Semantics of programs with strategy annotations

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

April 16, 2003

Abstract

Strategy annotations provide a simple strategy language which is used in a number of programming languages and rewriting-based systems for improving the termination behavior and avoiding useless computations. Examples are the E-strategies of Maude, OBJ2, OBJ3, and CafeOBJ, and the 'just-in-time' strategies of JITty and μCRL. We show that context-sensitive rewriting strategies provide an appropriate framework for modeling them: we show that the E-strategies and 'just-in-time' strategies are context-sensitive rewriting strategies. We study the semantics of context-sensitive rewriting strategies and give conditions and techniques guaranteeing correctness and completeness of computations regarding the usual semantics: head-normalization, normalization, functional evaluation, and infinitary normalization. Our results apply to the aforementioned kinds of strategy annotations and programming languages.

Keywords: CafeOBJ, Maude, OBJ, programming languages, programmable strategies, rewriting.

1 Introduction

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. Once a given reduction strategy is fixed in the implementation, every program will be executed according to that strategy. Eventually, this can arise problems with some programs, as each kind of strategy only behaves properly (i.e., it is normalizing, optimal, etc.) for particular classes of programs. Thus, the designers of some programming languages have included language constructs aimed at giving more flexible control of the program execution. For instance, programming languages such as Maude [5], OBJ2 [8], OBJ3

*Work partially supported by CICYT TIC2001-2705-C03-01 and MCYT Acción Integrada HU 2001-0019.

[13], and CafeOBJ [10] admit the *explicit* specification of a particular class of *strategy annotations* (called E-strategies [7]), which are lists of integers associated to function symbols which specify the ordering in which the arguments are (eventually) evaluated in function calls. This very simple strategy language provides quite a powerful way to control the program execution. Due to its simplicity, it also provides a simple interface for understanding and eventually modifying the execution of programs. The E-strategies permit us to completely avoid the evaluation of some arguments of function symbols. For instance, OBJ2 (and OBJ3) *built-in* conditional operator has the following (implicit) strategy annotation ([8], Sec. 4.4; [13], Sec. 2.4.4 & Ap. D.3):

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

which says to evaluate the first argument until it is reduced, and then apply rules at the top (indicated by '`0`'). No evaluation is performed on the second and third arguments[1]. The presence of such 'true' *replacement restrictions* (e.g., by forbidding replacements in the second and third arguments of `if_then_else_fi`) is often invoked to justify that OBJ programs[2] are able to *avoid nontermination* ([13], Section 2.4.4). Termination of E-strategies has been studied in [9, 23]. However, the use of E-strategies that forbid reductions on some arguments can also jeopardize the ability of programs to compute, e.g., normal forms.

Example 1 *Consider the following program:*

```
obj FIRST-FROM-LENGTH is
  sorts Nat LNat .
  op 0      : -> Nat .
  op s      : Nat -> Nat [strat (0)] .
  op nil    : -> LNat .
  op cons   : Nat LNat -> LNat [strat (1 0)] .
  op first  : Nat LNat -> LNat .
  op from   : Nat -> LNat .
  op add    : Nat Nat -> Nat .
  op length : LNat -> Nat .
  vars X Y  : Nat .
  var Z     : LNat .
  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))) .
  eq add(0,X) = X .
  eq add(s(X),Y) = s(add(X,Y)) .
  eq length(nil) = 0 .
  eq length(cons(X,Z))= s(length(Z)) .
endo
```

The strategy annotations for symbols `s` *and* `cons` *are both necessary to achieve a terminating behavior of the program (which can be formally proved). The evaluation of expression*

[1] The 'polymorphic' sort `Univ` for these arguments is denoted `Universal` in [13], Appendix D.3.

[2] As in [13], by OBJ we mean OBJ*, CafeOBJ, or Maude.

```
add(length(first(s(0),from(0))),length(first(0,from(0))))
```

using (command `red` *of) the* Maude *interpreter*[3] *yields:*

```
s(add(length(first(0, from(s(0)))), 0))
```

This term is not a normal form.

We especially note that, whenever the user provides no E-strategy for a given symbol, the (Maude, OBJ*, CafeOBJ) interpreter automatically assigns a *default* E-strategy. This can lead to sharp differences between different interpreters.

Example 2 *(Continuing Example 1) The evaluation of the previous expression using the* OBJ3 *interpreter yields:*

```
s(add(length(first(0,from(s(0)))),length(first(0,from(0)))))
```

This is because the default E-strategy for symbol `add` *in* Maude *is* (1 2 0) *whereas the default E-strategy in* OBJ3 *is* (1 0 2 0)*.*

Example 1 shows that computations using strategy annotations can eventually produce non-totally evaluated terms. Example 2 shows that different interpreters accepting the same program can behave quite differently due to the (internal) use of different E-strategies (i.e., there can be 'portability' problems). These scenarios show that strategy annotations are an essential component of the computational model of the language which needs to be undertaken in any semantic description of the programs.

In this paper we investigate the semantic description of rewriting computations controlled by strategy annotations and its connections with the usual semantics for rewriting computations. We show that context-sensitive rewriting (*CSR*, a simple restriction of rewriting that forbids reductions on selected arguments of functions [18]) provides a suitable framework for describing and analyzing computations with programs using strategy annotations.

1. By using the rewriting semantics of [24], we give a semantic description of computations using context-sensitive rewriting strategies [19] (Section 3).
2. We give conditions ensuring that semantics of context-sensitive rewriting (strategies) is correct and complete w.r.t. standard semantics: head-normalization, functional evaluation, normalization, infinitary normalization, etc. (Section 4).
3. We show that E-strategies can be considered as specifications of concrete CS-strategies. Thus, the results in previous sections immediately apply (Section 5).

[3] We use version 1.0.5 of Maude interpreter (available at `http://maude.cs.uiuc.edu/current/system/`).

4. We also consider Van de Pol's strategies [31] which are an interesting refinement of the E-strategies. We prove that Van de Pol's strategies can be viewed as CS-strategies. Again, the results in Sections 3 and 4 immediately apply (Section 6).

Semantic aspects of computations with these strategy annotations, also in connection to more standard semantics, have been studied in [26, 27, 31]. Comparisons to our approach are given below. Proofs of theorems are given in an appendix.

2 Preliminaries

Let $R \subseteq A \times A$ be a binary relation on a set A. We denote the transitive closure of R by R^+ and its reflexive and transitive closure by R^*. A finite R-sequence is a sequence $a_1, a_2, \ldots, a_n$ of elements taken from A such that $a_i \; R \; a_{i+1}$ 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 $a_1 \; R \; a_2 \; R \; a_3 \cdots$. Obviously, terminating relations are normalizing.

Throughout the paper, $\mathcal{X}$ denotes a countable set of variables and $\mathcal{F}$ denotes a set of function symbols $\{\mathtt{f}, \mathtt{g}, \ldots\}$, each having a fixed arity given by a function $ar : \mathcal{F} \to \mathbb{N}$. We denote the set of terms by $\mathcal{T}(\mathcal{F}, \mathcal{X})$. $\mathcal{V}ar(t)$ is the set of variables in t.

Terms are viewed as labelled trees in the usual way. Positions $p, q, \ldots$ are represented by chains of positive natural numbers which are used to address subterms of t. We denote the empty chain by 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 $\mathcal{P}os(t)$. Given terms t and s, $\mathcal{P}os_s(t)$ denotes the set of positions of s in t, i.e., $p \in \mathcal{P}os_s(t)$ iff $t|_p = s$. Positions of non-variable symbols in t are denoted as $\mathcal{P}os_{\mathcal{F}}(t)$ and $\mathcal{P}os_{\mathcal{X}}(t)$ 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 $root(t)$. A *context* is a term $C \in \mathcal{T}(\mathcal{F} \cup \{\Box\}, \mathcal{X})$ with zero or more 'holes' $\Box$ (a fresh constant symbol). We write $C[\,]_p$ to denote that there is a (usually single) hole $\Box$ at position p of C. Generally, we write $C[\,]$ to denote an arbitrary context (where the number and location of the holes is clarified 'in situ') and $C[t_1, \ldots, t_n]$ to denote the term obtained by filling the holes of a context $C[\,]$ with terms $t_1, \ldots, t_n$. $C[\,] = \Box$ is called the *empty* context.

A rewrite rule is an ordered pair (l, r), written $l \to r$, with $l, r \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, $l \notin \mathcal{X}$ and $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$. 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 $\mathcal{P}os_{\mathcal{R}}(t) = \{p \in \mathcal{P}os(t) \mid \exists l \in L(\mathcal{R}) \ : \ t|_p = \sigma(l)\}$.

A TRS $\mathcal{R}$ is left-linear if for all $l \in L(\mathcal{R})$, l is a linear term. Given a TRS $\mathcal{R} = (\mathcal{F}, R)$, we consider $\mathcal{F}$ 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 $\mathcal{D} = \{root(l) \mid l \to r \in R\}$ and $\mathcal{C} = \mathcal{F} - \mathcal{D}$.

A term t rewrites to s (at position p), written $t \xrightarrow{p}_{\mathcal{R}} s$ (or just $t \xrightarrow{p} s$, or $t \to_{\mathcal{R}} s$, or even $t \to s$) if $t|_p = \sigma(l)$ and $s = t[\sigma(r)]_p$, for some rule $l \to r \in R$, $p \in \mathcal{P}os(t)$ and substitution σ. The one-step rewrite relation for $\mathcal{R}$ is $\to$. A finite $\to$-sequence is called a rewrite sequence. If $t \to^* s$, then s is a reduct of t. A term t is a head-normal form if it cannot be rewritten to a redex. A term is said to be head-normalizing if it has reduct which is a head-normal form. In this paper, the $\to$-normal forms (resp. $\to$-reducible terms) are called normal forms (resp. reducible terms); $\to$-normalizing terms are said to be normalizing. A TRS is confluent (resp. normalizing, terminating) if $\to$ is confluent (resp. normalizing, terminating).

An infinite term on a signature $\mathcal{F}$ is an infinite ordered tree such that each node is labeled by a symbol $f \in \mathcal{F}$ and has a tuple of descendants, and the size of such a tuple is equal to $ar(f)$. The set of (ground) infinite or finite terms is denoted by $\mathcal{T}^\omega(\mathcal{F}, \mathcal{X})$ (resp. $\mathcal{T}^\omega(\mathcal{F})$). The set of finite or infinite normal forms of $\mathcal{R}$ is ω-$\mathsf{NF}_{\mathcal{R}}$.

By an *infinite* rewrite sequence S of terms we mean a mapping $A : \mathbb{N}_{>0} \to \mathcal{T}(\mathcal{F}, \mathcal{X})$ from positive integers into finite terms such that $A_i \to A_{i+1}$ for $i > 0$ (as usual, we write A_i rather than $A(i)$). Note that we do not consider here rewrite sequences which explicitly contain infinite terms; infinite terms only appear as limits of infinite rewrite sequences.

An infinite rewrite sequence $t_1 \to t_2 \to \cdots$ is strongly converging if for all $d \geq 0$, there is an index $i \geq 1$ such that the depth of every redex contracted in $t_i \to t_{i+1} \to \cdots$ is at least d. Also all finite sequences are strongly converging. Note that every infinite strongly converging sequence $t = t_1 \to t_2 \to \cdots$ has a limit s (written $t \to^\omega s$ which is necessarily an infinite term. If $t \to^* s$ or $t \to^\omega s$, we write $t \to^{\leq\omega} s$. A rewrite sequence is called *infinitary normalizing* if it strongly converges to a (possibly infinite) normal form [25]. An infinite rewrite sequence that is not infinitary normalizing is called *perpetual.*

2.1 Context-sensitive rewriting

Given a signature $\mathcal{F}$, a mapping $\mu : \mathcal{F} \to \mathcal{P}(\mathbb{N})$ is a *replacement map* (or $\mathcal{F}$-map) if for all $f \in \mathcal{F}$, $\mu(f) \subseteq \{1, \ldots, ar(f)\}$. The replacement map μ determines the *argument* positions which can be reduced for each symbol in $\mathcal{F}$ [18]. The set of all $\mathcal{F}$-maps is $M_{\mathcal{F}}$. When considering a TRS $\mathcal{R} = (\mathcal{F}, R)$, we also write $M_{\mathcal{R}}$ rather than $M_{\mathcal{F}}$. An ordering $\sqsubseteq$ is defined on $M_{\mathcal{F}}$, the set of all $\mathcal{F}$-maps: $\mu \sqsubseteq \mu'$ if for all $f \in \mathcal{F}$, $\mu(f) \subseteq \mu'(f)$. Thus, $\mu \sqsubseteq \mu'$ means that μ considers less positions than μ' for reduction, i.e., μ is 'more restrictive' than (or equally restrictive to) μ'. Given $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and $\mu \in M_{\mathcal{F}}$, the set of μ-*replacing* positions $\mathcal{P}os^{\mu}(t)$ of t is: $\mathcal{P}os^{\mu}(t) = \{\Lambda\}$, if $t \in \mathcal{X}$ and $\mathcal{P}os^{\mu}(t) = \{\Lambda\} \cup \bigcup_{i \in \mu(root(t))} i.\mathcal{P}os^{\mu}(t|_i)$ if $t \notin \mathcal{X}$. By abuse, the occurrence of subterm $t|_p$ at position p is called replacing if $p \in \mathcal{P}os^{\mu}(t)$ In context-sensitive rewriting (*CSR*), we rewrite subterms at *replacing* positions: t μ-rewrites to s, written $t \overset{p}{\hookrightarrow}_{\mathcal{R}(\mu)} s$ (or simply $t \hookrightarrow_{\mathcal{R}(\mu)} s$, $t \hookrightarrow_{\mu} s$ or $t \hookrightarrow s$) if $t \overset{p}{\to}_{\mathcal{R}} s$ and $p \in \mathcal{P}os^{\mu}(t)$. The $\hookrightarrow_{\mu}$-normal forms ($\hookrightarrow_{\mu}$-reducible terms) are called μ-normal forms (μ-reducible terms). Let $\mathsf{NF}^{\mu}_{\mathcal{R}}$ be the set of μ-normal forms of $\mathcal{R}$. A term is μ-normalizing if it is $\hookrightarrow_{\mu}$-normalizing.

A TRS is μ-confluent if $\hookrightarrow_{\mu}$ is confluent. Confluence of *CSR* has been investigated in [18]. A TRS $\mathcal{R}$ is μ-terminating if $\hookrightarrow_{\mu}$ is terminating. Termination of *CSR* has been studied in a number of papers, see [12, 21] for recent surveys.

With *innermost* context-sensitive rewriting $\overset{i}{\hookrightarrow}_{\mu}$, we only contract *maximal* positions (w.r.t. $\leq$) of replacing redexes (in $\mathcal{P}os^{\mu}_{\mathcal{R}}(t)$): $t \overset{i}{\hookrightarrow}_{\mu} s$ if $t \overset{p}{\to}_{\mathcal{R}} s$ and $p \in maximal_{\leq}(\mathcal{P}os^{\mu}_{\mathcal{R}}(t))$. We say that $\mathcal{R}$ is *innermost* μ-terminating if $\overset{i}{\hookrightarrow}_{\mu}$ is terminating. Termination of innermost *CSR* has been studied in [11, 23].

The *canonical replacement map* $\mu^{can}_{\mathcal{R}}$ for a TRS $\mathcal{R}$ 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 [18, 19]. Note that $\mu^{can}_{\mathcal{R}}$ can be automatically associated to $\mathcal{R}$ by means of a very simple calculus: for each symbol $f \in \mathcal{F}$ and $i \in \{1, \ldots, ar(f)\}$, $i \in \mu^{can}_{\mathcal{R}}(f)$ iff $\exists l \in L(\mathcal{R}), p \in \mathcal{P}os_{\mathcal{F}}(l), (root(l|_p) = f \wedge p.i \in \mathcal{P}os_{\mathcal{F}}(l))$. Given a TRS $\mathcal{R}$, $CM_{\mathcal{R}} = \{\mu \in M_{\mathcal{R}} \mid \mu^{can}_{\mathcal{R}} \sqsubseteq \mu\}$ is the set of replacement maps which are less restrictive than or equally restrictive to $\mu^{can}_{\mathcal{R}}$. One of the most important properties of the canonical replacement map is the following.

Theorem 1 [18, 19] *Let $\mathcal{R}$ be a left-linear TRS and $\mu \in CM_{\mathcal{R}}$. Every μ-normal form is a head-normal form.*

3 Semantics of (restricted) rewriting computations

As we will see, semantics of rewriting under strategy annotations is often given as a *rewriting semantics*. By a rewriting semantics, we mean a mapping $\mathsf{S} : \mathcal{T}(\mathcal{F}, \mathcal{X}) \to \mathcal{P}(\mathcal{T}^{\omega}(\mathcal{F}, \mathcal{X}))$ such that, for all $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and $s \in \mathsf{S}(t)$, $t \to^{\leq \omega} s$ (resp. $t \to^{*} s$) [24]. A semantics S is defined (deterministic) if $\mathsf{S}(t) \neq \varnothing$ ($|\mathsf{S}(t)| \leq 1$) for all terms t. Each rewriting semantics S defines a set $W_{\mathsf{S}} = \bigcup_{t \in \mathcal{T}(\mathcal{F}, \mathcal{X})} \mathsf{S}(t)$ (i.e., the *range* of S) of 'interesting' or 'canonical' values which the computation represented by the semantics obtains.

$$
\begin{array}{rcl}
\mathsf{eval}(t) & = & \{s \in \mathcal{T}(\mathcal{C},\mathcal{X}) \mid t \to^{*} s\} \\
\omega\text{-}\mathsf{eval}(t) & = & \{s \in \mathcal{T}^{\omega}(\mathcal{C},\mathcal{X}) \mid t \to^{\leq\omega} s\} \\
\mathsf{nf}(t) & = & \{s \in \mathsf{NF}_{\mathcal{R}} \mid t \to^{*} s\} \\
\omega\text{-}\mathsf{nf}(t) & = & \{s \in \omega\text{-}\mathsf{NF}_{\mathcal{R}} \mid t \to^{\leq\omega} s\} \\
\mathsf{hnf}(t) & = & \{s \in \mathsf{HNF}_{\mathcal{R}} \mid t \to^{*} s\}
\end{array}
$$

Figure 1: Rewriting semantics for a TRS $\mathcal{R}$

Example 3 *The usual semantics given to a (first-order) functional program $\mathcal{R}$ is the set of constructor terms*[4] *that $\mathcal{R}$ is able to produce in a finite number of rewriting steps (*eval*, see Figure 1). Lazy functional languages would rather consider ω-*eval *instead (i.e., the possibly infinite constructor values obtained as limits of infinitary computations). Programming languages such as* ELAN, CafeOBJ, OBJ3, *and* Maude *focus on the computation of (finitary) normalization semantics* nf*. Infinite normal forms can also be considered with ω-*nf*. A (more or less) auxiliary semantics often considered are, e.g., the set of all possible reducts of a term which are head-normal forms (*hnf*).*

Programs of real (rewriting-based) programming languages are better described by also considering the concrete strategy which is used to execute them. A (non-deterministic) *rewriting strategy* for a TRS $\mathcal{R}$ is a function $\mathbb{S}$ 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. If sequences in $\mathbb{S}(t)$ consists of at most one rewriting step for all term t, we say that $\mathbb{S}$ is a *one-step* rewriting strategy. $\mathbb{S}$ is called deterministic if $|\mathbb{S}(t)| \leq 1$. We write $t \to_{\mathbb{S}} s$ if $\mathbb{S}(t)$ contains a reduction sequence ending with s. An $\mathbb{S}$-*sequence* is a reduction sequence of the form $t_1 \to_{\mathbb{S}} t_2 \to_{\mathbb{S}} \cdots$. A strategy $\mathbb{S}$ is *normalizing* if, for all normalizing term t, there is no infinite $\mathbb{S}$-sequence starting from t. A reduction strategy $\mathbb{S}$ for a TRS is called *infinitary normalizing* if there are no perpetual $\mathbb{S}$-rewrite sequences starting from terms that admit an infinitary normalizing rewrite sequence.

We say that a rewriting strategy $\mathbb{S}$ is *terminating* if $\to_{\mathbb{S}}$ is terminating.

Remark 1 *Termination of strategies is closer to termination of programs than termination of rewriting. On the other hand, ensuring termination of a strategy is more interesting than just having normalization (traditionally the most important property of a strategy), since termination means that no infinite computation sequence ever arise.*

[4] Functional languages such as Haskell or ML would actually restrict the attention to *ground* constructor terms.

Given a strategy $\mathbb{S}$ for a TRS $\mathcal{R}$, we let $\mathsf{nf}_{\mathbb{S}}$ to be

$$\mathsf{nf}_{\mathbb{S}}(t) = \{s \in \mathcal{T}(\mathcal{F}, \mathcal{X}) \mid t \to^{!}_{\mathbb{S}} s\}$$

and ω-$\mathsf{nf}_{\mathbb{S}}$ for the corresponding infinitary version[5]

$$\omega\text{-}\mathsf{nf}_{\mathbb{S}}(t) = \mathsf{nf}_{\mathbb{S}}(t) \cup \{s \in \mathcal{T}^{\omega}(\mathcal{F}, \mathcal{X}) \mid t \to^{\omega}_{\mathbb{S}} s\}.$$

3.1 Correctness and completeness of rewriting semantics

A partial order $\sqsubseteq$ among rewriting semantics is given by $\mathsf{S} \sqsubseteq \mathsf{S}'$ if and only if $\forall t \in \mathcal{T}(\mathcal{F}, \mathcal{X}), \mathsf{S}(t) \subseteq \mathsf{S}'(t)$ [24]. The following diagram depicts the ordering among the semantics in Figure 1 (an arrow from S to S' means that $\mathsf{S} \sqsubseteq \mathsf{S}'$).

$$\begin{tikzcd}
\mathsf{hnf} & \\
& \omega\text{-}\mathsf{nf} \\
\mathsf{nf} \arrow[uu] \arrow[ur] & \\
& \omega\text{-}\mathsf{eval} \arrow[uu] \\
\mathsf{eval} \arrow[uu] \arrow[ur] &
\end{tikzcd}$$

Suitable notions of correctness and completeness of a given semantics w.r.t. a *reference* semantics can be defined by using this ordering.

Definition 1 *Given a* reference *semantics* S_0 *for a TRS* $\mathcal{R}$*, a particular rewriting semantics* S *for* $\mathcal{R}$ *is:*

partially correct *(w.r.t.* S_0*)*	*if*	$\lambda t.\mathsf{S}(t) \cap W_{\mathsf{S}_0}$	$\sqsubseteq$	S_0,
correct *(w.r.t.* S_0*)*	*if*	S	$\sqsubseteq$	S_0,
complete *(w.r.t.* S_0*)*	*if*	S_0	$\sqsubseteq$	S

Informally, S is *partially correct* if each *interesting* (referred to S_0) value which is computed using S is acceptable (regarding S_0); *correct* if each computed value using S is acceptable (for S_0); *complete* if each interesting value which is computed using S_0 is also computed using S. Note that correctness and completeness are dual properties in this order-theoretic setting. Correctness of a semantics S w.r.t. S_0 implies partial correctness of S w.r.t. S_0 (note that, whenever $\mathsf{S} \sqsubseteq \mathsf{S}_0$, we have $\mathsf{S}(t) \cap W_{\mathsf{S}_0} = \mathsf{S}(t)$ for all terms t). The opposite is not true: for instance, nf is only partially correct w.r.t. eval; also, ω-eval is partially correct (but not correct) w.r.t. nf.

Remark 2 *If* S *is partially correct w.r.t.* S_0*, then* $\mathsf{S}' = \lambda t.\mathsf{S}(t) \cap W_{\mathsf{S}_0}$ *is, by definition, correct w.r.t.* S_0*. Hence, ensuring partial correctness of a semantics gives us a systematic way to obtain a correct semantics.*

[5]Note that we only consider strongly converging $\mathbb{S}$-sequences.

When considering partial correctness and completeness together, we have the following interesting (and obvious) fact.

Proposition 1 *If* S *is partially correct and complete w.r.t.* S_0*, then* $\mathsf{S}_0 = \lambda t.\mathsf{S}(t) \cap W_{\mathsf{S}_0}$.

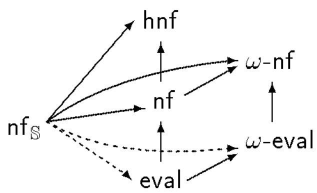
A rewriting semantics S which is correct and complete w.r.t. S_0 is, in fact, equivalent to S_0, i.e., $\mathsf{S} = \mathsf{S}_0$.

Remark 3 *Proposition 1 allows us to say that if* S *is partially correct and complete w.r.t.* S_0*, then* S *could be used to simulate* S_0*, in the sense that, for each term* t*, we only need to remove from* $\mathsf{S}(t)$ *the values which are not in* W_0 *to exactly obtain* $\mathsf{S}_0(t)$*.*

In contrast to correctness, partial correctness is *not* transitive. However, we have the following.

Proposition 2 *If* S *is correct w.r.t.* S_0 *and* S_0 *is partially correct w.r.t.* S'_0*, then* S *is partially correct w.r.t.* S'_0*.*

Since every strategy is forced to perform rewriting steps unless faced to a normal form, for every rewriting strategy $\mathbb{S}$, $\mathsf{nf}_{\mathbb{S}}$ is correct w.r.t. nf and hnf, and partially correct w.r.t. eval. According to the previous facts, semantics $\mathsf{nf}_{\mathbb{S}}$ is added to the previous diagram as follows (where a dashed arrow from S to S' means that S is partially correct w.r.t. S').



In contrast to $\mathsf{nf}_{\mathbb{S}}$ regarding nf, semantics ω-$\mathsf{nf}_{\mathbb{S}}$ is *not* correct w.r.t. ω-nf. This is because convergent infinite reduction sequences issued by $\mathbb{S}$ do not need to reach a normal form.

Example 4 *Consider the TRS* $\mathcal{R}$*:*

$\mathtt{a} \to \mathtt{f(a,a)}$

The evaluation of expression `a` *using the leftmost-outermost rewriting strategy yields the strongly converging sequence:*

$\underline{\mathtt{a}} \to \mathtt{f(}\underline{\mathtt{a}}\mathtt{,a)} \to \mathtt{f(f(}\underline{\mathtt{a}}\mathtt{,a),a)} \to \cdots$

which does not converge to a normal form.

In order to ensure correctness of ω-$\mathsf{nf}_{\mathbb{S}}$ w.r.t. ω-nf, we need to use an *infinitary normalizing* strategy $\mathbb{S}$ (see [16, 20, 25] for a discussion about the definition of such strategies.).

3.2 Semantics of context-sensitive computations

The rewriting semantics $\mathsf{cs\text{-}nf}_\mu$ for a TRS $\mathcal{R}$ computes the set of μ-normal forms of each term t:

$$\mathsf{cs\text{-}nf}_\mu(t) = \{s \in \mathsf{NF}^\mu_{\mathcal{R}} \mid t \hookrightarrow^*_\mu s\}$$

We also consider the infinitary version:

$$\omega\text{-}\mathsf{cs\text{-}nf}_\mu(t) = \{s \in \omega\text{-}\mathsf{NF}^\mu_{\mathcal{R}} \mid t \hookrightarrow^{\leq\omega}_\mu s\}$$

which computes the finite or infinite μ-normal forms of a term t (by using strongly converging μ-rewrite sequences). As for unrestricted rewriting, we can also consider *concrete* sequences of context-sensitive rewriting steps when modeling rewriting computations. A (non-deterministic) *μ-strategy* for $\mathcal{R}$ is a function $\mathbb{H}$ that assigns a non-empty set of non-empty finite μ-rewrite sequences each beginning with t to every term t which is not a μ-normal form [19]. Again, if sequences in $\mathbb{H}(t)$ consist of at most one μ-rewriting step for all term t, we say that $\mathbb{H}$ is a *one-step* μ-rewriting strategy; $\mathbb{H}$ is called deterministic if $|\mathbb{H}(t)| \leq 1$.

Remark 4 *Note that, by using μ-strategies, μ-normal forms cannot be further reduced. Thus, whenever $\mu \neq \mu_\top$ (where $\mu_\top(f) = \{1, \ldots, ar(f)\}$ for all symbol f), a μ-strategy is* not *necessarily a rewriting strategy. On the other hand, since $\to \, = \, \hookrightarrow_{\mu_\top}$, rewriting strategies can be thought of as a particular subclass of context-sensitive rewriting strategies.*

We write $t \hookrightarrow_{\mathbb{H}} s$ if $\mathbb{H}(t)$ contains a μ-reduction sequence ending with s. Given a μ-strategy $\mathbb{H}$, a μ-reduction sequence of the form $t_1 \hookrightarrow_{\mathbb{H}} t_2 \hookrightarrow_{\mathbb{H}} \cdots$ is called an $\mathbb{H}$-*sequence*. A μ-strategy $\mathbb{H}$ is *μ-normalizing* if, for all μ-normalizing term t, there is no infinite $\mathbb{H}$-sequence starting from t (see [19] for further information about how to define μ-normalizing μ-strategies). A μ-strategy is infinitary μ-normalizing if there are no perpetual $\mathbb{H}$-sequences starting from terms that admit an infinitary μ-normalizing μ-rewrite sequence. A μ-strategy $\mathbb{H}$ is *innermost* if $\hookrightarrow_{\mathbb{H}} \subseteq \overset{i}{\hookrightarrow}^+_\mu$.

A μ-strategy $\mathbb{H}$ is *μ-terminating* if $\hookrightarrow_{\mathbb{H}}$ is terminating. A proof of μ-termination for the TRS $\mathcal{R}$ (see [12, 21]) can be used to prove μ-termination of (any) μ-strategy for $\mathcal{R}$. The following result connects μ-normalization and μ-termination of TRSs and μ-strategies.

Proposition 3 *Let $\mathcal{R}$ be a TRS, $\mu \in M_{\mathcal{R}}$, and $\mathbb{H}$ be a μ-strategy for $\mathcal{R}$. Then, $\mathbb{H}$ is μ-terminating if and only if $\mathcal{R}$ is μ-normalizing and $\mathbb{H}$ is μ-normalizing.*

Given a μ-strategy $\mathbb{H}$ for a TRS $\mathcal{R}$, we define the following semantics:

$$\mathsf{cs\text{-}nf}_{\mathbb{H}}(t) = \{s \in \mathcal{T}(\mathcal{F}, \mathcal{X}) \mid t \hookrightarrow^!_{\mathbb{H}} s\}$$

and analogously for the infinitary counterpart, $\omega\text{-}\mathsf{cs\text{-}nf}_{\mathbb{H}}$. Since every μ-strategy is forced to perform μ-rewriting steps unless faced to a μ-normal form, we have the following:

Theorem 2 *Let $\mathcal{R}$ be a TRS, $\mu \in M_{\mathcal{R}}$, and $\mathbb{H}$ be a μ-strategy for $\mathcal{R}$. Then,* $\mathsf{cs\text{-}nf}_{\mathbb{H}}$ *is correct w.r.t.* $\mathsf{cs\text{-}nf}_{\mu}$.

Again, this result does not hold for ω-$\mathsf{cs\text{-}nf}_{\mathbb{H}}$ w.r.t. ω-$\mathsf{cs\text{-}nf}_{\mu}$ unless $\mathbb{H}$ is infinitary μ-normalizing. We also have:

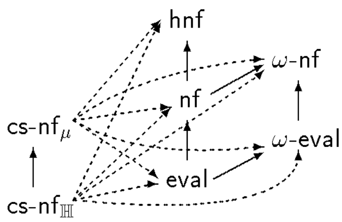
Theorem 3 *Let $\mathcal{R}$ be a TRS, $\mu \in M_{\mathcal{R}}$, and $\mathbb{H}$ be a μ-normalizing μ-strategy for $\mathcal{R}$. If $\mathcal{R}$ is μ-confluent, then* $\mathsf{cs\text{-}nf}_{\mathbb{H}} = \mathsf{cs\text{-}nf}_{\mu}$.

4 Computing canonical forms

In general, $\mathsf{cs\text{-}nf}_{\mu}$ or $\mathsf{cs\text{-}nf}_{\mathbb{H}}$ are *not* correct or complete w.r.t. eval, nf or hnf. However, since $\hookrightarrow_{\mu} \subseteq \to$ for every replacement map μ, we have the following obvious fact.

Proposition 4 *Let $\mathcal{R}$ be a TRS and $\mu \in M_{\mathcal{R}}$. Then,* $\mathsf{cs\text{-}nf}_{\mu}$ *is partially correct w.r.t.* eval, nf, *and* hnf.

As a consequence of Theorem 2, and Propositions 2 and 4, some new relationships between the different semantics arise:



For left-linear TRSs $\mathcal{R}$ and $\mu \in CM_{\mathcal{R}}$, the μ-normal forms are head-normal forms (Theorem 1). Thus, we have:

Theorem 4 *Let $\mathcal{R}$ be a left-linear TRS, $\mu \in CM_{\mathcal{R}}$, and $\mathbb{H}$ be a μ-strategy for $\mathcal{R}$. Then,* $\mathsf{cs\text{-}nf}_{\mu}$ *and* $\mathsf{cs\text{-}nf}_{\mathbb{H}}$ *are correct w.r.t.* hnf.

Correctness of $\mathsf{cs\text{-}nf}_{\mu}$ or $\mathsf{cs\text{-}nf}_{\mathbb{H}}$ w.r.t. eval or nf is more difficult to achieve. With regard to the computation of constructor terms, *CSR* exhibits some interesting properties. Given a TRS $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ and $\mathcal{B} \subseteq \mathcal{C}$, we let $\mu_{\mathcal{R}}^{\mathcal{B}}$ to be $\mu_{\mathcal{R}}^{\mathcal{B}}(c) = \{1, \ldots, ar(c)\}$ for all $c \in \mathcal{B}$ and $\mu_{\mathcal{R}}^{\mathcal{B}}(f) = \mu_{\mathcal{R}}^{can}(f)$ if $f \in \mathcal{F} - \mathcal{B}$. Note that $\mu_{\mathcal{R}}^{\mathcal{B}} \in CM_{\mathcal{R}}$. Now we let $EvM_{\mathcal{R},\mathcal{B}} = \{\mu \in M_{\mathcal{R}} \mid \mu_{\mathcal{R}}^{\mathcal{B}} \sqsubseteq \mu\}$.

Theorem 5 [18] *Let $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ be a left-linear TRS, $\mathcal{B} \subseteq \mathcal{C}$ and $\mu \in EvM_{\mathcal{R},\mathcal{B}}$. Let $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and $\delta \in \mathcal{T}(\mathcal{B}, \mathcal{X})$. Then, $t \to^* \delta$ iff $t \hookrightarrow_{\mu}^* \delta$.*

Theorem 5 is very easy to use in sorted signatures and TRSs (e.g., in OBJ programs): given a term t (of sort τ), we let $\mathcal{B} \subseteq \mathcal{C}$ to be the set of constructor symbols which are used to build constructor values of sort τ. As immediate consequence, we obtain the following.

Theorem 6 *Let $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ be a left-linear, confluent TRS, $\mathcal{B} \subseteq \mathcal{C}$, $\mu \in EvM_{\mathcal{R},\mathcal{B}}$ and $\mathbb{H}$ be a μ-normalizing μ-strategy for $\mathcal{R}$. Let $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and $\delta \in \mathcal{T}(\mathcal{B}, \mathcal{X})$. Then, $\delta \in \mathsf{eval}(t)$ if and only if $\delta \in \mathsf{cs\text{-}nf}_{\mathbb{H}}(t)$.*

Therefore, we can use any μ-normalizing μ-strategy to *directly* obtain the values (in $\mathcal{T}(\mathcal{B}, \mathcal{X})$) associated to a term t, provided that μ is taken from $EvM_{\mathcal{R},\mathcal{B}}$. Note that we do not need μ-confluence for ensuring this result.

Even though Theorems 5 and 6 do not express (full) correctness or completeness of $\mathsf{cs\text{-}nf}$ or $\mathsf{cs\text{-}nf}_{\mathbb{H}}$ w.r.t. eval, they are useful in practice.

A TRS $\mathcal{R} = (\mathcal{C} \uplus \mathcal{D}, R)$ is completely defined (CD) if no ground normal form contains defined symbols (see [15, 17] for methods for checking completely definedness of TRSs). Semantics eval and nf of a completely defined TRS $\mathcal{R}$ are equivalent over ground terms t: $\mathsf{eval}(t) = \mathsf{nf}(t)$. When considering context-sensitive computations, we have the following.

Proposition 5 *Let $\mathcal{R} = (\mathcal{C} \uplus \mathcal{D}, R)$ be a left-linear, completely defined TRS and $\mu \in EvM_{\mathcal{R},\mathcal{C}}$. There is no ground μ-normal form containing defined symbols.*

Left-linearity is necessary for Proposition 5 to hold.

Example 5 *Consider the following TRS $\mathcal{R}$ [19]:*

```
f(x,x) → b
a      → b
```

and $\mu(f) = \varnothing$. Note that $\mathcal{R}$ is completely defined, but there is a μ-normal form `f(a,b)` *containing a defined symbol* `f`*.*

As a consequence of Proposition 5, Theorem 5, and Theorem 6, we have the following.

Theorem 7 *Let $\mathcal{R} = (\mathcal{C} \uplus \mathcal{D}, R)$ be a left-linear TRS, $\mu \in EvM_{\mathcal{R},\mathcal{C}}$, and $\mathbb{H}$ be a μ-strategy for $\mathcal{R}$. Then,*

1. *If $\mathcal{R}$ is completely defined, then $\mathsf{cs\text{-}nf}_{\mu}$ and $\mathsf{cs\text{-}nf}_{\mathbb{H}}$ are correct (over ground terms) w.r.t. eval.*
2. *Semantics $\mathsf{cs\text{-}nf}_{\mu}$ is complete w.r.t. eval.*
3. *If $\mathcal{R}$ is confluent and $\mathbb{H}$ is μ-normalizing, then $\mathsf{cs\text{-}nf}_{\mathbb{H}}$ is complete w.r.t. eval.*

Although Theorem 7(1) requires to use CD TRSs to achieve correctness w.r.t. eval, we note that Theorems 7(2) and 7(3) together with Propositions 1 and 4 provide a suitable framework to compute constructor terms which does not require completely definedness.

4.1 Computing normal forms

The *maximal replacing context* $MRC^\mu(t)$ of t consists of the maximal part of t whose positions are μ-replacing in t, see [19]. When considering left-linear TRSs $\mathcal{R}$ and replacement maps $\mu \in CM_\mathcal{R}$, every μ-strategy $\mathbb{H}$ can be extended to a strategy $\mathbb{S}_\mathbb{H}$ as follows [19]:

$$\mathbb{S}_\mathbb{H}(t) = \begin{cases} \mathbb{H}(t) & \text{if } t \notin \mathsf{NF}^\mu_\mathcal{R} \\ C[\mathbb{S}_\mathbb{H}(t_1), \ldots, \mathbb{S}_\mathbb{H}(t_n)] & \text{if } t \in \mathsf{NF}^\mu_\mathcal{R} - \mathsf{NF}_\mathcal{R}, \\ \quad \text{where } C[\,] = MRC^\mu(t) \text{ and } t = C[t_1, \ldots, t_n] & \\ \varnothing & \text{otherwise} \end{cases}$$

Here, for a given context $C[\,]$ and sets of rewrite sequences $S_1, \ldots, S_n$, issued form terms $t_1, \ldots, t_n$, $C[S_1, \ldots, S_n]$ is the set of sequences from $C[t_1, \ldots, t_n]$ to $C[s_1, \ldots, s_n]$, where, for $1 \leq i \leq n$, either $s_i = t_i$ (during the whole sequence) or s_i is the end point of a sequence in S_i (also, at least one of the s_i must be taken in this way). We have the following:

Theorem 8 [19] *Let $\mathcal{R}$ be a left-linear, confluent TRS and $\mu \in CM_\mathcal{R}$. If $\mathbb{H}$ is a μ-normalizing μ-strategy, then $\mathbb{S}_\mathbb{H}$ is normalizing.*

Note that (again) we do *not* need μ-confluence for ensuring this result. Therefore, for left-linear, confluent TRSs $\mathcal{R}$ and $\mu \in CM_\mathcal{R}$, the 'normalization via μ-normalization' procedure expressed by the definition of $\mathbb{S}_\mathbb{H}$ for a given μ-strategy $\mathbb{H}$ provides a systematic way to obtain normalizing strategies from μ-normalizing μ-strategies $\mathbb{H}$.

Theorem 9 *Let $\mathcal{R} = (\mathcal{C} \uplus \mathcal{D}, R)$ be a left-linear TRS, $\mu \in CM_\mathcal{R}$, and $\mathbb{H}$ be a μ-strategy for $\mathcal{R}$. Then,*

1. *Semantics $\mathsf{nf}_{\mathbb{S}_\mathbb{H}}$ is correct w.r.t.* hnf *and* nf.
2. *If $\mathcal{R}$ is completely defined, then*
   (a) *$\mathsf{nf}_{\mathbb{S}_\mathbb{H}}$ is correct (over ground terms) w.r.t.* eval.
   (b) *If $\mu \in EvM_{\mathcal{R},\mathcal{C}}$, then $\mathsf{nf}_{\mathbb{S}_\mathbb{H}}$ is equivalent (over ground terms) to* $\mathsf{cs\text{-}nf}_\mathbb{H}$.
3. *If $\mathcal{R}$ is confluent and $\mathbb{H}$ is μ-normalizing, then $\mathsf{nf}_{\mathbb{S}_\mathbb{H}} = \mathsf{nf}$.*

In Figure 2 we summarize the conditions for achieving correctness and completeness in context-sensitive (based) finitary computations w.r.t. different semantics. Obvious conditions entailing determinism of the semantics are also given.

4.2 Computing infinite values and normal forms

In [24] we have shown that, if we do not want to consider transfinite rewrite sequences, infinitary strongly convergent sequences suffice for computing the

w.r.t.	Correctness of cs-nf$_\mu$	cs-nf$_{\mathbb{H}}$	nf$_{\mathbb{S}_{\mathbb{H}}}$
cs-nf$_\mu$	✓	✓	$\mu_\top$
hnf	LL, $CM_{\mathcal{R}}$	LL, $CM_{\mathcal{R}}$	LL, $CM_{\mathcal{R}}$
eval	LL, CD, $EvM_{\mathcal{R},\mathcal{C}}$	LL, CD, $EvM_{\mathcal{R},\mathcal{C}}$	LL, CD, $CM_{\mathcal{R}}$
nf	$\mu_\top$	$\mu_\top$	LL, $CM_{\mathcal{R}}$

w.r.t.	Completeness of cs-nf$_\mu$	cs-nf$_{\mathbb{H}}$	nf$_{\mathbb{S}_{\mathbb{H}}}$
cs-nf$_\mu$	✓	μ-CR, $\mathbb{H}$ μ-norm	CR, $\mathbb{H}$ $\mu_\top$-norm
eval	LL, $EvM_{\mathcal{R},\mathcal{C}}$	LL, CR, $EvM_{\mathcal{R},\mathcal{C}}$, $\mathbb{H}$ μ-norm	LL, CR, $CM_{\mathcal{R}}$, $\mathbb{H}$ μ-norm
nf	$\mu_\top$	CR, $\mathbb{H}$ $\mu_\top$-norm	LL, CR, $CM_{\mathcal{R}}$, $\mathbb{H}$ μ-norm

Determinism of cs-nf$_\mu$	cs-nf$_{\mathbb{H}}$	nf$_{\mathbb{S}_{\mathbb{H}}}$
μ-CR	μ-CR, or $\mathbb{H}$ deterministic	LL, CR, $CM_{\mathcal{R}}$

CR: confluence LL: Left-linearity

Figure 2: Correctness, completeness, and determinism for CS-computations (finitary case)

infinite values that can be obtained from finite terms[6] ([24], Theorem 1). Thus, restricting the attention to strongly convergent sequences in the definition of ω-eval does not entail any lose of generality[7]. Regarding the ability of *CSR* to compute infinite values, we prove the following 'infinitary version' of Theorem 5.

Theorem 10 *Let $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ be a left-linear TRS, $\mathcal{B} \subseteq \mathcal{C}$ and $\mu \in EvM_{\mathcal{R},\mathcal{B}}$. Let $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and $\delta \in \mathcal{T}^\omega(\mathcal{B}, \mathcal{X})$. Then, $t \to^{\leq\omega} \delta$ iff $t \hookrightarrow_\mu^{\leq\omega} \delta$.*

The role of (finitary) confluence to ensure good semantic properties of infinitary computations has been investigated in [24]. Confluence ensures unicity of infinite constructor normal forms ([24], Theorems 6 and 7). Unfortunately, with confluent TRSs we can still have terms with more than one (possibly infinite) normal form.

[6]For left-linear, confluent TRSs, there is no need to consider transfinite sequences at all [24], Theorem 4.

[7]Regarding the computation of infinite normal forms (i.e., ω-nf), we need to restrict the attention to left-bounded orthogonal TRSs, where a TRS is left-bounded (LB) if the depth of the left-hand sides of its rules is bounded [16, 24].

Example 6 *Consider the confluent TRS $\mathcal{R}$ [24]:*

$$\begin{aligned} \mathtt{f(a)} &\to \mathtt{f(f(a))} \\ \mathtt{f(a)} &\to \mathtt{a} \end{aligned}$$

where only symbols $\mathtt{a}$ *and* $\mathtt{f}$ *belong to the underlying signature. Term* $\mathtt{f(a)}$ *has a finite (constructor) normal form* $\mathtt{a}$, *and an infinite (non-constructor) normal form* $\mathtt{f}^\omega$.

Unicity of infinite normal forms obtained from infinitary strongly converging sequences is ensured for orthogonal TRSs ([16], Theorem 7.15). According to these results, we have the following.

Theorem 11 *Let $\mathcal{R} = (\mathcal{C} \uplus \mathcal{D}, R)$ be a left-linear TRS, $\mu \in EvM_{\mathcal{R},\mathcal{C}}$, and $\mathbb{H}$ be a μ-strategy for $\mathcal{R}$. Then,*

1. *Semantics ω-cs-nf_μ is complete w.r.t. ω-eval.*
2. *If $\mathcal{R}$ is orthogonal and $\mathbb{H}$ is infinitary μ-normalizing, then ω-cs-$\mathsf{nf}_\mathbb{H}$ is complete w.r.t. ω-eval.*

Note that the infinitary counterpart of Theorem 7(1) is missing in Theorem 11. TRS $\mathcal{R}$ in Example 6 can be used to show that completely definedness do *not* ensure correctness of ω-cs-nf_μ w.r.t. ω-eval. It also shows that confluence does not ensure completeness of ω-cs-$\mathsf{nf}_\mathbb{H}$ w.r.t. ω-eval.

Example 7 *(Continuing Ex. 6) Note that*

$$\omega\text{-}\mathsf{eval}(\mathtt{f(a)}) = \{\mathtt{a}\}, \ \textit{whereas}\ \omega\text{-}\mathsf{cs}\text{-}\mathsf{nf}_{\mu_\top}(\mathtt{f(a)}) = \{\mathtt{a}, \mathtt{f}^\omega\}.$$

Thus, ω-cs-$\mathsf{nf}_{\mu_\top}$ is not correct w.r.t. ω-eval. On the other hand, a $\mu_\top$-strategy $\mathbb{H}$ which only uses the first rule of $\mathcal{R}$ in Example 6 for reducing terms is infinitary $\mu_\top$-normalizing but $\mathbb{H}$ will never obtain the (finite) constructor normal form $\mathtt{a}$ *of* $\mathtt{f(a)}$. *Thus, ω-cs-$\mathsf{nf}_\mathbb{H}$ is not complete w.r.t. ω-eval. Note that $\mathcal{R}$ is completely defined.*

Regarding infinitary normalization with left-linear TRSs $\mathcal{R}$, replacement maps $\mu \in CM_\mathcal{R}$, and a μ-strategy $\mathbb{H}$ for $\mathcal{R}$, we can use the following strategy [19]:

$$\mathbb{S}^{\|}_{\mathbb{H}}(t) = \begin{cases} \mathbb{H}(t) & \text{if } t \notin \mathsf{NF}^\mu_\mathcal{R} \\ C^{\|}[\mathbb{S}^{\|}_{\mathbb{H}}(t_1), \ldots, \mathbb{S}^{\|}_{\mathbb{H}}(t_n)] & \text{if } t \in \mathsf{NF}^\mu_\mathcal{R} - \mathsf{NF}_\mathcal{R}, \\ \quad \text{where } C[\,] = MRC^\mu(t) \text{ and } t = C[t_1, \ldots, t_n] & \\ \varnothing & \text{otherwise} \end{cases}$$

Here, for a given context $C[\,]$ and sets of rewrite sequences $S_1, \ldots, S_n$, issued from terms $t_1, \ldots, t_n$, we let $C^{\|}[S_1, \ldots, S_n]$ denote the set of derivations from $C[t_1, \ldots, t_n]$ to $C[s_1, \ldots, s_n]$ such that there is $i \in \{1, \ldots, n\}$ such that t_i is not a normal form and for all $1 \leq j \leq n$, either t_j is not a normal form (and is $t_j \to^+ s_j \in S_j$ as well), or t_j is a normal form and $s_j = t_j$.

Theorem 12 *Let $\mathcal{R}$ be a left-linear TRS and $\mu \in CM_{\mathcal{R}}$. If $\mathbb{H}$ is a μ-terminating μ-strategy, then $\mathbb{S}^{\|}_{\mathbb{H}}$ is infinitary normalizing.*

Example 8 *Consider the TRS $\mathcal{R}$ [19]:*

$\mathtt{a} \to \mathtt{c(a,a)}$

and $\mu(\mathtt{c}) = \varnothing$. It is easy to see that $\mathcal{R}$ is μ-terminating. Therefore, every μ-strategy is also μ-terminating. Since the only μ-reducible term is a*, there is only one possible μ-strategy $\mathbb{H}$ for $\mathcal{R}$. We have the following $\mathbb{S}^{\|}_{\mathbb{H}}$-sequence*

$\underline{\mathtt{a}} \to_{\mathbb{S}^{\|}_{\mathbb{H}}} \mathtt{c}(\underline{\mathtt{a}},\underline{\mathtt{a}}) \to_{\mathbb{S}^{\|}_{\mathbb{H}}} \mathtt{c}(\mathtt{c}(\underline{\mathtt{a}},\underline{\mathtt{a}}),\mathtt{c}(\underline{\mathtt{a}},\underline{\mathtt{a}})) \to_{\mathbb{S}^{\|}_{\mathbb{H}}} \cdots$

which clearly converges to the infinite normal form of a*.*

Interestingly, μ-termination (rather than μ-normalization) of μ-strategies is helpful for achieving *infinitary normalization*. The following example shows that requiring μ-normalization does not suffice to ensure infinitary normalization with $\mathbb{S}^{\|}_{\mathbb{H}}$.

Example 9 *Consider the TRS $\mathcal{R}$ of Example 8 and $\mu(\mathtt{c}) = \{1\}$. Note that term* a *has no μ-normal form now. Again, there is only one possible μ-strategy $\mathbb{H}$ for $\mathcal{R}$ which is trivially μ-normalizing (for every term t, if t contains a replacing occurrence of* a*, then t has no μ-normal form; otherwise, t is a μ-normal form). Since* a *has no μ-normal form, both $\mathbb{S}^{\|}_{\mathbb{H}}$ and $\mathbb{H}$ perform the same reduction steps which correspond to the infinite sequence*

$\underline{\mathtt{a}} \hookrightarrow \mathtt{c}(\underline{\mathtt{a}},\mathtt{a}) \hookrightarrow \mathtt{c}(\mathtt{c}(\underline{\mathtt{a}},\mathtt{a}),\mathtt{a}) \hookrightarrow \cdots$

which does not converge to the infinite normal form of a*.*

Moreover, note that this is a proper feature of μ-strategies which does not apply to rewriting strategies; obviously, terminating rewriting strategies cannot be used to obtain *infinite* normal forms. Note, however, that if $\mathbb{H}$ is terminating, then ω-cs-$\mathsf{nf}_{\mathbb{H}} = \mathsf{cs}$-$\mathsf{nf}_{\mathbb{H}}$.

Remark 5 *Note that $\mathbb{S}_{\mathbb{H}}$ and $\mathbb{S}^{\|}_{\mathbb{H}}$ are equivalent regarding normalization, i.e., $\mathsf{nf}_{\mathbb{S}_{\mathbb{H}}} = \mathsf{nf}_{\mathbb{S}^{\|}_{\mathbb{H}}}$ for all μ-strategy $\mathbb{H}$. We also note that, if $\mathbb{H}$ is terminating, then $\mathbb{H}$-sequences and $\mathbb{S}^{\|}_{\mathbb{H}}$-sequences are strongly converging.*

By using these results, we can also prove the following.

Theorem 13 *Let $\mathcal{R}$ be a left-linear TRS, $\mu \in CM_{\mathcal{R}}$, and $\mathbb{H}$ be a μ-terminating μ-strategy for $\mathcal{R}$. Then,*

1. *ω-$\mathsf{nf}_{\mathbb{S}^{\|}_{\mathbb{H}}}$ is correct w.r.t. ω-nf.*
2. *If $\mathcal{R}$ is completely defined and $\mu \in EvM_{\mathcal{R},\mathcal{C}}$, then ω-$\mathsf{nf}_{\mathbb{S}^{\|}_{\mathbb{H}}}$ is equivalent (over ground terms) to cs-$\mathsf{nf}_{\mathbb{H}}$.*
3. *If $\mathcal{R}$ is orthogonal, then ω-$\mathsf{nf}_{\mathbb{S}^{\|}_{\mathbb{H}}}$ is complete w.r.t. ω-eval, and ω-$\mathsf{nf}_{\mathbb{S}^{\|}_{\mathbb{H}}} = \omega$-$\mathsf{nf}$*

Now, we introduce a new criterion for guaranteeing unique normal forms by infinitary rewriting by requiring confluence and termination of *CSR*.

Theorem 14 *Let $\mathcal{R}$ be a left-linear, confluent TRS and $\mu \in CM_{\mathcal{R}}$ be such that $\mathcal{R}$ is μ-terminating. Let $t \in \mathcal{T}(\mathcal{F},\mathcal{X})$. If $t \to^{\leq\omega} s_1$ and $t \to^{\leq\omega} s_2$ for normal forms s_1, s_2, then $s_1 = s_2$.*

Note that $\mathcal{R}$ in Example 6 is confluent but *not* μ-terminating for any $\mu \in CM_{\mathcal{R}}$ (note that $CM_{\mathcal{R}} = \{\mu_\top\}$). By lack of space, we refer the reader to [19], Section 11.1 for a non-trivial example of TRS to which Theorem 14 applies, whereas results in [6, 16] do not.

In [21], we prove that, for left-linear TRSs $\mathcal{R}$ and $\mu \in CM_{\mathcal{R}}$, μ-termination of $\mathcal{R}$ implies top-termination of $\mathcal{R}$ (a TRS is *top-terminating* if no infinitary reduction sequence performs infinitely many rewrites at topmost position Λ [6]). Interestingly, top-termination is not sufficient to ensure Theorem 14, as the following example shows:

Example 10 *Consider the TRS $\mathcal{R}$ [6]:*

$$\begin{array}{llll} \texttt{g(x,a)} \to \texttt{f(g(x,a))} & & \texttt{a} & \to \texttt{b} \\ \texttt{g(x,b)} \to \texttt{c} & & \texttt{f(c)} & \to \texttt{c} \end{array}$$

As remarked in [6], $\mathcal{R}$ is top-terminating and locally confluent. It is not difficult to see that it is confluent, indeed. However, `g(x,a)` *has two normal forms:* `c` *and* $\texttt{f}^\omega$.

We have the following corollary of Theorem 14.

Corollary 1 *Let $\mathcal{R}$ be a left-linear, confluent TRS, and $\mu \in CM_{\mathcal{R}}$. If $\mathcal{R}$ is μ-terminating, then, for all μ-strategy $\mathbb{H}$ for $\mathcal{R}$, $\omega\text{-}\mathsf{nf}_{\mathbb{S}^{\parallel}_{\mathbb{H}}}$ is complete w.r.t. $\omega\text{-}\mathsf{eval}$, and $\omega\text{-}\mathsf{nf}_{\mathbb{S}^{\parallel}_{\mathbb{H}}} = \omega\text{-}\mathsf{nf}$.*

5 E-strategies

A local strategy (or E-strategy) for a k-ary symbol $f \in \mathcal{F}$ is a sequence $\varphi(f)$ of non-negative integers[8] taken from $\{0, 1, \ldots, k\}$. In the OBJ family of languages, they are given as sequences of numbers in parentheses. A mapping φ that associates a local strategy $\varphi(f)$ to every $f \in \mathcal{F}$ is called an E-strategy map [27].

Eker [7] and Nagaya [26] largely motivate the interest of requiring that 0 be the last index of local strategies associated to defined symbols. Since this requirement is essential in our development, we introduce the following[9]:

Definition 2 (Regular E-strategy) *We say that an E-strategy map φ for a signature $\mathcal{F} = \mathcal{C} \uplus \mathcal{D}$ is regular if for all $f \in \mathcal{D}$, $\varphi(f)$ ends in 0.*

[8] The use of negative integers has also been proposed in local strategies aimed at relaxing the replacement restrictions, see [1] for a discussion on the topic.

[9] The adjective 'regular' that we adopt here is inspirated by [30], where it is used with a slightly different meaning.

The following definition is used below.

Definition 3 [23] *Given a TRS* $\mathcal{R} = (\mathcal{C} \uplus \mathcal{D}, R)$*, we say that an* E*-strategy map* φ *is* elementary *for* $\mathcal{R}$ *if for all* $f \in \mathcal{D}$*,* $\varphi(f) = (i_1 \cdots i_n\ \mathtt{0})$ *and* $i_j > 0$ *for* $1 \leq j \leq n$*.*

Given a regular E-strategy φ, we obtain an elementary E-strategy $\llcorner\varphi\lrcorner$ as follows: for all symbol $f \in \mathcal{D}$, $\llcorner\varphi\lrcorner(f)$ is the prefix of $\varphi(f)$ which contains a single occurrence of 0.

5.1 Default E-strategies

As remarked above, symbols without an explicit local strategy are given a *default* one whose concrete shape depends on the language considered:

1. In Maude, the default local strategy associated to a k-ary symbol f, is $(1\ 2\ \cdots k\ 0)$, see [5].
2. In OBJ3, the default local strategy associated to a k-ary symbol f 'is determined from its equations by requiring that all argument places that contain a non-variable term in some rule are evaluated before equations are applied at the top' and 'all arguments are reduced in some order' and 'either the operator has no rules or the strategy ends with a final zero' [13], Section 2.4.4. In other words: for defined symbols $f \in \mathcal{D}$, $\varphi(f) = (i_1, \ldots, i_m, 0, j_1, \ldots, j_n, 0)$ where $\{i_1, \ldots, i_m\} = \mu^{can}_{\mathcal{R}}(f)$ and $\{j_1, \ldots, j_n\} = \{1, \ldots, ar(f)\} - \mu^{can}_{\mathcal{R}}(f)$. For constructor symbols $c \in \mathcal{C}$, $\varphi(c) = (1\ 2\ \cdots k\ 0)$ (or $\varphi(c) = (1\ 2\ \cdots k)$ which is equivalent for constructor symbols).
3. In CafeOBJ, the default strategy is computed as follows in OBJ3 (see [28], Section 7.4.2).

Example 11 *The following table shows the* E*-strategies used in the program of Example 1 with* OBJ3 *and* Maude *interpreters (user defined strategies highlighted).*

FIRST-FROM-LENGTH		
Symbol	OBJ3	Maude
`0`	(0)	(0)
`s`	**(0)**	**(0)**
`nil`	(0)	(0)
`cons`	**(1 0)**	**(1 0)**
`first`	(1 2 0)	(1 2 0)
`from`	(0 1 0)	(1 0)
`add`	(1 0 2 0)	(1 2 0)
`length`	(1 0)	(1 0)

Note that default E-strategies in Maude, OBJ3, and CafeOBJ are regular. Moreover, Maude recently imposed that *all* E-strategies end in 0. In fact, the Maude interpreter automatically *adds* a zero at the end of every non-zero-ended local strategy (even for explicit ones) in order to achieve regularity. Indeed, the version of the OBJ3 interpreter that we used seems to do the same.

Remark 6 *Default strategies in* Maude *are elementary (the exception is annotation* (1 0 2 3 0) *for the* `if_then_else_fi` *operator). This is not the case for* OBJ3 *or* CafeOBJ *(see Example 11).*

5.2 Rewriting under E-strategies

Nagaya describes the operational semantics of term rewriting under E-strategy maps as follows [26]: Let $\mathcal{L}$ be the set of all lists consisting of natural numbers. By $\mathcal{L}_n$, we denote the set of all lists of natural numbers not exceding $n \in \mathbb{N}$. We use the signature $\mathcal{F}_\mathcal{L} = \{f_L \mid f \in \mathcal{F} \wedge L \in \mathcal{L}_{ar(f)}\}$ and labelled variables $\mathcal{X}_\mathcal{L} = \{x_{nil} \mid x \in \mathcal{X}\}$. The set of labelled terms is, then, $\mathcal{T}(\mathcal{F}_\mathcal{L}, \mathcal{X}_\mathcal{L})$.

An E-strategy map φ for $\mathcal{F}$ is extended to a mapping from $\mathcal{T}(\mathcal{F}, \mathcal{X})$ to $\mathcal{T}(\mathcal{F}_\mathcal{L}, \mathcal{X}_\mathcal{L})$ as follows:

$$\varphi(t) = \begin{cases} x_{nil} & \text{if } t = x \in \mathcal{X} \\ f_{\varphi(f)}(\varphi(t_1), \ldots, \varphi(t_k)) & \text{if } t = f(t_1, \ldots, t_k) \end{cases}$$

The mapping $erase : \mathcal{T}(\mathcal{F}_\mathcal{L}, \mathcal{X}_\mathcal{L}) \to \mathcal{T}(\mathcal{F}, \mathcal{X})$ removes labellings from symbols in the obvious way.

The binary relation $\to_\varphi$ on $\mathcal{T}(\mathcal{F}_\mathcal{L}, \mathcal{X}_\mathcal{L}) \times \mathbb{N}^*_+$ (i.e., pairs $\langle t, p\rangle$ of labelled terms t and positions p) is [27, 26]: $\langle t, p\rangle \to_\varphi \langle s, q\rangle$ if and only if $p \in \mathcal{P}os(t)$ and either

1. $root(t|_p) = f_{nil}$, $s = t$ and $p = q.i$ for some i; or
2. $t|_p = f_{i:L}(t_1, \ldots, t_k)$ with $i > 0$, $s = t[f_L(t_1, \ldots, t_k)]_p$ and $q = p.i$; or
3. $t|_p = f_{0:L}(t_1, \ldots, t_k)$, $erase(t|_p)$ is not a redex, $s = t[f_L(t_1, \ldots, t_k)]_p$, $q = p$; or
4. $t|_p = f_{0:L}(t_1, \ldots, t_k) = \sigma(l')$, $erase(l') = l$, $s = t[\sigma(\varphi(r))]_p$ for some $l \to r \in R$ and substitution σ, $q = p$.

Given pairs $\langle t, p\rangle$ and $\langle s, q\rangle$, if $\langle t, p\rangle \to_\varphi \langle s, q\rangle$ using one of the first three (*traversal*) steps above, we write $\langle t, p\rangle \stackrel{\text{T}}{\to}_\varphi \langle s, q\rangle$. On the other hand, if the last (rewriting) step is used, we write $\langle t, p\rangle \stackrel{\text{R}}{\to}_\varphi \langle s, q\rangle$. This defines two auxiliary reduction relations $\stackrel{\text{T}}{\to}_\varphi$ and $\stackrel{\text{R}}{\to}_\varphi$ on pairs of labelled terms and positions which we use later. Obviously, $\to_\varphi = \stackrel{\text{T}}{\to}_\varphi \cup \stackrel{\text{R}}{\to}_\varphi$. Note that $\stackrel{\text{T}}{\to}_\varphi$ is deterministic (i.e., if $\langle t, p\rangle \stackrel{\text{T}}{\to}_\varphi \langle s, q\rangle$ and $\langle t, p\rangle \stackrel{\text{T}}{\to}_\varphi \langle s', q'\rangle$, then $s = s'$ and $q = q'$), hence confluent. Moreover, $\stackrel{\text{T}}{\to}_\varphi$ is obviously terminating.

Given a TRS $\mathcal{R} = (\mathcal{F}, R)$ and a E-strategy map φ for $\mathcal{F}$, $eval_\varphi : \mathcal{T}(\mathcal{F}, \mathcal{X}) \to \mathcal{P}(\mathcal{T}(\mathcal{F}, \mathcal{X}))$ is defined as [26]:

$$eval_\varphi(t) = \{erase(s) \in \mathcal{T}(\mathcal{F}, \mathcal{X}) \mid \langle \varphi(t), \Lambda\rangle \to^!_\varphi \langle s, \Lambda\rangle\}.$$

Remark 7 *Command* `red` *(or* `reduce`*) which is available in existing* OBJ *interpreters implements the semantic function* $eval_\varphi$. *In principle, the evaluation of expressions in* OBJ *languages produces a single output expression of no expression at all (i.e., the evaluation does not terminate). It is, however, possible to obtain all possible values associated to an input expression* t *by explicitly activating such a multiple evaluation in the interpreter. This only has sense if the program, viewed as a TRS, contains* overlays, *i.e., critical pairs whose overlapping position is the root position* Λ.

A TRS $\mathcal{R}$ is *φ-terminating* if, for all $t \in \mathcal{T}(\mathcal{F},\mathcal{X})$, there is no infinite $\rightarrow_\varphi$-rewrite sequence starting from $\langle\varphi(t),\Lambda\rangle$ [22, 23]. The following proposition is used below.

Proposition 6 *Let* $\mathcal{R}$ *be a TRS and* φ *be an E-strategy map for* $\mathcal{R}$. *Then,*

$$eval_\varphi(t) = \{erase(s) \in \mathcal{T}(\mathcal{F},\mathcal{X}) \mid \langle\varphi(t),\Lambda\rangle(\stackrel{\mathrm{T}}{\longrightarrow}^!_\varphi \circ \stackrel{\mathrm{R}}{\rightarrow}_\varphi)^!\langle s,p\rangle\}.$$

5.3 E-strategies as context-sensitive rewriting strategies

We write $e \in L$ to denote that item e appears somewhere within the list L.

Definition 4 *Given a E-strategy map* φ *for* $\mathcal{F}$, *we define* $\mu^\varphi \in M_\mathcal{F}$ *as follows: for all* $f \in \mathcal{F}$, $\mu^\varphi(f) = \{i > 0 \mid i \in \varphi(f)\}$.

Remark 8 *Since we use* sets *of indices rather than* lists, *we loose the information concerning* repeated *indices and the* order *of reductions. Duplicated positive indices do not add computational power to E-strategies (Corollary 3.2 in [7]). Contiguous occurrences of zero can be simplified into a single one (Corollary 3.3 in [7]); we, however, drop all occurrences of zero as they are useless to configure a replacement map.*

By abuse, we write $\varphi \in CM_\mathcal{R}$, $\varphi \in EvM_{\mathcal{R},\mathcal{C}}$, etc., if $\mu^\varphi \in CM_\mathcal{R}$, $\mu^\varphi \in EvM_{\mathcal{R},\mathcal{C}}$, etc. There is a very close connection between $\rightarrow_\varphi$-reduction and $\hookrightarrow_\mu$-reduction: each computation step induced by an E-strategy map φ corresponds to a (possibly empty) context-sensitive reduction step using μ^φ [22, 23].

Theorem 15 *Let* $\mathcal{R}$ *be a TRS and* φ *be an E-strategy map. Let* $t \in \mathcal{T}(\mathcal{F}_\mathcal{L},\mathcal{X}_\mathcal{L})$, *and* $p \in \mathcal{P}os^{\mu^\varphi}(erase(t))$ *be s.t.* $root(t|_p) = f_L$ *for some suffix* L *of* $\varphi(f)$. *If* $\langle t,p\rangle \rightarrow_\varphi \langle s,q\rangle$, *then* $q \in \mathcal{P}os^{\mu^\varphi}(erase(s))$ *and*

1. *if* $\langle t,p\rangle \stackrel{\mathrm{T}}{\rightarrow}_\varphi \langle s,q\rangle$, *then* $erase(t) = erase(s)$.
2. *if* $\langle t,p\rangle \stackrel{\mathrm{R}}{\rightarrow}_\varphi \langle s,q\rangle$, *then* $erase(t) \hookrightarrow_{\mu^\varphi} erase(s)$.

Terms returned by $eval_\varphi$ are called *E-normal forms* (*ENF*s [7, 30]). We have the following.

Theorem 16 [22] *Let* $\mathcal{R}$ *be a TRS and* φ *be a regular E-strategy map for* $\mathcal{R}$. *If* $s \in eval_\varphi(t)$, *then* s *is a* μ*-normal form of* t.

Regularity of E-strategies ensures that ENFs are μ^φ-normal forms. For regular E-strategies φ, $eval_\varphi$ can be thought of as the specification of a μ^φ-rewriting strategy $\mathbb{H}_\varphi$ such that $\mathbb{H}_\varphi(t)$ contains a μ^φ-rewriting sequence

$$t = t_1 \hookrightarrow_{\mu^\varphi} t_2 \hookrightarrow_{\mu^\varphi} \cdots \hookrightarrow_{\mu^\varphi} t_n$$

for each sequence

$$\langle s_1, p_1\rangle \stackrel{\mathrm{T}}{\longrightarrow}^!_\varphi \circ \stackrel{\mathrm{R}}{\to}_\varphi \langle s_2, p_2\rangle \stackrel{\mathrm{T}}{\longrightarrow}^!_\varphi \circ \stackrel{\mathrm{R}}{\to}_\varphi \cdots \stackrel{\mathrm{T}}{\longrightarrow}^!_\varphi \circ \stackrel{\mathrm{R}}{\to}_\varphi \langle s_n, p_n\rangle$$

where $s_1 = \varphi(t)$, $p_1 = \Lambda$, $t_i = erase(s_i)$ for $i \geq 1$, and $\langle s_n, p_n\rangle$ is a $\stackrel{\mathrm{T}}{\longrightarrow}^!_\varphi \circ \stackrel{\mathrm{R}}{\to}_\varphi$-normal form. Note that the definition of $\mathbb{H}_\varphi$ is technically correct for φ-terminating TRSs; otherwise, there will be terms t which are not μ-normal forms for which $\mathbb{H}_\varphi(t) = \varnothing$. By construction of $\mathbb{H}_\varphi$, an using Proposition 6, Theorem 15, and Theorem 16, we have:

Theorem 17 *Let $\mathcal{R}$ be a TRS and φ be a regular E-strategy map for $\mathcal{R}$. If $\mathcal{R}$ is φ-terminating, then $eval_\varphi = \mathsf{cs\text{-}nf}_{\mathbb{H}_\varphi}$.*

We have the following immediate consequence of Theorems 16 and 17, and the results in Sections 3 and 4.

Theorem 18 *Let $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ be a TRS and φ be a regular E-strategy map for $\mathcal{R}$.*

1. *Semantics $eval_\varphi$ is correct w.r.t. $\mathsf{cs\text{-}nf}_{\mu^\varphi}$.*
2. *Semantics $eval_\varphi$ is partially correct w.r.t. hnf, eval, and nf.*
3. *If $\mathcal{R}$ is left-linear and $\varphi \in CM_\mathcal{R}$, then $eval_\varphi$ is correct w.r.t. hnf.*
4. *If $\mathcal{R}$ is left-linear and $\varphi \in EvM_{\mathcal{R},\mathcal{C}}$, then*
   - *(a) If $\mathcal{R}$ is completely defined, then $eval_\varphi$ is correct w.r.t. eval.*
   - *(b) If $\mathcal{R}$ is confluent and φ-terminating, then $eval_\varphi$ is complete w.r.t. eval.*
5. *Let $\mathcal{B} \subseteq \mathcal{C}$, $\varphi \in EvM_{\mathcal{R},\mathcal{B}}$, $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and $\delta \in \mathcal{T}(\mathcal{B}, \mathcal{X})$. If $\mathcal{R}$ is left-linear, confluent, and φ-terminating, then $\delta \in \mathsf{eval}(t)$ if and only if $\delta \in eval_\varphi(t)$.*

Some examples of use of Theorem 18 follow.

Example 12 *Consider the* OBJ3 *version of a TRS in [11]:*

```
obj DIFF is
  sort Nat .
  op 0 : -> Nat .
  ops p s : Nat -> Nat .
  op _<=_ : Nat Nat -> Bool .
  op _-_ : Nat Nat -> Nat .
```

```
  vars M N : Nat .
  eq p(s(N)) = N .
  eq 0 <= N = true .
  eq s(M) <= 0 = false .
  eq s(M) <= s(N) = M <= N .
  eq M - N = if M <= N then 0 else s(p(M) - N) fi .
endo
```

where Bool, true, false, *and* if_then_else_fi *are predefined in the 'prelude' of* OBJ3*. Termination of this program can be formally proved (see [11]); we note that the* OBJ3 *default annotation* (1 0) *for the* if_then_else_fi *operator is essential for this (in fact, the program is not terminating when executed using the* Maude *interpreter!). According to Theorem 18(2), $eval_\varphi$ is partially correct w.r.t.* eval*, i.e., every constructor term in $eval_\varphi(t)$ is a value of t. Also, $eval_\varphi$ is* complete *regarding* eval *(Theorem 18(4b)). This means that every constructor term of sort* Nat *that can be obtained from any expression t of sort* Nat*, can also be obtained using command* red *of* OBJ3*. Thus, terms s which are eventually obtained using* red *on a term t can be just disregarded if they are not constructor terms, since they cannot be further reduced to a constructor normal form of t (e.g., evaluate* p(0)*).*

Example 13 *The following program:*

```
obj SEL-FIRST-FROM is
  sorts Nat LNat .
  op 0     : -> Nat .
  op s     : Nat -> Nat .
  op nil   : -> LNat .
  op cons  : Nat LNat -> LNat [strat (1 0)] .
  op sel   : Nat LNat -> Nat .
  op first : Nat LNat -> LNat .
  op from  : Nat -> LNat .
  vars 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
```

specifies an explicit *strategy annotation* (1 0) *for the list constructor* cons *which disables reductions on the second argument. Termination of this program can also be formally proved. Program* SEL-FIRST-FROM *can eventually fail to fully evaluate expressions of sort* LNat*. However, according to Theorem 18(5), the program can indeed be used to obtain every constructor term of sort* Nat *that can be obtained from any expression t of sort* Nat *using command* red *of* OBJ3*.*

In some cases, $eval_\varphi$ can be more conveniently expressed as the semantics induced by the following (one-step) μ^φ-strategy:

$$\mathbb{H}'_\varphi(t) = \{t \hookrightarrow erase(s) \mid \langle\varphi(t),\Lambda\rangle \xrightarrow{\mathrm{T}}{}^!_\varphi \circ \xrightarrow{\mathrm{R}}_\varphi \langle s,p\rangle\}$$

Again, $\mathbb{H}'_\varphi$ is well-defined for regular E-strategies φ, but φ-termination is not necessary now.

Theorem 19 *Let $\mathcal{R}$ be a TRS and φ be a regular E-strategy for $\mathcal{R}$.*

1. *If $\mathcal{R}$ is orthogonal, then $eval_\varphi = \textsf{cs-nf}_{\mathbb{H}'_\varphi}$.*
2. *If φ is elementary, then $\mathbb{H}'_\varphi$ is an innermost μ^φ-strategy and $eval_\varphi = \textsf{cs-nf}_{\mathbb{H}'_\varphi}$.*

For non-elementary E-strategies (and non-orthogonal systems), Theorem 19 does not hold.

Example 14 *Consider the following TRS $\mathcal{R}$*

```
f(a) → c          a → c
b    → a
```

and $\varphi(\mathtt{f}) = (0\ 1\ 0)$, $\varphi(\mathtt{g}) = (1\ 0)$ *and* $\varphi(\mathtt{a}) = \varphi(\mathtt{b}) = (0)$. *Then,*

$\mathtt{f}(\underline{\mathtt{b}}) \hookrightarrow_{\mathbb{H}'_\varphi} \underline{\mathtt{f(a)}} \hookrightarrow_{\mathbb{H}'_\varphi} \mathtt{c}$

Thus, $\mathtt{c} \in \textsf{cs-nf}_{\mathbb{H}'_\varphi}(\mathtt{f(b)})$*, but* $eval_\varphi(\mathtt{f(b)}) = \{\mathtt{f(c)}\}$.

Remark 9 *The* elementary *part $\llcorner\varphi\lrcorner$ of any default E-strategy in* Maude *and* OBJ3 *is compatible with the canonical replacement map $\mu^{can}_{\mathcal{R}}$, i.e., $\mu^{\llcorner\varphi\lrcorner}(f) \subseteq \mu^{can}_{\mathcal{R}}(f)$ for all symbol f which is given a default strategy $\varphi(f)$.*

We have implemented the 'normalization via μ-normalization' procedure of Section 4.1 in the OnDemandOBJ interpreter[10] by introducing a new command `norm`.

Example 15 *Consider the program in Example 1. As shown in the example, command* `red` *fails to obtain the values of either sort* `Nat` *or* `LNat` *that correspond to some initial expressions. By using command* `norm` *of* OnDemandOBJ*:*

```
FIRST-FROM-LENGTH>
  norm add(length(first(s(0),from(0))),length(first(0,from(0)))) .
Normal form: s(0)
{ 0.0000 sec., 11 rewrites }
```

we obtain the desired normal form of the initial expression.

Roughly speaking, command `norm` of OnDemandOBJ implements the following semantics:

$$norm_\varphi = \textsf{nf}_{\mathbb{S}_{eval_\varphi}}$$

that is: the *ENF*s computed by $eval_\varphi$ are used in the normalization via μ-normalization process of Section 4.1. The following results are useful for using $norm_\varphi$.

[10] See `http://www.dsic.upv.es/users/elp/ondemandOBJ`.

Theorem 20 *Let $\mathcal{R} = (\mathcal{C} \uplus \mathcal{D}, R)$ be a TRS and φ be a regular E-strategy map for $\mathcal{R}$ such that $\varphi \in CM_{\mathcal{R}}$. Then,*

1. *Semantics $norm_\varphi$ is correct w.r.t.* hnf *and* nf.
2. *If $\mathcal{R}$ is completely defined, then*
   (a) *$norm_\varphi$ is correct (over ground terms) w.r.t.* eval.
   (b) *If $\varphi \in EvM_{\mathcal{R},\mathcal{C}}$, then $norm_\varphi$ is equivalent (over ground terms) to $eval_\varphi$.*
3. *If $\mathcal{R}$ is confluent and φ-terminating, then $norm_\varphi = \mathsf{nf}$.*

No command for 'infinitary normalization via μ-normalization' is available yet. We plan to use the results in Section 4.2 to address this task in the near future.

5.3.1 Related work

Correctness of OBJ computations regarding semantics nf has been studied in [26] and [27] (using a different, sometimes misleading, terminology). In principle, the strategy annotations can be fixed in such a way that E-normal forms are ensured to be normal forms, which guarantees correctness of OBJ computations w.r.t. nf:

Theorem 21 [26] *Let φ be a regular E-strategy map such that $\mu^\varphi(f) = \{1, \ldots, ar(f)\}$ for all symbol $f \in \mathcal{F}$. Then, $eval_\varphi$ is correct w.r.t.* nf.

Note that Theorem 21 does not apply to any program in this paper (remarkably, it does not apply to OBJ3 program `DIFF` in Example 12 as φ(`if_then_else`) = (1 0) is fixed by the interpreter).

Remark 10 *Default E-strategies in* Maude *fulfill the conditions in Theorem 21. In* OBJ3*, the* `if_then_else_fi` *built-in operator is the only exception.*

The following theorem by Nakamura and Ogata provides another interesting characterization of correctness regarding nf. In the following result, '++' appends two lists.

Theorem 22 [27] *Let φ be a regular E-strategy map such that $eval_\varphi$ is correct w.r.t.* hnf. *Then, for any φ' given by $\varphi'(f) = \varphi(f)\texttt{++}(i_1\ \cdots\ i_n)$ for all symbol $f \in \mathcal{F}$ and $\{i_1, \ldots, i_n\} = \{1, \ldots, ar(f)\} - \mu^\varphi(f)$, $eval_{\varphi'}$ is correct w.r.t.* nf.

Nevertheless, we do not consider that this is a very suitable way to achieve normal forms in practice (at least with non-terminating TRSs), as the apparent possibilities introduced by the relax of the replacement restrictions are often missed owing to problems of non-termination

Example 16 *Consider the program in Example 1. The evaluation of* t = `first(s(0),from(0))`*:*

```
Maude> red first(s(0),from(0)) .
reduce in FIRST-FROM-LENGTH : first(s(0), from(0)) .
rewrites: 2 in -10ms cpu (0ms real) (~ rewrites/second)
result LNat: cons(0, first(0, from(s(0))))
```

does not obtain the desired normal form `cons(0,nil)`*. By Theorem 18(3),* $eval_\varphi$ *is correct w.r.t.* hnf*. According to Theorem 22, we let* $\varphi'(\mathtt{s}) = (0\ 1)$*,* $\varphi'(\mathtt{cons}) = (1\ 0\ 2)$ *and* $\varphi'(f) = \varphi(f)$ *for every other symbol* f *(we label it* `SEL-FIRST-FROM-INF`*). We have:*

```
Maude> red first(s(0),from(0)) .
reduce in FIRST-FROM-LENGTH-INF : first(s(0), from(0)) .
Segmentation fault (core dumped)
```

The evaluation of t *does not terminate now; the* Maude *interpreter 'shows' this as a 'segment violation'. However, with* OnDemandOBJ *and* `norm`*, we obtain:*

```
FIRST-FROM-LENGTH> norm first(s(0),from(0)) .
Normal form: cons(0,nil)
{ 0.0000 sec., 4 rewrites }
```

With regard to the computation of normal forms by *directly* using the E-strategy, Nagaya gives conditions (on the TRS and the E-strategy φ) ensuring that φ is normalizing, i.e., it is able to compute a normal form of a term whenever it exists (i.e., he studies *completeness* of $eval_\varphi$ w.r.t. nf). However, these results concern quite a restricted subclass of orthogonal TRSs. Complementarily, Theorem 20 establishes that *completeness* of $norm_\varphi$ w.r.t. nf is possible for left-linear, confluent and φ-terminating TRSs.

Correctness or completeness w.r.t. eval has been not addressed before. Correctness or completeness w.r.t. infinitary semantics ω-eval and ω-nf either.

In [2] we discuss a transformational approach to achieve correct and complete computations under E-strategies regarding the evaluation semantics. In [1] we discuss the use of negative annotations to improve correctness and completeness of computations using E-strategies.

6 Van de Pol's strategy annotations

Van de Pol's strategies [31] are a refinement of the E-strategies: instead of using '0' to indicate that the application of *rules* must be attempted in some stage of computation, van de Pol permits to exactly specify what *a rule* should eventually be applied. Van de Pol's style of strategy annotations is the basis of the JITty system [32] which has been integrated in the μCRL tool set [3].

Example 17 *The following* JITty *program*

```
signature
  true(0)       if(3)
  false(0)
```

```
rules
  if1([x,y], if(true,x,y), x)
  if2([x,y], if(false,x,y), y)
  if3([x,y], if(x,y,y), y)
strategies
  if([1,if1,if2,2,3,if3])
end
```

indicates that, after evaluating the first argument of a call to `if`*, only the first and second rules (labelled* `if1` *and* `if2`*, respectively) can be attempted. If they fail, then the second and third arguments must be evaluated and finally only the third rule (with lable* `if3`*) is considered.*

Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS. According to van de Pol [31], a strategy annotation associated to a given symbol $f \in \mathcal{F}$ is a list $\varsigma(f)$ whose elements can be either

1. a number i with $1 \leq i \leq ar(f)$; or
2. a rule $l \to r \in R$ such that $root(t) = f$.

In principle, strategy annotations contain no duplicated items. We say that a strategy annotation ς is r-full if for all $l \to r \in R$, $l \to r \in \varsigma(root(l))$; we say that ς is full if it is r-full and for all $f \in \mathcal{F}$ and $i \in \{1, \ldots, ar(f)\}$, $i \in \varsigma(f)$. A strategy annotation ς is *in-time* if for all $f \in \mathcal{F}$, $\alpha : l \to r \in R$ such that $root(l) = f$, and $i \in \{1, \ldots, ar(f)\}$, whenever $\varsigma(f) = L_1 \alpha L_2 i L_3$, then i is not needed for α. Here, index i is needed for a rule $\alpha : l \to r$ if $l|_i \notin \mathcal{X}$ or $l|_i \in \mathcal{X}$ occurs in $l|_j$ for $i \neq j$. For example, the strategy of Example 17 is full and in-time.

6.1 Van de Pol's strategies as context-sensitive rewriting strategies

Given a strategy annotation, van de Pol describes the rewriting strategy that it specifies. A rewriting strategy is seen as a function that, given a term t, yields either some rewrite of t, i.e., a pair (p, s) such that $t \stackrel{p}{\to} s$, or $\perp$ if no rewrite step has been selected. Given a term t and a strategy annotation ς, $rewr_\varsigma$ indicates the (unique, if any) rewrite step that can be issued on t.

Definition 5 [31] *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, ς be a strategy annotation, and $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. Then, $rewr_\varsigma(t) = rewr_\varsigma(t, \varsigma(root(t)))$, where $rewr_\varsigma(t, nil) = \perp$; $rewr_\varsigma(t, (l \to r \ : L)) = (\Lambda, \sigma(r))$, if $t = \sigma(l)$ for some σ and $rewr_\varsigma(t, (l \to r \ : L)) = rewr_\varsigma(t, L)$ otherwise; and $rewr_\varsigma(t, (i : L)) = (i.p, t[s]_i)$, if $rewr_\varsigma(t|_i) = (p, s)$ for some p, s, and $rewr_\varsigma(t, L)$ otherwise.*

We write $t \stackrel{p}{\to}_\varsigma s$ (or just $t \to_\varsigma s$) if $(p, s) = rewr_\varsigma(t) \neq \perp$. Thus, t is a $\to_\varsigma$-normal form (or just a ς-normal form) if and only if $rewr_\varsigma(t) = \perp$.

Given a strategy annotation ς for $\mathcal{F}$, we define $\mu^\varsigma \in M_\mathcal{F}$ as follows: for all $f \in \mathcal{F}$, $\mu^\varsigma(f) = \{i \in \mathbb{N} \mid i \in \varsigma(f)\}$. We have:

Theorem 23 [23] *Let $\mathcal{R}$ be a TRS, ς be a strategy annotation, and $t, s \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. If $t \to_\varsigma s$, then $t \hookrightarrow_{\mu^\varsigma} s$.*

According to Theorem 23, μ-normal forms are always ς-normal forms. We also have the following.

Theorem 24 *Let $\mathcal{R}$ be a TRS, ς be an r-full strategy annotation, and $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. If t is a ς-normal form, then t is a μ-normal form.*

Thus, for r-full strategy annotations, ς-normal forms and μ-normal forms coincide. In this case, $\to_\varsigma$ can be thought of as the reduction relation $\hookrightarrow_{\mathbb{H}_\varsigma}$ associated to a (one-step) μ^ς-strategy $\mathbb{H}_\varsigma$ given by

$$\mathbb{H}_\varsigma(t) = \{t \to_\varsigma s\}.$$

For r-full strategy annotations ς, we can use semantics $\mathsf{cs\text{-}nf}_{\mathbb{H}_\varsigma}$ associated to the μ^ς-strategy $\mathbb{H}_\varsigma$.

According to the definition of $\to_\varsigma$, given terms t, s, and s', it follows that $t \to_\varsigma s$ and $t \to_\varsigma s'$ imply that $s = s'$, i.e., each $\to_\varsigma$-reduction step (hence $\mathbb{H}_\varsigma$ and $\mathsf{cs\text{-}nf}_{\mathbb{H}_\varsigma}$) is *deterministic*.

In fact, although Van de Pol's strategy annotations forms the basis of the aforementioned JITty system, the current implementation *only* permits full and in-time strategy annotations. Thus, the system is not able to exploit all possibilities of this kind of annotations. In any case, van de Pol did not investigate any semantic property of non-full strategies. Now, Theorems 4, 6, 7, 9, 11, and 13 can be used to describe computations with van de Pol's strategy annotations by taking $\mathbb{H}_\varsigma$ as the considered μ^ς-strategy and $\mathsf{cs\text{-}nf}_{\mathbb{H}_\varsigma}$ or $\mathsf{nf}_{\mathbb{S}_{\mathbb{H}_\varsigma}}$ as semantics.

6.1.1 Related work

The main concern of [31] is computing normal forms. Van de Pol provides a normalization (partial) function:

Definition 6 [31] *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, ς be a strategy annotation, and $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. Then, $norm_\varsigma(t) = norm_\varsigma(t, \varsigma(root(t)))$, where $norm_\varsigma(t, nil) = t$; $norm_\varsigma(t, (l \to r : L)) = norm_\varsigma(\sigma(r))$ if $t = \sigma(l)$ for some σ, or $norm_\varsigma(t, (l \to r : L)) = norm_\varsigma(t, L)$ otherwise; and $norm_\varsigma(t, (i : L)) = norm_\varsigma(t[norm_\varsigma(t|_i)]_i, L)$*

Then, he proves the following theorem.

Theorem 25 [31] *If ς is in-time, then $norm_\varsigma(t)$ is the last element of the maximal $\to_\varsigma$-reduction sequence starting from t.*

As a consequence of Theorem 25, we conclude the following.

Theorem 26 *Let $\mathcal{R}$ be a TRS and ς be r-full and in-time. If $s \in norm_\varsigma(t)$, then s is a μ-normal form of t. Moreover, $norm_\varsigma = \mathsf{cs\text{-}nf}_{\mathbb{H}_\varsigma}$.*

Van de Pol proves that semantics $norm_\varsigma$ for full and in-time strategies ς is correct w.r.t. the normalization semantics nf.

Theorem 27 (Correctness [31]**)** *Let ς be full and in-time. If $s \in norm_\varsigma(t)$, then s is a normal form of t.*

Unfortunately, no analysis of completeness is given for $norm_\varsigma$. As mentioned above, our results can be used to ensure correctness and completeness of computations with van de Pol's strategy annotations by taking $\mathbb{H}_\varsigma$ as the considered μ^ς-strategy and $\mathsf{cs\text{-}nf}_{\mathbb{H}_\varsigma}$ (i.e., $norm_\varsigma$ for r-full and just-in-time strategy annotations) or $\mathsf{nf}_{\mathbb{S}_{\mathbb{H}_\varsigma}}$ as semantics.

7 Conclusions

Strategy annotations have been used in the OBJ family of languages for many years. However, only recently (but quite intensively) has been addressed the formal analysis of computations under such kind of strategies (see, for instance, [7, 9, 22, 26, 27, 30, 31]). Also, other eager programmming languages such as ELAN [4] 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 (see [19]).

In this paper, we have given a semantic description of computations of rewriting under strategy annotations. Our description uses context-sensitive rewriting (strategies) and rewriting semantics as useful frameworks for modeling computations under strategy annotations and compare them w.r.t. the usual semantics: computation of (infinite) constructor terms, head-normal forms, and (infinite) normal forms. We have used some existing results in the literature about *CSR* to provide a complete description of the computational power of strategy annotations. We have also proved new results regarding the use of *CSR* with completely defined TRSs, the ability of *CSR* to compute infinite values, the possibility of guaranteeing unicity of infinite normal forms with non-orthogonal TRSs, etc. We have considered the E-strategies of Maude, OBJ*, and CafeOBJ. We proved that the semantic framework which has been developed for *CSR* can be used to predict the behavior of the implementations of such languages. Many new results have been given in this sense. We have also considered Van de Pol's strategy annotations. Van de Pol payed no much attention to the use of strategy annotations with incomplete specification of indices and rules. In this paper, we give a solid framework to deal with these kind of strategy annotations which can be used to improve the systems where they are used [3, 32]. We finally note that only standard restrictions (mainly left-linearity and confluence/orthogonality) are imposed on the TRSs in our results (even for the infinitary rewriting stuff).

References

[1] M. Alpuente, S. Escobar, B. Gramlich, and S. Lucas. Improving On-Demand Strategy Annotations. In Matthias Baaz and Andrei Voronkov, editors, *Proc. 9th Int. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning, LPAR'02*, LNAI 2514:1-18, Springer-Verlag, Berlin, 2002.

[2] M. Alpuente, S. Escobar, and S. Lucas. Correct and complete (positive) strategy annotations for OBJ. In F. Gadducci and U. Montanari, editors, *Proc. of the 4th International Workshop on Rewriting Logic and its Applications, WRLA'02, Electronic Notes in Theoretical Computer Science*, volume 71. Elsevier Sciences, to appear 2003.

[3] S. Blom, W. Fokkink, J. Groote, I. van Langevelde, B. Lisser, and J. van de Pol. μCRL: a toolset for analysing algebraic specifications. In G. Berry, H. Comon, A. Finkel, editors, *13th International Conference on Computer Aided Verification, CAV'01*, LNCS 2102:250-254, Springer-Verlag, Berlin, 2001.

[4] 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.

[5] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and J.F. Quesada. Maude: specification and programming in rewriting logic. *Theoretical Computer Science* 285(2):187-243, 2002.

[6] N. Dershowitz, S. Kaplan, and D. Plaisted. Rewrite, rewrite, rewrite, rewrite, rewrite. *Theoretical Computer Science* 83:71-96, 1991.

[7] 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.

[8] 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.

[9] O. Fissore, I. Gnaedig, and H. Kirchner. Induction for termination with local strategies. *Electronic Notes in Theoretical Computer Science*, volume 58(2), 2001.

[10] 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.

[11] J. Giesl and A. Middeldorp. Termination of Innermost Context-Sensitive Rewriting. In *Proc. of DLT'02*, LNCS to appear, Springer-Verlag, Berlin, 2003.

[12] J. Giesl and A. Middeldorp. Transformation Techniques for Context-Sensitive Rewrite Systems. *Journal of Functional Programming*, to appear, 2003.

[13] 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.

[14] 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.

[15] D. Kapur, P. Narendran, and Z. Zhang. On sufficient-completeness and related properties of term rewriting systems. *Acta Informatica* 24:395-416, 1987.

[16] 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.

[17] E. Kounalis. Completeness in data type specifications. In B.F. Caviness, editor, *Proc. of EUROCAL'85*, LNCS 204:348-362. Springer-Verlag, Berlin, 1985.

[18] S. Lucas. Context-sensitive computations in functional and functional logic programs. *Journal of Functional and Logic Programming*, 1998(1):1-61, January 1998.

[19] S. Lucas. Context-sensitive rewriting strategies. *Information and Computation*, 178(1):293-343, 2002.

[20] S. Lucas. Root-neededness and approximations of neededness. *Information Processing Letters*, 67(5):245-254, 1998.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] T. Nagaya. Reduction Strategies for Term Rewriting Systems. PhD Thesis, School of Information Science, Japan Advanced Institute of Science and Technology, March 1999.

[27] 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'00, Electronic Notes in Theoretical Computer Science*, volume 36, 17 pages, 2001.

[28] A.T. Nakagawa, T. Sawada, and K. Futatsugi. CafeOBJ User's Manual –ver 1.4–, 1999. Available at `http://www.ldl.jaist.ac.jp/Research/CafeOBJ/`.

[29] V. van Oostrom and R. de Vrijer. Strategies. In TeReSe, *Term Rewriting Systems*, Chapter 9. Cambridge University Press, 2003.

[30] K. Ogata and K. Futatsugi. Implementation of Term Rewritings with the Evaluation Strategy. In H. Glaser and P. Hartel, editors, *Proc of 9th International Symposium on Programming Languages, Implementations, Logics and Programs, PLILP'97*, LNCS 1292:225-239, Springer-Verlag, Berlin, 1997.

[31] J. van de Pol. Just-in-time: on Strategy Annotations. *Electronic Notes in Theoretical Computer Science*, volume 57, 2001.

[32] J. van de Pol. JITty: A Rewriter with Strategy Annotations. In Sophie Tison, editor, *Proc. 13th International Conference on Rewriting Techniques and Applications, RTA'02*, LNCS 2378:367-370, Springer-Verlag, Berlin, 2002.

Proofs of Section 3

Proposition 1 *If* S *is partially correct and complete w.r.t.* S_0*, then* $\mathsf{S}_0 = \lambda t.\mathsf{S}(t) \cap W_{\mathsf{S}_0}$.

Proof. If S is partially correct w.r.t. S_0, then $\mathsf{S}(t) \cap W_{\mathsf{S}_0} \subseteq \mathsf{S}_0(t)$ for all terms t. If S is complete w.r.t. S_0, then $\mathsf{S}_0(t) \subseteq \mathsf{S}(t)$, i.e., $\mathsf{S}_0(t) \cap W_{\mathsf{S}_0} \subseteq \mathsf{S}(t) \cap W_{\mathsf{S}_0}$. Since $\mathsf{S}_0(t) \cap W_{\mathsf{S}_0} = \mathsf{S}_0(t)$, the conclusion follows. $\square$

Proposition 3 *Let* $\mathcal{R}$ *be a TRS,* $\mu \in M_{\mathcal{R}}$*, and* $\mathbb{H}$ *be a* μ*-strategy for* $\mathcal{R}$*. Then,* $\mathbb{H}$ *is* μ*-terminating if and only if* $\mathcal{R}$ *is* μ*-normalizing and* $\mathbb{H}$ *is* μ*-normalizing.*

Proof. If $\mathcal{R}$ is not μ-normalizing, then there is a term t having no μ-normal form. Thus, any $\mathbb{H}$-sequence starting from t must perform an infinite number of rewritings. If the μ-strategy $\mathbb{H}$ is μ-terminating, then $\mathbb{H}$ is obviously μ-normalizing. On the other hand, if $\mathbb{H}$ is μ-normalizing, since every term t has a μ-normal form (due to $\mathcal{R}$ μ-normalizing), then there is no term issuing an infinite $\mathbb{H}$-sequence, i.e., $\mathbb{H}$ is μ-terminating. $\square$

Proofs of Section 4

Theorem 6 *Let* $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ *be a left-linear, confluent TRS,* $\mathcal{B} \subseteq \mathcal{C}$*,* $\mu \in EvM_{\mathcal{R},\mathcal{B}}$ *and* $\mathbb{H}$ *be a* μ*-normalizing* μ*-strategy. Let* $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ *and* $\delta \in \mathcal{T}(\mathcal{B}, \mathcal{X})$*. Then,* $\delta \in \mathsf{eval}(t)$ *if and only if* $\delta \in \mathsf{cs\text{-}nf}_{\mathbb{H}}(t)$*.*

Proof. The 'if' direction is obvious. For the 'only if' direction, if $\delta \in \mathsf{eval}(t)$, then, by confluence, δ is the only normal form of t. By Theorem 5, we have that $t \hookrightarrow^* \delta$. Since $\mathbb{H}$ is μ-normalizing, it must be $\delta \in \mathsf{cs\text{-}nf}_{\mathbb{H}}(t)$. $\square$

In the proof of the following proposition, we use the notion of *strong* head-normal form. A term is a *strong head-normal form* if $\omega(t) \neq \Omega$, where (following [14]), Ω is a new constant symbol which is introduced to represent arbitrary terms. Then, $\omega(t)$ is the (unique) normal form of t w.r.t. Huet and Levy's Ω-reduction (see [14, 19] for precise definitions).

Proposition 5 *Let* $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ *be a left-linear, completely defined TRS and* $\mu \in EvM_{\mathcal{R},\mathcal{C}}$*. There is no ground* μ*-normal form containing defined symbols.*

Proof. By induction of the structure of ground μ-normal forms t. If t is a constant symbol c, then c is a constructor symbol. Let $t = f(t_1, \ldots, t_k)$ be a μ-normal form.

1. If f is a k-ary constructor symbol, then, since $\mu \in EvM_{\mathcal{R},\mathcal{C}}$, we have $\mu(f) = \{1, \ldots, k\}$. Since replacing subterms of a μ-normal form are μ-normal forms, $t_1, \ldots, t_k$ are μ-normal forms. By the induction hypothesis, they do not contain defined symbols. Hence, t contains no defined symbol.

2. If f is a k-ary defined symbol and $\mu(f) = \varnothing$, then, since $\mu \in CM_{\mathcal{R}}$, every rule defining f has shape $f(x_1, \ldots, x_k) \to r$ for variables $x_1, \ldots, x_k$ and term r. Thus, t is not a normal form; hence, t is not a μ-normal form either. Hence, we assume $\mu(f) \neq \varnothing$ and take $i \in \mu(f)$. By the induction hypothesis, t_i is a ground term that contains no defined symbol. This implies that there is a constant constructor symbol $c \in \mathcal{C}$. By Corollary 1 in [19], $C[\overline{\Omega}]$ is a strong head-normal form for $C[\,] = MRC^{\mu}(t)$. Let s be the smallest subterm of $C[\overline{\Omega}]$ satisfying $root(s) \in \mathcal{D}$. Since every subterm of a strong head-normal form is also a strong head-normal form, s is a strong head-normal form. Let s' be s with all occurrences of Ω replaced by c. Note that s' is a strong head-normal form. Moreover, since only the root symbol of s' is a defined symbol, s' is a ground normal form containing a defined symbol. This contradicts the CD property for $\mathcal{R}$.

□

Theorem 9 *Let $\mathcal{R} = (\mathcal{C} \uplus \mathcal{D}, R)$ be a left-linear TRS, $\mu \in CM_{\mathcal{R}}$, and $\mathbb{H}$ be a μ-strategy for $\mathcal{R}$. Then,*

1. *Semantics $\mathsf{nf}_{\mathbb{S}_{\mathbb{H}}}$ is correct w.r.t. hnf and nf.*
2. *If $\mathcal{R}$ is completely defined, then*
   (a) *$\mathsf{nf}_{\mathbb{S}_{\mathbb{H}}}$ is correct (over ground terms) w.r.t. eval.*
   (b) *If $\mu \in EvM_{\mathcal{R},\mathcal{C}}$, then $\mathsf{nf}_{\mathbb{S}_{\mathbb{H}}}$ is equivalent (over ground terms) to $\mathsf{cs\text{-}nf}_{\mathbb{H}}$.*
3. *If $\mathcal{R}$ is confluent and $\mathbb{H}$ is μ-normalizing, then $\mathsf{nf}_{\mathbb{S}_{\mathbb{H}}} = \mathsf{nf}$.*

PROOF.

1. Since $\mathbb{S}_{\mathbb{H}}$ is a rewriting strategy, only normal forms can be considered irreducible for $\mathbb{S}_{\mathbb{H}}$. Thus, $\mathsf{nf}_{\mathbb{S}_{\mathbb{H}}}(t) \subseteq \mathsf{nf}(t)$ for all terms t, i.e., $\mathsf{nf}_{\mathbb{S}_{\mathbb{H}}}$ is correct w.r.t. nf. Since nf is correct w.r.t. hnf, it follows that $\mathsf{nf}_{\mathbb{S}_{\mathbb{H}}}$ also is.
2. Let t be an arbitrary ground term.
   (a) For completely defined TRSs, $\mathsf{nf}(t) = \mathsf{eval}(t)$. Since $\mathsf{nf}_{\mathbb{S}_{\mathbb{H}}}$ is correct w.r.t. nf, $\mathsf{nf}_{\mathbb{S}_{\mathbb{H}}}(t) \subseteq \mathsf{nf}(t) = \mathsf{eval}(t)$.
   (b) If $\mu \in EvM_{\mathcal{R},\mathcal{C}}$, then, by Proposition 5, no ground μ-normal form s of t contains defined symbols. Therefore, by construction of $\mathsf{nf}_{\mathbb{S}_{\mathbb{H}}}$ we cannot further rewrite on s when no further $\hookrightarrow_{\mathbb{H}}$-steps are possible on s. Thus, $\mathsf{nf}_{\mathbb{S}_{\mathbb{H}}}(t) = \mathsf{cs\text{-}nf}_{\mathbb{H}}(t)$.
3. By Theorem 8, $\mathbb{S}_{\mathbb{H}}$ is normalizing, i.e., there are no infinite $\mathbb{S}_{\mathbb{H}}$-sequences for terms t having a normal form. Thus, if $\mathsf{nf}(t)$ is not empty, then $\mathsf{nf}_{\mathbb{S}_{\mathbb{H}}}(t)$ contains a normal form which, by confluence, is the only one belonging to $\mathsf{nf}(t)$, i.e., $\mathsf{nf}_{\mathbb{S}_{\mathbb{H}}}(t) = \mathsf{nf}(t)$. On the other hand, if $\mathsf{nf}(t)$ is empty, by correctness of $\mathsf{nf}_{\mathbb{S}_{\mathbb{H}}}$ w.r.t. Snf, $\mathsf{nf}_{\mathbb{S}_{\mathbb{H}}}(t) = \varnothing$. Thus, for all term t, $\mathsf{nf}_{\mathbb{S}_{\mathbb{H}}}(t) = \mathsf{nf}(t)$.

$\square$

Theorem 28 [18] *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 $t \to^* x$, then $t \hookrightarrow^* x$. If $t \to^* s$, then there exists $s' = c(\overline{s'})$ such that $t \hookrightarrow^* s'$ and $s' \xrightarrow{>\Lambda}{}^* s$.*

A simple consequence of Theorem 28 is the following.

Proposition 7 *Let $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ be a left-linear TRS, $\mathcal{B} \subseteq \mathcal{C}$, and $\mu \in EvM_{\mathcal{R},\mathcal{B}}$. If $t \to^* s$ and $s = C[s_1, \ldots, s_n]$ for some $C[\,, \ldots, \,] \in \mathcal{T}(\mathcal{B} \cup \{\square\}, \mathcal{X})$ and $s_1, \ldots, s_n \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, then $t \hookrightarrow^*_\mu s'$, where $s' = C[s'_1, \ldots, s'_n]$ and $s'_i \to^* s_i$ for $1 \leq i \leq n$.*

Proposition 8 [24] *Let $\mathcal{R}$ be a left-linear TRS, $t \in \mathcal{T}^\omega(\mathcal{F}, \mathcal{X})$, and s be a finite term. If $t \to^{\leq\omega} s$, then $t \to^* s$.*

Theorem 29 [24] *Let $\mathcal{R}$ be a left-linear TRS, $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and $s \in \mathcal{T}^\omega(\mathcal{F}, \mathcal{X})$. If $t \to^\omega s$, then for all $\kappa \in]0, 1]$, there exists $s' \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ such that $t \to^* s'$ and $d(s', s) < \kappa$.*

Theorem 10 *Let $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ be a left-linear TRS, $\mathcal{B} \subseteq \mathcal{C}$ and $\mu \in EvM_{\mathcal{R},\mathcal{B}}$. Let $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and $\delta \in \mathcal{T}^\omega(\mathcal{B}, \mathcal{X})$. Then, $t \to^{\leq\omega} \delta$ iff $t \hookrightarrow^{\leq\omega}_\mu \delta$.*
Proof. The 'if' part is obvious. For the 'only if' part, we consider two cases: If δ is a finite term, by Proposition 8, $t \to^* \delta$. Thus, by Theorem 5, $t \hookrightarrow^* \delta$. If δ is an infinite term, then (by Theorem 1 in [24]) there is an infinite derivation

$$A : t = t_1 \to t_2 \to t_3 \to \cdots$$

that strongly converges to δ. By using A, we construct an infinitary μ-rewriting sequence A' starting from t and converging into δ.

Let d_i be the depth of the redex which is contracted in the rewriting step $t_i \to t_{i+1}$. Since A is a strongly converging derivation, for each $P \in \mathbb{N}$, there is a least index i_P such that $\mathsf{d}_j \geq P$ for all $j \geq i_P$. Obviously, for $P = 0$, we have $i_P = 1$. Note that there can be (many) $P \in \mathbb{N}$ such that $i_P = i_{P+1}$. Given $P \in \mathbb{N}$, we let $succ_A(P)$ be the least $Q \in \mathbb{N}$ such that $Q > P$ and $i_P \neq i_Q$. Obviously, $P < succ_A(P)$ and $i_P < i_{succ_A(P)}$ for all $P \in \mathbb{N}$. Let $\mathsf{P}_A = \{P_0, P_1, \ldots\}$, where $P_0 = 0$ and $P_k = succ^k_A(P_0)$ for $k > 0$. Note that, since δ is an infinite term and A is strongly converging, P_A is an infinite set. We have the following facts:

1. for all $k \in \mathbb{N} - \{0\}$, $sp_\square(MCP(t_{i_{P_k}})) = P_k$ where, for any term t, $MCP(t)$ is the maximal constructor prefix of t and $sp_\square(C[\,])$ is the length of the shortest path from the root to a hole in $C[\,]$. The proof of this claim is easy: Obviously, by definition of MCP, $sp_\square(MCP(t_{i_{P_k}})) \leq P_k$, Assume that $sp_\square(MCP(t_{i_{P_k}})) < P_k$. Then, by definition of $t_{i_{P_k}}$, the depth of all rewriting steps issued in sequence A after $t_{i_{P_k}}$ take place at depth greather than or equal to P_k. However, there is a subterm of $t_{i_{P_k}}$ at depth

$sp_{\Box}(MCP(t_{i_{P_k}})) < P_k$ which is rooted by a defined symbol (note that this is not necessarily true for $k = 0$; hence the exclusion of this case). In order to obtain δ, this subterm must be reduced in sequence A at its topmost position. This leads to a contradiction.

2. Since A strongly converges to an infinite constructor term, for all $k \in \mathbb{N}$, $MCP(t_{i_{P_k}})$ is a strict prefix of $MCP(t_{i_{P_{k+1}}})$, Moreover, according to 1 above, $2^{-P_k} \geq d(t_{i_{P_k}}, \delta)$.

Let $s = t_{i_{P_1}} = C[s_1, \ldots, s_n]$ where $C[] = MCP(s)$. By 1 above, $sp_{\Box}(C[]) = P_1 > 0$, i.e., $C[]$ is not empty. By Proposition 7, there is a term $u = C[u_1, \ldots, u_n]$ such that $t \hookrightarrow^+ u$ ($t = u$ is not possible here) and $u_m \to^* s_m$ for $1 \leq m \leq n$. Let B^1 be the (finite) μ-rewrite sequence starting from t and leading to u. Consider now $t_{P_2} = C[s'_1, \ldots, s'_n]$. Note that, by 1 above, $MCP(s'_m) \neq \Box$ for some $1 \leq m \leq n$ (whereas, necessarily, $MCP(s_m) = \Box$ for all $1 \leq m \leq n$). Since $s_m \to^* s'_m$ for $1 \leq m \leq n$, we have $u_m \to^* s'_m$ for $1 \leq m \leq n$. Again, we have finite (possibly empty) μ-rewrite sequences $B^1_1, \ldots, B^1_n$ leading from u_m to u'_m such that $u'_m \to^* s'_m$ and $MCP(u'_m) = MCP(s'_m)$ for $1 \leq m \leq n$. Moreover, at least one of the B^1_m must be non-empty for some $1 \leq m \leq n$. Now, we can build a single rewrite sequence B^2 from t to $u' = C[u'_1, \ldots, u'_m]$ as follows: $B^2 = B^1; B^1_1; \cdots; B^1_n$ (the ordering among sequences B_m is not important). Since $\delta \in \mathcal{T}^\omega(\mathcal{B}, \mathcal{X})$, $C[\,]$ is a prefix of δ, and $\mu(c) = \{1, \ldots, ar(c)\}$ for all $c \in \mathcal{B}$, the resulting rewriting sequence is, in fact, a μ-rewriting sequence. We proceed in this way for each P_k with $k > 2$ to obtain a set of finite μ-rewriting sequences $B^1, B^2, B^3, \ldots$ leading from t to terms $u^1, u^2, u^3, \ldots$ respectively (where $u^1 = u, u^2 = u', \ldots$). By construction, B^1 is a strict prefix of B^2 which is a strict prefix of B^3, etc. Moreover, since 2^{-P_m} (which tends to cero when m tends to ω) is an upper bound of $d(u^m, \delta)$, it follows that $d(u^m, \delta)$ also tends to cero. Now, we can define A' to be the union of the previous sequences: $A' = \bigcup_{k \leq 1} B^k$. Therefore, A' is an infinite sequence of μ-rewriting steps whose limit is δ. $\Box$

Theorem 11 *Let $\mathcal{R} = (\mathcal{C} \uplus \mathcal{D}, R)$ be a left-linear TRS, $\mu \in EvM_{\mathcal{R},\mathcal{C}}$, and $\mathbb{H}$ be a μ-strategy for $\mathcal{R}$. Then,*

1. *Semantics ω-cs-nf_μ is complete w.r.t. ω-eval.*

2. *If $\mathcal{R}$ is orthogonal and $\mathbb{H}$ is infinitary μ-normalizing, then ω-cs-$\mathsf{nf}_{\mathbb{H}}$ is complete w.r.t. ω-eval.*

Proof.

1. Use Theorem 10.

2. If $\mathcal{R}$ is orthogonal and ω-$\mathsf{eval}(t)$ contains a constructor term δ, then δ is only finite or infinite normal form of t ([16], Theorem 7.15). By Theorem 10 there is a μ-rewriting sequence leading to δ. Since $\mathbb{H}$ is infinitary μ-normalizing, every $\mathbb{H}$-sequence will end in a normal form which must be δ.

□

Theorem 12 *Let $\mathcal{R}$ be a left-linear TRS and $\mu \in CM_{\mathcal{R}}$. If $\mathbb{H}$ is a μ-terminating μ-strategy, then $\mathbb{S}^{\parallel}_{\mathbb{H}}$ is infinitary normalizing.*

PROOF. By Proposition 3, $\mathbb{H}$ is μ-terminating only if $\mathcal{R}$ is μ-normalizing and $\mathbb{H}$ is μ-normalizing. Thus, the conclusion follows by Corollary 13 in [19]. □

Theorem 13 *Let $\mathcal{R}$ be a left-linear TRS, $\mu \in CM_{\mathcal{R}}$, and $\mathbb{H}$ be a μ-terminating μ-strategy for $\mathcal{R}$. Then,*

1. *ω-$\mathsf{nf}_{\mathbb{S}^{\parallel}_{\mathbb{H}}}$ is correct w.r.t. ω-nf.*
2. *If $\mathcal{R}$ is completely defined and $\mu \in EvM_{\mathcal{R},\mathcal{C}}$, then ω-$\mathsf{nf}_{\mathbb{S}^{\parallel}_{\mathbb{H}}}$ is equivalent (over ground terms) to cs-$\mathsf{nf}_{\mathbb{H}}$.*
3. *If $\mathcal{R}$ is orthogonal, then ω-$\mathsf{nf}_{\mathbb{S}^{\parallel}_{\mathbb{H}}}$ is complete w.r.t. ω-eval, and ω-$\mathsf{nf}_{\mathbb{S}^{\parallel}_{\mathbb{H}}} = \omega$-$\mathsf{nf}$*

PROOF.

1. Theorem 12.
2. Since $\mathcal{R}$ is completely defined, every ground μ-normal form is a constructor term. Thus, by definition of ω-$\mathsf{nf}_{\mathbb{S}^{\parallel}_{\mathbb{H}}}$, the ω-$\mathsf{nf}_{\mathbb{S}^{\parallel}_{\mathbb{H}}}$-sequences issued from any ground term t coincide with the corresponding $\mathbb{H}$-sequences. Therefore, ω-$\mathsf{nf}_{\mathbb{S}^{\parallel}_{\mathbb{H}}}(t) = \mathsf{cs}$-$\mathsf{nf}_{\mathbb{H}}(t)$ for all ground term t.
3. Let $\delta \in \omega$-$\mathsf{eval}(t)$; there exists an infinitary rewriting sequence starting from t which converges to δ. By [16], Theorem 7.15, such δ is unique and there is no other finite or infinite normal form of t. By Theorem 12, there exists no perpetual ω-$\mathsf{nf}_{\mathbb{S}^{\parallel}_{\mathbb{H}}}$-sequence starting from t, i.e., $\delta \in \omega$-$\mathsf{nf}_{\mathbb{S}^{\parallel}_{\mathbb{H}}}(t)$.

□

Proposition 9 [18] *Let $\mathcal{R}$ be a left-linear TRS, $l \in L(\mathcal{R})$ and $\mu \in CM_{\mathcal{R}}$. If $t \to^* \sigma(l)$ for some substitution σ, then there is a substitution θ such that $t \hookrightarrow^* \theta(l)$ and $\theta(x) \to^* \sigma(x)$ for all $x \in Var(l)$.*

Proposition 10 *Let $\mathcal{R}$ be a left-linear TRS and $\mu \in CM_{\mathcal{R}}$. If $t(\to - \hookrightarrow)^* s$ and $t|_p = \sigma(l)$ for some substitution σ, $p \in \mathcal{P}os^{\mu}(t)$, and $l \in L(\mathcal{R})$, then $s|_p = \sigma'(l)$ for some substitution σ'.*

PROOF. Since t rewrites into s without giving any μ-rewriting step, we can write $t = C[t_1, \ldots, t_n]$ and $s = C[s_1, \ldots, s_n]$ for $C[\,] = MRC^{\mu}(t)$. Obviously, $p \in \mathcal{P}os_{\mathcal{F}\cup\mathcal{X}}(C[\,])$. If $s|_p$ is not a redex, then some reduction on a position $p.q \in \mathcal{P}os(t')$ for some $q \in \mathcal{P}os_{\mathcal{F}}(l)$ is performed during the rewrite sequence from t to s (let's say, on a term t' such that $t(\to - \hookrightarrow)^* t'(\to - \hookrightarrow)^* s$. However, since $\mu \in CM_{\mathcal{R}}$, $q \in \mathcal{P}os^{\mu}(l)$ and hence $p.q \in \mathcal{P}os^{\mu}(t')$, a contradiction. □

Proposition 11 *Let $\mathcal{R}$ be a left-linear TRS and $\mu \in CM_{\mathcal{R}}$. Let $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and s is a normal form. If $A : t = t_1 \to t_2 \to^{\omega} s$ and, for all $i \in \mathbb{N}$, t_i is not a head-normal form, then A contains an infinite number of μ-rewriting steps.*

PROOF. By contradiction. Let $i \in \mathbb{N}$ be such that for all $j \geq i$, the position $p_j \in \mathcal{P}os_{\mathcal{R}}(t_j)$ of the contracted redex is not replacing. Since t_i is not a head-normal form, by Theorem 1, t_i is not a μ-normal form and contains a replacing redex at a position $p \in \mathcal{P}os^{\mu}_{\mathcal{R}}(t_i)$. By Proposition 10, this redex is not destroyed by the derivation from t_i to s. Thus, it is not difficult to see that s cannot be a normal form. $\square$

Proposition 12 *Let $\mathcal{R}$ be a left-linear TRS and $\mu \in CM_{\mathcal{R}}$. Let $t, u, s \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. If $t(\to - \hookrightarrow)^{*}u$ and $u \hookrightarrow s$, then there exists $u' \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ such that $t \hookrightarrow u'$ and $u' \to^{*} s$.*

PROOF. By induction on the number of steps in derivation $t(\to - \hookrightarrow)^{*}u$. If $n = 0$, it is obvious. If $n > 0$, then let $t(\to - \hookrightarrow)t'(\to - \hookrightarrow)^{*}u$. By the induction hypothesis, there is a term w such that $t' \hookrightarrow w$ and $w \to^{*} s$. Therefore, we can write $t' = C[\sigma(l)]_p$ for some $p \in \mathcal{P}os^{\mu}(t')$ and rule $l \to r$ of $\mathcal{R}$. Also, $w = C[\sigma(r)]_p$. By reasoning as in the proof of Proposition 10, we conclude that there is a substitution σ' such that $t|_p = \sigma'(l)$. Let $q \in \mathcal{P}os_{\mathcal{R}}(t)$ be the position of (non-replacing) redex contracted in the rewriting step $t(\to - \hookrightarrow)t'$. Obviously, $t|_q = \theta(l')$ and $t' = t[\theta(r')]_q$ for some substitution θ and rule $l' \to r'$. Let $u' = t[\sigma'(r)]_p$. We consider two cases: $p \parallel q$ and $p < q$ (the case $p \geq q$ is not possible, since positions above p must be replacing). If $p \parallel q$, then $u' \to w$ by contracting redex $u'|_q$ using rule $l' \to r'$. If $p < q$, then there is a position $q' \in \mathcal{P}os_{\mathcal{X}}(l)$ such that $p.q' \leq q$. Assume that $x = l|'_q$. If $x \notin \mathcal{V}ar(r)$, then $u' = w$; otherwise, $u' \to^{+} w$ by contracting the descendant of redex $t|_q$ in u' as many times as variable occurrences of x are present in r. Thus, in all cases, $u' \to^{*} w$, i.e., $u' \to^{*} s$. $\square$

Theorem 30 *Let $\mathcal{R}$ be a left-linear TRS and $\mu \in CM_{\mathcal{R}}$ be such that $\mathcal{R}$ is μ-terminating. Let $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. Every finitary or infinitary rewrite sequence leading from t to a normal form contains a finite head-normal form.*

PROOF. For finitary sequences, it is obvious. For infinitary sequences, we proceed by contradiction. Consider an infinitary sequence A leading from t to s that does not contain any finite head-normal. By Proposition 11, A contains an infinite number of μ-rewriting steps. Thus, we can write A as follows:

$$A : t = t_1(\to - \hookrightarrow)^{*}t_2 \hookrightarrow^{+} t_3(\to - \hookrightarrow)^{*}t_4 \hookrightarrow^{+} t_5 \to \cdots \to^{\omega} s$$

where purely μ-rewriting and purely non-μ-rewriting steps are alternated. By a repeated application of Proposition 12, we are able to build an infinite μ-rewrite sequence starting from t which contradicts μ-termination of $\mathcal{R}$. $\square$

Proposition 13 *Let $\mathcal{R}$ be a left-linear TRS and $\mu \in CM_{\mathcal{R}}$ be such that $\mathcal{R}$ is μ-terminating. Let $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. If $t = t_1 \to t_2 \to^{\omega} s$ and s is a normal form, then, for all $d \geq 0$, there is $i \geq 1$ such that for all $j > i$, $|p| \geq d$ for every $p \in \mathcal{P}os(t_j)$ such that $t_j|_p$ is not a head-normal form.*

PROOF. First, we note that, since we are considering an infinite strongly converging sequence, s must be an infinite term. Now, we proceed by induction on d. If $d = 0$, it is obvious. If $d > 0$, by Theorem 30, there is $i \geq 1$ such that $t \to^* t_i$ and $t_i = f(u_1, \ldots, u_k)$ is a head-normal form. Then, $s = f(s_1, \ldots, s_k)$ for infinite normal forms $s_1, \ldots, s_k$, and there are sequences $u_i \to^{\omega} s_i$ for $1 \leq i \leq k$ which are extracted from the sequence $t_i \to^{\omega} s$ in the obvious way. By induction hypothesis, for $1 \leq m \leq k$, there is i_m such that for all $j_m > i_m$, $|p_m| \geq d - 1$ for every $p \in \mathcal{P}os(t_j)$ such that $t_j|_p$ is not a head-normal form. Now, if we take $i' = \sum_{m=1}^{k} i_m$, since t_i is a head-normal form, it follows that for all $j > i'$, $|p| \geq d$ for every $p \in \mathcal{P}os(t_j)$ such that $t_j|_p$ is not a head-normal form. □

Theorem 14 *Let $\mathcal{R}$ be a left-linear, confluent TRS and $\mu \in CM_{\mathcal{R}}$ be such that $\mathcal{R}$ is μ-terminating. Let $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. If $t \to^{\leq \omega} s_1$ and $t \to^{\leq \omega} s_2$ for normal forms s_1, s_2, then $s_1 = s_2$.*

PROOF. Assume that $s_1 \neq s_2$. By confluence, at least one of the derivations must be infinite; hence, we assume $t \to^{\omega} s_1$. Let $\kappa = d(s_1, s_2) \neq 0$ and $d = \lfloor log_2 \kappa \rfloor + 1$. By Proposition 13, we can write $t \to^* t_1 \to^{\leq \omega} s_1$ in such a way that, for all position $p \in \mathcal{P}os(t_1)$ such that $|p| \leq d$, $t_1|_p$ is a head-normal form. Note that $d(t_1, s_1) < \kappa$; hence there exists a context $C[]$ such that $t_1 = C[u_1, \ldots, u_n]$ and $s_2 = C[u'_1, \ldots, u'_n]$. If s_2 is a finite term, then we have $t \to^* s_2$. By confluence, there should be $t_1 \to^* s_2$, but this is not possible since there is u_i for some $1 \leq i \leq n$ such that u_i is a head-normal form and $root(u_i) \neq root(u'_i)$. This leads to a contradiction.

Now, if we assume that s_2 is an infinite term, then by Proposition 13 we can write $t \to^* t_2 \to^{\leq \omega} s_2$, in such a way that, for all position $p \in \mathcal{P}os(t_2)$ such that $|p| \leq d$, $t_2|_p$ is a head-normal form. Again, $d(t_2, s_2) < \kappa$ and we can write $t_2 = C[u''_1, \ldots, u''_n]$ for the same previous context $C[]$. By confluence, there is a term u such that $t_1 \to^* u$ and $t_2 \to^* u$. However, there is some j, $1 \leq j \leq n$ such that $root(u_j) \neq root(u''_j)$. Since u_i and u''_i are head-normal forms for all $1 \leq i \leq n$, subterms u_j and u''_j are not joinable, and t_1 and t_2 are not either. This, again, leads to a contradiction. □

Proofs of Section 5

Theorem 15 *Let $\mathcal{R}$ be a TRS and φ be an E-strategy map. Let $t \in \mathcal{T}(\mathcal{F}_{\mathcal{L}}, \mathcal{X}_{\mathcal{L}})$, and $p \in \mathcal{P}os^{\mu^{\varphi}}(erase(t))$ be s.t. $root(t|_p) = f_L$ for some suffix L of $\varphi(f)$. If $\langle t, p \rangle \to_{\varphi} \langle s, q \rangle$, then $q \in \mathcal{P}os^{\mu^{\varphi}}(erase(s))$ and*

1. *if $\langle t, p \rangle \xrightarrow{\mathrm{T}}_{\varphi} \langle s, q \rangle$, then $erase(t) = erase(s)$.*

2. *if* $\langle t,p\rangle \stackrel{\mathrm{R}}{\to}_{\varphi} \langle s,q\rangle$, *then* $erase(t) \hookrightarrow_{\mu^{\varphi}} erase(s)$.

PROOF. Implicit in the proof of Theorem 7 in [22] $\square$

Proposition 6 *Let* $\mathcal{R}$ *be a TRS and* φ *be an E-strategy map for* $\mathcal{R}$. *Then,*

$$eval_{\varphi}(t) = \{erase(s) \in \mathcal{T}(\mathcal{F},\mathcal{X}) \mid \langle \varphi(t), \Lambda\rangle (\stackrel{\mathrm{T}}{\longrightarrow}^{!}_{\varphi} \circ \stackrel{\mathrm{R}}{\to}_{\varphi})^{!}\langle s,p\rangle\}.$$

PROOF. Obvious, due to the mutually excluding cases of the definition of $\to_{\varphi}$. $\square$

Theorem 18 *Let* $\mathcal{R} = (\mathcal{F},R) = (\mathcal{C} \uplus \mathcal{D}, R)$ *be a TRS and* φ *be a regular E-strategy map for* $\mathcal{R}$.

1. *Semantics* $eval_{\varphi}$ *is correct w.r.t.* $\textsf{cs-nf}_{\mu^{\varphi}}$.
2. *Semantics* $eval_{\varphi}$ *is partially correct w.r.t.* $\textsf{hnf}$, $\textsf{eval}$, *and* $\textsf{nf}$.
3. *If* $\mathcal{R}$ *is left-linear and* $\varphi \in CM_{\mathcal{R}}$, *then* $eval_{\varphi}$ *is correct w.r.t.* $\textsf{hnf}$.
4. *If* $\mathcal{R}$ *is left-linear and* $\varphi \in EvM_{\mathcal{R},\mathcal{C}}$, *then*
   (a) *If* $\mathcal{R}$ *is completely defined, then* $eval_{\varphi}$ *is correct w.r.t.* $\textsf{eval}$.
   (b) *If* $\mathcal{R}$ *is confluent and* φ*-terminating, then* $eval_{\varphi}$ *is complete w.r.t.* $\textsf{eval}$.
5. *Let* $\mathcal{B} \subseteq \mathcal{C}$, $\varphi \in EvM_{\mathcal{R},\mathcal{B}}$, $t \in \mathcal{T}(\mathcal{F},\mathcal{X})$ *and* $\delta \in \mathcal{T}(\mathcal{B},\mathcal{X})$. *If* $\mathcal{R}$ *is left-linear, confluent, and* φ*-terminating, then* $\delta \in \textsf{eval}(t)$ *if and only if* $\delta \in eval_{\varphi}(t)$.

PROOF.

1. By Theorem 16, for all $s \in eval_{\varphi}(t)$, s is a μ-normal form of t, i.e., $s \in \textsf{cs-nf}_{\mu^{\varphi}}(t)$.
2. By Proposition 4, $\textsf{cs-nf}_{\mu^{\varphi}}$ is partially correct w.r.t. $\textsf{hnf}$, $\textsf{eval}$, and $\textsf{nf}$. Since $eval_{\varphi}$ is correct w.r.t. $\textsf{cs-nf}_{\mu^{\varphi}}$, by Proposition 2 the conclusion follows.
3. By Theorem 16, for all $s \in eval_{\varphi}(t)$, s is a μ-normal form of t. By Theorem 1 it is a head-normal form of t, i.e., $s \in \textsf{hnf}(t)$.
4.
   (a) Let t be a ground term. By Theorem 16, for all $s \in eval_{\varphi}(t)$, s is a μ-normal form of t. By Proposition 5, $s \in \mathcal{T}(\mathcal{C})$, i.e., $s \in \textsf{eval}(t)$.
   (b) Since $\mathcal{R}$ is φ-terminating, by Theorem 17, $eval_{\varphi} = \textsf{cs-nf}_{\mathbb{H}_{\varphi}}$. Obviously, $\mathbb{H}_{\varphi}$ is μ^{φ}-normalizing. Thus, by Theorem 7, $eval_{\varphi}$ is complete w.r.t. $\textsf{eval}$.
5. Theorem 17 and Theorem 6.

$\square$

Lemma 1 *Let $\mathcal{R}$ be a TRS and φ be an elementary E-strategy for $\mathcal{R}$. If $\langle\varphi(t),\Lambda\rangle \xrightarrow{\mathrm{T}}{}^{!}_{\varphi} \circ \xrightarrow{\mathrm{R}}_{\varphi} \langle s,p\rangle$, then $\langle\varphi(erase(s)),\Lambda\rangle \xrightarrow{\mathrm{T}}{}^{*}_{\varphi} \langle s,p\rangle$.*

PROOF. We can write $t = C[\sigma(l)]_p$ and $erase(s) = C[\sigma(r)]_p$ for some rule $l \to r \in R$ and substitution σ. Moreover, we have $\langle\varphi(t),\Lambda\rangle \xrightarrow{\mathrm{T}}{}^{!}_{\varphi} \langle t',p\rangle \xrightarrow{\mathrm{R}}_{\varphi} \langle s,p\rangle$ for some term t' such that $erase(t') = erase(t)$. Since $\xrightarrow{\mathrm{T}}_{\varphi}$ is deterministic, and reduction steps leading from $\langle\varphi(t),\Lambda\rangle$ to $\langle t',p\rangle$ only depend on (the annotations on) symbols in $\varphi(C[\,])$. Here elementarity is used to ensure that no new redex created on positions $q < p$ can be considered for reduction before reaching position p again. Thus, the conclusion follows. $\square$

Lemma 2 *Let $\mathcal{R}$ be an orthogonal TRS and φ be a regular E-strategy for $\mathcal{R}$. If $\langle\varphi(t),\Lambda\rangle \xrightarrow{\mathrm{T}}{}^{!}_{\varphi} \circ \xrightarrow{\mathrm{R}}_{\varphi} \langle s,p\rangle$, then $\langle\varphi(erase(s)),\Lambda\rangle \xrightarrow{\mathrm{T}}{}^{*}_{\varphi} \langle s,p\rangle$.*

PROOF. The proof is similar to that of Lemma 1 but orthogonality ensures now that no new redex is created above a position p by issuing a rewriting step on this position. $\square$

Theorem 19 *Let $\mathcal{R}$ be a TRS and φ be a regular E-strategy for $\mathcal{R}$.*

1. *If $\mathcal{R}$ is orthogonal, then $eval_{\varphi} = \textsf{cs-nf}_{\mathbb{H}'_{\varphi}}$.*
2. *If φ is elementary, then $\mathbb{H}'_{\varphi}$ is an innermost μ^{φ}-strategy and $eval_{\varphi} = \textsf{cs-nf}_{\mathbb{H}'_{\varphi}}$.*

PROOF. Consider a term $t \in \mathcal{T}(\mathcal{F},\mathcal{X})$. If $s \in eval_{\varphi}(t)$, then, by Proposition 6, there is a reduction sequence

$$\langle t_1,p_1\rangle \xrightarrow{\mathrm{T}}{}^{!}_{\varphi} \circ \xrightarrow{\mathrm{R}}_{\varphi} \langle t_2,p_2\rangle \xrightarrow{\mathrm{T}}{}^{!}_{\varphi} \circ \xrightarrow{\mathrm{R}}_{\varphi} \cdots \xrightarrow{\mathrm{T}}{}^{!}_{\varphi} \circ \xrightarrow{\mathrm{R}}_{\varphi} \langle t_n,p_n\rangle$$

where $t_1 = \varphi(t)$, $p_1 = \Lambda$, $erase(t_n) = s$, and $\langle t_n,p_n\rangle$ is a $\xrightarrow{\mathrm{T}}{}^{!}_{\varphi} \circ \xrightarrow{\mathrm{R}}_{\varphi}$-normal form. By Theorem 16, s is a μ^{φ}-normal form. We proceed by induction on n. If $n = 1$, then $t = s$. Hence, $\mathbb{H}'_{\varphi}(t) = \varnothing$. If $n > 1$, by definition of $\mathbb{H}_{\varphi}$, we have $t \hookrightarrow_{\mathbb{H}_{\varphi}} erase(t_2)$. By Lemmata 2 or 1, we can write $\langle\varphi(erase(t_2)),\Lambda\rangle \xrightarrow{\mathrm{T}}{}^{*}_{\varphi} \langle t_2,p_2\rangle$. Thus, by I.H., $s \in \textsf{cs-nf}_{\mathbb{H}'_{\varphi}}(erase(t_2))$, i.e., $erase(t_2) \hookrightarrow^{!}_{\mathbb{H}'_{\varphi}} s$. Therefore, $s \in \textsf{cs-nf}_{\mathbb{H}'_{\varphi}}(t)$.

Now, let $s \in \textsf{cs-nf}_{\mathbb{H}'_{\varphi}}(t)$. By definition of $\mathbb{H}'_{\varphi}$, we have $t = t_1 \hookrightarrow_{\mathbb{H}'_{\varphi}} t_2 \hookrightarrow_{\mathbb{H}'_{\varphi}} \cdots \hookrightarrow_{\mathbb{H}'_{\varphi}} t_n = s$. Again, we proceed by induction on n: if $n = 1$ it is obvious: t is a μ^{φ}-normal form and, then, $\langle\varphi(t),\Lambda\rangle$ is a $\xrightarrow{\mathrm{T}}{}^{!}_{\varphi} \circ \xrightarrow{\mathrm{R}}_{\varphi}$-normal form. If $n > 1$, by the induction hypothesis, $s \in eval_{\varphi}(t_2)$. Since $\langle\varphi(t_1),\Lambda\rangle \xrightarrow{\mathrm{T}}{}^{!}_{\varphi} \circ \xrightarrow{\mathrm{R}}_{\varphi} \langle t_2,p_2\rangle$, by Lemmata 2 or 1, again, it follows that $s \in eval_{\varphi}(t)$.

If φ is elementary, the innermost character of $\mathbb{H}_{\varphi}$ is easily proved by induction on the length n of the traversing derivation $\langle\varphi(t),\Lambda\rangle \xrightarrow{\mathrm{T}}{}^{!}_{\varphi} \langle t',p\rangle$ which is part of the $\mathbb{H}'_{varphi}$-step $t \hookrightarrow_{\mathbb{H}'_\varphi} s$, where $t|_p$ is the contracted redex. If $n = 0$, then, by elementarity of φ, $\varphi(root(t)) = (0)$ and $p = \Lambda$. Since $\mu^\varphi(root(t)) = \varnothing$, p is the only innermost replacing redex in t. If $n > 0$, then the conclusion easily follows by the induction hypothesis. $\square$

Proofs of Section 6

Theorem 24 *Let $\mathcal{R}$ be a TRS, ς be an r-full strategy annotation, and $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. If t is a $\to_\varsigma$-normal form, then t is a μ-normal form.*

PROOF. We proceed by induction on pairs (t, L) (where t is a term and L is a list) ordered by the (left-to-right) lexicographic extension $(\succeq, \geq)$ of orderings $\succeq$ and $\geq$, where $\succeq$ is the subterm ordering (i.e., $t \succeq s$ if $\exists p \in \mathcal{P}os(t), s = t|_p$), and $\geq$ is the *suffix* ordering on lists (i.e., $L \geq L'$ if L' is a suffix of L). Minimal elements of this ordering are pairs (c, nil), where c is a constant symbol or a variable. If t is a constant and $\varsigma(root(t)) = nil$, then, since ς is r-full, there is no rule associated to $root(t)$. Hence, t is a μ-normal form.

Otherwise, by induction hypothesis, the conclusion follows. $\square$