

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

Abstract

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.

Keywords: 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 $\mu : \mathcal{F} \rightarrow \mathcal{P}(\mathbb{N})$, called the *replacement map*, discriminates some argument positions $\mu(f) \subseteq \{1, \dots, k\}$ for each k -ary symbol f . Given a function call $f(t_1, \dots, t_k)$, 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.

*Work partially supported by CICYT TIC2001-2705-C03-01, Acciones Integradas HI 2000-0161, HA 2001-0059, HU 2001-0019, and Generalitat Valenciana GV01-424.

Example 1 Consider the TRS \mathcal{R} :

$$\begin{array}{llll} \text{sel}(0, x : y) & \rightarrow x & \text{first}(0, x) & \rightarrow [] \\ \text{sel}(s(x), y : z) & \rightarrow \text{sel}(x, z) & \text{first}(s(x), y : z) & \rightarrow y : \text{first}(x, z) \\ \text{from}(x) & \rightarrow x : \text{from}(s(x)) & & \end{array}$$

together with the replacement map

$$\mu(s) = \mu(:) = \mu(\text{from}) = \{1\} \quad \text{and} \quad \mu(\text{sel}) = \mu(\text{first}) = \{1, 2\}.$$

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

$$\begin{aligned} \text{sel}(s(0), \text{from}(s(0))) & \rightarrow \underline{\text{sel}(s(0), s(0) : \text{from}(s(s(0))))} \\ & \rightarrow \text{sel}(0, \text{from}(s(s(0)))) \\ & \rightarrow \underline{\text{sel}(0, s(s(0)) : \text{from}(s(s(0))))} \\ & \rightarrow s(s(0)) \end{aligned}$$

However, the infinite (meaningless) derivation

$$\begin{aligned} \text{sel}(s(0), \text{from}(s(0))) & \rightarrow \text{sel}(s(0), s(0) : \underline{\text{from}(s(s(0)))}) \\ & \rightarrow \text{sel}(s(0), s(0) : s(s(0)) : \underline{\text{from}(s(s(s(0)))})} \\ & \rightarrow \dots \end{aligned}$$

is avoided since $\mu(:) = \{1\}$ (the second argument of ‘:’ cannot be rewritten). For instance, the second reduction step is not allowed with CSR.

Context-sensitive computations under a replacement map μ obtain (at most) μ -normal forms, i.e., terms which cannot be further μ -rewritten. In general, the μ -normal forms of a TRS \mathcal{R} strictly include its normal forms (e.g., the subterm $s(0) : \text{from}(s(s(0)))$ which appears in Example 1 is a μ -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 forms¹. 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 values² have been established in [Luc98a]. In fact, given a TRS \mathcal{R} we are able to *automatically* provide replacement maps supporting such computations. In this setting, the *canonical replacement map* (denoted by $\mu_{\mathcal{R}}^{\text{can}}$) is specially important, as it specifies the

¹See, 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.

²Here, 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.

most restrictive replacement map which can be (automatically) associated to a TRS \mathcal{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) non-variable subterms of the left-hand sides of the rules of \mathcal{R} are replacing.

For instance, the replacement map μ in Example 1 is, in fact, *less* restrictive than the canonical replacement map $\mu_{\mathcal{R}}^{can}$ of TRS \mathcal{R} in the example (see Example 12 below). With the replacement map μ 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 *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

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

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

(terms n) \equiv **(cons (reciprocal (square n)) (terms (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].

Example 2 *With the TRS:*

sqr (0) \rightarrow 0	0 + x \rightarrow x
sqr (s(x)) \rightarrow s(sqr (x)+ dbl (x))	s (x) + y \rightarrow s(x+y)
dbl (0) \rightarrow 0	
dbl (s(x)) \rightarrow s(s(dbl (x)))	
terms (n) \rightarrow recip (sqr (n)):terms(s(n))	

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

Moreover, by adding the two rules for function **first** of Example 1:

first(0,x) \rightarrow []
first(s(x),y:z) \rightarrow y:first(x,z)

we can obtain (as `first(n,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 μ (with $\mu(\cdot) = \{1\}$). Then, \mathcal{R}_L^μ :

$$\begin{array}{llll} \text{sqr}(0) & \rightarrow 0 & 0 + x & \rightarrow x \\ \text{sqr}(s(x)) & \rightarrow s(\text{sqr}(x) + \text{dbl}(x)) & s(x) + y & \rightarrow s(x+y) \\ \text{dbl}(0) & \rightarrow 0 & \text{first}(0, x) & \rightarrow [] \\ \text{dbl}(s(x)) & \rightarrow s(\text{dbl}(x)) & \text{first}(s(x), : (y)) & \rightarrow : (y) \\ \text{terms}(n) & \rightarrow : (\text{recip}(\text{sqr}(n))) & & \end{array}$$

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

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

Friedman and Wise also use replacement restrictions to provide alternative (more efficient) definitions to logical connectives **and**, **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 **false** (resp. **true**) if connective **and** (resp. **or**) is considered (see page 277 of [FW76]). This is implemented (without using lists) with the TRS:

$$\begin{array}{ll} \text{and}(\text{false}, x) & \rightarrow \text{false} & \text{or}(\text{true}, x) & \rightarrow \text{true} \\ \text{and}(\text{true}, x) & \rightarrow x & \text{or}(\text{false}, x) & \rightarrow x \end{array}$$

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 languages³ 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 annotation⁴ ([FGJM85], Section 4.4, [GWMFJ00], Section 2.4.4)

```
op if_then_else_fi : Bool Int Int -> Int [strat (1 0)]
```

³As in [GWMFJ00], by OBJ we mean OBJ2, OBJ3, CafeOBJ, or Maude.

⁴A more precise and general definition can be found in OBJ3’s standard prelude, see Appendix D.3 of [GWMFJ00].

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 `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 (*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 [HPF99]. Since there are a number of overheads in the implementation of this feature (see [Pey87]), lazy functional languages like Gofer [Jon92] and Haskell [HPW92] allow for syntactic annotations on the arguments of datatype constructors, thus allowing an immediate evaluation.

Example 4 *The following definition in Haskell:*

```
data List a = Nil | Cons !a (List a)
```

declares a (polymorphic) type List a whose binary data constructor Cons evaluates the first argument (of type a). This is specified by using the symbol ‘!’ in the first argument of Cons.

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

Example 5 *The following specification*

```
if :: !Bool a a -> a
if True  x y = x
if False x y = y
```

is an annotated definition of the function if which forces the evaluation of the first argument of each if call (see the mark ‘!’ in the type declaration of if; this is called a 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 $f(t_1, \dots, t_k)$ 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 *secondary* role in the global execution mechanism: an underlying strategy is assumed. Program annotations are usually obtained

from some kind of strictness⁵ 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. 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, 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 $\mu_{\mathcal{R}}^{can}$) the (exclusive) use of strictness information for defining a replacement map (for instance, by letting $i \in \mu_{\mathcal{R}}^{str}(f)$ 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 $\mu_{\mathcal{R}}^{can}$ and $\mu_{\mathcal{R}}^{str}$.

Example 6 Consider the function \mathbf{f} defined by

$$\mathbf{f}(\mathbf{x}) \rightarrow \mathbf{x}$$

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

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

⁵Let D_1, \dots, D_k, D be ordered sets with least elements $\perp_1, \dots, \perp_k, \perp$ respectively, expressing undefinedness. A mapping $f : D_1 \times \dots \times D_k \rightarrow D$ is said to be strict in its i -th argument if $f(d_1, \dots, \perp_i, \dots, d_k) = \perp$ for all $d_1 \in D_1, \dots, d_k \in D_k$. See section 3.3.3 [Luc98a] for connections between strictness and *CSR*.

sequence), then it is also μ -terminating (no term initiates an infinite μ -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 sequentiality, 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, etc.).

In order to formalize the claim that ‘*CSR can be used to improve the termination behavior*’, we are going to show that μ -normalization (i.e., the computation of μ -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 μ -rewrite sequence by proving μ -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 μ -rewrite sequences starting from terms having a μ -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 μ -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 $\mu_{\mathcal{R}}^{can}$ -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 μ -strategies for a given replacement map μ), 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 μ -normalizing μ -strategies, i.e., μ -strategies that obtain a μ -normal form of a term whenever it exists (of course, without issuing infinite μ -rewrite sequences, even though they exist): We prove that,

for every left-linear, confluent TRS \mathcal{R} , every $\mu_{\mathcal{R}}^{can}$ -normalizing $\mu_{\mathcal{R}}^{can}$ -strategy for \mathcal{R} induces a normalizing strategy for \mathcal{R} .

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

Every μ -strategy is trivially μ -normalizing for μ -terminating TRSs. However, whenever a replacement map μ cannot ensure μ -termination (or we fail to prove that it can), we need to provide μ -normalizing μ -strategies. A well-known theory for defining (efficient) normalizing strategies is Huet and Lévy's theory of needed reductions⁶ [HL79, HL91]. However, reduction of needed redexes is not adequate for defining μ -normalizing μ -strategies. This is because there are terms that have a μ -normal form but have no normal form. Since each redex in a term which does not have a normal form is needed, neededness is not useful for discriminating the redexes which should be contracted to μ -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, μ -normal forms are root-stable [Luc98a], Middeldorp's theory provides an appropriate framework for the definition of μ -normalizing μ -strategies which we can summarize as follows:

orthogonal TRSs \mathcal{R} admit a (one-step) $\mu_{\mathcal{R}}^{can}$ -normalizing $\mu_{\mathcal{R}}^{can}$ -strategy.

This result does *not* provide an immediate 'operative' definition of $\mu_{\mathcal{R}}^{can}$ -normalizing $\mu_{\mathcal{R}}^{can}$ -strategies. This is because both root-stability and root-neededness are undecidable. By using the decidable notion of $\mu_{\mathcal{R}}^{can}$ -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 μ -normalizing μ -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) $\mu_{\mathcal{R}}^{can}$ -normalizing $\mu_{\mathcal{R}}^{can}$ -strategy.

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

⁶A *needed* redex of a term t is a redex which is contracted (either itself or its descendants) in every rewrite sequence which normalizes t .

1. Characterization of μ -normal forms and the μ -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 μ -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).

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, KJo92].

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, \dots, a_n of elements taken from A such that $a_i R a_{i+1}$ 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 $a R^* b$; 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 $a_1 R a_2 R a_3 \dots$. Obviously, terminating relations are normalizing.

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

Terms are viewed as labelled trees in the usual way. Positions p, q, \dots are represented by chains of positive natural numbers which are used to address subterms of t . We denote the empty chain by Λ . 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 $\mathcal{Pos}(t)$. Given terms t and s , $\mathcal{Pos}_s(t)$ denotes the set of positions of s in t , i.e., $p \in \mathcal{Pos}_s(t)$ iff $t|_p = s$. Positions of non-variable symbols in t are denoted as $\mathcal{Pos}_{\mathcal{F}}(t)$ and $\mathcal{Pos}_{\mathcal{X}}(t)$ 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 labelling the root of t is denoted as $root(t)$. The chain of symbols lying on positions above/on $p \in \mathcal{Pos}(t)$ is $prefix_t(\Lambda) = root(t)$, $prefix_t(i.p) = root(t).prefix_t(p)$. The strict prefix $sprefix$ is $sprefix_t(\Lambda) = \Lambda$, $sprefix_t(p.i) = prefix_t(p)$. A *context* is a term $C \in \mathcal{T}(\mathcal{F} \cup \{\square\}, \mathcal{X})$ 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 $C[t_1, \dots, t_n]$ to denote the term obtained by filling the holes of a context $C[\]$ with terms t_1, \dots, 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 $l, r \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, $l \notin \mathcal{X}$ 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 pair $\mathcal{R} = (\mathcal{F}, R)$ where R is a set of rewrite rules. $L(\mathcal{R})$ denotes the *lhs*’s of \mathcal{R} . Given a substitution σ , an instance $\sigma(l)$ of a *lhs* l of a rule is a redex. The set of redex positions in t is $\mathcal{Pos}_{\mathcal{R}}(t) = \{p \in \mathcal{Pos}(t) \mid \exists l \in L(\mathcal{R}) : t|_p = \sigma(l)\}$.

A TRS \mathcal{R} is left-linear if for all $l \in L(\mathcal{R})$, 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 non-variable position $p \in \mathcal{Pos}_{\mathcal{F}}(l)$ and a most-general unifier σ such that $\sigma(l|_p) = \sigma(l')$. The pair $\langle \sigma(l)[\sigma(r')]_p, \sigma(r) \rangle$ is called a critical pair and is also called an overlay if $p = \Lambda$. A critical pair $\langle t, s \rangle$ is trivial if $t = s$. The critical pairs of a TRS \mathcal{R} are the critical pairs between any two of its (renamed) rules; this includes overlaps of a rule with a renamed variant of itself, except at the root, i.e., if $p = \Lambda$. A left-linear TRS without critical pairs is called orthogonal. It is called almost orthogonal if its critical pairs are trivial overlays. If it only has trivial critical pairs, it is called weakly orthogonal.

A term t rewrites to s (at position p), written $t \xrightarrow{p}_{\mathcal{R}} s$ (or just $t \xrightarrow{p} s$, or $t \rightarrow_{\mathcal{R}} s$, or even $t \rightarrow s$) if $t|_p = \sigma(l)$ and $s = t[\sigma(r)]_p$, for some rule $l \rightarrow r \in R$, $p \in \mathcal{Pos}(t)$ and substitution σ . The one-step rewrite relation for \mathcal{R} is \rightarrow . A finite \rightarrow -sequence is called a rewrite sequence. If $t \rightarrow^* s$, then s is a reduct of t . The inner reduction relation is $\xrightarrow{\Lambda} = \rightarrow - \xrightarrow{\Lambda}$. A term t is root-stable if it cannot be rewritten to a redex. A term is said to be root-normalizing if it

has a root-stable reduct. In this paper, the \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 Ω -reduction

A new constant symbol Ω 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 Ω -terms and they are used to denote prefixes of terms. We denote by t_Ω the term t where all variables are replaced by Ω . Positions $p \in \mathcal{Pos}_\Omega(t)$ are said to be the Ω -positions of t . An ordering \leq on Ω -terms is given: $\Omega \leq t$ for all t , $x \leq x$ if $x \in \mathcal{X}$, and $f(\bar{t}) \leq f(\bar{s})$ if $t_i \leq s_i$ for all $1 \leq i \leq ar(f)$. 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 $s \leq u$. We note that, if $t \uparrow s$, there is a maximal context $C[\]$, such that $t = C[t_1, \dots, t_n]$, $s = C[s_1, \dots, s_n]$ 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^Ω the term t where all outermost redexes are replaced by Ω . An Ω -normal form is an Ω -term t such that $\mathcal{Pos}_\mathcal{R}(t) = \emptyset$ 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 $l \in L(\mathcal{R})$. Then, $t \geq l_\Omega$ iff t is a redex.*

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

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

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

$$\begin{array}{ll} \mathbf{f}(\mathbf{a}) & \rightarrow \mathbf{a} \\ \mathbf{b} & \rightarrow \mathbf{c} \end{array}$$

Then, we have,

$$\mathbf{f}(\mathbf{b}) \rightarrow_\Omega \mathbf{f}(\Omega)$$

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

$$\mathbf{f}(\Omega) \rightarrow_\Omega \Omega$$

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

Proposition 1 [KM91] *Let $\mathcal{R} = (\mathcal{F}, R)$ 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. \square

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

Example 8 *Consider the TRS of Example 7. The term $\mathbf{f}(\mathbf{b})$ 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 $C[\overline{\Omega}]$ is a rigid term.

Proposition 3 [KM91, Klo92] *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS. Every term $t \in \mathcal{T}_\Omega(\mathcal{F}, \mathcal{X})$ can be uniquely written as $t = C[t_1, \dots, t_n]$ such that $C[\Omega, \dots, \Omega]$ 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[\], \dots, C_n[\]$ be rigid contexts. Then, $C[C_1[\], \dots, C_n[\]]$ is a rigid context.*

PROOF. By contradiction. If $t = C[C_1[\], \dots, C_n[\]]$ is not rigid, since $C_1[\], \dots, C_n[\]$ are rigid, it follows that there is $p \in \text{Pos}_{\mathcal{F}}(C[\overline{\Omega}])$ such that $t|_p$ is compatible with l_Ω for some $l \in L(\mathcal{R})$. Therefore, since $C[\overline{\Omega}]|_p \leq t|_p$, we also get $C[\overline{\Omega}]|_p \uparrow l_\Omega$. Hence, $\omega(C[\overline{\Omega}]) < C[\overline{\Omega}]$ and $C[\overline{\Omega}]$ is not rigid. \square

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 $f \in \mathcal{F}$, $\mu(f) \subseteq \{1, \dots, \text{ar}(f)\}$. The replacement map μ determines the *argument positions* which can be reduced for each symbol in \mathcal{F} [Luc98a].

Example 9 *The conditional operator **if-then-else** can be given the following replacement map*

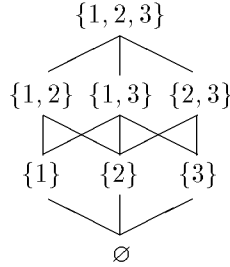
$$\mu(\mathbf{if-then-else}) = \{1\}$$

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 $(M_{\mathcal{F}}, \sqsubseteq, \mu_{\perp}, \mu_{\top}, \sqcup, \sqcap)$: The minimum element is μ_{\perp} , given by $\mu_{\perp}(f) = \emptyset$ for all $f \in \mathcal{F}$; the maximum element is μ_{\top} , given by $\mu_{\top}(f) = \{1, \dots, ar(f)\}$ for all $f \in \mathcal{F}$; and the least upper bound (*lub*), \sqcup , and greatest lower bound, \sqcap , are given by $(\mu \sqcup \mu')(f) = \mu(f) \cup \mu'(f)$ and $(\mu \sqcap \mu')(f) = \mu(f) \cap \mu'(f)$ for all $f \in \mathcal{F}$. Thus, $\mu \sqsubseteq \mu'$ means that μ considers less positions than μ' for reduction. We also say that μ is more restrictive than (or equally restrictive to) μ' .

Example 10 *The following figure depicts the lattice of replacing arguments that can be associated to symbol `if-then-else`:*



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

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

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

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

if $t \notin \mathcal{X}$. The *non- μ -replacing* positions⁷ are $\overline{\mathcal{Pos}^{\mu}}(t) = \mathcal{Pos}(t) - \mathcal{Pos}^{\mu}(t)$. By abuse, the occurrence of subterm $t|_p$ at position p is called replacing (resp. non-replacing) if $p \in \mathcal{Pos}^{\mu}(t)$ (resp. $p \in \overline{\mathcal{Pos}^{\mu}}(t)$).

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

The non- μ -replacing positions never lie above replacing ones.

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

⁷In [Zan97], Zantema uses ‘active/forbidden’ instead of μ -replacing/non- μ -replacing. However, this terminology has also been used elsewhere for formalizing different (but related) rewriting restrictions [FKW00, Luc02a].

Hence, the (ordered) set of replacing positions $(\mathcal{Pos}(t), \leq)$ is *downward closed*: for all $p \in \mathcal{Pos}^\mu(t)$, $q \leq p \Rightarrow q \in \mathcal{Pos}^\mu(t)$. 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 $p \in \mathcal{Pos}(t) \cap \mathcal{Pos}(s)$ and $\text{sprefix}_t(p) = \text{sprefix}_s(p)$, then $p \in \mathcal{Pos}^\mu(t) \Leftrightarrow p \in \mathcal{Pos}^\mu(s)$.*

In context-sensitive rewriting (CSR), we rewrite subterms at *replacing* positions: t μ -rewrites to s , written

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

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

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

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

However,

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

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

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

3.1 Canonical context-sensitive rewriting

The *canonical replacement map* for a TRS \mathcal{R} is⁸ [Luc98a]:

the most restrictive replacement map which ensures that the (positions of) non-variable subterms of the left-hand sides of the rules of \mathcal{R} are replacing.

Note that $\mu_{\mathcal{R}}^{\text{can}}$ can be automatically associated to \mathcal{R} by means of a very simple calculus: $\forall f \in \mathcal{F}, i \in \{1, \dots, \text{ar}(f)\}$,

$$i \in \mu_{\mathcal{R}}^{\text{can}}(f) \quad \text{iff} \quad \exists l \in L(\mathcal{R}), p \in \mathcal{Pos}_{\mathcal{F}}(l), (\text{root}(l|_p) = f \wedge p.i \in \mathcal{Pos}_{\mathcal{F}}(l))$$

⁸In [Luc98a], we used $\mu_{\mathcal{R}}^{\text{com}}$ instead of $\mu_{\mathcal{R}}^{\text{can}}$ to denote the canonical replacement map. We believe that the new notation is clearer.

Example 12 Consider the TRS \mathcal{R} of Example 1. Since we have the rules

$$\begin{aligned}\text{first}(0, \mathbf{x}) &\rightarrow \square \\ \text{first}(\mathbf{s}(\mathbf{x}), \mathbf{y}:\mathbf{z}) &\rightarrow \mathbf{y}:\text{first}(\mathbf{x}, \mathbf{z})\end{aligned}$$

we have $1 \in \mu_{\mathcal{R}}^{\text{can}}(\text{first})$ because, e.g., $\text{first}(0, \mathbf{x})|_1 = 0 \notin \mathcal{X}$; and $2 \in \mu_{\mathcal{R}}^{\text{can}}(\text{first})$ because $\text{first}(\mathbf{s}(\mathbf{x}), \mathbf{y}:\mathbf{z})|_2 = \mathbf{y}:\mathbf{z} \notin \mathcal{X}$. On the other hand, since we have the rules

$$\begin{aligned}\text{sel}(0, \mathbf{x}:\mathbf{y}) &\rightarrow \mathbf{x} \\ \text{sel}(\mathbf{s}(\mathbf{x}), \mathbf{y}:\mathbf{z}) &\rightarrow \text{sel}(\mathbf{x}, \mathbf{z})\end{aligned}$$

we have $\text{sel}(0, \mathbf{x}:\mathbf{y})|_1 = 0 \notin \mathcal{X}$ and $\text{sel}(0, \mathbf{x}:\mathbf{y})|_2 = \mathbf{x}:\mathbf{y} \notin \mathcal{X}$, i.e., $\mu_{\mathcal{R}}^{\text{can}}(\text{sel}) = \{1, 2\}$. Finally, $\mu_{\mathcal{R}}^{\text{can}}(f) = \emptyset$ for $f \in \{\mathbf{s}, \text{from}, :\}$. Therefore,

$$\mu_{\mathcal{R}}^{\text{can}}(\text{first}) = \mu_{\mathcal{R}}^{\text{can}}(\text{sel}) = \{1, 2\} \quad \text{and} \quad \mu_{\mathcal{R}}^{\text{can}}(\mathbf{s}) = \mu_{\mathcal{R}}^{\text{can}}(:) = \mu_{\mathcal{R}}^{\text{can}}(\text{from}) = \emptyset$$

Example 13 Consider the TRS \mathcal{R} :

$$\begin{aligned}\text{half}(0) &\rightarrow 0 & \text{half}(\mathbf{s}(\mathbf{s}(\mathbf{x}))) &\rightarrow \mathbf{s}(\text{half}(\mathbf{x})) \\ \text{half}(\mathbf{s}(0)) &\rightarrow 0\end{aligned}$$

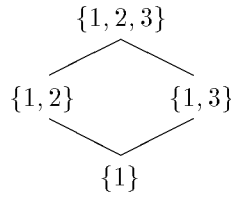
Then, we have $\mu_{\mathcal{R}}^{\text{can}}(\text{half}) = \{1\}$, because, e.g., $\text{half}(0)|_1 = 0 \notin \mathcal{X}$. On the other hand, $\mu_{\mathcal{R}}^{\text{can}}(\mathbf{s}) = \{1\}$ because, e.g., $\text{root}(\text{half}(\mathbf{s}(0))|_1) = \mathbf{s}$ and $\text{half}(\mathbf{s}(0))|_{1.1} = 0 \notin \mathcal{X}$.

Given a TRS \mathcal{R} , $CM_{\mathcal{R}} = \{\mu \in M_{\mathcal{R}} \mid \mu_{\mathcal{R}}^{\text{can}} \sqsubseteq \mu\}$ is the set of replacement maps which are less restrictive than or equally restrictive to $\mu_{\mathcal{R}}^{\text{can}}$. By abuse, if $\mu \in CM_{\mathcal{R}}$, we say that μ is a canonical replacement map of \mathcal{R} . We say that \hookrightarrow_{μ} is a canonical context-sensitive rewrite relation if $\mu \in CM_{\mathcal{R}}$. It is not difficult to see that $(CM_{\mathcal{R}}, \sqsubseteq, \mu_{\mathcal{R}}^{\text{can}}, \mu_{\top}, \sqcup, \sqcap)$ is a complete sublattice of $(M_{\mathcal{R}}, \sqsubseteq, \mu_{\perp}, \mu_{\top}, \sqcup, \sqcap)$.

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

$$\begin{aligned}\text{if true then } \mathbf{x} \text{ else } \mathbf{y} &\rightarrow \mathbf{x} \\ \text{if false then } \mathbf{x} \text{ else } \mathbf{y} &\rightarrow \mathbf{y}\end{aligned}$$

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:



Canonical replacement maps make CSR complete for root-normalization.

Proposition 7 [Luc98a] Let \mathcal{R} be a left-linear TRS, $l \in L(\mathcal{R})$ and $\mu \in CM_{\mathcal{R}}$. If $t \rightarrow^* \sigma(l)$ for some substitution σ , then there is a substitution θ such that $t \hookrightarrow^* \theta(l)$ and $\theta(x) \rightarrow^* \sigma(x)$ for all $x \in \text{Var}(l)$.

Theorem 1 [Luc98a] *Let $\mathcal{R} = (\mathcal{F}, R)$ be a left-linear TRS and $\mu \in CM_{\mathcal{R}}$. Let $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and s be a root-stable term. If $t \rightarrow^* s$, then there exists s' such that $t \hookrightarrow^* s'$, $\text{root}(s) = \text{root}(s')$, and $s' \xrightarrow{\Lambda}^* s$.*

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

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

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

4 Characterization of μ -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 μ -replacing in t .

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

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

If $C[] = MRC^{\mu}(t)$, then $t = C[t_1, \dots, t_n]$ is such that

$$\{p_1, \dots, p_n\} = \text{minimal}(\overline{\mathcal{Pos}^{\mu}(t)})$$

are the positions of t_1, \dots, t_n in t , i.e., $t_i = t|_{p_i}$ for $1 \leq i \leq n$. Downward closedness of $\mathcal{Pos}^{\mu}(t)$ in $(\mathcal{Pos}(t), \leq)$, together with Proposition 6, implies that every subterm of $MRC^{\mu}(t)$ is μ -replacing, except for the holes in the context. This justifies the name ‘maximal replacing context’.

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

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 μ -normal forms are rigid contexts.

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

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

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

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} :*

$$\begin{array}{lcl} \mathbf{f}(\mathbf{x}, \mathbf{x}) & \rightarrow & \mathbf{c}(\mathbf{x}) \\ \mathbf{a} & \rightarrow & \mathbf{b} \end{array}$$

and μ be given by $\mu(\mathbf{f}) = \{1, 2\}$ and $\mu(\mathbf{c}) = \emptyset$. Note that $\mu_{\mathcal{R}}^{can}(\mathbf{f}) = \mu_{\mathcal{R}}^{can}(\mathbf{c}) = \emptyset$; thus, $\mu \in CM_{\mathcal{R}}$. Hence, $\mathbf{f}(\mathbf{c}(\square), \mathbf{c}(\square))$ is the maximal μ -replacing context of $\mathbf{f}(\mathbf{c}(\mathbf{a}), \mathbf{c}(\mathbf{b}))$. However, despite the fact that $\mathbf{f}(\mathbf{c}(\mathbf{a}), \mathbf{c}(\mathbf{b}))$ is a μ -normal form, the term $\mathbf{f}(\mathbf{c}(\Omega), \mathbf{c}(\Omega))$ is neither rigid, nor strongly root-stable, since $\mathbf{f}(\mathbf{c}(\Omega), \mathbf{c}(\Omega)) \rightarrow_{\Omega} \Omega$. Note that variables in the left-hand side of the non-left-linear rule are μ -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 $\mu(\mathbf{first}) = \emptyset$, we have $\mu \notin CM_{\mathcal{R}}$. Then, $\mathbf{first}(t, \mathbf{x})$ is a μ -normal form for any redex t . The maximal replacing context is $\mathbf{first}(\square, \square)$. However, $\mathbf{first}(\Omega, \Omega)$ is not rigid, since $\mathbf{first}(\Omega, \Omega) \uparrow \mathbf{first}(\mathbf{0}, \Omega)$. Hence, $\omega(\mathbf{first}(\Omega, \Omega)) = \Omega$.*

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

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

Maximal replacing contexts of μ -normal forms are stable parts of further reducts from those μ -normal forms (using unrestricted rewriting).

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

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

Corollary 1 Let \mathcal{R} be a left-linear TRS and $\mu \in \text{CM}_{\mathcal{R}}$. Every μ -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., $\mathbf{f}(\mathbf{c}(\mathbf{a}), \mathbf{c}(\mathbf{b}))$ in Example 15 and $\mathbf{first}(\mathbf{t}, \mathbf{x})$ in Example 16 are μ -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 $\mu \in \text{CM}_{\mathcal{R}}$. Every μ -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 $\mu \in \text{CM}_{\mathcal{R}}$. Every μ -normalizing term is root-normalizing.

4.1 μ -normalization

The following proposition establishes conditions to ensure that the set of μ -normal forms is closed under (unrestricted) rewriting.

Proposition 11 Let \mathcal{R} be a left-linear TRS, $\mu \in \text{CM}_{\mathcal{R}}$, and $t \in \text{NF}_{\mathcal{R}}^{\mu}$. If $t \rightarrow^* s$, then $s \in \text{NF}_{\mathcal{R}}^{\mu}$ and $\text{MRC}^{\mu}(t) = \text{MRC}^{\mu}(s)$.

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

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

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

Left-linearity cannot be dropped in Proposition 11.

Example 18 Consider the TRS \mathcal{R} in Example 15. Note that $\mathbf{f}(\mathbf{a}, \mathbf{b})$ is a $\mu_{\mathcal{R}}^{can}$ -normal form and that $\mathbf{f}(\underline{\mathbf{a}}, \mathbf{b}) \rightarrow \mathbf{f}(\mathbf{b}, \mathbf{b})$, which is not a μ -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 μ -normal form* property:

Example 19 Consider the confluent TRS \mathcal{R} :

$$\begin{array}{lcl} \mathbf{f}(\mathbf{x}, \mathbf{y}) & \rightarrow & \mathbf{c}(\mathbf{x}) \\ \mathbf{a} & \rightarrow & \mathbf{b} \end{array}$$

and the replacement map μ such that $\mu(\mathbf{f}) = \{1\}$ and $\mu(\mathbf{c}) = \emptyset$. Note that $\mu \in CM_{\mathcal{R}}$. However, we have the following μ -normalizing derivations:

$$\underline{\mathbf{f}(\mathbf{a}, \mathbf{b})} \hookrightarrow \mathbf{c}(\mathbf{a}) \quad \text{and} \quad \mathbf{f}(\underline{\mathbf{a}}, \mathbf{b}) \hookrightarrow \underline{\mathbf{f}(\mathbf{b}, \mathbf{b})} \hookrightarrow \mathbf{c}(\mathbf{b})$$

which yield different μ -normal forms $\mathbf{c}(\mathbf{a})$ and $\mathbf{c}(\mathbf{b})$.

Nevertheless, we have the following result.

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

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

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 $\mu(\mathbf{f}) = \{1\}$ (note that $\mu \in CM_{\mathcal{R}}$), we have the following (μ -) derivations:

$$\underline{\mathbf{f}(\mathbf{a}, \mathbf{a})} \hookrightarrow \mathbf{b} \quad \text{and} \quad \mathbf{f}(\underline{\mathbf{a}}, \mathbf{a}) \hookrightarrow \mathbf{f}(\mathbf{b}, \mathbf{a})$$

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

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

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

The unique μ -normal form property is ensured by μ -confluence, i.e., confluence of $\hookrightarrow_{\mathcal{R}(\mu)}$. The following proposition establishes that every (unrestricted) reduct of a μ -normalizing term is also μ -normalizing.

Proposition 13 *Let \mathcal{R} be a left-linear, confluent TRS and $\mu \in CM_{\mathcal{R}}$. If t is μ -normalizing and $t \rightarrow^* s$, then s is μ -normalizing.*

PROOF. Let u be a μ -normal form of t , i.e., $t \hookrightarrow^! u$. 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 μ -normal form. Let $C[\] = MRC^{\mu}(s')$. We prove, by induction on the structure of $C[\]$, that there exists a μ -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, $s \hookrightarrow^* s'$ and we let $v = s'$. If $C[\] = f(C_1[\], \dots, C_k[\])$, since $s' = f(s'_1, \dots, s'_k)$ is root-stable (Theorem 4), by Theorem 1 $s \hookrightarrow^* f(s''_1, \dots, s''_k)$ and $s''_i \rightarrow^* s'_i$ for $1 \leq i \leq k$. Since each s'_i for $i \in \mu(f)$ is a μ -normal form and $MRC^{\mu}(s'_i) = C_i[\]$, by the induction hypothesis there are μ -normal forms v_i such that $s''_i \hookrightarrow^! v_i$ for $i \in \mu(f)$. Hence, since $MRC^{\mu}(s')$ is rigid (Proposition 9), $v = f(v_1, \dots, v_k)$ (where $v_j = s'_j$ if $j \notin \mu(f)$) is a μ -normal form and $s \hookrightarrow^! v$. \square

Confluence alone is not sufficient to ensure Proposition 13.

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

$$\begin{array}{ll} \mathbf{f}(\mathbf{x}, \mathbf{x}) \rightarrow \mathbf{a} & \mathbf{b} \rightarrow \mathbf{b} \\ \mathbf{c} & \rightarrow \mathbf{b} \end{array}$$

and $\mu(\mathbf{f}) = \{1\}$. Note that $\mu \in CM_{\mathcal{R}}$. Term $\mathbf{f}(\mathbf{c}, \mathbf{c})$ has a μ -normal form:

$$\underline{\mathbf{f}(\mathbf{c}, \mathbf{c})} \hookrightarrow \mathbf{a}$$

but $\mathbf{f}(\mathbf{b}, \mathbf{c})$, which is a (μ) -reduct of $\mathbf{f}(\mathbf{c}, \mathbf{c})$, has no μ -normal form.

5 Context-sensitive rewriting strategies

A (non-deterministic) *rewriting strategy* for a TRS \mathcal{R} is a function S that assigns a non-empty set of non-empty 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* (non-deterministic) rewriting strategy for a TRS \mathcal{R} we mean a function S that assigns a non-empty set $S(t) \subseteq \mathcal{Pos}_{\mathcal{R}}(t)$ 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 non-empty sets of (non-empty) 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 $t \xrightarrow{p} s$ and $p \in S(t)$ for one-step strategies). An *S-sequence* is a reduction sequence of the form $t_1 \rightarrow_S t_2 \rightarrow_S \dots$. If $t \rightarrow_S^* s$, we say that s is an *S-reduct* of t .

A strategy S is *root-normalizing* if for all root-normalizing term t , every possible infinite S -sequence starting from t contains a root-stable term; S is *μ -normalizing* if for all μ -normalizing term t , every infinite S -sequence starting from t contains a μ -normal form whose S -reducts are always μ -normal forms⁹; S is *normalizing* if, for all normalizing term t , there is no infinite S -sequence starting from t .

Proposition 14 *Let \mathcal{R} be a left-linear TRS, $\mu \in CM_{\mathcal{R}}$, and S be a rewriting strategy such that every infinite S -sequence contains a μ -normal form. Then, S is μ -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 .

Computability: To provide an effective¹⁰ 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 performed¹¹.

⁹This is required to ensure that reaching a μ -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.

¹⁰Here, ‘effective’ is used in the sense of [KM91]: a reduction strategy S is *effective* if $S(t)$ can be computed from every term t .

¹¹Actually, 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]).

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 context-sensitive rewriting strategy.

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

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

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

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

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

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

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

Every μ -strategy is μ -normalizing for μ -terminating TRSs. Then, we have:

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

In general, we cannot extend Theorem 6 to non- μ -normalizing TRSs.

Example 22 *Consider the following (orthogonal) TRS \mathcal{R} :*

$$\begin{array}{lcl} \mathbf{f}(\mathbf{x}) & \rightarrow & \mathbf{g}(\mathbf{b}) \\ \mathbf{b} & \rightarrow & \mathbf{b} \end{array}$$

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

$$\mathbf{f}(\mathbf{b}) \hookrightarrow_{\mathbf{H}} \mathbf{f}(\mathbf{b}) \hookrightarrow_{\mathbf{H}} \dots$$

while the reduction step

$$\mathbf{f}(\mathbf{b}) \rightarrow \mathbf{g}(\mathbf{b})$$

root-normalizes $\mathbf{f}(\mathbf{b})$.

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

Every one-step μ -strategy \mathbf{H} extends to a one-step strategy $S_{\mathbf{H}}$ as follows:

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

The generalization to arbitrary μ -strategies \mathbf{H} is immediate:

$$S_{\mathbf{H}}(t) = \begin{cases} \mathbf{H}(t) & \text{if } t \notin \text{NF}_{\mathcal{R}}^{\mu} \\ C[S_{\mathbf{H}}(t_1), \dots, S_{\mathbf{H}}(t_n)] & \text{if } t \in \text{NF}_{\mathcal{R}}^{\mu} - \text{NF}_{\mathcal{R}}, \text{ where} \\ & C[\] = \text{MRC}^{\mu}(t) \text{ and } t = C[t_1, \dots, t_n] \\ \emptyset & \text{otherwise} \end{cases}$$

Here, for a given context $C[\]$ and sets of rewrite sequences S_1, \dots, S_n , issued from terms t_1, \dots, t_n , $C[S_1, \dots, S_n]$ is the set of sequences from $C[t_1, \dots, t_n]$ to $C[s_1, \dots, s_n]$, 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 $C[u_1, \dots, u_n] \rightarrow C[u'_1, \dots, u'_n]$ 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 \mathcal{R} be a left-linear TRS and $\mu \in CM_{\mathcal{R}}$. If \mathbf{H} is a μ -normalizing (resp. root-normalizing) μ -strategy, then $S_{\mathbf{H}}$ is μ -normalizing (resp. root-normalizing).*

PROOF. By Proposition 14, if S_H is not μ -normalizing (root-normalizing), then there exists an infinite S_H -sequence

$$t_1 \rightarrow_{S_H} t_2 \rightarrow_{S_H} \dots$$

issued from a μ -normalizing (root-normalizing) term t_1 which does not contain a μ -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 μ -normal forms, see Theorem 4), this corresponds to an infinite H -sequence

$$t_1 \hookrightarrow_H t_2 \hookrightarrow_H \dots$$

which contradicts that H is μ -normalizing (root-normalizing). \square

6 Definition of μ -normalizing μ -strategies

Since μ -normal forms do not need to be normal forms, needed reduction is not useful for defining μ -normalizing μ -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 \mathcal{R} [Mid97]:

$$\begin{array}{lcl} \mathbf{f}(\mathbf{x}) & \rightarrow & \mathbf{g}(\mathbf{f}(\mathbf{x})) \\ \mathbf{b} & \rightarrow & \mathbf{b} \end{array}$$

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

$$\underline{\mathbf{f}(\mathbf{b})} \hookrightarrow \mathbf{g}(\mathbf{f}(\mathbf{b}))$$

However, $\mathbf{f}(\mathbf{b})$ has no normal form. Thus, redex \mathbf{b} of $\mathbf{f}(\mathbf{b})$ is needed but repeated μ -reductions of this redex:

$$\mathbf{f}(\underline{\mathbf{b}}) \hookrightarrow \mathbf{f}(\underline{\mathbf{b}}) \hookrightarrow \dots$$

do not μ -normalize $\mathbf{f}(\mathbf{b})$.

Theorem 4 shows that μ -normal forms of a left-linear TRS \mathcal{R} are root-stable (if $\mu \in CM_{\mathcal{R}}$). 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 μ -normalizing computations, as every derivation leading to a μ -normal form yields a root-stable term in some step of the derivation. In fact, according to Corollary 2, whenever $\mu \in CM_{\mathcal{R}}$, the μ -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 μ -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 \mathcal{R} [Mid97]:

$$\begin{array}{lcl} \mathbf{f}(\mathbf{x}) & \rightarrow & \mathbf{g}(\mathbf{a}) \\ \mathbf{b} & \rightarrow & \mathbf{b} \end{array}$$

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

$$g(\mathbf{f}(\mathbf{b})) \rightarrow g(\mathbf{f}(\mathbf{b})) \rightarrow \dots$$

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

$$g(\mathbf{f}(\mathbf{b})) \rightarrow g(g(\mathbf{a}))$$

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 term $t = f(t_1, \dots, t_i, \dots, t_k)$ and $i \in \{1, \dots, k\}$, such that $t \rightarrow_S f(t_1, \dots, t'_i, \dots, t_k)$, 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 $t = f(t_1, \dots, t_i, \dots, t_k)$ and $i \in \{1, \dots, k\}$ such that $t \rightarrow_S f(t'_1, \dots, t'_i, \dots, t'_k)$ and the subsequence from t_i to t'_i is non-empty, 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 $t \rightarrow_S t' \rightarrow_S \dots$ for a normalizing term t . Since S' extends S , there is an infinite reduction S' -sequence, thus contradicting Theorem 7. \square

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

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

$$\begin{array}{ll} \mathbf{f}(\mathbf{x}) \rightarrow \mathbf{c}(\mathbf{x}, \mathbf{a}) & \mathbf{b} \rightarrow \mathbf{b} \\ \mathbf{a} \rightarrow \mathbf{d} & \end{array}$$

together with $\mu(\mathbf{f}) = \emptyset$ and $\mu(\mathbf{c}) = \{2\}$. 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

$$\underline{\mathbf{f}}(\underline{\mathbf{b}}) \rightarrow_S \mathbf{c}(\underline{\mathbf{b}}, \underline{\mathbf{a}}) \rightarrow_S \mathbf{c}(\underline{\mathbf{b}}, \underline{\mathbf{a}}) \rightarrow_S \dots$$

does not compute the μ -normal form $\mathbf{c}(\underline{\mathbf{b}}, \underline{\mathbf{d}})$ which can be obtained by the μ -reduction sequence

$$\underline{\mathbf{f}}(\underline{\mathbf{b}}) \hookrightarrow \mathbf{c}(\underline{\mathbf{b}}, \underline{\mathbf{a}}) \hookrightarrow \mathbf{c}(\underline{\mathbf{b}}, \underline{\mathbf{d}})$$

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 μ -normalization. Note that, in the realm of *CSR*, the notion of one-step *context-free* μ -strategy could be equivalently formulated by imposing the condition that for all root-stable terms $t = f(t_1, \dots, t_i, \dots, t_k)$ and $i \in \mu(f)$, such that $t \hookrightarrow_H f(t_1, \dots, t'_i, \dots, t_k)$, we have $t_i \hookrightarrow_H t'_i$ (with the analogous generalization for arbitrary μ -strategies).

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

PROOF. Let H be a context-free root-normalizing reduction μ -strategy and t be a μ -normalizing term having a μ -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 μ -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 H is root-normalizing, there is no infinite H -sequence starting from t .

If $s = f(s_1, \dots, s_k)$, assume that $A : t = t_1 \hookrightarrow_H t_2 \hookrightarrow_H \dots$ is infinite. Since H is root-normalizing, A contains a root-stable term $t' = f(t'_1, \dots, t'_k)$, i.e., $t' = t_j$ for some $j \geq 1$. By Proposition 13, t' also has a μ -normal form s' which, by Theorem 5, has the same maximal replacing context, $C[\] = MRC^\mu(s) = MRC^\mu(s')$. Since t' is root-stable, it must be $s' = f(s'_1, \dots, s'_k)$; hence, we can write $C[\] = f(C_1[\], \dots, C_k[\])$. Moreover, $t'_i \hookrightarrow^* s'_i$, i.e., each t'_i μ -normalizes in s'_i , and $C_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 μ -normalizing on t'_i . \square

Corollary 5 *Let \mathcal{R} be a left-linear, confluent TRS and $\mu \in CM_{\mathcal{R}}$. Every μ -strategy for \mathcal{R} that can be extended to a context-free root-normalizing μ -strategy for \mathcal{R} is μ -normalizing.*

Corollary 5 formalizes the use of root-normalizing μ -strategies for defining μ -normalizing μ -strategies. In the following sections, we investigate how to (effectively) define them.

6.1 Root-neededness and context-sensitive rewriting

The notion of *root-needed computation* [Ken94, Mid97] provides a suitable formal framework for the definition of root-normalizing, normalizing, and infinitary normalizing reduction sequences [Mid97]. A redex in a term t is *root-needed* if

it is contracted (either itself or its descendants) in every rewrite sequence from t to a root-stable term.

Example 26 Consider the TRS \mathcal{R} :

$$\begin{array}{ll} g(a, x, y) \rightarrow x & b \rightarrow a \\ g(d, x, a) \rightarrow a & c \rightarrow d \end{array}$$

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

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

Note that redex \underline{b} at position 2 of t is root-needed. On the other hand, the redex \underline{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 μ -normalizing μ -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 , $p \in \text{minimal}(P \cap \text{Pos}_{\mathcal{F}}(l))$, and $q \in \text{Pos}_{\mathcal{R}}(t|_p)$ 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]. \square

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

PROOF. Assume $p \in \text{minimal}(P \cap \text{Pos}_{\mathcal{F}}(l))$. Since $t \xrightarrow{\Lambda}^+ \sigma(l)$, we have $\text{sprefix}_t(p) = \text{sprefix}_l(p)$. 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 $p \in \text{Pos}_{\mathcal{F}}(l)$, $q \in P \cap \text{Pos}_{\mathcal{F}}(l)$, but, since $q < p$, p is not minimal in $P \cap \text{Pos}_{\mathcal{F}}(l)$ and this leads to a contradiction. Since $\text{sprefix}_t(p) = \text{sprefix}_l(p)$ and $\mu \in CM_{\mathcal{R}}$, we have $p \in \text{Pos}_{\mathcal{F}}(l) \subseteq \text{Pos}^{\mu}(l)$. By Proposition 6, $p \in \text{Pos}^{\mu}(t)$. \square

Theorem 10 *Let \mathcal{R} be an orthogonal TRS and $\mu \in CM_{\mathcal{R}}$. Every non-root-stable term has a μ -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 non-redex. The first case is the same for us, since we always have $\Lambda \in \text{Pos}^{\mu}(t)$ and thus $\Lambda \in \text{Pos}_{\mathcal{R}}^{\mu}(t)$ 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 $t \xrightarrow{\Lambda}^+ \sigma(l) \xrightarrow{\Lambda} t'' \rightarrow^* t'$. The redex pattern l for the reduction $\sigma(l) \xrightarrow{\Lambda} t''$ 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 $p \in \text{minimal}(P \cap \text{Pos}_{\mathcal{F}}(l))$, where P is as in Lemma 5. We must first prove that $p \in \text{Pos}^{\mu}(t)$. 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 μ -replacing in $t|_p$ and $p \in \text{Pos}^{\mu}(t)$, by Proposition 4, s is a μ -replacing root-needed redex of t . \square

Note that $\mu_{\mathcal{R}}^{\text{can}}$ 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 \mathbf{b} at the occurrence 2 of $t = \mathbf{g}(\mathbf{a}, \mathbf{b}, \mathbf{c})$ is root-needed.*

Since $2 \notin \text{Pos}_{\mathcal{R}}^{\mu_{\mathcal{R}}^{can}}(g(a, b, c)) = \{\Lambda, 1, 3\}$, this redex is not $\mu_{\mathcal{R}}^{can}$ -replacing. Thus, there are root-needed redexes that are not $\mu_{\mathcal{R}}^{can}$ -replacing.

Example 26 also shows that the replacing redex c in t is not root-needed. Therefore, there are $\mu_{\mathcal{R}}^{can}$ -replacing redexes that are not root-needed.

Theorem 10 establishes the *possibility* of defining the following one-step μ -strategy for a given orthogonal TRS \mathcal{R} and whenever $\mu \in CM_{\mathcal{R}}$:

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

By construction, H_{needed} is context-free and Theorem 8 can be invoked to justify its μ -normalizing character. Therefore,

Every orthogonal TRS admits a one-step μ -normalizing μ -strategy.

Unfortunately, the definition of H_{needed} 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 $I \subseteq \text{Pos}_{\mathcal{R}}^{\mu}(t)$ 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 $\mu \in CM_{\mathcal{R}}$. If t is not root-stable, then, $\text{minimal}(\text{Pos}_{\mathcal{R}}^{\mu}(t))$ is a root-necessary set of redexes.*

PROOF. We proceed by structural induction. If t is a redex, it is immediate. If t is not a redex, as in the proof of Theorem 10, $\text{minimal}(\text{Pos}_{\mathcal{R}}^{\mu}(t|_p))$ is a root-necessary set of redexes for each $p \in \text{minimal}(P \cap \text{Pos}_{\mathcal{F}}(l))$. If $q \in \text{minimal}(\text{Pos}_{\mathcal{R}}^{\mu}(t|_p))$ is a position of a root-needed redex of $t|_p$, by Lemma 4, then $t|_{p.q}$ is also a root-needed redex of t . Therefore, $p.\text{minimal}(\text{Pos}_{\mathcal{R}}^{\mu}(t|_p))$ is a root-necessary set of redexes for all $p \in \text{minimal}(P \cap \text{Pos}_{\mathcal{F}}(l))$. Since $p \in \text{Pos}^{\mu}(t)$, by Proposition 4, $p.\text{minimal}(\text{Pos}_{\mathcal{R}}^{\mu}(t|_p)) \subseteq \text{minimal}(\text{Pos}_{\mathcal{R}}^{\mu}(t))$, and the conclusion follows. \square

Given a replacement map μ , the *parallel outermost μ -strategy* H_{po} is:

$$H_{po}(t) = \{t = t_1 \xrightarrow{p_1} t_2 \hookrightarrow \dots \hookrightarrow t_n \xrightarrow{p_n} t_{n+1} \mid \{p_1, \dots, p_n\} = \text{minimal}(\text{Pos}_{\mathcal{R}}^{\mu}(t))\}$$

Note that, since $p_i \parallel p_j$ for every $1 \leq i < j \leq n$, every possible H_{po} -step $t \hookrightarrow_{H_{po}} s$ can be thought of as a single step of parallel (context-sensitive) rewriting (see [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:

Theorem 12 *Let \mathcal{R} be an orthogonal TRS and $\mu \in CM_{\mathcal{R}}$. Then, $H_{p\circ}$ is μ -normalizing.*

PROOF. By Theorem 11 and Theorem 9, $H_{p\circ}$ is root-normalizing. Since $H_{p\circ}$ is context-free, by Theorem 8 the conclusion follows. \square

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

7 Effective definition of one-step μ -normalizing μ -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 [DM97]. Decidable approximations to neededness have been extensively explored [Com00, DM97, HL91, Jac96, JS94, KM91, NST95, NT99, Oya93, TKS00, Toy92]. Recently, we have investigated the use of these approximations to capture root-neededness for *almost orthogonal* TRSs [Luc98b]. We have demonstrated that, among them, NV-sequentiality [Oya93] (hence strong sequentiality [HL91], a particular case of NV-sequentiality) is the most general approximation to root-neededness.

7.1 Sequentiality

Sequentiality is based on the notion of index. An Ω -position $p \in \mathcal{Pos}_{\Omega}(t)$ of an Ω -term $t \in \mathcal{T}_{\Omega}(\mathcal{F}, \mathcal{X})$ is an *index* with respect to a predicate P on Ω -terms if, for every Ω -term s with $s \geq t$, $P(s)$ implies $s|_p \neq \Omega$ [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 Ω -occurrences), we can test whether a redex position $p \in \mathcal{Pos}_{\mathcal{R}}(t)$ ‘is an index’ by applying the previous definition to $t[\Omega]_p$. A monotone predicate P is *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 *sequential* if the predicate $nf_{\mathcal{R}}$ on Ω -terms (where $nf_{\mathcal{R}}(t)$ holds if and only if t has a normal form in $\mathcal{T}(\mathcal{F}, \mathcal{X})$) is sequential.

Example 28 *Consider the TRS \mathcal{R} [HL91]:*

$$\begin{array}{ll} \mathbf{f}(\mathbf{a}, \mathbf{b}, \mathbf{x}) \rightarrow \mathbf{k} & \mathbf{c} \rightarrow \mathbf{a} \\ \mathbf{f}(\mathbf{b}, \mathbf{x}, \mathbf{a}) \rightarrow \mathbf{k} & \mathbf{c} \rightarrow \mathbf{b} \end{array}$$

\mathcal{R} is not sequential because the term $t = \mathbf{f}(\mathbf{c}, \Omega, \Omega)$ has no index: Note that $s = \mathbf{f}(\mathbf{c}, \Omega, \mathbf{a}) > \mathbf{f}(\mathbf{c}, \Omega, \Omega)$ and $\mathbf{f}(\mathbf{c}, \Omega, \mathbf{a})$ has a normal form (without Ω ’s):

$$\mathbf{f}(\underline{\mathbf{c}}, \Omega, \mathbf{a}) \rightarrow \underline{\mathbf{f}(\mathbf{b}, \Omega, \mathbf{a})} \rightarrow \mathbf{k}$$

but $s|_2 = \Omega$, i.e., 2 is not a sequential index of t . Also, $s' = \mathbf{f}(\mathbf{c}, \mathbf{b}, \Omega) > \mathbf{f}(\mathbf{c}, \Omega, \Omega)$ and

$$\mathbf{f}(\mathbf{c}, \mathbf{b}, \Omega) \rightarrow \mathbf{f}(\mathbf{a}, \mathbf{b}, \Omega) \rightarrow \mathbf{k}$$

but $s'|_3 = \Omega$, i.e., 3 is not a sequential index of t . Hence, t has no sequential index. Since $\text{nf}_{\mathcal{R}}(t)$ does not hold whereas both $\text{nf}_{\mathcal{R}}(s)$ and $\text{nf}_{\mathcal{R}}(s')$ hold, and t has no sequential index, \mathcal{R} is not sequential.

For orthogonal TRSs, sequential indices with respect to predicate $\text{nf}_{\mathcal{R}}$ serve to approximate needed redexes of $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$: if $p \in \text{Pos}_{\mathcal{R}}(t)$ is an index of $t[\Omega]_p$ w.r.t. $\text{nf}_{\mathcal{R}}$, 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 \mathcal{R} , the reduction relation $\rightarrow_?$ (arbitrary reduction) on $\mathcal{T}(\mathcal{F}, \mathcal{X})$ is defined as follows: $t \rightarrow_? s$ if there are $p \in \text{Pos}_{\mathcal{R}}(t)$ and s' such that $s = t[s']_p$ [KM91]. Clearly, the $\rightarrow_?$ -normal forms and the normal forms coincide. A TRS \mathcal{R} is *strongly sequential* [HL91, KM91] if predicate $\text{nf}_?$ is sequential (where $\text{nf}_?(t)$ holds if there exists an arbitrary reduction sequence $t \rightarrow_?^* s$ to some normal form $s \in \mathcal{T}(\mathcal{F}, \mathcal{X})$). Indices with respect to predicate $\text{nf}_?$ are said to be *strong* indices (and they are sequential indices, i.e., indices of $\text{nf}_{\mathcal{R}}$). The set of strong indices of a term t is denoted by $\mathcal{I}_s(t)$ (rather than $\mathcal{I}_{\text{nf}_?}(t)$). Strong indices can also be effectively computed by using Ω -reduction in Section 2.1: Given a fresh symbol \bullet and $p \in \text{Pos}_{\Omega}(t)$, we have that $p \in \mathcal{I}_s(t)$ 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 \mathcal{R} :

$$\mathbf{f}(\mathbf{x}, \mathbf{a}) \rightarrow \mathbf{c}$$

$$\mathbf{g}(\mathbf{a}, \mathbf{x}) \rightarrow \mathbf{c}$$

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

$$\mathbf{f}(\mathbf{g}(\Omega, \mathbf{x}), \mathbf{g}(\bullet, \mathbf{x})) \rightarrow_{\Omega} \mathbf{f}(\Omega, \mathbf{g}(\bullet, \mathbf{x}))$$

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

$$\mathbf{f}(\mathbf{g}(\bullet, \mathbf{x}), \mathbf{g}(\Omega, \mathbf{x})) \rightarrow_{\Omega} \mathbf{f}(\mathbf{g}(\bullet, \mathbf{x}), \Omega) \rightarrow_{\Omega} \Omega$$

that is, $1.1 \notin \mathcal{I}_s(t)$.

A TRS \mathcal{R} is *strongly sequential* if $\mathcal{I}_s(t) \neq \emptyset$ for every Ω -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 $\mathcal{R} = (\mathcal{F}, R)$ be a TRS. Let $t \in \mathcal{T}_{\Omega}(\mathcal{F}, \mathcal{X})$. If $p, q \in \mathcal{I}_s(t)$, then $q \in \mathcal{I}_s(t|_p)$. If $p \in \mathcal{I}_s(t)$ and $t \leq s$, then $p \in \mathcal{I}(s[\Omega]_p)$.

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

7.1.2 NV-sequentiality

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

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

Proposition 18 [Oya93] *Strong indices are nv-indices.*

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

We write $t \xrightarrow{p}_\omega s$ (or just $t \rightarrow_\omega s$) if $\exists p \in \mathcal{Pos}(t)$ 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 following \rightarrow_ω -reduction sequence:*

$$\mathbf{f}(\mathbf{g}(\Omega, \mathbf{x}), \mathbf{g}(\Omega, \mathbf{x})) \rightarrow_\omega \mathbf{f}(\mathbf{g}(\Omega, \mathbf{x}), \mathbf{c}) \rightarrow_\omega \mathbf{f}(\mathbf{c}, \mathbf{c})$$

that can be compared with the following \rightarrow_Ω -reduction sequence:

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

The following result connects \rightarrow_{nv} -reduction (used for defining the notion of *nv-index*) and \rightarrow_ω -reduction (used for finding *nv-indices*, see Lemma 7 below).

Lemma 6 [Oya93] *Let $t \rightarrow_{nv} t'$ 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 $s \rightarrow_\omega s'$ and $s' \leq t'$. (This implies that $\exists s'$ such that $s \rightarrow_\omega^* s'$ and $s' \leq t'$.)*

Oyamaguchi characterizes *nv-indices* as follows.

Lemma 7 [Oya93] *Let $t \in \mathcal{T}_\Omega(\mathcal{F}, \mathcal{X})$ and $p \in \mathcal{Pos}_\Omega(t)$. Then $p \notin \mathcal{I}_{nv}(t)$ if and only if there exist $q \in \mathcal{Pos}(t)$, where $q < p$, and $s \in \mathcal{T}_\Omega(\mathcal{F}, \mathcal{X}) - \{\Omega\}$ such that $t[\bullet]_p|_q \rightarrow_\omega^* s$ 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 *nv-index*, since $t|_1 = \mathbf{g}(\bullet, \mathbf{x})$ is not compatible with $\mathbf{g}(\mathbf{a}, \Omega)$ and the only \rightarrow_ω -reduction step which can be given on $t = t|_\Lambda$ is:*

$$\mathbf{f}(g(\bullet, \mathbf{x}), \underline{g(\Omega, \mathbf{x})}) \rightarrow_{\omega} \mathbf{f}(g(\bullet, \mathbf{x}), \mathbf{c})$$

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

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

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

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

The following lemma establishes that redexes placed on nv -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 $p \in \mathcal{I}_{nv}(t[\Omega]_p)$, then $q \not\leq p$.*

The following lemma proves that nv -indices are preserved by \rightarrow_{ω} -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 $p \in \mathcal{I}_{nv}(t)$. If $t \xrightarrow{q}_{\omega} t'$, then $p \in \mathcal{I}_{nv}(t')$.*

PROOF. Note that, since $t|_p = \Omega$ and $p \parallel q$, we have that $t'|_p = \Omega$. If $p \notin \mathcal{I}_{nv}(t')$, by Lemma 7 there exist $p' < p$ and $s \neq \Omega$ such that $t'[\bullet]_{p|p'} \rightarrow_{\omega}^* s$ 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 $t[\bullet]_{p|p'} \rightarrow_{\omega}^* s$ thus contradicting that $p \in \mathcal{I}_{nv}(t)$. \square

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

Example 32 *Consider the TRS \mathcal{R} :*

$$\begin{array}{ll} \mathbf{f}(\mathbf{x}, \mathbf{a}) \rightarrow \mathbf{c} & \mathbf{h}(\mathbf{x}) \rightarrow \mathbf{x} \\ \mathbf{g}(\mathbf{a}, \mathbf{x}) \rightarrow \mathbf{c} & \mathbf{b} \rightarrow \mathbf{c} \end{array}$$

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

$$\mathbf{f}(g(\Omega, \mathbf{x}), \mathbf{b}) \rightarrow_{\omega} \mathbf{f}(\mathbf{c}, \mathbf{x})$$

and 1.1 is not an nv -index of $\mathbf{f}(\mathbf{c}, \mathbf{x})$.

The following result ensures that nv -indices of an Ω -term are preserved under arbitrary reductions.

Lemma 12 *Let \mathcal{R} be a TRS, $t \in \mathcal{T}_{\Omega}(\mathcal{F}, \mathcal{X})$, and $p \in \mathcal{I}_{nv}(t)$. If $t \xrightarrow{q}_{\omega} t'$, then $p \in \mathcal{I}_{nv}(t')$.*

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

Therefore, $p \parallel q$ holds. Note that $t'|_p = \Omega$. Since $\rightarrow \subseteq \rightarrow_{nv}$, by Lemma 6, either $t \leq t'$ or $t \xrightarrow{q}_\omega t''$ and $t'' \leq t'$. In the first case, by Lemma 9, $p \in \mathcal{I}_{nv}(t')$. In the second case, by Lemma 11, $p \in \mathcal{I}_{nv}(t'')$ and by Lemma 9, $p \in \mathcal{I}_{nv}(t')$. Thus, the conclusion follows. \square

7.2 NV-indices and replacement restrictions

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

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

In order to distinguish between replacing and non-replacing indices of a term t , we denote *replacing indices* by $\mathcal{I}_\alpha^\mu(t) = \mathcal{I}_\alpha(t) \cap \mathcal{Pos}^\mu(t)$ for $\alpha \in \{s, nv\}$. Our first result establishes that the canonical replacement map captures *all* *nv*-indices in non-root-stable terms.

Theorem 14 *Let $\mathcal{R} = (\mathcal{F}, R)$ be an almost orthogonal TRS and $\mu \in CM_{\mathcal{R}}$. If $t \in \mathcal{T}_\Omega(\mathcal{F}, \mathcal{X})$ is not root-stable and $p \in \mathcal{Pos}_{\mathcal{R}}(t)$, then $p \in \mathcal{I}_{nv}(t[\Omega]_p) \Leftrightarrow p \in \mathcal{I}_{nv}^\mu(t[\Omega]_p)$.*

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

$$t = t_1 \xrightarrow{p_1} t_2 \hookrightarrow \dots \hookrightarrow t_{n-1} \xrightarrow{p_{n-1}} t_n = \sigma(l)$$

for some substitution σ . We can assume that, for all p_i , $1 \leq i \leq n-1$, $p_i \neq \Lambda$, i.e., μ -rewriting steps from t to $\sigma(l)$ only contract inner positions. We prove, by induction on the length of A , that for all $1 \leq i \leq n$, $p \in \mathcal{I}_{nv}(t_i[\Omega]_p)$ and $p \notin \mathcal{Pos}^\mu(t_i)$. 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 $p_1 \in \mathcal{Pos}^\mu(t_1)$ and $p \in \overline{\mathcal{Pos}^\mu}(t_1)$, by Proposition 5, $p \not\leq p_1$; moreover, $p_1 \not\leq p$; otherwise we contradict Lemma 10. Thus, $p_1 \parallel p$. Hence $t_1 \xrightarrow{p_1} t_2$ implies $t_1[\Omega]_p \xrightarrow{p_1} t_2[\Omega]_p$. By Lemma 12, $p \in \mathcal{I}_{nv}(t_2[\Omega]_p)$. Since $p_1 \parallel p$, by Proposition 6, $p \in \overline{\mathcal{Pos}^\mu}(t_2)$. By the induction hypothesis, the conclusion follows.

Hence, in particular, $p \in \overline{\mathcal{Pos}^\mu}(\sigma(l))$ and $p \in \mathcal{I}_{nv}(\sigma(l)[\Omega]_p)$. Since $\sigma(l)$ itself is a redex, this contradicts Lemma 10. \square

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 nv -index of t . However, 1.1 is a non-replacing position of t : $1.1 \in \overline{\mathcal{Pos}^{\mu_{\mathcal{R}}^{can}}}(t)$.

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

PROOF. If $p \in \mathcal{I}_{nv}(t^{\Omega})$, then, since $t^{\Omega} \leq t[\Omega]_p$, by (repeated application of) Lemma 9, $p \in \mathcal{I}_{nv}(t[\Omega]_p)$. By Theorem 14, $p \in \mathcal{I}_{nv}^{\mu}(t[\Omega]_p)$, i.e., $p \in \mathcal{Pos}^{\mu}(t[\Omega]_p)$. By Proposition 6, $p \in \mathcal{Pos}^{\mu}(t^{\Omega})$, hence $p \in \mathcal{I}_{nv}^{\mu}(t^{\Omega})$. \square

Remark 4 Note the usefulness of these results: By Theorem 14, only $\mu_{\mathcal{R}}^{can}$ -replacing nv -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 $MRC^{\mu_{\mathcal{R}}^{can}}(t)$ of a non-root-stable term t for finding out nv -indices.

7.3 Strong indices and replacement restrictions

Strong indices are nv -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 $t_1 \xrightarrow{p_1}_{\Omega} t_2 \xrightarrow{p_2}_{\Omega} \dots \rightarrow_{\Omega} t_n \xrightarrow{p_n}_{\Omega} t_{n+1}$, then for all i, j , $1 \leq i \leq n$, $j \leq i$, $\mathcal{Pos}(t_i) \subseteq \mathcal{Pos}(t_j)$ and $\mathcal{Pos}^{\mu}(t_i) \subseteq \mathcal{Pos}^{\mu}(t_j)$.

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

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

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

PROOF. We proceed by induction on the length of a derivation $t = t_1 \xrightarrow{p_1}_{\Omega} t_2 \rightarrow_{\Omega} \dots \rightarrow_{\Omega} t_n \xrightarrow{p_n}_{\Omega} \Omega$. If $n = 1$, then $t \xrightarrow{\Lambda}_{\Omega} \Omega$. Since $\Lambda \in \mathcal{Pos}^{\mu}(t)$, it is immediate. If $n > 1$, let us consider the first step $t_i \xrightarrow{p_i}_{\Omega} t_{i+1}$ such that $p_i \in \mathcal{Pos}^{\mu}(t_i)$. Since $\Lambda \in \{p_1, \dots, p_n\}$ such a step will actually be performed. Let $P_i = \{p_j \mid p_i < p_j, 1 \leq j < i\}$. This set is well defined because, by Lemma 13, $p_i \in \mathcal{Pos}(t_j)$ for all $j < i$. If $P_i = \emptyset$, then, since $p_j \in \overline{\mathcal{Pos}^{\mu}}(t_j)$ and $p_i \in \mathcal{Pos}(t_j)$, 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_Ω -reduction directly on $t_1|_{p_i}$. Moreover, by Lemma 13, $p_i \in \mathcal{Pos}^\mu(t_1)$. Hence, we build the derivation

$$t = t_1 \xrightarrow{p_i} t'_1 \xrightarrow{p_1} t'_2 \rightarrow_\Omega \cdots \rightarrow_\Omega t'_{i-1} \xrightarrow{p_{i-1}} t_{i+1} \xrightarrow{p_{i+1}} \cdots \rightarrow_\Omega t_n \xrightarrow{p_n} \Omega$$

where $t'_j = t_j[\Omega]_{p_i}$ for all j , $1 \leq j \leq i-1$. Since t'_1 is soft, we apply the I.H. to the derivation $t'_1 \rightarrow_\Omega^* \Omega$ which takes $n-1$ steps and the conclusion follows.

If $P_i = \{p_i.p'_1, \dots, p_i.p'_m\}$, and $m > 0$, then, since, by Lemma 13, $p_i \in \mathcal{Pos}(t)$, let us consider $s = t|_{p_i}$, and the $l \in L(\mathcal{R})$ such that $t_i|_{p_i} \uparrow l_\Omega$. We prove $s \uparrow l_\Omega$ by contradiction. Hence, reductions performed using positions in $\{p_1, \dots, p_n\} - P_i$ are disjoint to ours and are not relevant here (since we assume $p_j \in \overline{\mathcal{Pos}^\mu}(t_j)$ for all $j < i$, we cannot have $p_j < p_i$). Since $t_i|_{p_i} \uparrow l_\Omega$, and $p_i \in \mathcal{Pos}^\mu(t_i)$, previous reductions (at non-replacing positions in P_i) cannot modify the root of $t|_{p_i}$. Hence, we write $s = C[s_1, \dots, s_p]$ and $l_\Omega = C[l_1, \dots, l_p]$ for some maximal, non-empty context $C[]$. If s and l_Ω 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 $\text{root}(l_k) \neq \Omega$. Hence, since $\mu_{\mathcal{R}}^{can} \sqsubseteq \mu$, we have $q_k \in \mathcal{Pos}^\mu(l_\Omega)$ if $l_\Omega|_{q_k} = l_k$. However, no \rightarrow_Ω -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 $q_k \in \mathcal{Pos}^\mu(s)$ and every \rightarrow_Ω -reduction is given at non-replacing positions, and our choice of s_k (hence of q_k) is arbitrary. However, this means that $t_i|_{p_i}$ will not be compatible with l_Ω , thus leading to a contradiction. Therefore, we can eliminate every \rightarrow_Ω -reduction step $t_j \xrightarrow{p_j} t_{j+1}$ for $p_j \in P_i$ because they are overridden by the reduction step $t_i \xrightarrow{p_i} t_{i+1}$ and are therefore useless. Thus, we obtain a shorter derivation (because $m > 0$) and by I.H., the conclusion follows. \square

The following proposition establishes that ‘softness’ of terms is preserved under replacements on non-replacing positions.

Proposition 21 *Let \mathcal{R} be a TRS and $\mu \in CM_{\mathcal{R}}$. If $t \hookrightarrow_\Omega^* \Omega$, then for all $p \in \overline{\mathcal{Pos}^\mu}(t)$ and $t' \in \mathcal{T}_\Omega(\mathcal{F}, \mathcal{X})$, $t[t']_p \hookrightarrow_\Omega^* \Omega$ in at most the same number of steps.*

PROOF. By induction on the length n of the derivation $t \hookrightarrow_\Omega^* \Omega$. If $n = 0$, then $t = \Omega$ and $\overline{\mathcal{Pos}^\mu}(t) = \emptyset$. Thus, the conclusion follows.

For the induction step, let $t \xrightarrow{q} u \hookrightarrow_\Omega^* \Omega$. Thus, there exists $l \in L(\mathcal{R})$ such that $t|_q \uparrow l_\Omega$ and $q \in \mathcal{Pos}^\mu(t)$. Hence, there is a maximal context $C[]$ such that $t = C[t_1, \dots, t_m]$ and $l_\Omega = C[l_1, \dots, l_m]$ 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, $p \not\leq q$. If $q < p$, we let $p = q.q'$. If there is no t_i for $1 \leq i \leq m$ such that $q' \in \mathcal{Pos}(t_i)$, then, by Proposition 6, $q' \in \mathcal{Pos}_{\mathcal{F}}(l)$. Since $\mu_{\mathcal{R}}^{can} \sqsubseteq \mu$, we have $q' \in \mathcal{Pos}^\mu(l)$. However, by Proposition 4, $q' \in \overline{\mathcal{Pos}^\mu}(t|_q)$ and by Proposition 6 $q' \in \overline{\mathcal{Pos}^\mu}(l)$ thus leading to a contradiction. Therefore, let t_i be such that $t|_{p_i} = t_i$ and $p = p_i.p'$ for some $1 \leq i \leq m$. If $t_i = \Omega$, then $p = p_i$. Since $t_i \leq l_i$ and $p \in \overline{\mathcal{Pos}^\mu}(t)$, by Proposition 6, $p \in \overline{\mathcal{Pos}^\mu}(l)$. Since $\mu_{\mathcal{R}}^{can} \sqsubseteq \mu$, we have $l_i = \Omega$. Therefore, if

we let $s' = t[t']_p|_q$, we have that $s' \uparrow l_\Omega$. If $t_i \neq \Omega$, then $l_i = \Omega$ and, again, $s' \uparrow l_\Omega$. In both cases, we have that $t[t']_p|_q \uparrow l_\Omega$. Hence, since $u = t[\Omega]_q$, we have that $t[t']_p \xrightarrow{q}_\Omega u \hookrightarrow_\Omega^* \Omega$ in n steps and the conclusion follows. If $p \parallel q$, then $t[t']_p \xrightarrow{q}_\Omega u[t']_p$. By the induction hypothesis, $u[t']_p \hookrightarrow_\Omega^* \Omega$ in at most $n - 1$ steps. Therefore, $t[t']_p \hookrightarrow_\Omega^* \Omega$ in at most n steps and the conclusion follows. \square

Theorem 15 *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $\mu \in CM_{\mathcal{R}}$. If $t \in \mathcal{T}_\Omega(\mathcal{F}, \mathcal{X})$ is a soft term, then $\mathcal{I}_s^\mu(t) = \mathcal{I}_s(t)$.*

PROOF. If $\mathcal{Pos}_\Omega(t) = \emptyset$, then $\mathcal{I}_s(t) = \emptyset = \mathcal{I}_s^\mu(t)$. Assume $\mathcal{Pos}_\Omega(t) \neq \emptyset$ and that there is a non-replacing strong index $p \in \mathcal{I}_s(t) - \mathcal{I}_s^\mu(t)$. Since $\omega(t) = \Omega$, by Proposition 20, $t \hookrightarrow_\Omega^* \Omega$. By Proposition 21, $t[\bullet]_p \hookrightarrow_\Omega^* \Omega$ thus contradicting that p is a strong index of t . \square

Theorem 15 does not hold for nv -indices.

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

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

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

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 Ω -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 non-soft 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 $2 \notin \mathcal{Pos}^\mu(t)$, i.e. $2 \notin \mathcal{I}_s^\mu(t)$.

7.4 Using indices for defining context-sensitive strategies

The previous results allow us to establish the following.

Corollary 7 *Let $\mathcal{R} = (\mathcal{F}, R)$ be an almost orthogonal, NV-sequential TRS and $\mu \in CM_{\mathcal{R}}$. If $t \in \mathcal{T}_\Omega(\mathcal{F}, \mathcal{X})$ is not root-stable, then $\mathcal{I}_{nv}^\mu(t^\Omega) \neq \emptyset$.*

PROOF. Since t^Ω is an Ω -normal form, by NV-sequentiality, $\mathcal{I}_{nv}(t^\Omega) \neq \emptyset$ and by Corollary 6, $\mathcal{I}_{nv}^\mu(t^\Omega) \neq \emptyset$. \square

Theorem 13 and Corollary 7 entail a result that complements Theorem 10.

Theorem 16 *Let \mathcal{R} be an almost orthogonal, NV-sequential TRS, and $\mu \in CM_{\mathcal{R}}$. Every non-root-stable term has a replacing root-needed redex.*

PROOF. Let t be a non-root-stable term. By Corollary 7, $\mathcal{I}_{nv}^\mu(t^\Omega) \neq \emptyset$. Let $p \in \mathcal{I}_{nv}^\mu(t^\Omega)$. Since $t^\Omega \leq t[\Omega]_p$, by Lemma 9 and Proposition 6, $p \in \mathcal{I}_{nv}^\mu(t[\Omega]_p)$. By Theorem 13, $t|_p$ is a root-needed redex of t . Proposition 6, $t|_p$ is a replacing root-needed redex. \square

Corollary 7 allows us to define the following one-step μ -strategy for almost orthogonal, NV-sequential TRSs (whenever $\mu_{\mathcal{R}}^{zan} \sqsubseteq \mu$):

$$H_{rv}(t) = \begin{cases} \mathcal{I}_{nv}^\mu(t^\Omega) & \text{if } t \text{ is not root-stable} \\ \bigcup_{i \in \mu(f)} i.H_{rv}(t_i) & \text{if } t = f(\bar{t}) \text{ is root-stable} \\ \emptyset & \text{if } t \text{ is a variable} \end{cases}$$

According to Corollary 7, H_{rv} is actually a μ -strategy. By Theorems 13 and 9, H_{rv} is root-normalizing. By construction, H_{rv} is context-free hence μ -normalizing (Theorem 8). Unfortunately, since root-stability remains undecidable, this definition of H_{rv} 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 nv -index of t^Ω .

8 Context-sensitive index reduction strategies

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

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

$$\mathbf{f}(\mathbf{g}(\mathbf{h}(\mathbf{a}), \mathbf{x}), \mathbf{h}(\mathbf{c})) \hookrightarrow_{H_{rv}} \mathbf{f}(\mathbf{g}(\mathbf{a}, \mathbf{x}), \mathbf{h}(\mathbf{c}))$$

However, 1.1 is not an nv -index of $\mathbf{f}(\mathbf{g}(\Omega, \mathbf{x}), \mathbf{h}(\mathbf{c}))$:

$$\mathbf{f}(\mathbf{g}(\bullet, \mathbf{x}), \mathbf{h}(\mathbf{c})) \rightarrow_\omega \mathbf{f}(\mathbf{g}(\bullet, \mathbf{x}), \Omega)$$

and $\mathbf{f}(\mathbf{g}(\bullet, \mathbf{x}), \Omega) \uparrow \mathbf{f}(\Omega, \mathbf{a})$.

We can (try to) define *arbitrary* (strong or nv -) index μ -reduction as follows:

$$H_s(t) = \mathcal{I}_s^\mu(t^\Omega) \quad H_{nv}(t) = \mathcal{I}_{nv}^\mu(t^\Omega)$$

that is, only redexes occurring on replacing strong or nv -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 μ -reduction.

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

Now we prove that it is possible to define α -index reduction μ -strategies; in particular, we prove the existence of both H_s and H_{nv} .

In the following, we write $t \hookrightarrow_\omega s$ if $t \xrightarrow{p}_\omega s$ and $p \in \mathcal{Pos}^\mu(t)$. We have the following result.

Proposition 22 *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, $\mu \in CM_{\mathcal{R}}$, $l \in L(\mathcal{R})$ and let $s \in \mathcal{T}_\Omega(\mathcal{F}, \mathcal{X}) - \{\Omega\}$ be such that $s \uparrow l_\Omega$. If $t \hookrightarrow_\omega^* s$, then for all $p \in \overline{\mathcal{Pos}^\mu(t)}$ and $t' \in \mathcal{T}_\Omega(\mathcal{F}, \mathcal{X}) - \{\Omega\}$, there exists $s' \in \mathcal{T}_\Omega(\mathcal{F}, \mathcal{X}) - \{\Omega\}$ such that $t[t']_p \hookrightarrow_\omega^* s'$ in at most the same number of steps and $s' \uparrow l_\Omega$.*

PROOF. By induction on the length n of the derivation $t \hookrightarrow_\omega^* s$. If $n = 0$, then $t = s \neq \Omega$ and $t \uparrow l_\Omega$. 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 $p \in \overline{\mathcal{Pos}^\mu(t)}$, we have that $p \neq \Lambda$) and $s' \uparrow l_\Omega$.

For the induction step, let $t \xrightarrow{q}_\omega u \hookrightarrow_\omega^* s$. Thus, there exists $l' \rightarrow r' \in R$ such that $t|_q \uparrow l'_\Omega$ and $q \in \mathcal{Pos}^\mu(t)$. By Proposition 5, $p \not\leq q$. If $q < p$, then, by reasoning as in the base case, we have that $t[t']_p|_q \uparrow l'_\Omega$. Hence, since $u = t[r'_\Omega]_q$, we have that $t[t']_p \xrightarrow{q}_\omega u \hookrightarrow_\omega^* s$ and the conclusion follows. If $p \parallel q$, then $t[t']_p \xrightarrow{q}_\omega u[t']_p$. By the induction hypothesis, there exists $s' \neq \Omega$ such that $u[t']_p \hookrightarrow_\omega^* s'$ in at most $n - 1$ steps and $s' \uparrow l_\Omega$. Therefore, $t[t']_p \hookrightarrow_\omega^* s'$ in at most n steps and the conclusion follows. \square

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

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

Now, we can ensure the existence of nv -index μ -strategies for NV-sequential TRSs.

Theorem 17 *Let $\mathcal{R} = (\mathcal{F}, R)$ be an NV-sequential TRS and $\mu \in CM_{\mathcal{R}}$. If $t \in \mathcal{T}_\Omega(\mathcal{F}, \mathcal{X})$ is an Ω -normal form such that $\mathcal{Pos}_\Omega^\mu(t) \neq \emptyset$, then $\mathcal{I}_{nv}^\mu(t) \neq \emptyset$.*

PROOF. By structural induction. If $t = \Omega$, the proof is immediate. If $t = f(t_1, \dots, t_k)$ since t is an Ω -normal form and $\mathcal{Pos}_\Omega^\mu(t) \neq \emptyset$, there exists an Ω -normal form t_i , for some $i \in \mu(f)$ such that $\mathcal{Pos}_\Omega^\mu(t_i) \neq \emptyset$. By the induction hypothesis, $\mathcal{I}_{nv}^\mu(t_i) \neq \emptyset$. Assume that $\mathcal{I}_{nv}^\mu(t) = \emptyset$. Then for any $p \in \mathcal{I}_{nv}^\mu(t_i)$ and using Lemma 7, there are $p' < i.p$, $s \neq \Omega$ and $l \in L(\mathcal{R})$ such that $t[\bullet]_{i.p|p'} \rightarrow_\omega^* s$ and $s \uparrow l_\Omega$. Since $p \in \mathcal{I}_{nv}^\mu(t_i)$, the only possibility is $p' = \Lambda$; hence we assume

$t[\bullet]_{i.p} \rightarrow_{\omega}^* s$. Obviously, this means that there exists $s' \neq \Omega$ and a derivation $t \rightarrow_{\omega}^* s'$ (which reduces at the same positions that derivation $t[\bullet]_{i.p} \rightarrow_{\omega}^* s$) such that $s' \uparrow l_{\Omega}$. By Proposition 23, there exists $s'' \neq \Omega$ such that $t \hookrightarrow_{\omega}^* s''$ and $s'' \uparrow l_{\Omega}$. By NV-sequentiality, $\mathcal{I}_{nv}(t) \neq \emptyset$; let $q \in \mathcal{I}_{nv}(t)$. Since we assume $\mathcal{I}_{nv}^{\mu}(t) = \emptyset$, we have $q \in \overline{\mathcal{Pos}^{\mu}}(t)$. By Proposition 22, there exists $s''' \neq \Omega$ such that $t[\bullet]_q \hookrightarrow_{\omega}^* s'''$ and $s''' \uparrow l_{\Omega}$. This contradicts that $q \in \mathcal{I}_{nv}(t)$. \square

We need that $\mu \in CM_{\mathcal{R}}$ to ensure the result:

Example 36 Consider the TRS \mathcal{R} of Example 1 and assume $\mu(\mathbf{first}) = \{2\}$. Note that $\mu \notin CM_{\mathcal{R}}$. Consider the Ω -normal form $t = \mathbf{first}(\Omega, \Omega)$. Since $\mathbf{first}(\Omega, \bullet) \uparrow \mathbf{first}(\Omega, \Omega)$, we have that $2 \notin \mathcal{I}_{nv}(t)$. Note that $1 \in \mathcal{I}_{nv}(t)$, but since $1 \in \overline{\mathcal{Pos}^{\mu}}(t)$, $\mathcal{I}_{nv}^{\mu}(t) = \emptyset$.

Theorem 17 formalizes the existence of H_{nv} for NV-sequential TRSs. With regard to H_s , the following result justifies the existence of strong index reduction μ -strategies for strongly sequential TRSs.

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

PROOF. According to Proposition 3, we consider the maximal rigid context $C[\]$ such that $t = C[t_1, \dots, t_n]$ and t_1, \dots, t_n are soft terms. Since $\mathcal{Pos}_{\Omega}^{\mu}(t) \neq \emptyset$, there must be some i , $1 \leq i \leq n$ such that $t_i = t|_p$ is an Ω -normal form (or t itself if $C[\] = \square$), such that $p \in \mathcal{Pos}^{\mu}(t)$, and $\mathcal{Pos}_{\Omega}^{\mu}(t_i) \neq \emptyset$. By strong sequentiality, $\mathcal{I}_s(t_i) \neq \emptyset$ and by Theorem 15, $\mathcal{I}_s^{\mu}(t_i) = \mathcal{I}_s(t_i) \neq \emptyset$. Let $q \in \mathcal{I}_s^{\mu}(t_i)$. Since every Ω -position of a rigid context is trivially an index, $p \in \mathcal{I}_s(C[\Omega])$. By Proposition 17, $p.q \in \mathcal{I}_s(t)$. By Proposition 4, $p.q \in \mathcal{Pos}^{\mu}(t)$; hence, $p.q \in \mathcal{I}_s^{\mu}(t)$ and the conclusion follows. \square

8.1 Properties of context-sensitive index reduction strategies

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

Proposition 24 Arbitrary μ -reduction of strong or nv -indices is context-free.

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

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

Theorem 19 Let \mathcal{R} be an almost orthogonal TRS, $\mu \in CM_{\mathcal{R}}$, and $\alpha \in \{s, nv\}$. Every α -index reduction μ -strategy for \mathcal{R} is root-normalizing.

PROOF. Let H be an α -index reduction μ -strategy and t be a root-normalizing term. Maximal finite H -sequences end in μ -normal forms thus containing a root-stable term (by Theorem 4). If there is an infinite reduction sequence $t = t_0 \hookrightarrow_H t_1 \hookrightarrow_H \dots$ 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. \square

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 \mathcal{R} and whenever $\mu \in CM_{\mathcal{R}}$.

Theorem 20 *Let \mathcal{R} be an almost orthogonal, strongly sequential TRS and $\mu \in CM_{\mathcal{R}}$. Every strong index reduction μ -strategy for \mathcal{R} is μ -normalizing.*

PROOF. By Theorem 18, H_s is a strong index reduction μ -strategy. Since every strong index reduction μ -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. \square

Theorem 21 *Let \mathcal{R} be an almost orthogonal, NV sequential TRS and $\mu \in CM_{\mathcal{R}}$. Every nv -index reduction μ -strategy for \mathcal{R} is μ -normalizing.*

PROOF. By Theorem 17, H_{nv} is a nv -index reduction μ -strategy. Since every nv -index reduction μ -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. \square

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

9 μ -normalization and normalization

In this section, we discuss the use of context-sensitive strategies for defining *normalizing* 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 μ -normal forms. The fact that, in general, there are terms having normal forms which do not have μ -normal forms (and terms having μ -normal forms which have no normal form) is shown by the following example.

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

$$\begin{array}{ll} f(x, a) \rightarrow a & b \rightarrow b \\ g(a) \rightarrow a & \end{array}$$

If we take $\mu(f) = \{1\}$ and $\mu(g) = \emptyset$, then:

1. The term $t = \mathbf{f}(\mathbf{b}, \mathbf{g}(\mathbf{a}))$ is normalizing:

$$\mathbf{f}(\mathbf{b}, \mathbf{g}(\mathbf{a})) \rightarrow \mathbf{f}(\mathbf{b}, \mathbf{a}) \rightarrow \mathbf{a}$$

However, t is not μ -normalizing, as there is only the following infinite μ -rewriting sequence:

$$\mathbf{f}(\mathbf{b}, \mathbf{g}(\mathbf{a})) \hookrightarrow \mathbf{f}(\mathbf{b}, \mathbf{g}(\mathbf{a})) \hookrightarrow \dots$$

2. The term $t = \mathbf{g}(\mathbf{b})$ is μ -normalizing (it is a μ -normal form), but t is not normalizing:

$$\mathbf{g}(\mathbf{b}) \rightarrow \mathbf{g}(\mathbf{b}) \rightarrow \dots$$

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

Theorem 22 *Let $\mathcal{R} = (\mathcal{F}, R)$ 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 $t \hookrightarrow_{\mu}^! t' \rightarrow^! s$.*

PROOF. Since normal forms are root-stable, by Theorem 1, there is a term s' such that $t \hookrightarrow^* s' \xrightarrow{\geq \Delta}^* s$, 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., $\xrightarrow{\geq \Delta}$) from s' to s is not possible and we have $t' = s' = s$ as the desired μ -normal form.

If $s = f(s_1, \dots, s_k)$, let $s' = f(s'_1, \dots, s'_k)$. 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 μ -normal forms u_i of each s'_i such that $u_i \rightarrow^* s_i$ for $1 \leq i \leq k$. We let $t' = f(t'_1, \dots, t'_k)$ 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 $t \hookrightarrow^* t'$ and $t' \rightarrow^* s$. In order to prove that t' is a μ -normal form, we proceed by contradiction. If it is not, then, since each t'_i for $i \in \mu(f)$ is a μ -normal form, t' must be a redex of a rewrite rule $l \rightarrow r \in R$. This means that $t' = \sigma(l)$ for some substitution σ . However, since $\mu_{\mathcal{R}}^{\text{can}} \subseteq \mu$, the subterms at the non-replacing occurrences of l are variables. By Proposition 11, $MRC^{\mu}(t') = MRC^{\mu}(s)$. Thus, $s = \sigma'(l)$ for some substitution σ' . Therefore, s is not a normal form, thus leading to a contradiction. \square

Theorem 22 ensures that, whenever a term has a normal form, it also has a μ -normal form which is a ‘prelude’ to the normal form. In fact, we can use ordering \leq on Ω -terms to give a more standard formulation of ‘approximation’: by using Proposition 10, we can obtain the following: there is $C[\]$ such that $t \hookrightarrow^! t'$, $s = C[t'_1, \dots, t'_n]$, and $C[\overline{\Omega}] \leq s$ (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 μ -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 μ -normalizing.*

These results ensure that a μ -normalizing μ -strategy will stop giving a μ -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 μ -normalization) *Let \mathcal{R} be a left-linear, confluent TRS and $\mu \in CM_{\mathcal{R}}$. If H is μ -normalizing, then S_H is normalizing.*

PROOF. Let H be a μ -normalizing μ -strategy and t be a normalizing term. By Corollary 8, t is μ -normalizing. Since H is μ -normalizing, there is no infinite H -sequence starting from t . Thus, by definition of S_H , every S_H -sequence issued from a normalizing term t can be written as a (possibly empty) *finite* H -sequence:

$$t = t_1 \hookrightarrow_H t_2 \hookrightarrow_H \cdots \hookrightarrow_H t_i = s$$

leading to a μ -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 μ -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 $t\downarrow = C[t_1\downarrow, \dots, t_n\downarrow]$ (with $t_1\downarrow, \dots, t_n\downarrow$ being the normal forms of t_1, \dots, t_n , respectively) and $s = C[s_1, \dots, s_n]$ for $C[\] = MRC^\mu(s) \neq \square$ 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. \square

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

$$\begin{aligned} \mu_{\mathcal{R}}^{can}(s) &= \mu_{\mathcal{R}}^{can}(:) = \mu_{\mathcal{R}}^{can}(\text{recip}) = \mu_{\mathcal{R}}^{can}(\text{terms}) = \emptyset, \\ \mu_{\mathcal{R}}^{can}(\text{sqr}) &= \mu_{\mathcal{R}}^{can}(\text{dbl}) = \mu_{\mathcal{R}}^{can}(+) = \{1\}, \text{ and } \mu_{\mathcal{R}}^{can}(\text{first}) = \{1, 2\} \end{aligned}$$

Consider the one-step $\mu_{\mathcal{R}}^{can}$ -strategy H_{lo} that contracts the leftmost-outermost $\mu_{\mathcal{R}}^{can}$ -replacing redex of terms, i.e.,

$$H_{lo}(t) = \min_{\leq_L} (Pos_{\mathcal{R}}^{\mu_{\mathcal{R}}^{can}}(t))$$

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 $i < j \vee (i = j \wedge p' \leq_L q')$. Since \mathcal{R} is $\mu_{\mathcal{R}}^{can}$ -terminating¹², this strategy is $\mu_{\mathcal{R}}^{can}$ -normalizing; moreover, it is a strong index reduction $\mu_{\mathcal{R}}^{can}$ -strategy¹³, which means that no useless reduction is performed when $\mu_{\mathcal{R}}^{can}$ -normalizing terms. As an example of use, we show how to obtain $\text{first}(\text{dbl}(1), \text{terms}(1))$ (again, n abbreviates $s^n(0)$).

$$\begin{aligned} \text{first}(\text{dbl}(1), \text{terms}(1)) &\hookrightarrow_{H_{lo}} \text{first}(s(s(\text{dbl}(0))), \text{terms}(1)) \\ &\hookrightarrow_{H_{lo}} \frac{\text{first}(s(s(\text{dbl}(0))), \text{recip}(\text{sqr}(1)) : \text{terms}(2))}{\text{recip}(\text{sqr}(1)) : \text{first}(s(\text{dbl}(0)), \text{terms}(2))} \\ &\hookrightarrow_{H_{lo}} \text{recip}(\text{sqr}(1)) : \text{first}(s(\text{dbl}(0)), \text{terms}(2)) \end{aligned}$$

At this point, the $\mu_{\mathcal{R}}^{can}$ -strategy H_{lo} stops yielding a $\mu_{\mathcal{R}}^{can}$ -normal form. However, we can continue with $S_{H_{lo}}$:

$$\begin{aligned} &\text{recip}(\text{sqr}(1)) : \text{first}(s(\text{dbl}(0)), \text{terms}(2)) \\ &\rightarrow_{S_{H_{lo}}} \text{recip}(s(\text{sqr}(0) + \text{dbl}(0))) : \text{first}(s(\text{dbl}(0)), \text{terms}(2)) \\ &\rightarrow_{S_{H_{lo}}} \text{recip}(s(\text{0} + \text{dbl}(0))) : \text{first}(s(\text{dbl}(0)), \text{terms}(2)) \\ &\rightarrow_{S_{H_{lo}}} \text{recip}(s(\text{dbl}(0))) : \text{first}(s(\text{dbl}(0)), \text{terms}(2)) \\ &\rightarrow_{S_{H_{lo}}} \text{recip}(1) : \text{first}(s(\text{dbl}(0)), \text{terms}(2)) \\ &\rightarrow_{S_{H_{lo}}} \text{recip}(1) : \text{first}(s(\text{dbl}(0)), \text{recip}(\text{sqr}(2)) : \text{terms}(3)) \\ &\rightarrow_{S_{H_{lo}}} \text{recip}(1) : \text{recip}(\text{sqr}(2)) : \text{first}(\text{dbl}(0), \text{terms}(3)) \\ &\rightarrow_{S_{H_{lo}}} \text{recip}(1) : \text{recip}(s(\text{sqr}(1) + \text{dbl}(1))) : \text{first}(\text{dbl}(0), \text{terms}(3)) \\ &\rightarrow_{S_{H_{lo}}} \text{recip}(1) : \text{recip}(s(s(\text{sqr}(0) + \text{dbl}(0)) + \text{dbl}(1))) : \text{first}(\text{dbl}(0), \text{terms}(3)) \\ &\rightarrow_{S_{H_{lo}}} \text{recip}(1) : \text{recip}(s(s(\text{sqr}(0) + \text{dbl}(0) + \text{dbl}(1)))) : \text{first}(\text{dbl}(0), \text{terms}(3)) \\ &\rightarrow_{S_{H_{lo}}} \text{recip}(1) : \text{recip}(s(s(\text{0} + \text{dbl}(0) + \text{dbl}(1)))) : \text{first}(\text{dbl}(0), \text{terms}(3)) \\ &\rightarrow_{S_{H_{lo}}} \text{recip}(1) : \text{recip}(s(s(\text{dbl}(0) + \text{dbl}(1)))) : \text{first}(\text{dbl}(0), \text{terms}(3)) \\ &\rightarrow_{S_{H_{lo}}} \text{recip}(1) : \text{recip}(s(s(\text{0} + \text{dbl}(1)))) : \text{first}(\text{dbl}(0), \text{terms}(3)) \\ &\rightarrow_{S_{H_{lo}}} \text{recip}(1) : \text{recip}(s(s(\text{dbl}(1)))) : \text{first}(\text{dbl}(0), \text{terms}(3)) \\ &\rightarrow_{S_{H_{lo}}} \text{recip}(1) : \text{recip}(s(s(s(s(\text{dbl}(0)))))) : \text{first}(\text{dbl}(0), \text{terms}(3)) \\ &\rightarrow_{S_{H_{lo}}} \text{recip}(1) : \text{recip}(4) : \text{first}(\text{dbl}(0), \text{terms}(3)) \\ &\rightarrow_{S_{H_{lo}}} \text{recip}(1) : \text{recip}(4) : \text{first}(0, \text{terms}(3)) \\ &\rightarrow_{S_{H_{lo}}} \text{recip}(1) : \text{recip}(4) : \square \end{aligned}$$

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 non-confluent TRSs.

¹²In Example 3, we proved that \mathcal{R} is μ -terminating for a less restrictive replacement map μ ; hence the conclusion.

¹³This 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 ' $\mu_{\mathcal{R}}^{can}$ -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'Do85]. Similarly, the leftmost-outermost $\mu_{\mathcal{R}}^{can}$ -replacing redex is a strong index for this TRS.

Example 39 Consider the (non-confluent) TRS \mathcal{R} :

$$\begin{aligned} \mathbf{a} &\rightarrow \mathbf{b} \\ \mathbf{a} &\rightarrow \mathbf{g}(\mathbf{a}) \end{aligned}$$

If $\mu(\mathbf{g}) = \emptyset$, then \mathcal{R} is μ -terminating and every μ -strategy \mathbf{H} is μ -normalizing. In particular, if \mathbf{H} only uses the second rule for reducing redex \mathbf{a} , we have:

$$\underline{\mathbf{a}} \rightarrow_{\mathbf{S}_\mathbf{H}} \mathbf{g}(\underline{\mathbf{a}}) \rightarrow_{\mathbf{S}_\mathbf{H}} \mathbf{g}(\mathbf{g}(\underline{\mathbf{a}})) \rightarrow_{\mathbf{S}_\mathbf{H}} \dots$$

which does not normalize \mathbf{a} . However, \mathbf{a} normalizes into \mathbf{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 μ -terminating, then $\mathbf{S}_\mathbf{H}$ is normalizing for every μ -strategy \mathbf{H} .

10 μ -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 \mathcal{R} in Example 1. The term $\mathbf{from}(0)$ has no normal form, since each application of the \mathbf{from} rule always introduces a new function call for the function \mathbf{from} . However, the reduction sequence

$$\mathbf{from}(0) \rightarrow 0:\mathbf{from}(1) \rightarrow 0:1:\mathbf{from}(2) \rightarrow \dots$$

suggests that the ‘infinite value’

$$0:1:2:3:\dots$$

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

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, KKS95, Luc01c, Mid97].

By an *infinite sequence* S of elements taken from a set A we mean a mapping $S : \mathbb{N}^+ \rightarrow A$. 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 $d : A \times A \rightarrow \mathbb{R}$ 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 $\forall \epsilon > 0, \exists n \in \mathbb{N}^+, \forall p \geq n, \forall q \geq n, d(S_p, S_q) < \epsilon$. 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 transfinite¹⁴ rewriting frameworks and their semantic correspondences). An *infinite rewrite sequence* is an infinite sequence t_1, t_2, \dots of (finite) terms such that $t_n \rightarrow t_{n+1}$ for all $n \geq 1$. The depth d of (an occurrence of) a subterm $s = t|_p$ 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 $k = mch(t, s)$, (*maximal common height*) where:

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

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

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

The metric completion of $(\mathcal{T}(\mathcal{F}, \mathcal{X}), d)$ yields the set $\mathcal{T}^\infty(\mathcal{F}, \mathcal{X})$ 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 ω (the first limit ordinal) whose limit is a (possibly infinite) normal form*. Kennaway et al. have developed the notion of *strongly converging* (infinite) rewrite sequence [KKS95]. In sequences of this kind, the depth of the contracted redexes tends to infinite.

Definition 4 [Mid97] *An infinite rewrite sequence $t_1 \rightarrow t_2 \rightarrow \dots$ 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 $t_i \rightarrow t_{i+1} \rightarrow \dots$ is at least d . Also all finite sequences are strongly converging.*

¹⁴Transfinite rewrite sequences are obtained by considering mappings $S : \alpha \rightarrow \mathcal{T}^\infty(\mathcal{F}, \mathcal{X})$ for an arbitrary ordinal number α (and possibly involving infinite terms from the beginning) rather than mappings $S : \omega \rightarrow \mathcal{T}(\mathcal{F}, \mathcal{X})$ for representing infinite sequences (of finite terms, which only become infinite at the limit). Remind that $\omega = \mathbb{N}$ is the first ordinal limit.

Note that every infinite strongly converging sequence $t_1 \rightarrow t_2 \rightarrow \dots$ has a limit t_ω which is necessarily an infinite term. If t_ω is a normal form¹⁵, 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 infinitary normalizing if it strongly converges to a (possibly infinite) normal form. An infinite rewrite sequence that is not infinitary normalizing is called perpetual.*

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

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

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

Theorem 24 *Let \mathcal{R} be a left-linear TRS, and $\mu \in CM_{\mathcal{R}}$ be such that \mathcal{R} is μ -normalizing. If there exists a μ -normalizing μ -strategy for \mathcal{R} , then \mathcal{R} is infinitary normalizing.*

PROOF. Let H be a μ -normalizing μ -strategy for \mathcal{R} . Given a term t , we show that it is possible to use H to build a strongly converging sequence of reductions starting from t . Since \mathcal{R} is μ -normalizing, t has a μ -normal form. Assume that $t \hookrightarrow_H^* t_1$ and t_1 is a μ -normal form. This is well defined because H is μ -normalizing. Let $C_1[\] = MRC^\mu(t_1)$, $t_1 = C_1[t_{11}, \dots, t_{1n_1}]$ and $d_1 = sphole(C_1[\])$. By Proposition 9, $C_1[\]$ is rigid. Since $C_1[\] \neq \square$, $d_1 > 0$. Now we use H for μ -reducing each t_{1i} up to a μ -normal form s_{1i} ; we obtain the corresponding maximal replacing contexts $C_{1i}[\] = MRC^\mu(s_{1i})$ which are also rigid for $1 \leq i \leq n_1$. By Lemma 2, context $C_2[\] = C_1[C_{11}[\], \dots, C_{1n_1}[\]]$ is also rigid and we have obtained $t_1 \rightarrow^* C_2[t_{21}, \dots, t_{2n_2}] = t_2$. We let d_2 to be $d_2 = d_1 + \min(\{d_{1i} \mid 1 \leq i \leq n_1\})$ where $d_{1i} = sphole(C_{1i}[\])$ (note that $d_{1i} > 0$) 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 $t \rightarrow^* t_1 \rightarrow^* t_2 \rightarrow^* \dots$ constructed in this way is a strongly converging sequence: if $d \geq 0$ there always is $j \geq 1$ such that $d_{j-1} \leq d < d_j$ (we let $d_0 = 0$). By construction of the derivation, the depth d' of any redex contracted in the tail subderivation $t_j \rightarrow^* t_{j+1} \rightarrow^* \dots$ 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. \square

The following corollary connects μ -termination and infinitary normalization.

¹⁵Notice that the notion of normal form of [DKP91] (a term t such that $t = t'$ whenever $t \rightarrow t'$) differs from the standard one (which we use here).

Corollary 11 *Let \mathcal{R} be a left-linear TRS and $\mu \in CM_{\mathcal{R}}$. If \mathcal{R} is μ -terminating, then \mathcal{R} is infinitary normalizing.*

This result only makes sense if $\mu \neq \mu_{\top}$ since, in this case, termination and μ -termination differ (terminating TRSs do not admit infinite rewrite sequences). Thus, we apply the result to non-terminating TRSs which are μ -terminating. In this way, μ -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 $\mu_{\mathcal{R}}^{can}$ -terminating.

Example 41 *Consider the nonterminating TRS \mathcal{R} :*

$$\mathbf{f}(\mathbf{a}) \rightarrow \mathbf{f}(\mathbf{f}(\mathbf{a}))$$

This TRS is infinitary normalizing. However, since $\mu_{\mathcal{R}}^{can} = \mu_{\top}$, \mathcal{R} is not μ -terminating for any $\mu \in CM_{\mathcal{R}}$.

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 $t_1 \rightarrow t_2 \rightarrow \dots$ 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 $A : t = t_1 \xrightarrow{p_1} t_2 \xrightarrow{p_2} \dots$, and $p \in \mathcal{Pos}(t)$, we say that $s_1 \xrightarrow{q_1} s_2 \xrightarrow{q_2} \dots$ 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 $k \in \{1, \dots, j-1\} - \{i\}$, $p_i = p.q_1$, $p_j = p.q_2$, $s_1 = t_i|_p$, $s_2 = t_{i+1}|_p$, and $s_2 \xrightarrow{q_2} \dots$ is a p -subsequence of $t_j \xrightarrow{p_j} t_{j+1} \xrightarrow{p_{j+1}} \dots$. The following lemma is used below.

Lemma 14 *Let \mathcal{R} be an infinitary normalizing TRS. Let $t = C[s_1, \dots, s_n]$ be such that $C[\]$ is rigid, let $p_i \in \mathcal{Pos}(t)$ be such that $s_i = t|_{p_i}$ for $1 \leq i \leq n$. Every rewrite p_i -subsequence of a fair infinite rewrite sequence $A : t = t_1 \rightarrow t_2 \rightarrow \dots$ is either normalizing or fair infinite.*

PROOF. Let $p \in \{p_1, \dots, p_n\}$. If there is a finite p -subsequence B of A starting from $t|_p$ 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 $q \in \mathcal{Pos}(t_i)$ such that $p \leq q$ and $t_i|_q = s'$. Since $C[\]$ is rigid, s' is a maximal non-root-stable subterm of t_i . Since \mathcal{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. \square

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 μ -normalizing μ -strategies.

Proposition 25 *Let \mathcal{R} be a left-linear TRS and $\mu \in CM_{\mathcal{R}}$ be such that \mathcal{R} is μ -normalizing. Let H be a μ -normalizing μ -strategy. For all infinite fair S_H -sequence*

$$A : t_1 \rightarrow_{S_H} t_2 \rightarrow_{S_H} \dots$$

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

PROOF. By induction on d . If $d = 0$, we take $i = 1$ and $C[\] = \square$. If $d > 0$, we note that, by Proposition 15, S_H is μ -normalizing. Thus, since \mathcal{R} is μ -normalizing, t_1 has a μ -normal form and A can be written as follows:

$$t_1 \rightarrow_{S_H} t_2 \rightarrow_{S_H} \dots \rightarrow_{S_H} t_i \rightarrow_{S_H} \dots$$

where t_i is a μ -normal form (and, by Proposition 14, every t_j for $j \geq i$). Let $C_0[\] = MRC^\mu(t_i)$, $t_i = C_0[s_1, \dots, s_n]$ and p_j be such that $t|_{p_j} = t_j$ for $1 \leq j \leq n$. Let $d_0 = sphole(C_0[\])$ and $d' = d - d_0$. Since $C_0[\] \neq \square$, 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.,

$$A_j : s_j = s_j^1 \rightarrow_{S_H} s_j^2 \rightarrow_{S_H} \dots$$

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 to be the length of A_j . If A_j is infinite, by the induction hypothesis, there is $k_j \geq 1$ such that $s_j^{k_j} = C'_j[s'_1, \dots, s'_{m_j}]$, $C'_j[\]$ is rigid, s'_1, \dots, s'_{m_j} are not normal forms, and $d' \leq sphole(C'_j[\])$. Let $C[\] = C_0[C_1[\], \dots, 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, $k = i + \sum_{j=1}^n k_j$ is such that $t_k = C[s'_1, \dots, s'_m]$. Since normal forms u_j are rigid contexts (having no hole), by Lemma 2, $C[\]$ is rigid. Note that

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

Hence,

$$\begin{aligned}
d &= d_0 + d' \\
&= \text{sphole}(C_0[]) + d' \\
&\leq \text{sphole}(C_0[]) + \min(\{\text{sphole}(C'_j[]) \mid A_j \text{ is infinite}\}) \\
&= \text{sphole}(C[])
\end{aligned}$$

□

Theorem 25 *Let \mathcal{R} be a left-linear TRS and $\mu \in CM_{\mathcal{R}}$ be such that \mathcal{R} is μ -normalizing. If H is a μ -normalizing μ -strategy, then S_H is infinitary fair-normalizing.*

PROOF. We must show that every infinite fair S_H -sequence A is infinitary normalizing, i.e., derivation A strongly converges to a (possibly infinite) normal form. This is an immediate consequence of Proposition 25. □

Theorem 25 shows that, in contrast to (most) normalizing strategies, μ -normalizing μ -strategies are useful for obtaining infinitary normal forms.

Corollary 12 *Let \mathcal{R} be a left-linear TRS and $\mu \in CM_{\mathcal{R}}$. If \mathcal{R} is μ -terminating, then S_H is infinitary fair-normalizing for every μ -strategy H .*

Theorem 25 and Corollary 12 complement the results on infinitary normalizing strategies given in [Mid97, KKS95] since they do not apply to left-linear TRSs but only to orthogonal TRSs. As a counterpart, we require μ -termination.

For non- μ -terminating TRSs, we can still use μ -normalizing strategies.

Theorem 26 [Mid99] *Let \mathcal{R} be a confluent TRS. Every reduction strategy S for \mathcal{R} that can be extended to a context-free root-normalizing reduction strategy for \mathcal{R} is infinitary fair-normalizing.*

Proposition 26 *If H is a context-free μ -strategy, then S_H is context-free.*

Theorem 27 *Let \mathcal{R} be a left-linear, confluent TRS and $\mu \in CM_{\mathcal{R}}$. If H is a μ -strategy for \mathcal{R} that can be extended to a context-free root-normalizing μ -strategy for \mathcal{R} , then S_H is infinitary fair-normalizing.*

PROOF. Let H' be the context-free root-normalizing extension of H . By Proposition 15, $S_{H'}$ is root-normalizing. By Proposition 26, $S_{H'}$ is context-free. Since S_H can be extended to $S_{H'}$, by Theorem 26 the conclusion follows. □

According to Theorem 27, Theorem 19, and Proposition 24, whenever $\mu \in CM_{\mathcal{R}}$, strong and nv -index reduction μ -strategies are infinitary fair-normalizing for every almost orthogonal, strong and NV-sequential TRS \mathcal{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 \mathcal{R} [Mid97]:

$$\begin{aligned} \mathbf{a} &\rightarrow \mathbf{f}(\mathbf{a}, \mathbf{a}) \\ \mathbf{f}(\mathbf{b}, \mathbf{x}) &\rightarrow \mathbf{c} \end{aligned}$$

and $\mu = \mu_{\mathcal{R}}^{can}$. Note that \mathbf{a} has no normal form. The H_{po} -sequence:

$$\underline{\mathbf{a}} \hookrightarrow_{H_{po}} \mathbf{f}(\underline{\mathbf{a}}, \mathbf{a}) \hookrightarrow_{H_{po}} \mathbf{f}(\mathbf{f}(\underline{\mathbf{a}}, \mathbf{a}), \mathbf{a}) \hookrightarrow_{H_{po}} \dots$$

is not fair, since the redex \mathbf{a} in the second term $\mathbf{f}(\mathbf{a}, \mathbf{a})$ of the derivation is never considered for reduction as it is placed in a non-replacing position ($2 \notin \text{Pos}^\mu(\mathbf{f}(\mathbf{a}, \mathbf{a}))$).

Example 42 shows that $S_{H_{po}}$ is *not* infinitary normalizing. This is because whenever $S_{H_{po}}$ applies on non- μ -normalizing terms, it behaves exactly like H_{po} (which is not fair). In this way, ensuring μ -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 μ -replacing positions of maximal non-root-stable subterms would not be useful without ensuring the μ -normalizing character of the TRS: if we cannot ensure that H eventually stops giving a μ -normal form, then S_H cannot explore the non μ -replacing positions; thus, infinitary normalization is not ensured.

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

Example 43 Consider the TRS \mathcal{R} :

$$\mathbf{a} \rightarrow \mathbf{c}(\mathbf{a}, \mathbf{a})$$

Note that \mathcal{R} is $\mu_{\mathcal{R}}^{can}$ -terminating. Since the only μ -reducible term is \mathbf{a} , there is only one possible $\mu_{\mathcal{R}}^{can}$ -strategy H for \mathcal{R} . The S_H -sequence

$$\underline{\mathbf{a}} \rightarrow_{S_H} \mathbf{c}(\underline{\mathbf{a}}, \mathbf{a}) \rightarrow_{S_H} \mathbf{c}(\mathbf{c}(\underline{\mathbf{a}}, \mathbf{a}), \mathbf{a}) \rightarrow_{S_H} \dots$$

is not fair.

When dealing with μ -normalizing μ -strategies in μ -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 \text{NF}_{\mathcal{R}}^\mu \\ C^\parallel[S_H^\parallel(t_1), \dots, S_H^\parallel(t_n)] & \text{if } t \in \text{NF}_{\mathcal{R}}^\mu - \text{NF}_{\mathcal{R}}, \text{ where:} \\ & C[\] = \text{MRC}^\mu(t) \text{ and } t = C[t_1, \dots, t_n] \\ \emptyset & \text{otherwise} \end{cases}$$

Here, for a given context $C[]$ and sets of rewrite sequences S_1, \dots, S_n , issued from terms t_1, \dots, t_n , we let $C^\parallel[S_1, \dots, S_n]$ denote the set of derivations from $C[t_1, \dots, t_n]$ to $C[s_1, \dots, s_n]$ such that there is $i \in \{1, \dots, n\}$ 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 $\mu \in CM_{\mathcal{R}}$ be such that \mathcal{R} is μ -normalizing. If H is a μ -normalizing μ -strategy, then S_H^\parallel is infinitary fair-normalizing.*

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

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

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

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

$$\underline{a} \rightarrow_{S_H^\parallel} c(\underline{a}, \underline{a}) \rightarrow_{S_H^\parallel} c(c(\underline{a}, \underline{a}), c(\underline{a}, \underline{a})) \rightarrow_{S_H^\parallel} \dots$$

which is fair.

Note that S_H^\parallel 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 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 μ -strategies of Section 8 are optimal and can be used to provide optimal infinitary normalizing strategies by using S_H^\parallel to extend their computational scope (with left-linear, μ -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 than 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 $S_{H_{lo}}$ 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 sequential¹⁶. Therefore, it admits a computable normalizing strategy. The appealing point of our normalizing strategy $S_{H_{lo}}$ 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 μ -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} :

$$\begin{array}{llll} \text{sqr}(0) & \rightarrow 0 & 0 + x & \rightarrow x \\ \text{sqr}(s(x)) & \rightarrow s(\text{sqr}(x) + \text{dbl}(x)) & s(x) + y & \rightarrow s(x+y) \\ \text{dbl}(0) & \rightarrow 0 & \text{first}(0, x) & \rightarrow \square \\ \text{dbl}(s(x)) & \rightarrow s(s(\text{dbl}(x))) & \text{first}(s(x), y : z) & \rightarrow y : \text{first}(x, z) \\ \text{half}(0) & \rightarrow 0 & \text{half}(s(s(x))) & \rightarrow s(\text{half}(x)) \\ \text{half}(s(0)) & \rightarrow 0 & \text{half}(\text{dbl}(x)) & \rightarrow x \\ \text{terms}(n) & \rightarrow \text{recip}(\text{sqr}(n)) : \text{terms}(s(n)) \end{array}$$

is not terminating and we need a normalizing strategy for computing normal forms. Note that \mathcal{R} is not even weakly orthogonal: it has two critical pairs

$$\langle \text{half}(0), 0 \rangle \quad \text{and} \quad \langle \text{half}(s(s(\text{dbl}(x)))) , s(x) \rangle$$

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 \mathcal{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., $\xrightarrow{\Lambda}$ -steps). Note that, e.g., weakly orthogonal TRSs are trivially root-balanced joinable. Unfortunately, the previous critical pairs for \mathcal{R} are *not* root-balanced joinable, as

¹⁶Indeed, \mathcal{R} is *inductively sequential* in the sense of [Ant92]. These TRSs are strongly sequential, see [HLM98].

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

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

$$\underline{\text{half}(0)} \rightarrow 0$$

and

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

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

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

Therefore, by Newman's lemma, \mathcal{S} is confluent. On the other hand, the TRS \mathcal{T} consisting of the rule for **terms** is also confluent (by orthogonality). Now, we can use the following fact from¹⁷ [RV80]:

Confluence is preserved under the combination of left-linear TRSs (\mathcal{R}_1 and \mathcal{R}_2) satisfying that there are no critical pairs between rules of \mathcal{R}_1 and of \mathcal{R}_2 .

Therefore, by taking $\mathcal{R}_1 = \mathcal{S}$ and $\mathcal{R}_2 = \mathcal{T}$, we conclude that \mathcal{R} is confluent.

The TRS \mathcal{R} is μ -terminating: by using the contractive transformation of [Luc96], we obtain:

$\text{sqr}(0)$	$\rightarrow 0$	$0 + x$	$\rightarrow x$
$\text{sqr}(s(x))$	$\rightarrow s(\text{sqr}(x) + \text{dbl}(x))$	$s(x) + y$	$\rightarrow s(x+y)$
$\text{dbl}(0)$	$\rightarrow 0$	$\text{first}(0, x)$	$\rightarrow []$
$\text{dbl}(s(x))$	$\rightarrow s(s(\text{dbl}(x)))$	$\text{first}(s(x), : (y))$	$\rightarrow : (y)$
$\text{half}(0)$	$\rightarrow 0$	$\text{half}(s(s(x)))$	$\rightarrow s(\text{half}(x))$
$\text{half}(s(0))$	$\rightarrow 0$	$\text{half}(\text{dbl}(x))$	$\rightarrow x$
$\text{terms}(n)$	$\rightarrow : (\text{recip}(\text{sqr}(n)))$		

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

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

Therefore, according to Corollary 10, we can use *any* μ -strategy H as a basis for obtaining a normalizing strategy S_H . Similarly, we could also use Corollary 13 for ensuring that S_H^{\parallel} is an infinitary normalizing strategy for \mathcal{R} . Also note that existing infinitary normalizing strategies require (at least) almost orthogonality¹⁸ (see [KKSV95, Luc98b, Mid97]). Thus, they do not apply to \mathcal{R} .

¹⁷I thank Bernhard Gramlich for pointing out this result which permits a formal proof of confluence of \mathcal{R} [Gra02].

¹⁸Whether strong index reduction strategies are infinitary (fair) normalizing for left-linear, strongly sequential, root-balanced joinable TRSs is an open problem.

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 integers¹⁹ in parentheses, given as an operator attribute following the keyword `strat` [GWMFJ00].

Example 45 *The following specification:*

```
obj EXAMPLE is
  sorts Nat LNat .
  op 0      : -> Nat .
  op s      : Nat -> Nat .
  op nil    : -> LNat .
  op cons   : Nat LNat -> LNat [strat (1 0)] .
  op from   : Nat -> LNat .
  op sel    : Nat LNat -> Nat .
  op first  : Nat LNat -> LNat .
  var X Y : Nat .
  var Z : LNat .
  eq sel(s(X),cons(Y,Z)) = sel(X,Z) .
  eq sel(0,cons(X,Z)) = X .
  eq first(0,Z) = nil .
  eq first(s(X),cons(Y,Z)) = cons(Y,first(X,Z)) .
  eq from(X) = cons(X,from(s(X))) .
endo
```

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 `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 $f(t_1, \dots, t_k)$, 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.

Nagaya describes the operational semantics of term rewriting under *strategy maps* φ (which map each symbol f to its individual local strategy $\varphi(f)$) by using a reduction relation on *labelled* 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 labelling $\varphi(t)$ induced by the strategy map φ (which decorates each symbol $f \in \mathcal{F}$ with

¹⁹Here we only consider non-negative integers.

$\varphi(f)$), the evaluation $eval_\varphi(t) \subseteq \mathcal{T}(\mathcal{F}, \mathcal{X})$ of $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ is modeled as normalization of $\varphi(t)$ under this reduction relation followed by the ‘unlabelling’ 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 μ_φ 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 μ as in Example 1. The μ -termination of \mathcal{R} can be ensured by proving termination of the following TRS \mathcal{R}_Z^μ (see [Zan97]):

$$\begin{array}{llll} \text{first}(0, x) & \rightarrow \square & \text{sel}(0, x : y) & \rightarrow x \\ \text{first}(s(x), y : z) & \rightarrow y : \text{first}'(x, a(z)) & \text{sel}(s(x), y : z) & \rightarrow \text{sel}(x, a(z)) \\ \text{from}(x) & \rightarrow x : \text{from}'(s(x)) & & \\ \text{first}(x, y) & \rightarrow \text{first}'(x, y) & \text{from}(x) & \rightarrow \text{from}'(x) \\ a(\text{first}'(x, y)) & \rightarrow \text{first}(x, y) & a(\text{from}'(x)) & \rightarrow \text{from}(x) \\ a(x) & \rightarrow x & & \end{array}$$

where **first'**, **from'**, and **a** are new symbols introduced by the transformation. Termination of \mathcal{R}_Z^μ is proved by using an rpo based on precedence

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

and giving **sel** the usual (left-to-right) lexicographic status.

Since $\mu = \mu_\varphi$ for φ 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 $eval_\varphi(t)$ are μ_φ -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 μ_φ -rewriting strategy, since our requirement of ‘being active as long as a μ_φ -normal form is not reached’ (see Definition 2) is fulfilled:

$$H_\varphi(t) = \begin{cases} \{t \hookrightarrow_{\mu_\varphi}^+ s \mid s \in eval_\varphi(t)\} & \text{if } t \notin \text{NF}_{\mathcal{R}}^{\mu_\varphi} \\ \emptyset & \text{otherwise} \end{cases}$$

where the ‘rough’ description $t \hookrightarrow_{\mu_\varphi}^+ s$ could be exactly given (as a μ_φ -reduction sequence) from the concrete **OBJ** evaluation sequence by using Theorem 1 in [Luc01b]. Note that a single H_φ -step achieves the complete evaluation of a term t as done by using $eval_\varphi$.

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 φ -normalization) Let $\mathcal{R} = (\mathcal{C} \uplus \mathcal{D}, R)$ be a left-linear, confluent TRS and φ be an *E*-strategy map such that for all $f \in \mathcal{D}$, $\varphi(f)$ ends in **0**. If $\mu^\varphi \in CM_{\mathcal{R}}$ and \mathcal{R} is φ -terminating, then S_{H_φ} is normalizing.

Obtaining normal forms in OBJ computations is also guaranteed as follows: given a strategy map φ ensuring that terms in $eval_\varphi(t)$ are root-stable (for all $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$), any φ' given by $\varphi'(f) = \varphi(f) ++ (i_1 \dots i_n)$ for all symbol $f \in \mathcal{F}$ (where ‘++’ appends two lists, and for all $i \in \{1, \dots, ar(f)\} - \mu_\varphi(f)$, $i \in \{i_1, \dots, i_n\}$) ensures that terms in $eval_{\varphi'}(t)$ 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 φ' . For instance, we are able to obtain the normal form $[0, 1, 2, 3, 4]$ of term $\mathbf{first}(5, \mathbf{from}(0))$ (for φ as in Example 45) by using S_{H_φ} . In contrast, with $\varphi'(\mathbf{cons}) = (1 \ 0 \ 2)$, this is *not* possible with OBJ evaluation strategy which would realize a computation equivalent to the following infinite $\mu_{\varphi'}$ -sequence:

$$\begin{aligned} \mathbf{first}(5, \mathbf{from}(0)) &\hookrightarrow_{\mu_{\varphi'}} \mathbf{first}(5, 0: \mathbf{from}(1)) \\ &\hookrightarrow_{\mu_{\varphi'}} \mathbf{first}(5, 0: 1: \mathbf{from}(2)) \\ &\hookrightarrow_{\mu_{\varphi'}} \dots \end{aligned}$$

Thus, the μ_φ -termination of \mathcal{R} (see Example 46) does *not* ensure φ' -normalization. Moreover, even though $eval_\varphi$ is root-normalizing, now $eval_{\varphi'}$ 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 φ) the value²⁰ of any expression of the sort **Nat** without entering into infinite computations. For instance, it is possible to evaluate $\mathbf{sel}(\mathbf{s}(0), \mathbf{from}(0))$ to $\mathbf{s}(0)$, i.e., $\mathbf{s}(0) \in eval_\varphi(\mathbf{sel}(\mathbf{s}(0), \mathbf{from}(0)))$.

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 non-determinism 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 labelled and unlabelled rules. The operational semantics of ELAN takes advantage of this difference: the evaluation of a term proceeds in two steps [BCDK⁺00]:

²⁰We mean a term built from constructor symbols, rather than just a normal form.

1. First, a *leftmost-innermost* reduction strategy is applied to attempt the normalization with respect to the *unlabelled* rules. The user is recommended to provide a confluent and terminating unlabelled rewrite system in order to ensure termination and unicity of the result.
2. As for the normalized term (with respect to unlabelled rules), one first tries to apply a *labelled* 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 non-labelled 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 ς to a term t is denoted by $[\varsigma](t)$. If $[\varsigma](t) = \emptyset$, we say that the strategy ς *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 $\varsigma_1, \dots, \varsigma_k$ and a symbol f of the signature, the strategy $f(\varsigma_1, \dots, \varsigma_k)$ is defined as follows [KKV95, BKK98]:

$$[f(\varsigma_1, \dots, \varsigma_k)](t) = \begin{cases} f(\varsigma_1(t_1), \dots, \varsigma_k(t_k)) & \text{if } t = f(t_1, \dots, t_k) \\ \emptyset & \text{if } \text{root}(t) \neq f \end{cases}$$

where f is overloaded. In particular, when it is applied to sets of terms S_1, \dots, S_k , we have that $f(S_1, \dots, S_k) = \{f(s_1, \dots, s_k) \mid s_1 \in S_1, \dots, s_k \in S_k\}$. The **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. **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 *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; **first** is renamed as **fst** as **first** is used by ELAN itself)

```
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 (0)     : (nat) nat;
    from (0)  : (nat) listNat;
    0 . 0     : (nat listNat) listNat;
    sel (0,0) : (nat listNat) nat;
    fst (0,0) : (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
end

```

This specification corresponds to the leftmost-innermost evaluation strategy restricted to μ -replacing redexes, where $\mu(\mathbf{from}) = \emptyset$, $\mu(\mathbf{s}) = \mu(\cdot) = \{1\}$, and $\mu(\mathbf{sel}) = \mu(\mathbf{fst}) = \{1, 2\}$. Note that $\mu \in CM_{\mathcal{R}}$. This strategy fails if no μ -replacing redex is available for reductions.

The definition is easy to understand (see a similar definition for leftmost-outermost reductions in λ -calculus in [BKK98]): every argument of a symbol of the considered sort is considered for reduction, starting from left to right and skipping non-replacing arguments. Notice that, in some cases (for instance, for the operator **from** which has no replacing argument), there is no strategy **from(rli-nat)** within the list associated to **rli-listNat**. Since reductions in the argument of **from** are not allowed, it is not necessary to include the argument. Thus, this reflects the fact that $\mu(\mathbf{from}) = \emptyset$. 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 $\mu(\cdot) = \{1\}$. 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 **sel(s(zero), from(zero))** to **s(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 μ -normal forms regarding root-normalization and normalization. We proved that, for left-linear TRSs \mathcal{R} and canonical replacement maps $\mu \in CM_{\mathcal{R}}$:

1. the μ -normal forms are *strongly* root-stable. This refines a previous result that identified such μ -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 μ -normal forms of a term t , but it does ensure that the *maximal replacing context* of such μ -normal forms is unique.
3. (unrestricted) reducts of μ -normal forms are μ -normal forms, and the maximal replacing context of a μ -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 μ -strategies, for a given replacement map μ). The effective definition of μ -normalizing μ -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 $\mu \in CM_{\mathcal{R}}$, every orthogonal TRS \mathcal{R} admits a one-step μ -normalizing strategy. Moreover, for almost orthogonal NV-sequential TRSs, such strategies can be effectively given. We have also shown that the restricted parallel outermost μ -strategy is μ -normalizing for every orthogonal TRS \mathcal{R} (whenever $\mu \in CM_{\mathcal{R}}$).

Finally, we have shown how to use μ -normalizing strategies for defining efficient normalizing and infinitary normalizing strategies. These results can be summarized as follows: (1) each left-linear, confluent, TRS \mathcal{R} which has a μ -normalizing μ -strategy (in particular μ -terminating TRSs) admits a (one-step) normalizing strategy (where $\mu \in CM_{\mathcal{R}}$ is assumed), and (2) every left-linear, μ -normalizing TRS \mathcal{R} which admits a μ -normalizing strategy also admits an infinitary normalizing strategy. In both cases, optimality of the underlying μ -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 $\mu_{\mathcal{R}}^{can}$ -replacing redexes. This means that *cs*-restrictions provide a first, simple, and correct bound to root-neededness.
2. Since *nv*-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 $\mu_{\mathcal{R}}^{can}$ -replacing *nv*-indices which suffice for approximating root-needed redexes (without losing any *nv*-index).
3. We have demonstrated that, whenever $\mu \in CM_{\mathcal{R}}$, left-linear μ -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 *nv*-indices. In practice, strong index reduction strategies are not implemented by means of Ω -reduction, i.e., Ω -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 systematically adapt these methods to ease the specification and implementation of context-sensitive rewriting strategies in our setting.

Acknowledgements. I would like to 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

- [AG00] T. Arts and J. Giesl. Termination of Term Rewriting Using Dependency Pairs *Theoretical Computer Science*, 236:133-178, 2000.
- [AM96] S. Antoy and A. Middeldorp. A Sequential Reduction Strategy. *Theoretical Computer Science* 165:75-95, 1996.
- [AN80] A. Arnold and M. Nivat. The metric space of infinite trees. Algebraic and topological properties. *Fundamenta Informaticae* 4:445-476, 1980.
- [Ant92] S. Antoy. Definitional Trees. In H. Kirchner and G. Levi, editors, *Proc. of 3rd International Conference on Algebraic and Logic Programming, ALP'92*, LNCS 632:143-157, Springer-Verlag, Berlin, 1992.
- [BCDK⁺00] P. Borovanský, H. Cirtea, H. Dubois, C. Kirchner, H. Kirchner, P.-E. Moreau, C. Ringeissen, and M. Vittek. ELAN User Manual (version 3.4). January 2000.

- [BEGK⁺87] H.P. Barendregt, M.C.J.D. van Eekelen, J.R.W. Glauert, J.R. Kennaway, M.J. Plasmeijer, and M.R. Sleep. Term Graph Rewriting. In J.W. de Bakker, A.J. Nijman, and P.C. Treleaven, editors, *Proc. of 2nd Conference on Parallel Architectures and Languages Europe, PARLE'87*, LNCS 259:141-158, Springer-Verlag, Berlin, 1987.
- [BFR00] C. Borralleras, M.C.F. Ferreira, and A. Rubio. Complete Monotonic Semantic Path Ordering. In D.A. McAllester, editor, *Proc. of 17th International Conference on Automated Deduction, CADE'00*, LNCS 1831:346-364, Springer-Verlag, Berlin, 2000.
- [BKK98] P. Borovanský, C. Kirchner, and H. Kirchner. A functional view of rewriting and strategies for a semantics of ELAN. In M. Sato and Y. Toyama, editors, *Proc. of 3rd Fuji International Symposium on Functional and Logic Programming, FLOPS'98*, pages 143-166, World Scientific Press, Singapore, 1998.
- [BKKMR98] P. Borovanský, C. Kirchner, H. Kirchner, P.-E. Moreau, and C. Ringeissen. An Overview of ELAN. In C. Kirchner and H. Kirchner, editors, *Proc. of 2nd International Workshop on Rewriting Logic and its Applications, WRLA'98*, Electronic Notes in Computer Science, 15(1998):1-16, 1998.
- [BLR02] C. Borralleras, S. Lucas, and A. Rubio. Recursive Path Orderings can be Context-Sensitive. In A. Voronkov, editor *Proc. of 18th International Conference on Automated Deduction, CADE'02*, LNAI 2392:314-331, Springer-Verlag, Berlin, 2002.
- [BN98] F. Baader and T. Nipkow. Term Rewriting and All That. Cambridge University Press, 1998.
- [Bur91] G.L. Burn. Lazy Functional Languages: Abstract Interpretation and Compilation. Pitman, London and The MIT Press, Cambridge, MA, 1991.
- [CELM96] M. Claver, S. Eker, P. Lincoln, and J. Meseguer. Principles of Maude. In *Proc. 1st International Workshop on Rewriting Logic and its Applications*, Electronic Notes in Theoretical Computer Science, Elsevier Sciences, 1996.
- [CG99] A. Corradini and F. Gadducci. Rational Term Rewriting. In M. Nivat, editor, *Proc. of 1st International Conference on Foundations of Software Science and Computation Structures, FOSACS'98*, LNCS 1378:156-171, Springer-Verlag, Berlin, 1998.
- [Com00] H. Comon. Sequentiality, Monadic Second-Order Logic and Tree Automata. *Information and Computation*, 157(1-2): 25-51, 2000.

- [Cor93] A. Corradini. Term Rewriting in CT_{Σ} . In M.-C. Gaudel and J.-P. Jouannaud, editors, *Proc. of 1st International Joint Conference CAAP/FASE on Theory and Practice of Software Development, TAPSOFT'93*, LNCS 668:468-484, Springer-Verlag, Berlin, 1993.
- [CP85] C. Clack and S.L. Peyton-Jones. Strictness analysis – a practical approach. In J.-P. Jouannaud, editor, *Functional Programming and Computer Architecture*, LNCS 201:35-39, Springer-Verlag, Berlin, 1985.
- [Der87] N. Dershowitz. Termination of rewriting. *Journal of Symbolic Computation*, 3:69-115, 1987.
- [DJ90] N. Dershowitz and J.-P. Jouannaud. Rewrite Systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics, pages 243-320. Elsevier, Amsterdam and The MIT Press, Cambridge, MA, 1990.
- [DKP91] N. Dershowitz, S. Kaplan, and D. Plaisted. Rewrite, rewrite, rewrite, rewrite, rewrite. *Theoretical Computer Science* 83:71-96, 1991.
- [DM97] I. Durand and A. Middeldorp. Decidable Call by Need Computations in Term Rewriting (Extended Abstract). In W. McCune, editor, *Proc. of 14th International Conference on Automated Deduction, CADE'97*, LNAI 1249:4-18, Springer-Verlag, Berlin, 1997.
- [Dur94] I. Durand. Bounded, Strongly Sequential and Forward-Branching Term Rewriting Systems. *Journal of Symbolic Computation*, 18:319-352, 1994.
- [Eke98] S. Eker. Term Rewriting with Operator Evaluation Strategies. In C. Kirchner and H. Kirchner, editors, *Proc. of 2nd International Workshop on Rewriting Logic and its Applications, WRLA'98*, Electronic Notes in Computer Science, 15(1998):1-20, 1998.
- [ENPS92] M.C.J.D. van Eekelen, E.G.J.M.H. Nöcker, M.J. Plasmeijer, and J.E.W. Smetsers. Concurrent Clean. Language Manual - Version 0.8. Technical Report 92-18, University of Nijmegen, 1991.
- [FGJM85] K. Futatsugi, J. Goguen, J.-P. Jouannaud, and J. Meseguer. Principles of OBJ2. In *Conference Record of the 12th Annual ACM Symposium on Principles of Programming Languages, POPL'85*, pages 52-66, ACM Press, 1985.
- [FH88] A.J. Field and P.G. Harrison. Functional Programming. Addison-Wesley Publishing Company, 1988.

- [FKW00] W. Fokkink, J. Kamperman, and P. Walters. Lazy Rewriting on Eager Machinery. *ACM Transactions on Programming Languages and Systems*, 22(1):45-86, 2000.
- [FN97] K. Futatsugi and A. Nakagawa. An Overview of CAFE Specification Environment – An algebraic approach for creating, verifying, and maintaining formal specification over networks –. In *Proc. of 1st International Conference on Formal Engineering Methods*, 1997.
- [FW76] D.P. Friedman and D.S. Wise. CONS should not evaluate its arguments. In S. Michaelson and R. Milner, editors, *Automata, Languages and Programming*, pages 257-284, Edinburgh University Press, 1976.
- [FR99] M.C.F. Ferreira and A.L. Ribeiro. Context-Sensitive AC-Rewriting. In P. Narendran and M. Rusinowitch, editors, *Proc. of 10th International Conference on Rewriting Techniques and Applications, RTA '99*, LNCS 1631:286-300, Springer-Verlag, Berlin, 1999.
- [GL02] B. Gramlich and S. Lucas. Modular termination of context-sensitive rewriting. In C. Kirchner, editor, *Proc. of 4th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming, PPDP'02*, ACM Press, New York, to appear, 2002.
- [GM99] J. Giesl and A. Middeldorp. Transforming Context-Sensitive Rewrite Systems. In P. Narendran and M. Rusinowitch, editors, *Proc. of 10th International Conference on Rewriting Techniques and Applications, RTA '99*, LNCS 1631:271-285, Springer-Verlag, Berlin, 1999.
- [GM02] J. Giesl and A. Middeldorp. Transformation Techniques for Context-Sensitive Rewrite Systems. Technical Report AIB-2002-02, Department of Computer Science, RWTH Aachen, 2002.
- [Gra02] B. Gramlich. Personal communication. April 2002.
- [GWMFJ00] J.A. Goguen, T. Winkler, J. Meseguer, K. Futatsugi, and J.-P. Jouannaud. Introducing OBJ. In J. Goguen and G. Malcolm, editors, *Software Engineering with OBJ: algebraic specification in action*, Kluwer, 2000.
- [GWMFJ93] J.A. Goguen, T. Winkler, J. Meseguer, K. Futatsugi, and J.-P. Jouannaud. Introducing OBJ*. In D. Coleman, R. Gallimore, and J. Goguen, editors, *Applications of Algebraic Specifications Using OBJ*, 1993.

- [HL79] G. Huet and J.J. Lévy. Call by need computations in nonambiguous linear term rewriting systems. Technical Report 359, IRIA Laboria, LeChesnay, France, 1979.
- [HL91] G. Huet and J.J. Lévy. Computations in orthogonal term rewriting systems. In J.L. Lassez and G. Plotkin, editors, *Computational logic: essays in honour of J. Alan Robinson*, pages 395-414 and 415-443. The MIT Press, Cambridge, MA, 1991.
- [HLM98] M. Hanus, S. Lucas, and A. Middeldorp. Strongly sequential and inductively sequential term rewriting systems. *Information Processing Letters*, 67(1):1-8, 1998.
- [HM76] P. Henderson and J. Morris, Jr. A Lazy Evaluator. In *Conference record of the 3rd. ACM Symposium on Principles of Programming Languages, POPL'76*, ACM Press, pages 95-103, 1976.
- [HPF99] P. Hudak, J. Peterson, and J.H. Fasel. A Gentle Introduction to Haskell 98. *Yale University, Department of Computer Science* Technical Report YALEU, October 1999.
- [HPW92] P. Hudak, S.J. Peyton-Jones, and P. Wadler. Report on the Functional Programming Language Haskell: a non-strict, purely functional language. *Sigplan Notices*, 27(5):1-164, 1992.
- [Hue80] G. Huet. Confluent reductions: abstract properties and applications to term rewriting systems. *Journal of the ACM* 27:797-821, 1980.
- [Jac96] F. Jacquemard. Decidable approximations to Term Rewriting Systems. In H. Ganzinger, editor, *Proc. of 7th International Conference on Rewriting Techniques and Applications, RTA'96*, LNCS 1103:362-376, Springer-Verlag, Berlin, 1996.
- [Jon92] M.P. Jones. Introduction to Gofer 2.20. Oxford Programming Research Group, 1992.
- [JS94] J.P. Jouannaud and W. Sadfi. Strong Sequentiality of Left-Linear Overlapping Rewrite Systems. In N. Dershowitz and N. Lindenstrauss, editors, *Proc. of Workshop on Conditional Term Rewriting Systems, CTRS'94*, LNCS 968:235-246, Springer-Verlag, Berlin, 1995.
- [Ken89] R. Kennaway. Sequential Evaluation Strategies for Parallel-Or and Related Reduction Systems. *Journal of Pure and Applied Algebra*, 43:31-56, 1989.
- [Ken94] R. Kennaway. A Conflict Between Call-By-Need Computation and Parallelism. In N. Dershowitz and N. Lindenstrauss, editors, *Proc. of 4th Workshop on Conditional Term Rewriting Systems, CTRS'94*, LNCS 968:247-261, Springer-Verlag, Berlin, 1994.

- [KKV95] C. Kirchner, H. Kirchner, and M. Vittek. Designing Constraint Logic Programming Languages using Computational Systems. In P. van Hentenryck and S. Saraswat, editors, *Principles and Practice of Constraint Programming*. The MIT Press, 1995.
- [Klo92] J.W. Klop. Term Rewriting Systems. In S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum. *Handbook of Logic in Computer Science*, volume 2, pages 1-116. Oxford University Press, 1992.
- [KKS95] R. Kennaway, J.W. Klop, M.R. Sleep, and F.-J. de Vries. Transfinite reductions in Orthogonal Term Rewriting Systems. *Information and Computation* 119(1):18-38, 1995.
- [KM91] J.W. Klop and A. Middeldorp. Sequentiality in Orthogonal Term Rewriting Systems. *Journal of Symbolic Computation* 12:161-195, 1991.
- [KW95] J.F.Th. Kamperman and H.R. Walters. Lazy Rewriting and Eager Machinery. In J. Hsiang, editor, *Proc. of the 6th International Conference on Rewriting Techniques and Applications, RTA'95*, LNCS 914:147-162, Springer-Verlag, Berlin, 1995.
- [Luc96] S. Lucas. Termination of context-sensitive rewriting by rewriting. In F. Meyer auf der Heide and B. Monien, editors, *Proc. of 23rd. International Colloquium on Automata, Languages and Programming, ICALP'96*, LNCS 1099:122-133, Springer-Verlag, Berlin, 1996.
- [Luc98a] S. Lucas. Context-sensitive computations in functional and functional logic programs. *Journal of Functional and Logic Programming*, 1998(1):1-61, January 1998.
- [Luc98b] S. Lucas. Root-neededness and approximations of neededness. *Information Processing Letters*, 67(5):245-254, 1998.
- [Luc01a] S. Lucas. Termination of on-demand rewriting and termination of OBJ programs. In *Proc. of 3rd International Conference on Principles and Practice of Declarative Programming, PPDP'01*, pages 82-93, ACM Press, 2001.
- [Luc01b] S. Lucas. Termination of Rewriting With Strategy Annotations. In R. Nieuwenhuis and A. Voronkov, editors, *Proc. of 8th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, LPAR'01*, LNAI 2250:669-684, Springer-Verlag, Berlin, 2001.
- [Luc01c] S. Lucas. Transfinite Rewriting Semantics for Term Rewriting Systems. In A. Middeldorp, editor, *Proc. of 12th International Conference on Rewriting Techniques and Applications, RTA'01*, LNCS 2051:216-230. Springer-Verlag, Berlin, 2001.

- [Luc02a] S. Lucas. Lazy Rewriting and Context-Sensitive Rewriting. *Electronic Notes in Theoretical Computer Science*, volume 64, Elsevier, 2002.
- [Luc02b] S. Lucas. MU-TERM version 1.0. User's manual. Technical Report DSIC-II/01/02, DSIC, Universidad Politécnica de Valencia, 2002.
- [Luc02c] S. Lucas. Termination of (Canonical) Context-Sensitive Rewriting. In S. Tison, editor, *Proc. of 13th International Conference on Rewriting Techniques and Applications, RTA'02*, LNCS 2378:296-310, Springer-Verlag, Berlin, 2002.
- [Mar90] L. Maranget. Optimal Derivations in Weak Lambda-calculi and in Orthogonal Term Rewriting Systems. In *Conference Record of the 18th Annual ACM Symposium on Principles of Programming Languages, POPL'90*, pages 255-269, ACM Press, 1990.
- [McC60] J. McCarthy. Recursive Functions of Symbolic Expressions and their Computations by Machine, Part I. *Communications of the ACM*, 3(4):184-195, 1960.
- [McC78] J. McCarthy. History of Lisp. In R.L. Wexelblat, editor, *History of Programming Languages*, Academic Press, New York, 1978.
- [Mid97] A. Middeldorp. Call by Need Computations to Root-Stable Form. In *Conference Record of the 24th Annual ACM Symposium on Principles of Programming Languages, POPL'97*, pages 94-105, ACM Press, 1997.
- [Mid99] A. Middeldorp. Context-Freeness and Infinitary Normalization. Presented at the 14th Japanese Term Rewriting Meeting, Nara Institute of Science and Technology, March 15-16, 1999.
- [MN92] A. Mycroft and A.C. Norman. Optimising Compilation. Part II: lazy functional languages. In *Proc. of XIX Seminar on Current Trends in Theory and Practice of Informatics, SOFSEM'92*, 1992.
- [Myc80] A. Mycroft. The theory and practice of transforming call-by-need into call-by-value. In B. Robinet, editor, *Proc. of the 4th International Symposium on Programming*, LNCS 83:269-281, Springer-Verlag, Berlin, 1980.
- [Nag99] T. Nagaya. Reduction Strategies for Term Rewriting Systems. PhD Thesis, School of Information Science, Japan Advanced Institute of Science and Technology, March 1999.
- [Ngu01] Q.-H. Nguyen. Compact Normalisation Trace via Lazy Rewriting. *Electronic Notes in Theoretical Computer Science*, volume 57, Elsevier, 2001.

- [NO01] M. Nakamura and K. Ogata. The evaluation strategy for head normal form with and without on-demand flags. In K. Futatsugi, editor, *Proc. of 3rd International Workshop on Rewriting Logic and its Applications, WRLA'2000, Electronic Notes in Theoretical Computer Science*, volume 36, 17 pages, 2001.
- [NSEP92] E.G.J.M.H. Nöcker, J.E.W. Smetsers, M.C.J.D. van Eekelen, and M.J. Plasmeijer. Concurrent Clean. In E.H.L. Aarts, J. Leeuwen, and M. Rem, editors, *Proc. of Parallel Architectures and Languages Europe, PARLE'91*, LNCS 506:202-219, Springer-Verlag, Berlin, 1992.
- [NST95] T. Nagaya, M. Sakai, and Y. Toyama. NVNF-Sequentiality of Left-Linear Term Rewriting Systems. RIMS Technical Report 918, pages 109-117, University of Kyoto, 1995.
- [NT99] T. Nagaya and Y. Toyama. Decidability for Left-Linear Growing Term Rewriting Systems. In P. Narendran and M. Rusinowitch, editors, *Proc. of 10th International Conference on Rewriting Techniques and Applications, RTA'99*, LNCS 1631:256-270, Springer-Verlag, Berlin, 1999.
- [O'Do77] M.J. O'Donnell. Computing in Systems Described by Equations. LNCS 58, Springer-Verlag, Berlin, 1977.
- [O'Do85] M.J. O'Donnell. Equational Logic as a Programming Language. The MIT Press, Cambridge, Massachusetts, 1985.
- [O'Do95] M.J. O'Donnell. Equational Logic Programming. In D.M. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 4, chapter 3. Oxford University Press, 1995.
- [OV02] V. van Oostrom and R. de Vrijer. Strategies. In TeReSe, *Term Rewriting Systems*, Chapter 9. Cambridge University Press, 2002.
- [Oya93] M. Oyamaguchi. NV-Sequentiality: a decidable condition for call-by-need computations in term-rewriting systems. *SIAM Journal of Computing*, 22(1):114-135, 1993.
- [PE93] R. Plasmeijer and M. van Eekelen. Functional Programming and Parallel Graph Rewriting. Addison-Wesley, Reading, MA, 1993.
- [Pey87] S. L. Peyton-Jones. The Implementation of Functional Programming Languages. Prentice-Hall International, London, 1987.
- [Pol01] J. van de Pol. Just-in-time: On Strategy Annotations. *Electronic Notes in Theoretical Computer Science*, volume 57, Elsevier, 2001.

- [Rea93] C. Reade. Elements of Functional Programming. Addison-Wesley Publishing Company, 1993.
- [Red85] U.S. Reddy. Narrowing as the Operational Semantics of Functional Languages In *Proc. of IEEE International Symposium on Logic Programming*, pages 138-151, 1985.
- [RV80] J.C. Raoult and J. Vuillemin. Operational and Semantic Equivalence Between Recursive Programs. *Journal of the ACM* 27(4):772-796, 1980.
- [Sla74] J.R. Slagle. Automated theorem-proving for theories with simplifiers, commutativity, and associativity. *Journal of the ACM*, 21(4):622-642, 1974.
- [SR93] R.C. Sekar and I.V. Ramakrishnan. Programming in Equational Logic: Beyond Strong Sequentiality. *Information and Computation* 104:78-109, 1993.
- [Str89] R. Strandh. Classes of Equational Programs that Compile into Efficient Machine Code. In N. Dershowitz, editor, *Proc. of 3rd International Conference on Rewriting Techniques and Applications, RTA '89*, LNCS 355:449-461, Springer-Verlag, Berlin, 1989.
- [SX98] J. Steinbach and H. Xi. Freezing – Termination Proofs for Classical, Context-Sensitive and Innermost Rewriting. Institut für Informatik, T.U. München, January 1998.
- [TKS00] T. Takai, Y. Kaji, and H. Seki. Right-Linear Finite Path Overlapping Term Rewriting Systems Effectively Preserve Recognizability. In L. Bachmair, editor, *Proc. of 11th International Conference on Rewriting Techniques and Applications, RTA '00*, LNCS 1833:246-260, Springer-Verlag, Berlin, 2000.
- [Toy92] Y. Toyama. Strong Sequentiality of Left-Linear Overlapping Rewrite Systems. In *Proc. of 7th Annual IEEE Symposium on Logic in Computer Science, LICS'92*, pages 274-284, IEEE Computer Society Press, 1992.
- [Zan97] H. Zantema. Termination of Context-Sensitive Rewriting. In H. Comon, editor, *Proc. of 8th International Conference on Rewriting Techniques and Applications, RTA '97*, LNCS 1232:172-186, Springer-Verlag, Berlin, 1997.
- [Zan02] H. Zantema. Termination. In TeReSe, *Term Rewriting Systems*, Chapter 6. Cambridge University Press, 2002.