \Lambda \), we obtain a contradiction of Lemma 10. For the induction step, first we note that, since \( p_1 \in \text{Pos}_p(t_1) \) and \( p \notin \text{Pos}_p(t_1) \), by Proposition 5, \( p \neq p_1 \); moreover, \( p_1 \neq p \); otherwise we contradict Lemma 10. Thus, \( p_1 \parallel p \). Hence \( t_1 \overset{p_1}{\rightarrow} t_2 \) implies \( t_1[\Omega]_p \overset{p_1}{\rightarrow} t_2[\Omega]_p \). By Lemma 12, \( p \in \text{I}_m(t_2[\Omega]_p) \). Since \( p_1 \parallel p \), by Proposition 6, \( p \in \text{Pos}_p(t_2) \). By the induction hypothesis, the conclusion follows. Hence, in particular, \( p \in \text{Pos}_p^\omega(\sigma(l)) \) and \( p \in \text{I}_m(\sigma(l)[\Omega]_p) \). Since \( \sigma(l) \) itself is a redex, this contradicts Lemma 10. \( \blacksquare \)

In general, Theorem 14 does not hold for root-stable terms.
Example 33. Consider $\mathcal{R}$ and $t$ as in Example 32. Note that $t$ is root-stable. Recall that 1.1 is an $n$-index of $t$. However, 1.1 is a nonreplacing position of $t$: $1.1 \in P_{\text{os}^\Omega(t)}(t)$.

Corollary 6. Let $\mathcal{R} = (\mathcal{F}, R)$ be an almost orthogonal TRS and $\mu \in CM_\mathcal{R}$. If $t \in T_\Omega(\mathcal{F}, \mathcal{X})$ is not root-stable, then $T_{\mathcal{R}}(t) = T_{\mathcal{R}}^\mu(t)$. 

Proof. If $p \in T_{\mathcal{R}}(t)$, then, since $t \subseteq t[R]$, by (repeated application of) Lemma 9, $p \in T_{\mathcal{R}}(t[\Omega])$. By Theorem 14, $p \in T_{\mathcal{R}}^\mu(t[\Omega])$; i.e., $p \in P_{\text{os}^\Omega(t[\Omega])}$. By Proposition 6, $p \in P_{\text{os}^\Omega(t)}$; hence $p \in T_{\mathcal{R}}^\mu(t)$. 

Remark 4. Note the usefulness of these results: By Theorem 14, only $\mu_{\mathcal{R}}$-replacing $n$-indices are considered in non-root-stable terms $t$ (even without an explicit consideration of replacement restrictions). Hence, we only need to test redex occurrences inside $MRC_{\mathcal{R}}^\Omega(t)$ of a non-root-stable term $t$ for finding out $n$-indices.

7.3. Strong Indices and Replacement Restrictions

Strong indices are $n$-indices (Proposition 18) and strongly sequential TRSs are NV-sequential (Proposition 19). Hence, the previous results also apply to strong indices and strongly sequential TRSs. Nevertheless, for strong indices, we can improve Theorem 14 and Corollary 6 as follows.

Lemma 13. Let $\mathcal{R}$ be a TRS and $\mu \in M_\mathcal{R}$. If $t_1 \xrightarrow{p_1} t_2 \xrightarrow{p_2} \cdots \xrightarrow{p_n} t_n \xrightarrow{p_n} t_{n+1}$, then for all $i, j, 1 \leq i \leq n, j \leq i$, $\text{Pos}(t_i) \subseteq \text{Pos}(t_j)$ and $\text{Pos}^\mu(t_i) \subseteq \text{Pos}^\mu(t_j)$.

Proof. By induction on the length of the derivation. If $n = 1$, then $t_1 = C[s_1][p_1]$, $t_2 = C[\Omega][p_1]$, and $s \uparrow l_2$ for some $l \in L(R)$. Clearly $\text{Pos}(t_2) \subseteq \text{Pos}(t_1)$ and, since $\text{Pos}^\mu(t_1)$ is downward closed in $(\text{Pos}(t_1), \leq)$, $\text{Pos}^\mu(t_2) \subseteq \text{Pos}^\mu(t_1)$. By I.H., the conclusion follows.

The next result establishes that, whenever we deal with a soft term, we can define an $\Omega$-reduction sequence which only considers compatible terms at replacing positions; we write $t \xrightarrow{\Omega} \Omega$ if $t \xrightarrow{p} s$ and $p \in P_{\text{os}^\Omega(t)}$.

Proposition 20. Let $\mathcal{R}$ be a TRS and $\mu \in CM_\mathcal{R}$. If $t \xrightarrow{\Omega} \Omega$, then $t \xrightarrow{\Omega} \Omega$ in at most the same number of steps.

Proof. We proceed by induction on the length of a derivation $t = t_1 \xrightarrow{p_1} t_2 \xrightarrow{p_2} \cdots \xrightarrow{p_n} t_n \xrightarrow{p_n} \Omega$. If $n = 1$, then $t \xrightarrow{\Omega} \Omega$. Since $\Lambda \in P_{\text{os}^\Omega(t)}$, it is immediate. If $n > 1$, let us consider the first step $t_i \xrightarrow{p_i} t_{i+1}$ such that $p_i \in P_{\text{os}^\mu(t_i)}$. Since $\Lambda \in \{p_1, \ldots, p_n\}$ such a step will actually be performed. Let $P_i = \{p_j | p_i < p_j, 1 \leq j < i\}$. This set is well defined because, by Lemma 13, $p_i \in P_{\text{os}^\mu(t_i)}$ for all $j < i$. If $P_i = \emptyset$, then, since $p_i \in P_{\text{os}^\mu(t_i)}$ and $p_i \in P_{\text{os}^\mu(t_j)}$, for $1 \leq j < i$, by Proposition 5, $p_i \parallel p_j$ for all $1 \leq j < i$. Then, we can perform a $\Rightarrow \Omega$-reduction directly on $t_i[p_i]$. Moreover, by Lemma 13, $p_i \in P_{\text{os}^\mu(t_i)}$. Hence, we build the derivation

$$t = t_1 \xrightarrow{p_1} t_1' \xrightarrow{p_1} t_1' \xrightarrow{p_2} \cdots \xrightarrow{p_{n-1}} t_n' \xrightarrow{p_n} \Omega,$$

where $t_j' = t_j[\Omega]$ for all $j, 1 \leq j \leq i - 1$. Since $t_j'$ is soft, we apply the I.H. to the derivation $t_j' \xrightarrow{\Omega} \Omega$, which takes $n - 1$ steps and the conclusion follows.

If $P_i = \{p_1, p_2, \ldots, p_n\}$ and $m > 0$, then, since, by Lemma 13, $p_i \in P_{\text{os}^\mu(t_i)}$, let us consider $s = t_i[p_i]$ and the $l \in L(R)$ such that $t_i[p_i] \uparrow l_2$. We prove $s \uparrow l_2$ by contradiction. Hence, reductions performed using positions in $\{p_1, \ldots, p_n\} = P_i$ are disjoint to ours and are not relevant here (since we assume $p_j \notin P_{\text{os}^\mu(t_i)}$ for all $j < i$, we cannot have $p_j < p_i$). Since $t_i[p_i] \uparrow l_2$, and $p_i \in P_{\text{os}^\mu(t_i)}$, previous reductions (at nonreplacing positions in $P_i$) cannot modify the root of $t_i[p_i]$. Hence, we write $s = C[s_1, \ldots, s_m]$ and $l_2 = C[l_1, \ldots, l_2]$ for some maximal, nonempty context $C[]$. If $s$ and $l_2$ are not compatible, there is $k, 1 \leq k \leq p$ such that $s_k$ and $l_k$ are not compatible. Therefore, it must be root$(l_k) \neq \Omega$. Hence, since $\mu_{\mathcal{R}} \subseteq \mu$, we have $q_k \in P_{\text{os}^\mu(t_i)}$ if $l_2[q_k] = l_k$. However, no $\Rightarrow \Omega$-reduction performed below the root of subterm $s_k$ using positions in $P_i$ can make $s_k$ compatible with $l_k$, since, by Proposition 6, we have $q_k \notin P_{\text{os}^\mu(s)}$ and every $\Rightarrow \Omega$-reduction is given at nonreplacing positions, and
our choice of $s_k$ (hence of $q_k$) is arbitrary. However, this means that $t_i|_{P_j}$ will not be compatible with $l_\Omega$, thus leading to a contradiction. Therefore, we can eliminate every $\rightarrow\Omega$-reduction step $t_i \rightarrow^{\Omega}_{P_j} t_{i+1}$ for $P_j \in P_i$ because they are overridden by the reduction step $t_i \rightarrow^{\Omega}_{P_i} t_{i+1}$ and are therefore useless. Thus, we obtain a shorter derivation (because $m > 0$) and by I.H., the conclusion follows. 

The following proposition establishes that softness of terms is preserved under replacements on nonreplacing positions.

**Proposition 21.** Let $R$ be a TRS and $\mu \in CM_R$. If $t \rightarrow^*_{\Omega} t'$, then for all $p \in Pos^\Omega(t)$ and $t' \in T_\Omega(F, \lambda')$, $t'[\ p] \rightarrow^*_{\Omega} t'$ in at most the same number of steps.

**Proof.** By induction on the length $n$ of the derivation $t \rightarrow^*_{\Omega} t'$. If $n = 0$, then $t = t'$ and $\overline{Pos^\Omega(t)} = \emptyset$. Thus, the conclusion follows.

For the induction step, let $t \rightarrow_{\Omega}^q u \rightarrow_{\Omega}^q t'$. Thus, there exists $l \in L(R)$ such that $t|_{lq} \uparrow_{\Omega} l_\Omega$ and $q \in Pos^\Omega(t)$. Hence, there is a maximal context $C$ such that $t = C[t_1, \ldots, t_m]$ and $l_\Omega = C[l_1, \ldots, l_m]$ and either $t_i \leq l_i$ (and $t_i = \Omega$) or $t_i \geq l_i$ (and $l_i = \Omega$) for all $1 \leq i \leq m$. By Proposition 5, $p \not\in q$. If $q < p$, we let $p = qq'$. If there is no $t_i \leq l_i \leq m$ such that $q' \in Pos(t'_i)$, then, by Proposition 6, $q' \in Pos^\Omega(t)$. Hence, by Proposition 4, $q' \in Pos^\Omega(t')$. However, by Proposition 6, $q' \in Pos^\Omega(t')$ and by Proposition 6, $q' \in Pos^\Omega(t')$, thus leading to a contradiction. Therefore, let $t_i$ be such that $t|_{l}|_{l_i}$ and $p = p_i, p'_i$ for some $1 \leq i \leq m$. If $t_i = \Omega$, then $p = p_i$. Since $t_i \leq l_i \leq p \in \overline{Pos^\Omega(t)}$, by Proposition 6, $p \in \overline{Pos^\Omega(t)}$. Since $\mu_{\overline{R}} \subseteq \mu$, we have $q \in Pos^\Omega(t)$. Therefore, we let $l_i = \Omega$ and, again, $q' \rightarrow_{\Omega} l_i$. In both cases, we have that $t'[\ l_i] \rightarrow_{\Omega} l_i$. Hence, $l_i = |_{l_i}$, we have that $t'[\ l_i] \rightarrow_{\Omega} u \rightarrow_{\Omega} \Omega$ in $n$ steps and the conclusion follows. If $p \not\in q$, then $t'[\ p] \rightarrow_{\Omega} u'[\ p]. By the induction hypothesis, $u'[\ p] \rightarrow_{\Omega} \Omega$ in at most $n - 1$ steps. Therefore, $t'[\ p] \rightarrow_{\Omega} \Omega$ in at most $n$ steps and the conclusion follows. 

**Theorem 15.** Let $R = (F, R)$ be a TRS and $\mu \in CM_R$. If $t \in T_\Omega(F, \lambda')$ is a soft term, then $I^\mu_{\Omega}(t) = I_{\Omega}(t)$.

**Proof.** If $Pos^\Omega(t) \neq \emptyset$, then $I_{\Omega}(t) = \emptyset = I^\mu_{\Omega}(t)$. Assume $Pos^\Omega(t) \neq \emptyset$ and that there is a nonreplacing strong index $p \in I_{\Omega}(t) - I^\mu_{\Omega}(t)$. Since $0(\lambda') = \emptyset$, by Proposition 20, $t \rightarrow_{\Omega}^\mu \Omega$. By Proposition 21, $t[\ l_i] \rightarrow_{\Omega}^\mu \Omega$, thus contradicting that $p$ is a strong index of $t$.

Theorem 15 does not hold for $m$-indices.

**Example 34.** Consider the TRS $R$ and term $t$ as in Example 32. Note that $t = f(g(\Omega, x), b)$ is soft:

$$f(g(\Omega, x), b) \rightarrow_{\Omega} f(g(\Omega, x), \Omega) \rightarrow_{\Omega} \Omega.$$ 

In Example 32 we have shown that 1.1 is an $m$-index of $t$. Nevertheless, if we take $\mu = \mu_{\overline{R}}$, then 1.1 $\in Pos^\Omega(t)$ and we have that $I^\mu_{m}(t) = \emptyset$ but $I_{m}(t) \neq \emptyset$.

The advantage of Theorem 15 w.r.t. Theorem 14 is that, whereas root-stability is undecidable, deciding whether a term is soft is easy by using $\Omega$-reduction. Moreover, non-root-stable terms are soft, but the opposite is not true. Thus, Theorem 15 is more general.

Unfortunately, Theorem 15 cannot be extended to nonsoft terms: For instance, if we consider the TRS in Example 1, then term $t = \Omega: \Omega$ is rigid. However, both 1 and 2 are strong indices of $t$, but 2 $\notin Pos^\Omega(t)$; i.e., 2 $\notin I^\mu_{\Omega}(t)$.

### 7.4. Using Indices for Defining Context-Sensitive Strategies

The previous results allow us to establish the following.

**Corollary 7.** Let $R = (F, R)$ be an almost orthogonal, NV-sequential TRS and $\mu \in CM_R$. If $t \in T_\Omega(F, \lambda')$ is not root-stable, then $I^\mu_{m}(t^\Omega) \neq \emptyset$.

**Proof.** Since $t^\Omega$ is an $\Omega$-normal form, by NV-sequentiality, $I_{m}(t^\Omega) \neq \emptyset$ and by Corollary 6, $I^\mu_{m}(t^\Omega) \neq \emptyset$.
Theorem 13 and Corollary 7 entail a result that complements Theorem 10.

**THEOREM 16.** Let \( R \) be an almost orthogonal, NV-sequential TRS, and \( \mu \in CM_R \). Every non-root-stable term has a replacing root-needed redex.

**Proof.** Let \( t \) be a non-root-stable term. By Corollary 7, \( T^\mu_t(t^\Omega) \neq \emptyset \). Let \( p \in T^\mu_t(t^\Omega) \). Since \( t^\Omega \subseteq t[\Omega]_p \), by Lemma 9 and Proposition 6, \( p \in T^\mu_1(t[\Omega]_p) \). By Theorem 13, \( t_p \) is a root-needed redex of \( t \). By Proposition 6, \( t_p \) is a replacing root-needed redex. □

Corollary 7 allows us to define the following one-step \( \mu \)-strategy for almost orthogonal, NV-sequential TRSs (whenever \( \mu_{R,m} \subseteq \mu \)):

\[
H_{m}(t) = \begin{cases} 
T^\mu_m(t^\Omega) & \text{if } t \text{ is not root-stable} \\
\bigcup_{I \in \mu(f)} i.H_{m}(t_i) & \text{if } t = f(\vec{t}) \text{ is root-stable} \\
\emptyset & \text{if } t \text{ is a variable.}
\end{cases}
\]

According to Corollary 7, \( H_{m} \) is actually a \( \mu \)-strategy. By Theorems 9 and 13, \( H_{m} \) is root-normalizing. By construction, \( H_{m} \) is context-free and hence \( \mu \)-normalizing (Theorem 8). Unfortunately, since root-stability remains undecidable, this definition of \( H_{m} \) is not completely effective. In the following section, we overcome this problem by using the fact that, when considering NV-sequential TRSs, every reducible term \( t \) contains a redex which is addressed by a \( m \)-index of \( t^\Omega \).

### 8. CONTEXT-SENSITIVE INDEX REDUCTION STRATEGIES

We say that a one-step \( \mu \)-strategy \( H \) is an index reduction \( \mu \)-strategy if it always reduces (replacing) redexes pointed by indices. Notice that \( H_{m} \) is not an \( m \)-index reduction \( \mu \)-strategy.

**Example 35.** Consider \( R \) in Example 32 and \( f(g(h(a),x),h(c)) \) which is a root-stable term. Assume that \( \mu(f) = \mu(g) = \{1,2\} \). Since \( g(h(a),x) \) is not root-stable and \( 1 \in T^\mu_m(g(\Omega,x)) \), we have the following \( H_{m} \)-reduction step:

\[
f(g(h(a),x),h(c)) \leftrightarrow_{H_{m}} f(g(a,x),h(c)).
\]

However, 1.1 is not an \( m \)-index of \( f(g(\Omega,x),h(c)) \):

\[
f(g(\bullet,x),h(c)) \rightarrow_{\omega} f(g(\bullet,x),\Omega)
\]

and \( f(g(\bullet,x),\Omega) \uparrow f(\Omega,a) \).

We can (try to) define arbitrary (strong or \( m \)-) index \( \mu \)-reduction as

\[
H_{\mu}(t) = T^\mu_{\mu}(t^\Omega) \quad H_{m}(t) = T^\mu_{m}(t^\Omega);
\]

that is, only redexes occurring on replacing strong or \( m \)-index positions can be selected for reduction. The advantage of these definitions is that there is no mention of root-stability of term \( t \) that is considered for \( \mu \)-reduction.

**Remark 5.** Note that Proposition 16 and Lemma 9 are essential for making sense of the use of \( H_{\mu} \) and \( H_{m} \): they ensure that whenever a redex position \( p \in H_{\mu}(t)(\text{resp. } p \in H_{m}(t)) \) is selected for reduction, redex \( t_p \) actually occurs on a strong (resp. \( m \)-) index of \( t[\Omega]_p \) (note that indices are computed w.r.t. \( t^\Omega \)).

Now we prove that it is possible to define \( \alpha \)-index reduction \( \mu \)-strategies; in particular, we prove the existence of both \( H_{\mu} \) and \( H_{m} \).

In the following, we write \( t \rightarrow_{\omega} s \) if \( t \rightarrow_{\omega} s \) and \( p \in Pos^\alpha(t) \). We have the following result.
Proposition 22. Let \( R = (F, R) \) be a TRS, \( \mu \in CM_R, l \in L(R) \), and let \( s \in \mathcal{T}_\Omega(F, \mathcal{X}) - \{\Omega\} \) be such that \( s \uparrow l_2 \). If \( t \xrightarrow{\omega} s \), then for all \( p \in \overline{P}\text{os}^\mu(t) \) and \( t' \in \mathcal{T}_\Omega(F, \mathcal{X}) \), there exists \( s' \in \mathcal{T}_\Omega(F, \mathcal{X}) - \{\Omega\} \) such that \( t[t']_p \xrightarrow{\omega} s' \) in at most the same number of steps and \( s' \uparrow l_2 \).

Proof. By induction on the length of the derivation \( t \xrightarrow{\omega} s \). If \( n = 0 \), then \( s \neq \Omega \) and \( t \uparrow l_2 \). By reasoning as in the proof of Proposition 21, we conclude that if we let \( s' = t[t']_p \), we have that \( s' \neq \Omega \) (note that, since \( p \in \overline{P}\text{os}^\mu(t) \), we have that \( p \neq \Lambda \) and \( s' \uparrow l_2 \).

For the induction step, let \( t \xrightarrow{\omega} u \xrightarrow{\omega} s \). Thus, there exists \( l' \rightarrow r' \in R \) such that \( t_q \uparrow l_2 \) and \( q \in \text{Pos}^\mu(t) \). By Proposition 5, \( p \neq q \). If \( q < p \), then, by reasoning as in the base case, we have that \( t[t']_p \xrightarrow{\omega} u \xrightarrow{\omega} s \) and the conclusion follows. If \( p \equiv q \), then \( t[t']_p \xrightarrow{\omega} u(t[p])_p \). By the induction hypothesis, there exists \( s' \neq \Omega \) such that \( u(t[p])_p \xrightarrow{\omega} s' \) in at most \( n - 1 \) steps and \( s' \uparrow l_2 \). Therefore, \( t[t']_p \xrightarrow{\omega} s' \) in at most \( n \) steps and the conclusion follows. \( \blacksquare \)

Proposition 23. Let \( R = (F, R) \) be a TRS, \( \mu \in CM_R, l \in L(R) \), and let \( s \in \mathcal{T}_\Omega(F, \mathcal{X}) - \{\Omega\} \) such that \( s \uparrow l_2 \). If \( t \xrightarrow{\omega} s \), then there exists \( s' \in \mathcal{T}_\Omega(F, \mathcal{X}) - \{\Omega\} \) such that \( t \xrightarrow{\omega} s' \) in at most the same number of steps and \( s' \uparrow l_2 \).

Proof. By induction on the length of the derivation \( t \rightarrow^* s \). If \( n = 0 \), then the proof is immediate. If \( n > 0 \), let \( t \xrightarrow{\omega} u \rightarrow^* s \). By the induction hypothesis, there exists \( s'' \) such that \( u \xrightarrow{\omega} s'' \) in at most \( n - 1 \) steps and \( s'' \uparrow l_2 \). If \( p \in \text{Pos}^\mu(t) \), then \( t \xrightarrow{\omega} u \xrightarrow{\omega} s'' \) and, by taking \( s'' = s' \), the conclusion follows. If \( p \overline{\mathcal{T}} \text{Pos}^\mu(t) \), then, by Proposition 22, there is \( s' \neq \Omega \) such that \( t[u(t[p])]_p \rightarrow^* s' \) at most \( n - 1 \) steps and \( s' \uparrow l_2 \). Hence, the conclusion follows. \( \blacksquare \)

Now, we can ensure the existence of \( m \)-index \( \mu \)-strategies for NV-sequential TRSs.

Theorem 17. Let \( R = (F, R) \) be an NV-sequential TRS and \( \mu \in CM_R \). If \( t \in \mathcal{T}_\Omega(F, \mathcal{X}) \) is an \( \Omega \)-normal form such that \( \text{Pos}^\mu_\Omega(t) \neq \emptyset \), then \( \mathcal{T}^m_\Omega(t) \neq \emptyset \).

Proof. By structural induction. If \( t = \Omega \), the proof is immediate. If \( t = f(t_1, \ldots, t_k) \) since \( t \) is an \( \Omega \)-normal form and \( \text{Pos}^\mu_\Omega(t) \neq \emptyset \), there exists an \( \Omega \)-normal form \( t_i \), for some \( i \in \mu(f) \) such that \( \text{Pos}^\mu_\Omega(t_i) \neq \emptyset \). By the induction hypothesis, \( \mathcal{T}^m_\Omega(t_i) \neq \emptyset \). Assume that \( \mathcal{T}^m_\Omega(t) = \emptyset \). Then for any \( p \in \mathcal{T}^m_\Omega(t) \) and using Lemma 7, there are \( p' < i, p, s \neq \Omega \), and \( l \in L(R) \) such that \( t[t_i]_p \rightarrow^* s \) and \( s \uparrow l_2 \). Since \( p \in \mathcal{T}^m_\Omega(t) \), the only possibility is \( p' = \Lambda \); hence we assume \( t[t_i]_p \rightarrow^* s \). Obviously, this means that there exists \( s' \neq \Omega \) and a derivation \( t \rightarrow^* s' \) (which reduces at the same positions that derivation \( t[t_i]_p \rightarrow^* s \)) such that \( s' \uparrow l_2 \). By Proposition 23, there exists \( s'' \neq \Omega \) such that \( t \xrightarrow{\omega} s'' \) and \( s'' \uparrow l_2 \). By NV-sequentiality, \( \mathcal{T}^m_\Omega(t) \neq \emptyset \); let \( q \in \mathcal{T}^m_\Omega(t) \). Since we assume \( \mathcal{T}^m_\Omega(t) = \emptyset \), we have \( q \in \overline{P}\text{os}^\mu(t) \). By Proposition 22, there exists \( s''' \neq \Omega \) such that \( t[t_i]_p \rightarrow^* s''' \) and \( s''' \uparrow l_2 \). This contradicts that \( q \in \mathcal{T}^m_\Omega(t) \). \( \blacksquare \)

We need that \( \mu \in CM_R \) to ensure the result:

Example 36. Consider the TRS \( R \) of Example 1 and assume \( \mu(\text{first}) = \{2\} \). Note that \( \mu \notin CM_R \). Consider the \( \Omega \)-normal form \( t = \text{first}(\Omega, \Omega) \). Since \( \text{first}(\Omega, \bullet) \uparrow \text{first}(\Omega, \Omega) \), we have that \( 2 \notin \mathcal{T}^m_\Omega(t) \). Note that \( 1 \in \mathcal{T}^m_\Omega(t) \), but since \( 1 \in \overline{P}\text{os}^\mu(t) \), \( \mathcal{T}^m_\Omega(t) = \emptyset \).

Theorem 17 formalizes the existence of \( H_m \) for NV-sequential TRSs. With regard to \( H_1 \), the following result justifies the existence of strong index reduction \( \mu \)-strategies for strongly sequential TRSs.

Theorem 18. Let \( R = (F, R) \) be a strongly sequential TRS and \( \mu \in CM_R \). If \( t \in \mathcal{T}_\Omega(F, \mathcal{X}) \) is an \( \Omega \)-normal form such that \( \text{Pos}^\mu_\Omega(t) \neq \emptyset \), then \( \mathcal{T}^1_\Omega(t) \neq \emptyset \).

Proof. According to Proposition 3, we consider the maximal rigid context \( C[i] \) such that \( t = C[i_1, \ldots, i_n] \) and \( i_1, \ldots, i_n \) are soft terms. Since \( \text{Pos}^\mu_\Omega(t) \neq \emptyset \), there must be some \( i, 1 \leq i \leq n \) such that \( t_i = t_i[p] \) is an \( \Omega \)-normal form (or \( t_i \) itself if \( C[i] = \emptyset \)), such that \( p \in \text{Pos}^\mu(t) \), and \( \text{Pos}^\mu(t_i[p]) \neq \emptyset \). By strong sequentiality, \( \mathcal{T}^m_\Omega(t_i) \neq \emptyset \) and \( \mathcal{T}^m_\Omega(t) = \emptyset \). Let \( q \in \mathcal{T}^m_\Omega(t) \). Since every \( \Omega \)-position of a rigid context is trivially an index, \( p \in \mathcal{T}^m_\Omega(t) \). By Proposition 17, \( p.q \in \mathcal{T}^m_\Omega(t) \). By Proposition 4, \( p.q \in \text{Pos}^\mu(t) \); hence, \( p.q \in \mathcal{T}^1_\Omega(t) \) and the conclusion follows. \( \blacksquare \)
8.1. Properties of Context-Sensitive Index Reduction Strategies

We establish the main properties of (strong and \( nv \)-) index reduction \( \mu \)-strategies.

**Proposition 24.** Arbitrary \( \mu \)-reduction of strong or \( nv \)-indices is context-free.

*Proof.* Let \( t = f(t_1, \ldots, t_k) \) be root-stable and \( t \xrightarrow{\mu} t'_1, \ldots, t'_k \). Then, \( p \in I^{\mu}(t[\Omega]_p) \) and \( p = i.q \) for some \( 1 \leq i \leq k \). By Lemma 8 and Proposition 4, \( q \in I^{\mu}(t_i[\Omega]_q) \), i.e., \( t_i \xrightarrow{\mu} n \), thus showing context-freeness of \( H_{nv} \).

Concerning \( H_s \), we use Proposition 16 instead of Lemma 8.

**Theorem 19.** Let \( R \) be an almost orthogonal TRS, \( \mu \in CM_R \), and \( \alpha \in \{s, nv\} \). Every \( \alpha \)-index reduction \( \mu \)-strategy for \( R \) is root-normalizing.

*Proof.* Let \( H \) be an \( \alpha \)-index reduction \( \mu \)-strategy and \( t \) be a root-normalizing term. Maximal finite \( H \)-sequences end in \( \mu \)-normal forms thus containing a root-stable term (by Theorem 4). If there is an infinite reduction sequence \( t = t_0 \xrightarrow{H} t_1 \xrightarrow{H} \cdots \) that does not contain a root-stable term, by Theorem 13 (after considering Proposition 18 for strong indices), it is an infinite root-necessary reduction sequence whose existence contradicts Theorem 9.

In particular, Theorem 19 implies that both \( H_s \) and \( H_{nv} \) (that, according to Theorems 18 and 17, actually exist for strongly and NV-sequential TRSs respectively) are root-normalizing for every almost orthogonal (strongly) NV-sequential TRS \( R \) and whenever \( \mu \in CM_R \).

**Theorem 20.** Let \( R \) be an almost orthogonal, strongly sequential TRS and \( \mu \in CM_R \). Every strong index reduction \( \mu \)-strategy for \( R \) is \( \mu \)-normalizing.

*Proof.* By Theorem 18, \( H_s \) is a strong index reduction \( \mu \)-strategy. Since every strong index reduction \( \mu \)-strategy \( H \) can be extended to \( H_s \) (which, by Proposition 24, is context-free and by Theorem 19, root-normalizing), Corollary 5 entails the conclusion.

**Theorem 21.** Let \( R \) be an almost orthogonal, NV-sequential TRS and \( \mu \in CM_R \). Every \( nv \)-index reduction \( \mu \)-strategy for \( R \) is \( \mu \)-normalizing.

*Proof.* By Theorem 17, \( H_{nv} \) is an \( nv \)-index reduction \( \mu \)-strategy. Since every \( nv \)-index reduction \( \mu \)-strategy \( H \) can be extended to \( H_{nv} \) (which, by Proposition 24, is context-free and by Theorem 19, root-normalizing), by Corollary 5 the conclusion follows.

Therefore, \( H_s \) and \( H_{nv} \) are \( \mu \)-normalizing for every almost orthogonal (strongly, resp. NV-) sequential TRS \( R \), whenever \( \mu \in CM_R \).

9. \( \mu \)-NORMALIZATION AND NORMALIZATION

In this section, we discuss the use of context-sensitive strategies for defining normalization strategies. The first problem that we address here concerns the ability of CSR to approximate the (possibly many) normal form(s) of a term by means of \( \mu \)-normal forms. The fact that, in general, there are terms having normal forms which do not have \( \mu \)-normal forms (and terms having \( \mu \)-normal forms which have no normal form) is shown by the following example.

**Example 37.** Let us consider the (nonterminating) TRS

\[
\begin{align*}
  f(x, a) & \to a \\
  b & \to b \\
  g(a) & \to a.
\end{align*}
\]

If we take \( \mu(f) = \{1\} \) and \( \mu(g) = \emptyset \), then:
1. The term \( t = f(b, g(a)) \) is normalizing:

\[
f(b, g(a)) \rightarrow f(b, a) \rightarrow a.
\]

However, \( t \) is not \( \mu \)-normalizing, as there is only the following infinite \( \mu \)-rewriting sequence:

\[
f(b, g(a)) \leftarrow f(b, g(a)) \leftarrow \cdots.
\]

2. The term \( t = g(b) \) is \( \mu \)-normalizing (it is a \( \mu \)-normal form), but \( t \) is not normalizing:

\[
g(b) \rightarrow g(b) \rightarrow \cdots.
\]

Now, we establish conditions to ensure that any normal form \( s \) of a term \( t \) has a corresponding \( \mu \)-normal form \( t' \) of \( t \) that rewrites to \( s \).

**Theorem 22.** Let \( \mathcal{R} = (\mathcal{F}, \mathcal{R}) \) be a left-linear TRS, \( \mu \in CM_{\mathcal{R}} \), and \( t, s \in \mathcal{T}(\mathcal{F}, \mathcal{X}) \). If \( t \rightarrow^* s \), then there exists \( t' \in \mathcal{T}(\mathcal{F}, \mathcal{X}) \) such that \( t \rightarrow^*_\mu t' \rightarrow^* s \).

**Proof.** Since normal forms are root-stable, by Theorem 1, there is a term \( s' \) such that \( t \rightarrow^* s' \rightarrow^*_A s \), and \( \text{root}(s') = \text{root}(s) \). We prove the existence of \( t' \) by induction on the structure of \( s' \). If \( s \) is a constant or a variable, then (strict) inner reduction (i.e., \( \rightarrow^*_A \)) from \( s' \) to \( s \) is not possible and we have \( t' = s' = s \) as the desired \( \mu \)-normal form.

If \( s = f(s_1, \ldots, s_k) \), let \( s' = f(s'_1, \ldots, s'_k) \). We have \( s'_i \rightarrow^* s_i \), for \( 1 \leq i \leq k \). Since each \( s'_i \) has \( s_i \) as a normal form, we apply the I.H. to conclude that there are \( \mu \)-normal forms \( u_i \) of each \( s'_i \) such that \( u_i \rightarrow^* s_i \) for \( 1 \leq i \leq k \). We let \( t' = f(t'_1, \ldots, t'_k) \) be

\[
t'_i = \begin{cases} u_i & \text{if } i \in \mu(f) \\ s'_i & \text{if } i \notin \mu(f) \end{cases}
\]

for \( 1 \leq i \leq k \). Note that \( t \rightarrow^* t' \) and \( t' \rightarrow^* s \). In order to prove that \( t' \) is a \( \mu \)-normal form, we proceed by contradiction. If it is not, then, since each \( t'_i \) for \( i \in \mu(f) \) is a \( \mu \)-normal form, \( t' \) must be a redex of a rewrite rule \( l \rightarrow r \in \mathcal{R} \). This means that \( t' = \sigma(l) \) for some substitution \( \sigma \). However, since \( \mu_{\leq \infty} \subseteq \mu \), the subterms at the nonreplacing occurrences of \( l \) are variables. By Proposition 11, \( MRC^\mu(s') = MRC^\mu(s) \). Thus, \( s = \sigma'(l) \) for some substitution \( \sigma' \). Therefore, \( s \) is not a normal form, thus leading to a contradiction. \( \blacksquare \)

Theorem 22 ensures that, whenever a term has a normal form, it also has a \( \mu \)-normal form which is a prelude to the normal form. In fact, we can use ordering \( \leq \) on \( \Omega \)-terms to give a more standard formulation of approximation: by using Proposition 10, we can obtain the following: there is \( C[\ ] \) such that \( t \rightarrow^* t', s = C[t'_1, \ldots, t'_k] \), and \( C[\hat{\Omega}] \leq s \) (in fact, \( C[\ ] = MRC^\mu(t') \)). Immediate consequences of Theorem 22 are the following.

**Corollary 8.** Let \( \mathcal{R} \) be a left-linear TRS and \( \mu \in CM_{\mathcal{R}} \). Every normalizing term is \( \mu \)-normalizing.

**Corollary 9.** Let \( \mathcal{R} \) be a left-linear TRS and \( \mu \in CM_{\mathcal{R}} \). If \( \mathcal{R} \) is normalizing, then \( \mathcal{R} \) is \( \mu \)-normalizing.

These results ensure that a \( \mu \)-normalizing \( \mu \)-strategy will stop giving a \( \mu \)-normal form whenever it is applied to a normalizing term, i.e., a term having a normal form. This is the basis for proving the main result of this section.

**Theorem 23 (Normalization via \( \mu \)-normalization).** Let \( \mathcal{R} \) be a left-linear, confluent TRS and \( \mu \in CM_{\mathcal{R}} \). If \( \mathcal{H} \) is \( \mu \)-normalizing, then \( S_{\mathcal{H}} \) is normalizing.

**Proof.** Let \( \mathcal{H} \) be a \( \mu \)-normalizing \( \mu \)-strategy and \( t \) be a normalizing term. By Corollary 8, \( t \) is \( \mu \)-normalizing. Since \( \mathcal{H} \) is \( \mu \)-normalizing, there is no infinite \( \mathcal{H} \)-sequence starting from \( t \). Thus, by definition of \( S_{\mathcal{H}} \), every \( S_{\mathcal{H}} \)-sequence issued from a normalizing term \( t \) can be written as a (possibly
empty) finite H-sequence
\[ t = t_1 \rightarrow_H t_2 \rightarrow_H \cdots \rightarrow_H t_i = s \]
leading to a \( \mu \)-normal form \( s \) followed by a (possibly empty) \( S_H \)-sequence
\[ A : s = t_i \rightarrow_{S_H} t_{i+1} \rightarrow_{S_H} \cdots. \]
We proceed by induction on the structure of \( t \downarrow \), the normal form of \( t \).
If \( t \downarrow \) is a constant or a variable, then \( t \downarrow \) is also the unique \( \mu \)-normal form of \( t \) and \( s = t \downarrow \). Thus, sequence \( A \) is empty and every \( S_H \)-sequence issued from \( t \) is, in fact, an H-sequence. Hence, there is no infinite \( S_H \)-sequence starting from \( t \).

By confluence, we have \( s \rightarrow^* t \downarrow \) and by Proposition 11, we can write \( t \downarrow = C[t_1 \downarrow, \ldots, t_n \downarrow] \) (with \( t_1 \downarrow, \ldots, t_n \downarrow \) being the normal forms of \( t_1, \ldots, t_n \), respectively) and \( s = C[s_1, \ldots, s_n] \) for \( C[\ ] = \text{MRC}^n(s) \neq \emptyset \) and \( s_i \rightarrow^* t_i \downarrow \) for \( 1 \leq i \leq n \). By definition of \( S_H \), if \( A \) is infinite, then there exists \( s_i \) for some \( 1 \leq i \leq n \) such that
\[ s_i \rightarrow_{S_H} \cdots \]
is infinite, thus contradicting the induction hypothesis. ■

**Example 38.** Consider the orthogonal TRS \( \mathcal{R} \) of Example 2 (including the rules for \texttt{first}). Since \( \mathcal{R} \) is not terminating, a normalizing strategy is necessary for computing normal forms. According to Theorem 23, we can use a \( \mu^\text{can}_\mathcal{R} \)-normalizing \( \mu^\text{can}_\mathcal{R} \)-strategy \( \mathcal{H} \) for building a normalizing strategy \( S_H \).

Here,
\[
\mu^\text{can}_\mathcal{R}(::) = \mu^\text{can}_\mathcal{R}((\text{recip})) = \mu^\text{can}_\mathcal{R}((\text{terms})) = \emptyset,
\mu^\text{can}_\mathcal{R}(\text{sqr}) = \mu^\text{can}_\mathcal{R}(\text{dbl}) = \mu^\text{can}_\mathcal{R}(+) = \{1\}, \quad \text{and} \quad \mu^\text{can}_\mathcal{R}(\text{first}) = \{1, 2\}.
\]
Consider the one-step \( \mu^\text{can}_\mathcal{R} \)-strategy \( H_{lo} \) that contracts the leftmost-outermost \( \mu^\text{can}_\mathcal{R} \)-replacing redex of terms, i.e.,
\[
H_{lo}(t) = \text{min}_{\leq_L} (\text{Pos}_{\mu^\text{can}_\mathcal{R}}(t)),
\]
where \( \leq_L \) is the lexicographic ordering on positions: \( p \leq_L q \) if either \( p = \Lambda \), or \( p = i.p', q = j.q' \) for \( i, j \in \mathbb{N} \), and \( i < j \) or \( (i = j \wedge p' \leq_L q') \). Since \( \mathcal{R} \) is \( \mu^\text{can}_\mathcal{R} \)-terminating\(^{13}\), this strategy is \( \mu^\text{can}_\mathcal{R} \)-normalizing; moreover, it is a strong index reduction \( \mu^\text{can}_\mathcal{R} \)-strategy\(^{14}\), which means that no useless reduction is performed when \( \mu^\text{can}_\mathcal{R} \)-normalizing terms. As an example of use, we show how to obtain the first two terms of the infinite series \( \text{terms}(1) \) by evaluating the expression \( \text{first}(\text{dbl}(1), \text{terms}(1)) \) (again, \( n \) abbreviates \( s^n(0) \)).
\[
\begin{align*}
\text{first}(\text{dbl}(1), \text{terms}(1)) & \rightarrow_{H_{lo}} \text{first}(s(s(\text{dbl}(0))), \text{terms}(1)) \\
& \rightarrow_{H_{lo}} \text{first}(s(s(\text{dbl}(0))), \text{recip}(\text{sqr}(1)) : \text{terms}(2)) \\
& \rightarrow_{H_{lo}} \text{recip}(\text{sqr}(1)) : \text{first}(s(s(\text{dbl}(0))), \text{terms}(2))
\end{align*}
\]
At this point, the \( \mu^\text{can}_\mathcal{R} \)-strategy \( H_{lo} \) stops yielding a \( \mu^\text{can}_\mathcal{R} \)-normal form. However, we can continue with \( S_{H_{lo}} \):

\(^{13}\) In Example 3, we proved that \( \mathcal{R} \) is \( \mu \)-terminating for a less restrictive replacement map \( \mu \), hence the conclusion.

\(^{14}\) This could be justified as follows: due to the shape of the rules, the use of the canonical replacement map makes this TRS a kind of \( \mu^\text{can}_\mathcal{R} \)-left-normal TRS (remember that with left-normal TRSs, in all left-hand sides the function symbols occur to the left of all variables, in linear notation [BN98, p. 272]). It is well known that the position of the leftmost outermost redex of a term is a strong index for left-normal orthogonal TRSs [O’Do05]. Similarly, the leftmost-outermost \( \mu^\text{can}_\mathcal{R} \)-replacing redex is a strong index for this TRS.
\[
\text{recip(sqr(1))}: \text{first}(s(dbl(0)), \text{terms}(2)) \rightarrow_{s_{\mu}} \text{recip}(s(0+dbl(0))): \text{first}(s(dbl(0)), \text{terms}(2)) \rightarrow_{s_{\mu}} \text{recip}(s(dbl(0))): \text{first}(s(dbl(0)), \text{terms}(2)) \rightarrow_{s_{\mu}} \text{recip}(1): \text{first}(s(dbl(0)), \text{terms}(2)) \rightarrow_{s_{\mu}} \text{recip}(1): \text{recip}(s(sqr(2))): \text{first}(s(dbl(0)), \text{terms}(3)) \rightarrow_{s_{\mu}} \text{recip}(1): \text{recip}(s(sqr(1))+dbl(1))): \text{first}(s(dbl(0)), \text{terms}(3)) \rightarrow_{s_{\mu}} \text{recip}(1): \text{recip}(s(sqr(0)+dbl(0)+dbl(1))): \text{first}(s(dbl(0)), \text{terms}(3)) \rightarrow_{s_{\mu}} \text{recip}(1): \text{recip}(s(dbl(0)+dbl(1))): \text{first}(s(dbl(0)), \text{terms}(3)) \rightarrow_{s_{\mu}} \text{recip}(1): \text{recip}(4): \text{first}(s(dbl(0)), \text{terms}(3)) \rightarrow_{s_{\mu}} \text{recip}(1): \text{recip}(4): \text{first}(0, \text{terms}(3)) \rightarrow_{s_{\mu}} \text{recip}(1): \text{recip}(4): []
\]

Note that the previous sequence does not correspond to a standard leftmost-outermost reduction sequence: for instance, the second reduction step (in the first segment of the derivation) should contract \(\text{dbl}(0)\) rather than \(\text{terms}(1)\). Moreover, note that \(\text{dbl}(0)\) is not a strongly sequential redex, i.e., leftmost-outermost is not an index reduction strategy for this TRS.

In general, Theorem 23 does not hold for nonconfluent TRSs.

**EXAMPLE 39.** Consider the (nonconfluent) TRS \(\mathcal{R}\):
\[
a \rightarrow \textit{b} \\
a \rightarrow g(a).
\]
If \(\mu(g) = \emptyset\), then \(\mathcal{R}\) is \(\mu\)-terminating and every \(\mu\)-strategy \(H\) is \(\mu\)-normalizing. In particular, if \(H\) only uses the second rule for reducing redex \(a\), we have
\[
\overline{a} \rightarrow_{s_{\mu}} g(\overline{a}) \rightarrow_{s_{\mu}} g(g(\overline{a})) \rightarrow_{s_{\mu}} \cdots
\]
which does not normalize \(a\). However, \(a\) normalizes into \(b\).

The following corollary expresses the formal connection between termination theory and that of normalizing strategies.

**Corollary 10.** Let \(\mathcal{R}\) be a left-linear, confluent TRS, and \(\mu \in CM_\mathcal{R}\). If \(\mathcal{R}\) is \(\mu\)-terminating, then \(S_H\) is normalizing for every \(\mu\)-strategy \(H\).

10. \(\mu\)-NORMALIZATION AND INFINITARY NORMALIZATION

Lazy functional languages admit giving *infinite values* as the meaning of some expressions [FH88, Rea93]. Infinite values are defined as limits of converging infinite sequences of *partially defined* values which are more and more defined.
Example 40. Consider the TRS $R$ in Example 1. The term $\text{from}(0)$ has no normal form, since each application of the $\text{from}$ rule always introduces a new function call for the function $\text{from}$. However, the reduction sequence

$$\text{from}(0) \to 0: \text{from}(1) \to 0:1: \text{from}(2) \to \cdots$$

suggests that the infinite value

$$0:1:2:3: \cdots$$

(the infinite list of all natural numbers) could be considered as the (infinite) value of $\text{from}(0)$.

According to this situation, some research has been done concerning infinitary rewriting, i.e., rewriting that also considers infinite reduction sequences, probably involving infinite terms, and even term rewriting systems built from infinite terms [Cor93, CG99, DKP91, KKSV95, Luc01c, Mid97].

By an infinite sequence $S$ of elements taken from a set $A$ we mean a mapping $S : \mathbb{N}^+ \to A$. We denote the $n$th element of the sequence as $S_n$ rather than as $S(n)$. The definition of a notion of convergence to a limit of infinitary sequences on a set $A$ can be done by introducing a distance on elements of $A$. A distance is a function $d : A \times A \to \mathbb{R}$ such that, for all $x, y, z \in A$, $d(x, y) \geq 0$, $d(x, y) = 0$ iff $x = y$, $d(x, y) = d(y, x)$, and $d(x, z) \leq d(x, y) + d(y, z)$ [AN80]. A set $A$ together with a distance $d$ is a metric space $(A, d)$.

Let $(A, d)$ be a metric space. A sequence $S$ of elements of $A$ is said to be convergent if there exists $a \in A$ such that, for all $\epsilon > 0$, $\exists n \in \mathbb{N}^+, \forall p \geq n, d(S_p, a) < \epsilon$. If such an element $a$ exists, it is unique and it is called the limit of the sequence. A sequence $S$ is said to be a Cauchy sequence if $\forall \epsilon > 0, \exists n \in \mathbb{N}^+, \forall p, q \geq n, d(S_p, S_q) < \epsilon$. Every convergent sequence is a Cauchy sequence but there are Cauchy sequences which do not have a limit in $A$. A metric space is said to be complete if every Cauchy sequence is convergent. It is well known that every metric space $(A, d)$ can be embedded into a complete metric space $(\hat{A}, \hat{d})$ by a standard procedure called metric completion [AN80].

10.1. Infinitary Normalization

To discuss infinitary (term) rewriting, we follow Middeldorp’s approach [Mid97] which is simpler but still adequate for programming purposes (see [Luc01c] for a comparison of different transfinite rewriting frameworks and their semantic correspondences). An infinite rewrite sequence is an infinite sequence $t_1, t_2, \ldots$ of (finite) terms such that $t_n \to t_{n+1}$ for all $n \geq 1$. The depth $d$ of (an occurrence of) a subterm $s = t_1$ of a term $t$ is the length of position $p : d = |p|$. Given terms $t, s$, the largest natural number $k$ such that all nodes of $t$ and $s$ at a depth less than or equal to $k$ have the same label is given by $k = \text{mch}(t, s)$, (maximal common height) where:

$$\text{mch}(t, s) = \begin{cases} 0 & \text{if root}(t) \neq \text{root}(s) \\ 1 + \min(|\text{mch}(t_i, s_i) | 1 \leq i \leq ar(f)) & \text{if } t = f(\bar{t}) \text{ and } s = f(\bar{s}) \end{cases}$$

A distance $d : T(\mathcal{F}, \mathcal{X}) \times T(\mathcal{F}, \mathcal{X}) \to \mathbb{R}$ on terms is given as follows:

$$d(t, s) = \begin{cases} 0 & \text{if } t = s \\ 2^{-\text{mch}(t, s)} & \text{otherwise.} \end{cases}$$

The metric completion of $(T(\mathcal{F}, \mathcal{X}), d)$ yields the set $T^\omega(\mathcal{F}, \mathcal{X})$ of infinite terms. Thus, every infinitary Cauchy convergent rewriting sequence has a limit which is either a finite or an infinite term.

In infinitary normalization, we consider infinite sequences of length $\omega$ (the first limit ordinal) whose limit is a (possibly infinite) normal form. Kennaway et al. have developed the notion of strongly

15 Transfinite rewrite sequences are obtained by considering mappings $S : \alpha \to T^\omega(\mathcal{F}, \mathcal{X})$ for an arbitrary ordinal number $\alpha$ (and possibly involving infinite terms from the beginning) rather than mappings $S : \omega \to T(\mathcal{F}, \mathcal{X})$ for representing infinite sequences (of finite terms, which only become infinite at the limit). Recall that $\omega = \mathbb{N}$ is the first ordinal limit.
converging (infinite) rewrite sequence [KKSV95]. In sequences of this kind, the depth of the contracted redexes tends to infinite.

**Definition 4 [Mid97].** An infinite rewrite sequence \( t_1 \to t_2 \to \cdots \) is strongly converging if for all \( d \geq 0 \), there is an index \( i \geq 1 \) such that the depth of every redex contracted in \( t_i \to t_{i+1} \to \cdots \) is at least \( d \). Also all finite sequences are strongly converging.

Note that every infinite strongly converging sequence \( t_1 \to t_2 \to \cdots \) has a limit \( t_\infty \) which is necessarily an infinite term. If \( t_\infty \) is a normal form\(^\text{16} \), then, for all \( d \geq 0 \) there exists an index \( i \geq 0 \) such that the depth of every subterm which is not root-stable in \( t_i \) is at least \( d \).

**Definition 5 [Mid97].** A rewrite sequence is called **infinite normalizing** if it strongly converges to a (possibly infinite) normal form. An infinite rewrite sequence that is not infinite normalizing is called **perpetual**.

**Definition 6 [Mid97].** A reduction strategy \( S \) for a TRS is called **infinite normalizing** if there are no perpetual \( S \)-rewrite sequences starting from terms that admit an infinitary normalizing rewrite sequence.

**Definition 7.** A TRS is **infinite normalizing** if every (finite) term \( t \) admits an infinitary normalizing sequence.

For a given context \( C[\ ] \) having at least a hole, we denote by \( \text{sphole}(C[\ ]) \) the length of the **shortest path** from the root of \( C[\ ] \) to a hole (excluding the hole): \( \text{sphole}(C[\ ]) = \min(|\text{spfix}_{C[\ ]}(p)| \mid \text{such that } C[\ ][p] = \emptyset) \).

**Theorem 24.** Let \( \mathcal{R} \) be a left-linear TRS, and \( \mu \in \text{CM}_R \) be such that \( \mathcal{R} \) is \( \mu \)-normalizing. If there exists a \( \mu \)-normalizing \( \mu \)-strategy for \( \mathcal{R} \), then \( \mathcal{R} \) is infinitary normalizing.

**Proof.** Let \( \mathcal{H} \) be a \( \mu \)-normalizing \( \mu \)-strategy for \( \mathcal{R} \). Given a term \( t \), we show that it is possible to use \( \mathcal{H} \) to build a strongly converging sequence of reductions starting from \( t \). Since \( \mathcal{R} \) is \( \mu \)-normalizing, \( t \) has a \( \mu \)-normal form. Assume that \( t \leftarrow \mathcal{H}_t \to t_1 \) and \( t_1 \) is a \( \mu \)-normal form. This is well defined because \( \mathcal{H} \) is \( \mu \)-normalizing. Let \( C_1[\ ] = \text{MRC}^{\mu}(t_1) \), \( t_1 = C_1[t_1, \ldots , t_{m_1}] \) and \( d_1 = \text{sphole}(C_1[\ ]) \). By Proposition 9, \( C_1[\ ] \) is rigid. Since \( C_1[\ ] \neq \emptyset \), \( d_1 > 0 \). Now we use \( \mathcal{H} \) for \( \mu \)-reducing each \( t_{i_1} \) up to a \( \mu \)-normal form \( \text{sp}@_1 \); we obtain the corresponding maximal replacing contexts \( C_{i_1}[\ ] = \text{MRC}^{\mu}(t_{i_1}) \) which are also rigid for \( 1 \leq i \leq n_1 \). By Lemma 2, context \( C_2[\ ] = C_1[C_1[\ ] \ldots , C_{m_1}[\ ]] \) is also rigid and we have obtained \( t_1 \to \cdots \to C_2[t_{21}, \ldots , t_{2n_2}] = t_2 \). We let \( d_2 = d_1 + \min(d_{i_1}, 1 \leq i \leq n_1) \) where \( d_{i_1} = \text{sphole}(C_i[\ ]) \) (note that \( d_{i_1} > 0 \)) for \( 1 \leq i \leq n_1 \). Notice that \( 0 < d_1 < d_2 \). Reductions starting from \( t_2 \) take place on redexes whose depth is greater than or equal to \( d_2 \). Subterms whose depth is lower than \( d_2 \) are stable, since they overlap the rigid context \( C_2[\ ] \). We repeat this process. It is easy to show that the sequence \( t \to \cdots \to t_1 \to t_2 \to \cdots \) constructed in this way is a strongly converging sequence: if \( d \geq 0 \) there always is \( j \geq 1 \) such that \( d_{j-1} < d < d_j \) (we let \( d_0 = 0 \)). By construction of the derivation, the depth \( d' \) of any redex contracted in the tail subderivation \( t_j \to \cdots \to t_{j+1} \to \cdots \) satisfies \( d' \geq d_j \) and therefore \( d' > d \) as desired. Since any strongly converging derivation has a limit (which in this case is a normal form), the conclusion follows.

The following corollary connects \( \mu \)-termination and infinitary normalization.

**Corollary 11.** Let \( \mathcal{R} \) be a left-linear TRS and \( \mu \in \text{CM}_R \). If \( \mathcal{R} \) is \( \mu \)-terminating, then \( \mathcal{R} \) is infinitary normalizing.

This result only makes sense if \( \mu \neq \mu_+ \) since, in this case, termination and \( \mu \)-termination differ (terminating TRSs do not admit infinite rewrite sequences). Thus, we apply the result to nonterminating TRSs which are \( \mu \)-terminating. In this way, \( \mu \)-termination criteria [BLR02, FR99, GL02, GM99, GM02, Luc96, Luc02c, SX98, Zan97] can also be used for proving infinitary normalization. Finally, there are infinitary normalizing TRSs \( \mathcal{R} \) which are not \( \mu^\text{can}_\mathcal{R} \)-terminating.

\(^{16}\) Notice that the notion of normal form of [DKP91] (a term \( t \) such that \( t = t' \) whenever \( t \to t' \)) differs from the standard one (which we use here).
Example 41. Consider the nonterminating TRS \( R \):

\[ f(a) \rightarrow f(f(a)). \]

This TRS is infinitary normalizing. However, since \( \mu_{CM}^R = \mu_T \), \( R \) is not \( \mu \)-terminating for any \( \mu \in CM_R \).

10.2. Fairness and Infinitary Normalization

As remarked in [DKP91, KKS95, Mid97], we cannot hope to achieve infinitary normalization without imposing a fairness condition [Mid97].

Definition 8 [Mid97]. An infinite rewrite sequence \( t_1 \rightarrow t_2 \rightarrow \ldots \) is called fair if for every \( i \geq 1 \) and every maximal non-root-stable subterm \( s \) of \( t_i \) that has a root-stable reduct there is a \( j \geq i \) such that the position of the redex contracted in the step \( t_j \rightarrow t_{j+1} \) is below the position of \( s \) in \( t_i \).

Given a (possibly infinite) rewrite sequence \( A : t = t_1 \xrightarrow{p_1} t_2 \xrightarrow{p_2} \ldots \) and \( p \in Pos(t) \), we say that \( s_1 \xrightarrow{d_1} s_2 \xrightarrow{d_2} \ldots \) is a \( p \)-subsequence of \( A \) if it is empty or there are integers \( j > i \geq 1 \) such that \( p_k \parallel p \) for all \( k \in \{1, \ldots, j-1\} - \{i\} \), \( p_j = q_1, p_j = q_2, s_1 = t_i, s_2 = t_{i+1}, \) and \( s_2 \xrightarrow{d_2} \ldots \) is a \( p \)-subsequence of \( t_j \xrightarrow{p_j} t_{j+1} \xrightarrow{p_{j+1}} \ldots \). The following lemma is used below.

Lemma 14. Let \( R \) be an infinitary normalizing TRS. Let \( t = C[s_1, \ldots, s_n] \) be such that \( C[] \) is rigid and let \( p \in Pos(t) \) be such that \( s_i = t_i \) for \( 1 \leq i \leq n \). Every rewrite \( p \)-subsequence of a fair infinite rewrite sequence \( A : t = t_1 \rightarrow t_2 \rightarrow \ldots \) is either normalizing or fair infinite.

Proof. Let \( p \in \{p_1, \ldots, p_n\} \). If there is a finite \( p \)-subsequence \( B \) of \( A \) starting from \( t_1 \) and yielding a term \( s \) which is not a normal form, let \( s' \) be a maximal non-root-stable subterm of \( s \). By definition of \( p \)-subsequence, there are \( i \geq 1 \) and \( q \in Pos(t_i) \) such that \( \forall p \parallel q \), \( t_i \parallel q = s' \). Since \( C[] \) is rigid, \( s' \) is a maximal non-root-stable subterm of \( t_i \). Since \( R \) is infinitary normalizing, \( t_i \) admits an infinitary normalizing sequence. Then, \( s' \) has a root-stable reduct. Since \( A \) is fair, there is \( j \geq i \) such that the redex contracted in the step \( t_j \rightarrow t_{j+1} \) is below the position of \( s' \) in \( t_i \). Hence, \( s \) should be further reduced in \( B \) thus contradicting that \( s \) ends \( B \). If \( B \) is infinite, by definition of \( p \)-subsequence, it is fair.

Definition 9 [Mid97]. A reduction strategy \( S \) for a TRS is called infinitary fair-normalizing if there are no perpetual fair \( S \)-sequences starting from terms that admit an infinitary normalizing rewrite sequence.

The proof of Theorem 24 suggests that it is possible to build infinitary normalizing strategies which are based on \( \mu \)-normalizing \( \mu \)-strategies.

Proposition 25. Let \( R \) be a left-linear TRS and \( \mu \in CM_R \) be such that \( R \) is \( \mu \)-normalizing. Let \( H \) be a \( \mu \)-normalizing \( \mu \)-strategy. For all infinite fair \( S_H \)-sequence

\[ A : t_1 \rightarrow s_1, t_2 \rightarrow s_2, \ldots \]

and every \( d \in \mathbb{N} \), there is \( i \geq 1 \) such that \( t_i = C[s_1, \ldots, s_n], C[] \) is rigid, \( s_1, \ldots, s_n \) are not normal forms, and \( d \leq \text{spole}(C[]) \).

Proof. By induction on \( d \). If \( d = 0 \), we take \( i = 1 \) and \( C[] = \Box \). If \( d > 0 \), we note that, by Proposition 15, \( S_H \) is \( \mu \)-normalizing. Thus, since \( R \) is \( \mu \)-normalizing, \( t_1 \) has a \( \mu \)-normal form and \( A \) can be written as

\[ t_1 \rightarrow s_1, t_2 \rightarrow s_2, \ldots \rightarrow s_n, t_i \rightarrow s_n, \ldots, \]

where \( t_i \) is a \( \mu \)-normal form (and, by Proposition 14, every \( t_j \) for \( j \geq i \)). Let \( C_0[] = MRC^\mu(t_i), \) \( t_i = C_0[s_1, \ldots, s_n], \) and \( p_j \) be such that \( t_i \xrightarrow{p_j} t_i \) for \( 1 \leq j \leq n \). Let \( d_0 = \text{spole}(C_0[]) \) and \( d' = d - d_0 \). Since \( C_0[] \neq \Box \), we have that \( d_0 > 0 \); hence \( d' < d \). For each \( 1 \leq j \leq n \), let \( A_j \) be the (finite or infinite) rewrite \( p_j \)-subsequence extracted from the tail of \( A \) which starts in \( t_i \). In fact, by
definition of $S_H$, each $A_j$ is an $S_H$-sequence, i.e.,

$$A_j : s_j = s^1_j \rightarrow s^2_j \rightarrow s^3_j \cdots.$$ 

Since Theorem 24 ensures that every term is infinitary normalizing and Proposition 9 ensures that $C_0[ ]$ is rigid, by Lemma 14, each $A_j$ is either finite (and normalizes $s_j$ into $u_j$) or a fair infinite $S_H$-sequence. If $A_j$ is finite, we let $k_j$ be the length of $A_j$. If $A_j$ is infinite, by the induction hypothesis, there is $k_j \geq 1$ such that $s^k_j = C'_j[s'_{j_1}, \ldots, s'_{j_{m_j}}]$. $C_j[ ]$ is rigid, $s'_1, \ldots, s'_{m_j}$ are not normal forms, and $d' \leq \text{sphole}(C'_j[ ]).$

Let $C[ ] = C_0[C_1[ ], \ldots, C_n[ ]]$ where, for $1 \leq j \leq n$,

$$C_j[ ] = \begin{cases} u_j & \text{if } A_j \text{ is finite, and} \\ C'_j[ ] & \text{if } A_j \text{ is infinite}. \end{cases}$$

Note that, by definition of subsequence, $k = i + \sum_{j=m}^n k_j$ is such that $t_k = C[s'_1, \ldots, s'_{m_j}]$. Since normal forms $u_j$ are rigid contexts (having no hole), by Lemma 2, $C[ ]$ is rigid. Note that

$$\text{sphole}(C[ ]) = \text{sphole}(C_0[ ]) + \min(\{\text{sphole}(C'_j[ ]) | A_j \text{ is infinite}\}).$$

Hence,

$$d = d_0 + d'$$

$$= \text{sphole}(C_0[ ]) + d'$$

$$\leq \text{sphole}(C_0[ ]) + \min(\{\text{sphole}(C'_j[ ]) | A_j \text{ is infinite}\})$$

$$= \text{sphole}(C[ ]).$$

\[\begin{aligned}
\text{THEOREM 25.} & \quad \text{Let } R \text{ be a left-linear TRS and } \mu \in CM_R \text{ be such that } R \text{ is } \mu\text{-normalizing. If } H \text{ is a } \mu\text{-normalizing } \mu\text{-strategy, then } S_H \text{ is infinitary fair-normalizing.} \\
\text{Proof.} & \quad \text{We must show that every infinite fair } S_H\text{-sequence } A \text{ is infinitary normalizing, i.e., derivation } A \text{ strongly converges to a (possibly infinite) normal form. This is an immediate consequence of Proposition 25.} \\
\text{Theorem 25 shows that, in contrast to (most) normalizing strategies, } \mu\text{-normalizing } \mu\text{-strategies are useful for obtaining infinitary normal forms.} \\
\text{COROLLARY 12.} & \quad \text{Let } R \text{ be a left-linear TRS and } \mu \in CM_R. \text{If } R \text{ is } \mu\text{-terminating, then } S_H \text{ is infinitary fair-normalizing for every } \mu\text{-strategy } H. \\
\text{Theorem 25 and Corollary 12 complement the results on infinitary normalizing strategies given in [Mid97, KKV95] since they do not apply to left-linear TRSs but only to orthogonal TRSs. As a counterpart, we require } \mu\text{-termination.} \\
\text{For non- } \mu\text{-terminating TRSs, we can still use } \mu\text{-normalizing strategies.} \\
\text{THEOREM 26 [Mid99].} & \quad \text{Let } R \text{ be a confluent TRS. Every reduction strategy } S \text{ for } R \text{ that can be extended to a context-free root-normalizing reduction strategy for } R \text{ is infinitary fair-normalizing.} \\
\text{PROPOSITION 26.} & \quad \text{If } H \text{ is a context-free } \mu\text{-strategy, then } S_H \text{ is context-free.} \\
\text{THEOREM 27.} & \quad \text{Let } R \text{ be a left-linear, confluent TRS and } \mu \in CM_R. \text{If } H \text{ is a } \mu\text{-strategy for } R \text{ that can be extended to a context-free root-normalizing } \mu\text{-strategy for } R, \text{ then } S_H \text{ is infinitary fair-normalizing.} \\
\text{Proof.} & \quad \text{Let } H' \text{ be the context-free root-normalizing extension of } H. \text{ By Proposition 15, } S_{H'} \text{ is root-normalizing. By Proposition 26, } S_H \text{ is context-free. Since } S_H \text{ can be extended to } S_{H'}, \text{ by Theorem 26 the conclusion follows.} \quad \blacksquare}
\end{aligned} \]
According to Theorem 27, Theorem 19, and Proposition 24, whenever $\mu \in CM_R$, strong and $m$-index reduction $\mu$-strategies are infinitary fair-normalizing for every almost orthogonal, strong, and NV-sequential TRS $R$ (respectively).

10.3. Using Context-Sensitive Rewriting for Infinitary Normalization

The parallel outermost strategy, $S_{po}$, is proved infinitary normalizing for almost orthogonal TRSs [Mid97, Corollary 7.6]. This is because $S_{po}$ is infinitary fair-normalizing and fair; unfortunately, $H_{po}$ is not fair.

**Example 42.** Consider the following (orthogonal) TRS $R$ [Mid97]

$$a \rightarrow f(a, a)$$
$$f(b, x) \rightarrow c$$

and $\mu = \mu_R^{\text{can}}$. Note that $a$ has no normal form. The $H_{po}$-sequence

$$a \leftarrow_{n_{po}} f(a, a) \leftarrow_{n_{po}} f(f(a, a), a) \leftarrow_{n_{po}} \cdots$$

is not fair, since the redex $a$ in the second term $f(a, a)$ of the derivation is never considered for reduction as it is placed in a nonreplacing position ($2 \notin Pos^R(f(a, a))$).

Example 42 shows that $S_{H_{po}}$ is not infinitary normalizing. This is because whenever $S_{H_{po}}$ applies on non-$\mu$-normalizing terms, it behaves exactly like $H_{po}$ (which is not fair). In this way, ensuring $\mu$-normalization of TRSs turns out to be very important for achieving good behavior of $S_H$ in infinitary normalization.

**Remark 6.** A weaker notion of fairness that does not consider non-$\mu$-replacing positions of maximal non-root-stable subterms would not be useful without ensuring the $\mu$-normalizing character of the TRS: if we cannot ensure that $H$ eventually stops giving a $\mu$-normal form, then $S_H$ cannot explore the non-$\mu$-replacing positions; thus, infinitary normalization is not ensured.

Unfortunately, we do not obtain fairness with $S_H$ even with $\mu$-normalizing TRSs and a $\mu$-normalizing $\mu$-strategy $H$.

**Example 43.** Consider the TRS $R$:

$$a \rightarrow c(a, a)$$

Note that $R$ is $\mu_R^{\text{can}}$-terminating. Since the only $\mu$-reducible term is $a$, there is only one possible $\mu_R^{\text{can}}$-strategy $H$ for $R$. The $S_{H}$-sequence

$$a \rightarrow s_n \ c(a, a) \rightarrow s_n \ c(c(a, a), a) \rightarrow s_n \cdots$$

is not fair.

When dealing with $\mu$-normalizing $\mu$-strategies in $\mu$-normalizing TRSs, fairness is obtained without any specific effort when using the following strategy

$$S_{H}^{\parallel}(t) = \begin{cases} H(t) & \text{if } t \notin NF^\mu_R \\ C[\mathbb{S}_H(t_1), \ldots, \mathbb{S}_H(t_n)] & \text{if } t \in NF^\mu_R - NF_R, \text{ where:} \\ C[ ] = MRC^\mu(t) \text{ and } t = C[t_1, \ldots, t_n] \\ \emptyset & \text{otherwise.} \end{cases}$$

Here, for a given context $C[ ]$ and sets of rewrite sequences $S_1, \ldots, S_n$, issued from terms $t_1, \ldots, t_n$, we let $C[\mathbb{S}_j(t_1), \ldots, \mathbb{S}_j(t_n)]$ denote the set of derivations from $C[t_1, \ldots, t_n]$ to $C[s_1, \ldots, s_n]$ such that there is $i \in \{1, \ldots, n\}$ such that $t_i$ is not a normal form and for all $1 \leq j \leq n$, either $t_j$ is not a normal form (and is $t_j \rightarrow^+ s_j \in S_j$ as well), or $t_j$ is a normal form and $s_j = t_j$. 
According to this definition, we have the following results which can be proved in a way similar to previous ones.

**Theorem 28.** Let $\mathcal{R}$ be a left-linear TRS and $\mu \in \text{CM}_\mathcal{R}$ be such that $\mathcal{R}$ is $\mu$-normalizing. If $H$ is a $\mu$-normalizing $\mu$-strategy, then $S_H \parallel H$ is infinitary fair-normalizing.

**Theorem 29.** Let $\mathcal{R}$ be a left-linear TRS and $\mu \in \text{CM}_\mathcal{R}$ be such that $\mathcal{R}$ is $\mu$-normalizing. If $H$ is a $\mu$-normalizing $\mu$-strategy, then $S_H \parallel H$ is fair.

According to Theorems 28 and 29, we have the following.

**Corollary 13 (Infinitary normalization via $\mu$-normalization).** Let $\mathcal{R}$ be a left-linear TRS and $\mu \in \text{CM}_\mathcal{R}$ be such that $\mathcal{R}$ is $\mu$-normalizing. If $H$ is a $\mu$-normalizing $\mu$-strategy, then $S_H \parallel H$ is in infinitary normalizing.

**Example 44.** Continuing Example 43, we now only have the following $S_H \parallel H$-sequence

$$a \rightarrow_{S_H} \text{c}(a, a) \rightarrow_{S_H} \text{c}(\text{c}(a, a), \text{c}(a, a)) \rightarrow_{S_H} \cdots$$

which is fair.

Note that $S_H \parallel H$ has some advantages with respect to $S_{po}$: in general, $S_{po}$ is wasteful; i.e., it can perform useless reductions since (for almost orthogonal TRSs) it rewrites a root-necessary set of redexes rather than a set of root-needed redexes (see [Mid97, SR93] for further details about this). However, whenever $H$ is optimal (in the sense that it only reduces root-needed redexes), it is easy to show that $S_H \parallel H$ is also optimal. This is due to the rigidity of the maximal replacing context where the reducible parts to which the strategy jumps are placed. For instance, (one-step) index reduction $\mu$-strategies of Section 8 are optimal and can be used to provide optimal infinitary normalizing strategies by using $S_H \parallel H$ to extend their computational scope (with left-linear, $\mu$-normalizing TRSs).

11. APPLICATIONS

In this section, we use the theory that has been developed for the definition of normalizing strategies for TRSs which do not admit a normalizing strategy based on the usual techniques for doing so. We also show that the theory is suitable for analyzing the computational properties of certain types of strategies that can be specified within programming languages such as OBJ and ELAN.

11.1. Normalizing Strategies for Left-Linear (Possibly Overlapping) TRSs

Example 38 shows the use of CSR for defining an optimal normalizing strategy $S_{H_{\mu}}$ for a given TRS $\mathcal{R}$. Indeed, other techniques can be used for defining a normalizing strategy for $\mathcal{R}$. For instance, it is not difficult to see that $\mathcal{R}$ is strongly sequential$^{17}$. Therefore, it admits a computable normalizing strategy. The appealing point of our normalizing strategy $S_{H_{\mu}}$ in Example 38 is its simplicity (based on reducing replacing leftmost-outermost redexes) which may eventually drive to a simpler implementation. In this section, we go one step beyond and consider a TRS which cannot be given a normalizing strategy by using the usual techniques for doing so. Fortunately, we can define a normalizing strategy based on a $\mu$-normalizing context-sensitive strategy.

When considering the interaction between functions half (see Example 13) and dbl (see Example 2), the introduction of the rule

$$\text{half}(\text{dbl}(x)) \rightarrow x$$

immediately arises as a suitable optimization which eventually permits more efficient computations when executed using a suitable strategy. Consider the left-linear TRS $\mathcal{R}$ which is obtained by joining the rules of the TRSs of Examples 2 and 13 together with the previous rule. Thus, $\mathcal{R}$

$^{17}$ Indeed, $\mathcal{R}$ is inductively sequential in the sense of [Ant92]. These TRSs are strongly sequential, see [HLM98].
sqr(0) → 0
sqr(s(x)) → s(sqr(x)+dbl(x))
dbl(0) → 0
dbl(s(x)) → s(s(dbl(x)))
half(0) → 0
half(s(0)) → 0
terms(n) → recip(sqr(n)):terms(s(n))

This system is not terminating and we need a normalizing strategy for computing normal forms. Note that \( R \) is not even weakly orthogonal: it has two critical pairs

\[ \langle \text{half}(0), 0 \rangle \text{ and } \langle \text{half}(s(s(dbl(x)))), s(x) \rangle \]

which are not trivial. Therefore, most existing results describing normalizing strategies for weakly or almost orthogonal TRSs (e.g., [Ant92, AM96, DM97, HL91, Ken89, NT99, O'Do77, O'Do85, SR93, Toy92]) do not apply to \( R \).

As far as the author knows, only [Toy92] provides some results ensuring normalization of (strong) index reduction strategies for left-linear (possibly overlapping) strongly sequential TRSs. However, in this case, normalization is ensured only for the so-called root-balanced joinable TRSs (see Theorem 6.8 in [Toy92]) for which the critical pairs are root-balanced joinable (Definition 6.2 in [Toy92]). A critical pair \( \langle t, s \rangle \) is root-balanced joinable if both \( t \) and \( s \) reduce to a common term \( u \) using the same number \( k \geq 0 \) of root-reduction steps (i.e., \( \rightarrow \)-steps). Note that, e.g., weakly orthogonal TRSs are trivially root-balanced joinable. Unfortunately, the previous critical pairs for \( R \) are not root-balanced joinable, as components 0 or \( s(x) \) are normal forms which cannot be reduced. Moreover, root-reduction is not able to join the components of the second pair; see below. Thus, Toyama's results do not apply, either.

Consider \( R \) together with the replacement map \( \mu \) given by \( \mu(\cdot) = \{1\} \) and \( \mu(f) = \{1, \ldots, k\} \) for any other \( k \)-ary symbol \( f \). Note that \( \mu \in CM_R \). Corollary 10 allows us to define a normalizing strategy for this TRS. First, we need to prove that \( R \) (which is left-linear) is confluent and \( \mu \)-terminating.

Let \( S \) be the TRS containing all rules of \( R \) but the last one (for \text{terms}). Note that \( S \) “contains” all critical pairs in \( R \). Those critical pairs are convergent:

\[ \text{half}(0) \rightarrow 0 \]

and

\[ \text{half}(s(s(dbl(x)))) \rightarrow s(\text{half}(dbl(x))) \rightarrow s(x). \]

Hence, by Huet's critical pairs theorem, \( S \) is locally confluent. \( S \) is terminating: use a recursive path ordering (rpo) based on the precedence


\[ \text{sqr} > \text{dbl}, + > \text{s}; \text{ first} > [], ; \text{ and } \text{half} > \text{s}. \]

Therefore, by Newman's lemma, \( S \) is confluent. On the other hand, the TRS \( T \) consisting of the rule for \text{terms} is also confluent (by orthogonality). Now, we can use the following fact from [RV80]: “Confluence is preserved under the combination of left-linear TRSs (\( R_1 \) and \( R_2 \)) satisfying that there are no critical pairs between rules of \( R_1 \) and of \( R_2 \).”

Therefore, by taking \( R_1 = S \) and \( R_2 = T \), we conclude that \( R \) is confluent.
The TRS $\mathcal{R}$ is $\mu$-terminating: by using the contractive transformation of [Luc96], we obtain

\[
\begin{align*}
\text{sqr}(0) & \rightarrow 0 \\
\text{sqr}(\text{s}(x)) & \rightarrow \text{s}(\text{sqr}(x)+\text{dbl}(x)) \\
\text{dbl}(0) & \rightarrow 0 \\
\text{dbl}(\text{s}(x)) & \rightarrow \text{s}(\text{dbl}(x)) \\
\text{half}(0) & \rightarrow 0 \\
\text{half}(\text{s}(0)) & \rightarrow 0 \\
\text{terms}(n) & \rightarrow :\left(\text{recip}(\text{sqr}(n))\right)
\end{align*}
\]

which is terminating: again use the $rpo$ which is based on precedence

\[
\text{terms} > : \text{recip}, \text{sqr}; \quad \text{sqr} > \text{dbl}, + > \text{s}; \quad \text{first} > [ ] \quad \text{and} \quad \text{half} > \text{s}.
\]

Therefore, according to Corollary 10, we can use any $\mu$-strategy $H$ as a basis for obtaining a normalizing strategy $S_H$. Similarly, we could also use Corollary 13 for ensuring that $S_H|_1$ is an infinitary normalizing strategy for $\mathcal{R}$. Also note that existing infinitary normalizing strategies require (at least) almost orthogonality$^{19}$ (see [KKSV95, Luc98b, Mid97]). Thus, they do not apply to $\mathcal{R}$.

11.2. Context-Sensitive Rewriting and the Evaluation Strategy of OBJ

Algebraic languages, such as OBJ2 [FGJM85], OBJ3 [GWMFJ93, GWMFJ00], CafeOBJ [FN97], or Maude [CELM96], admit the specification of local strategies which are associated to function symbols. Syntactically, they are sequences of integers$^{20}$ in parentheses, given as an operator attribute following the keyword $\text{strat}$ [GWMFJ00].

**Example 45.** The specification

\[
\text{obj \ EXAMPLE is}
\begin{align*}
\text{sorts} \ & \text{Nat LNat.} \\
\text{op} \ & \text{0 :} \rightarrow \text{Nat.} \\
\text{op} \ & \text{s : Nat} \rightarrow \text{Nat.} \\
\text{op} \ & \text{nil :} \rightarrow \text{LNat.} \\
\text{op} \ & \text{cons : Nat LNat} \rightarrow \text{LNat [strat (1 0)]}. \\
\text{op} \ & \text{from : Nat} \rightarrow \text{LNat.} \\
\text{op} \ & \text{sel : Nat LNat} \rightarrow \text{Nat.} \\
\text{op} \ & \text{first : Nat LNat} \rightarrow \text{LNat.} \\
\text{var} \ & \text{X} \text{ Y : Nat.} \\
\text{var} \ & \text{Z : LNat.} \\
\text{eq} \ & \text{sel(s(X),cons(Y,Z)) = sel(X,Z).} \\
\text{eq} \ & \text{sel(0,cons(X,Z)) = X.} \\
\text{eq} \ & \text{first(0,Z) = nil.} \\
\text{eq} \ & \text{first(s(X),cons(Y,Z)) = cons(Y,first(X,Z)).} \\
\text{eq} \ & \text{from(X) = cons(X,from(s(X))).}
\end{align*}
\]

is an OBJ version of the TRS of Example 1 (with the natural typing expressed by sorts Nat and LNat). Note the local strategy (1 0) for the list constructor $\text{cons}$.

If a given symbol $f$ has no explicit local strategy, a default local strategy is determined according to each particular language. Local strategies serve to completely guide OBJ $E$-strategy ($E$ for evaluation): When considering a function call $f(t_1, \ldots, t_k)$, only the arguments whose indices are present in the list associated to the local strategy of $f$ are evaluated (following the ordering which has been specified in the list). If an index 0 is found, then the reduction of the external function call is attempted.

$^{19}$ Whether strong index reduction strategies are infinitary (fair) normalizing for left-linear, strongly sequential, root-balanced joinable TRSs is an open problem.

$^{20}$ Here we only consider non-negative integers.
Nagaya describes the operational semantics of term rewriting under strategy maps \( \varphi \) (which map each symbol \( f \) to its individual local strategy \( \varphi(f) \)) by using a reduction relation on labeled terms which helps to implement the necessary control of the arguments which must be evaluated and the order on which the evaluations must be performed [Nag99]. Starting from the canonical labeling \( \varphi(t) \) induced by the strategy map \( \varphi \) (which decorates each symbol \( f \in \mathcal{F} \) with \( \varphi(f) \)), the evaluation \( \text{eval}_\varphi(t) \subseteq T(\mathcal{F}, \mathcal{X}) \) of \( t \in T(\mathcal{F}, \mathcal{X}) \) is modeled as normalization of \( \varphi(t) \) under this reduction relation followed by the unlabeling of the obtained term(s). In [Luc01a, Luc01b], we have demonstrated that this evaluation process can be completely described by using CSR under the replacement map \( \mu_\varphi \) obtained by collecting as \( \mu_\varphi(f) \) all positive indices appearing in \( \varphi(f) \) for each symbol \( f \) (see Theorem 1 in [Luc01b]). For instance, CSR can be used to analyze termination of OBJ programs (Theorems 2 and 4 of [Luc01b]).

**Example 46.** Consider \( \mathcal{R} \) and \( \mu \) as in Example 1. The \( \mu \)-termination of \( \mathcal{R} \) can be ensured by proving termination of the following TRS \( \mathcal{R}''_\mathcal{Z} \) (see [Zan97])

\[
\begin{align*}
\text{first}(0, x) & \rightarrow [] & \text{sel}(0, x: y) & \rightarrow x \\
\text{first}(s(x), y: z) & \rightarrow y: \text{first}'(x, a(z)) & \text{sel}(s(x), y: z) & \rightarrow \text{sel}(x, a(z)) \\
\text{from}(x) & \rightarrow x: \text{from}'(s(x)) \\
\text{first}(x, y) & \rightarrow \text{first}'(x, y) & \text{from}(x) & \rightarrow \text{from}'(x) \\
a(\text{first}'(x, y)) & \rightarrow \text{first}(x, y) & a(\text{from}'(x)) & \rightarrow \text{from}(x) \\
a(x) & \rightarrow x \\
\end{align*}
\]

where \( \text{first}' \), \( \text{from}' \), and \( a \) are new symbols introduced by the transformation. Termination of \( \mathcal{R}''_\mathcal{Z} \) is proved by using an rpo based on precedence

\[
\text{sel} > a \approx \text{first} > \text{from}, :, \text{first}', \text{nil} \text{ and from} > :, \text{from}', s,
\]

and giving \( \text{sel} \) the usual (left-to-right) lexicographic status.

Since \( \mu = \mu_\varphi \) for \( \varphi \) as given in Example 45, according to [Luc01a, Luc01b] this means that the OBJ program of Example 45 is terminating.

We have also established conditions ensuring that terms in \( \text{eval}_\varphi(t) \) are \( \mu_\varphi \)-normal forms. This happens whenever the local strategies \( \varphi(f) \) for defined symbols \( f \in \mathcal{D} \) end in 0 (Theorem 9 in [Luc01a]; see [Eke98] for a discussion on the problems arising when such a requirement is not fulfilled). In this case, Nagaya’s formalization of OBJ evaluation strategy can be thought of as the specification of a \( \mu_\varphi \)-rewriting strategy, since our requirement of “being active as long as a \( \mu_\varphi \)-normal form is not reached” (see Definition 2) is fulfilled:

\[
H_\varphi(t) = \begin{cases} 
\{t \leftarrow^+ \mu_\varphi s \mid s \in \text{eval}_\varphi(t)\} & \text{if } t \not\in \text{NF}^\mu_\varphi \\
\emptyset & \text{otherwise}
\end{cases}
\]

where the rough description \( t \leftarrow^+ \mu_\varphi s \) could be exactly given (as a \( \mu_\varphi \)-reduction sequence) from the concrete OBJ evaluation sequence by using Theorem 1 in [Luc01b]. Note that a single \( H_\varphi \)-step achieves the complete evaluation of a term \( t \) as done by using \( \text{eval}_\varphi \).

As occurs for CSR, OBJ computations do not obtain normal forms (unless the local strategies contain all indices for the arguments of symbols, see [Nag99]). By using Theorem 23, we easily conclude the following.

**Theorem 30 (Normalization via \( \varphi \)-normalization).** Let \( \mathcal{R} = (\mathcal{C} \cup \mathcal{D}, \mathcal{R}) \) be a left-linear, confluent TRS and \( \varphi \) be an E-strategy map such that for all \( f \in \mathcal{D}, \varphi(f) \) ends in 0. If \( \mu_\varphi \in \text{CM}_\mathcal{R} \) and \( \mathcal{R} \) is \( \varphi \)-terminating, then \( S_\varphi \) is normalizing.

Obtaining normal forms in OBJ computations is also guaranteed as follows: given a strategy map \( \varphi \) ensuring that terms in \( \text{eval}_\varphi(t) \) are root-stable (for all \( t \in T(\mathcal{F}, \mathcal{X}) \)), any \( \varphi' \) given by \( \varphi'(f) = \varphi(f)++(i_1, \ldots, i_n) \) for all symbols \( f \in \mathcal{F} \) (where ++ appends two lists, and for all \( i \in \{1, \ldots, \ar(f) - \mu_\varphi(f), i \in \{i_1, \ldots, i_n\} \) ensures that terms in \( \text{eval}_{\varphi'}(t) \) are normal forms (Theorem 3.2 in [NO01]). In principle, this appears to be similar to the lifting of computational activity that \( S_H \) performs regarding
H. Unfortunately (in contrast to Theorem 30), this does not ensure a normalizing behavior for \( \psi' \). For instance, we are able to obtain the normal form \([0, 1, 2, 3, 4]\) of term \( \text{first}(5, \text{from}(0)) \) (for \( \psi \) as in Example 45) by using \( S_{\text{N}} \). In contrast, with \( \psi'(\text{cons}) = (1 \ 0 \ 2) \), this is not possible with the OBJ evaluation strategy which would realize a computation equivalent to the following infinite \( \mu_{\psi'} \)-sequence:

\[
\text{first}(5, \text{from}(0)) \leftrightarrow_{\mu_{\psi'}} \text{first}(5, 0: \text{from}(1)) \\
\leftrightarrow_{\mu_{\psi'}} \text{first}(5, 0:1: \text{from}(2)) \\
\leftrightarrow_{\mu_{\psi'}} \ldots
\]

Thus, the \( \mu_{\psi'} \)-termination of \( \mathcal{R} \) (see Example 45) does not ensure \( \psi' \)-normalization. Moreover, even though \( \text{eval}_{\psi} \) is root-normalizing, now \( \text{eval}_{\psi'} \) is no longer root-normalizing (also in contrast to Proposition 15).

Still, by using the results in [Luc98a, Luc01a], we can prove that it is possible to obtain (using \( \psi \)) the value\(^{21}\) of any expression of the sort \( \text{Nat} \) without entering into infinite computations. For instance, it is possible to evaluate \( \text{sel}(s(0), \text{from}(0)) \) to \( s(0) \), i.e., \( s(0) \in \text{eval}_{\psi}(\text{sel}(s(0), \text{from}(0))) \).

Similar kinds of annotations have been utilized in term (graph) rewriting [FKW00, KW95, Mar90,Ngu01, Pol01]. They have mainly been used to define restrictions of rewriting that permit the implementation of lazy reductions via eager rewritings in a transformed TRS [FKW00, KW95, Ngu01]. In [Luc98a, Luc01b, Luc02a] we have analyzed how these proposals relate to CSR.

11.3. Context-Sensitive Rewriting and Evaluation Strategies of ELAN

Most computational systems whose operational principle is based on reduction (e.g., functional, algebraic, and equational programming languages as well as theorem provers based on rewriting techniques) incorporate a predefined reduction strategy which is used to break down the nondeterminism which is inherent to reduction relations. The ELAN system provides an environment for specifying and prototyping deduction systems in a language based on rules controlled by strategies [BKKMR98]. In the context of rewriting, user-defined strategies were first introduced in ELAN [BKK98].

In ELAN, there are labeled and unlabeled rules. The operational semantics of ELAN takes advantage of this difference: the evaluation of a term proceeds in two steps [BCDK+00]:

1. First, a leftmost-innermost reduction strategy is applied to attempt the normalization with respect to the unlabeled rules. The user is recommended to provide a confluent and terminating unlabeled rewrite system in order to ensure termination and unicity of the result.

2. As for the normalized term (with respect to unlabeled rules), one first tries to apply a labeled rule following the strategy described in the logic description. This leads to a possibly empty collection of terms. If this set is empty, the evaluation backtracks to the last choice point; if it is not empty, then the evaluation continues after setting a new choice point and evaluating one of the returned terms by starting from the first step.

According to this description, the user can completely control the evaluation by specifying an adequate strategy only when the set of nonlabeled rules is empty. ELAN provides a language for the definition of strategies whose semantics is given in a functional style: a strategy is considered to be a mapping from terms to sets of terms which are obtained as a consequence of the application of the rewriting steps indicated by the strategy (see [BKK98]). The application of a strategy \( \varsigma \) to a term \( t \) is denoted by \( [\varsigma](t) \). If \( [\varsigma](t) = \emptyset \), we say that the strategy \( \varsigma \) fails on \( t \). The most elementary strategy, called a primal strategy, is a rewrite rule; a rule can be considered as a function which maps a term to its reduct at the top position [BKK98]. Actually, if different rules share the same label, then such a label can be also considered a mapping from terms to sets of terms (each of which comes from different rules). According to this, ELAN incorporates a rich suite of primitive strategies as well as operators for combining them (see [BKKMR98, BCDK+00]). For instance, given the strategies \( \varsigma_1, \ldots, \varsigma_k \) and a symbol \( f \) of the

\(^{21}\) We mean a term built from constructor symbols, rather than just a normal form.
signature, the strategy \( f(\varsigma_{1}, \ldots, \varsigma_{k}) \) is defined as [KKV95, BKK98]

\[
[f(\varsigma_{1}, \ldots, \varsigma_{k})](t) = \begin{cases} 
 f(\varsigma_{1}(t_{1}), \ldots, \varsigma_{k}(t_{k})) & \text{if } t = f(t_{1}, \ldots, t_{k}) \\
 \emptyset & \text{if } \text{root}(t) \neq f,
\end{cases}
\]

where \( f \) is overloaded. In particular, when it is applied to sets of terms \( S_{1}, \ldots, S_{k} \), we have that \( f(S_{1}, \ldots, S_{k}) = \{f(s_{1}, \ldots, s_{k}) \mid s_{1} \in S_{1}, \ldots, s_{k} \in S_{k}\} \). The \texttt{first} operator is applied to a list of strategies and selects the first strategy which does not fail among its arguments and returns all of its results. \texttt{id} is the identity strategy that does nothing and never fails. Strategies can also be defined by means of rewrite rules involving terms built from the preceding operators.

It is possible to specify context-sensitive rewriting strategies as an ELAN strategy. We exemplify the procedure by defining a \textit{leftmost-innermost restricted strategy}.

**Example 47.** Consider the following ELAN specification of the TRS in Example 1 (symbols for lists are included for better comprehension; \texttt{first} is renamed as \texttt{fst} as \texttt{first} is used by ELAN itself)

```plaintext
module restrictedLeftmostInnermost

// The module identity is part of the standard library

import
  local identity[X];
end

sort nat listNat; end

operators
  global
    zero : nat;
    nil : listNat;
    s (@) : (nat) nat;
    from (@) : (nat) listNat;
    @ . @ : (nat listNat) listNat;
    sel (@,@) : (nat listNat) nat;
    fst (@,@) : (nat listNat) listNat;
end

stratop
  global
    ev-nat : <nat -> nat>
    ev-listNat: <listNat -> listNat>
end

rules for listNat
  x,y: nat;
  z : listNat;
  global
    [ruleLNat] from(x) => x.from(s(x)) end
    [ruleLNat] fst(zero,z) => nil end
    [ruleLNat] fst(s(x),y.z) => y.fst(x,z) end
end

rules for nat
  x,y: nat;
  z : listNat;
  global
    [ruleNat] sel(zero,x.z) => x end
    [ruleNat] sel(s(x),y.z) => sel(x,z) end
```
end
strategies for nat
implicit
    [.] rli-nat => first([s(rli-nat), sel(rli-nat, id),
                          sel(id, rli-listNat), ruleNat]) end
end
strategies for listNat
implicit
    [.] rli-listNat => first([rli-nat.id, fst(rli-nat, id),
                             fst(id, rli-listNat), ruleLNat]) end
end

This specification corresponds to the leftmost-innermost evaluation strategy restricted to \( \mu \)-replacing redexes, where \( \mu(\text{from}) = \emptyset \), \( \mu(\text{sel}) = \mu(.) = \{1\} \), and \( \mu(\text{fst}) = \mu(\text{fst}) = \{1, 2\} \). Note that \( \mu \in \text{CM}_R \).

This strategy fails if no \( \mu \)-replacing redex is available for reductions.

The definition is easy to understand (see a similar definition for leftmost-outermost reductions in \( \lambda \)-calculus in [BKK98]): every argument of a symbol of the considered sort is considered for reduction, starting from left to right and skipping nonreplacing arguments. Notice that, in some cases (for instance, for the operator \text{from} which has no replacing argument), there is no strategy \text{from}(rli-nat) within the list associated to rli-listNat. Since reductions in the argument of \text{from} are not allowed, it is not necessary to include the argument. Thus, this reflects the fact that \( \mu(\text{from}) = \emptyset \). A similar remark applies to “.”; no component id.rli-listNat is needed for defining rli-listNat since reductions on the second argument of “.” are not allowed. Hence, this reflects the fact that \( \mu(.) = \{1\} \). Finally, the system attempts to apply every rule of the considered sort on the top position.

The theory of CSR can be also used to study the computational properties of this ELAN-strategy: Following the discussion at the end of Section 11.2, we conclude that it is possible to use this ELAN-strategy to obtain the complete evaluation of expressions of the sort nat without entering into an infinite computation. For instance, the evaluation of \text{sel}(s(\text{zero}), \text{from}(\text{zero})) to \text{s(\text{zero})} would not be possible by using the default (unrestricted) leftmost-innermost evaluation strategy of ELAN since it leads to an infinite computation.

12. CONCLUSIONS AND FUTURE WORK

We have investigated the main computational properties of \( \mu \)-normal forms regarding root-normalization and normalization. We proved that, for left-linear TRSs \( \mathcal{R} \) and canonical replacement maps \( \mu \in \text{CM}_R \):

1. the \( \mu \)-normal forms are strongly root-stable. This refines a previous result that identified such \( \mu \)-normal forms as root-stable terms [Luc98a]. Indeed, this fact can be viewed now as a consequence of the results established in this paper.

2. the confluence of a (left-linear) TRS does not ensure the unicity of the \( \mu \)-normal forms of a term \( t \), but it does ensure that the maximal replacing context of such \( \mu \)-normal forms is unique.

3. (unrestricted) reducts of \( \mu \)-normal forms are \( \mu \)-normal forms, and the maximal replacing context of a \( \mu \)-normal form remains unchanged under further (unrestricted) reductions.

We have formalized the notion of context-sensitive rewriting strategy. We have investigated the definition, properties, and use of context-sensitive rewriting strategies (or \( \mu \)-strategies, for a given replacement map \( \mu \)). The effective definition of \( \mu \)-normalizing \( \mu \)-strategies relies on the notions of root-normalization, and root-neededness [Mid97] and its decidable approximations [Luc98b]. This provides a measure for the efficiency of context-sensitive strategies, by taking the theory of root-needed reductions as a reference. We have proven that, whenever \( \mu \in \text{CM}_R \), every orthogonal TRS \( \mathcal{R} \) admits a one-step \( \mu \)-normalizing strategy. Moreover, for almost orthogonal NV-sequential TRSs, such
strategies can be effectively given. We have also shown that the restricted parallel outermost \( \mu \)-strategy is \( \mu \)-normalizing for every orthogonal TRS \( R \) (whenever \( \mu \in CM_R \)).

Finally, we have shown how to use \( \mu \)-normalizing strategies for defining efficient normalizing and infinitary normalizing strategies. These results can be summarized as follows: (1) each left-linear, confluent, TRS \( R \) which has a \( \mu \)-normalizing \( \mu \)-strategy (in particular \( \mu \)-terminating TRSs) admits a (one-step) normalizing strategy (where \( \mu \in CM_R \) is assumed), and (2) every left-linear, \( \mu \)-normalizing TRS \( R \) which admits a \( \mu \)-normalizing strategy also admits an infinitary normalizing strategy. In both cases, optimality of the underlying \( \mu \)-strategy is also inherited by the induced normalizing or infinitary normalizing strategy.

The theory is applied to define (infinitary) normalizing strategies for TRSs which do not admit a (infinitary) normalizing strategy based on the usual techniques for doing so. We also apply the theory to the analysis of computational properties of the evaluation strategies used in OBJ or ELAN. We believe that our work makes a contribution to the practical use and understanding of these languages.

We conclude by summarizing a number of results presented in the paper which complement or improve some results of the standard theory of rewriting:

1. We refine Middeldorp’s result: “every non-root-stable term has a root-needed redex” by considering \( \mu \)-replacing redexes. This means that \( cs \)-restrictions provide a first, simple, and correct bound to root-neededness.

2. Since \( mv \)-sequential indices have been proved to be (currently) the best decidable approximation to root-neededness (see [Luc98b]), we are able to refine this result by taking into account \( \mu \)-replacing \( mv \)-indices which suffice for approximating root-needed redexes (without losing any \( mv \)-index).

3. We have demonstrated that, whenever \( \mu \in CM_R \), left-linear \( \mu \)-terminating TRSs are infinitary normalizing.

4. We have shown that the use of computational restrictions of rewriting such as CSR can be helpful for defining normalizing strategies. Moreover, we have presented two complementary approaches to achieve this: (1) the analysis of termination of the computational restriction (that, for CSR, is formally analogous to the standard case, see [Zan97] for an abstract description), and (2) the definition of “good” strategies for the computational restriction (which, for CSR, we have based on the notions of root-normalization and root-neededness). We believe this is an interesting link between the two approaches (and theoretical fields) which has probably not yet been sufficiently explored.

The problem of implementing computational systems based on rewriting redexes pointed by strong indices was considered in [Dur94, HL79, O’Do95, Str89]. Unfortunately, no comparable effort has been devoted to \( mv \)-indices. In practice, strong index reduction strategies are not implemented by means of \( \Omega \)-reduction, i.e., \( \Omega \)-reduction is not used for identifying redexes that should be reduced at each computation step. Instead, there are two main approaches. The first one is to describe a class of TRSs for which the calculus of strong indices is very simple; for instance, in weakly orthogonal, left-normal TRSs (i.e., TRSs where the function and constant symbols occur to the left of variables in all left-hand sides [BN98]), the leftmost-outermost redex of any term is addressed by a strong index [O’Do85, Toy92]. The second approach is to provide an adequate data structure which is able to combine the pattern matching operation with the search for a strong index, thus finding not only a redex within a term but more precisely a redex addressed by a strong index (e.g., matching dags [HL79], index trees [Dur94, Str89], or even definitional trees [Ant92] for constructor systems [HLM98]). As for future work, we plan to systemically adapt these methods to ease the specification and implementation of context-sensitive rewriting strategies in our setting.

ACKNOWLEDGMENTS

I thank María Alpuente, Bernhard Gramlich, and Aart Middeldorp for their comments on preliminary versions of this paper. I also thank the anonymous referees for their helpful remarks.

REFERENCES


Friedman, D. P., and Wise, D. S. (1976), CONS should not evaluate its arguments,


Ferreira, M. C. F., and Ribeiro, A. L. (1999), Context-sensitive AC-rewriting,


