Context-Sensitive Rewriting, Lazy Rewriting, and On-demand Rewriting*

Salvador Lucas

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

Abstract. On-demand rewriting provides an operational model for term rewriting controlled by annotations consisting of (sets of) positive and negative integers referring the arguments of function symbols. Context-sensitive rewriting and lazy rewriting provide (different) operational models for positive annotations. In this paper we prove that, under certain conditions, the three operational models coincide. In this case, it makes sense using context-sensitive rewriting as it is the simplest one. This fact also permits proving termination of lazy rewriting by proving termination of context-sensitive rewriting for a transformed rewrite system.

Keywords: Declarative programming, replacement restrictions, term rewriting, termination.

1 Introduction

Syntactic annotations (i.e., associated to the arguments of symbols) have been used in programming languages such as Lisp, Haskell, Clean, OBJ*, CafeOBJ, Maude, etc., to improve termination and efficiency of computations. Lazy languages (e.g., Haskell, Clean) interpret them as *strictness annotations* to become 'more eager' and efficient. Eager languages (e.g., Lisp, OBJ*, CafeOBJ, Maude) use them as *replacement restrictions* to become 'more lazy' thus (hopefully) avoiding nontermination. For instance, [FW76] studied implementations of Lisp where the list constructor operator *cons* did not evaluate its arguments, during certain stages of the computation. Algebraic languages, such as OBJ2 [FGJM85], OBJ3 [GWMFJ00], CafeOBJ [FN97], or Maude [CELM96] admit the specification of *local strategies* which are sequences of integers in parentheses.

Replacement restrictions have demonstrated their usefulness in practice, since they actually improve efficiency of computations by reducing the number of attempted matchings (see [FGJM85]). However, the possibility of introducing replacement restrictions should be made available together with some additional formal support allowing the programmer a clear (and simple) understanding of their influence on the operational behavior of the program. In [Luc01] we have introduced *on-demand rewriting* (*ODR*), an extension of context-sensitive rewriting (*CSR* [Luc98]) which can be used to provide an operational model for

* Work partially supported by Spanish CICYT and Conselleria de Cultura y Educació de la Generalitat Valenciana.

computations issued with OBJ (like) programs. In *ODR*, positive and negative annotations are managed by using a pair $\langle\mu,\mu_D\rangle$ of replacement maps μ (for positive annotations) and μ_D (for absolute values of negative ones). Following [Luc98], a replacement map μ for a signature Σ is a mapping $\mu : \Sigma \to \mathcal{P}(\mathbb{N})$ that satisfies $\mu(f) \subseteq \{1,\ldots,ar(f)\}$. Given a TRS $\mathcal{R}$ and a pair $\langle\mu,\mu_D\rangle$, the on-demand reduction relation $\hookrightarrow_{\langle\mu,\mu_D\rangle}$ is associated to $\mathcal{R}$ (see Section 3). A positive annotation i for f permits reductions on the i-th argument of f; a negative annotation $-i$ activates reductions 'on-demand'. The absence of both i and $-i$ forbids reductions on the i-th argument of f. Termination of *ODR* is investigated in [Luc01]. In this paper, we study monotony of *ODR*: in Section 4, we define an ordering $\sqsubseteq_D$ on pairs of replacement maps which is compatible with the on-demand rewrite relation, i.e., if $\langle\mu,\mu_D\rangle \sqsubseteq_D \langle\mu',\mu'_D\rangle$, then $\hookrightarrow_{\langle\mu,\mu_D\rangle} \subseteq \hookrightarrow_{\langle\mu',\mu'_D\rangle}$. In this way, the programmer can compare different annotations for the program.

In Section 5, we consider Fokkink et al.'s lazy graph rewriting (*LR* [FKW00]), a restriction of (graph) rewriting that provides a (non-deterministic) operational model for using replacement restrictions specified by a single replacement map μ. Lazy rewriting is intended to *improve the termination behavior of TRSs* [FKW00]. Unfortunately, no analysis of termination of *LR* is available yet. In this paper, we adapt Fokkink et al.'s framework to lazy *term* rewriting and provide a transformation for proving termination of *LR* by using the techniques for proving termination of *CSR* (see, e.g., [GM99,Luc98,Zan97]). We show that under some conditions (namely, that all non-variable subterms of the left-hand sides of rules are μ-replacing), *CSR*, *LR*, and *ODR* coincide. In this case, it makes sense using *CSR* as it is the simplest one. This is interesting since, looking for better implementations of *LR*, [FKW00,Ngu01] pay some attention to develop transformation techniques for achieving this condition thus (silently) using *CSR* rather than *LR*. Section 6 concludes.

2 Preliminaries

Given a set A, $\mathcal{P}(A)$ denotes the set of all subsets of A. Given a binary relation R on a set A, we denote its transitive closure by R^+ and its reflexive and transitive closure by R^*. An element $a \in A$ is an R-normal form, if there is no b such that $a\ R\ b$. NF_R is the set of normal forms of R. We say that b is an R-normal form of a, if b is an R-normal form and $a\ R^* b$. We say that R is *terminating* iff there is no infinite sequence $a_1\ R\ a_2\ R\ a_3 \cdots$. Throughout the paper, $\mathcal{X}$ denotes a countable set of variables and Σ denotes a signature, i.e., a set of function symbols $\{\mathtt{f},\mathtt{g},\ldots\}$, each having a fixed arity given by a mapping $ar : \Sigma \to \mathbb{N}$. The set of terms built from Σ and $\mathcal{X}$ is $\mathcal{T}(\Sigma,\mathcal{X})$. A term is said to be linear if it has no multiple occurrences of a single variable. Terms are viewed as labelled trees in the usual way. Positions $p,q,\ldots$ are represented by chains of positive natural numbers used to address subterms of t. Given positions p,q, we denote its concatenation as $p.q$. Positions are ordered by the standard prefix ordering $\leq$. Given a set of positions P, $minimal_{\leq}(P)$ is the set of minimal positions of P w.r.t. $\leq$. If p is a position, and Q is a set of positions, $p.Q =$

$\{p.q \mid q \in Q\}$. We denote the empty chain by Λ. The set of positions of a term t is $\mathcal{P}os(t)$. Positions of non-variable symbols in t are denoted as $\mathcal{P}os_{\Sigma}(t)$, and $\mathcal{P}os_{\mathcal{X}}(t)$ are the positions of variables. 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 by s. The symbol labelling the root of t is denoted as $root(t)$. The chain of symbols lying on positions above/on $p \in \mathcal{P}os(t)$ is $prefix_t(\Lambda) = root(t)$, $prefix_t(i.p) = root(t).prefix_{t|_i}(p)$. The strict prefix is $sprefix_t(\Lambda) = \Lambda$, $sprefix_t(p.i) = prefix_t(p)$.

A rewrite rule is an ordered pair (l, r), written $l \to r$, with $l, r \in \mathcal{T}(\Sigma, \mathcal{X})$, $l \notin \mathcal{X}$ and $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$. The left-hand side (*lhs*) of the rule is l and r is the right-hand side (*rhs*). A TRS is a pair $\mathcal{R} = (\Sigma, R)$ where R is a set of rewrite rules. $L(\mathcal{R})$ denotes the set of *lhs*'s of $\mathcal{R}$. A TRS $\mathcal{R}$ is left-linear if for all $l \in L(\mathcal{R})$, l is a linear term. A term $t \in \mathcal{T}(\Sigma, \mathcal{X})$ rewrites to s (at position p), written $t \xrightarrow{p}_{\mathcal{R}} s$ (or just $t \to s$), 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.

3 Context-sensitive rewriting and on-demand rewriting

Given a signature Σ, a mapping $\mu : \Sigma \to \mathcal{P}(\mathbb{N})$ is a *replacement map* (or Σ-map) if for all $f \in \Sigma$, $\mu(f) \subseteq \{1, \ldots, ar(f)\}$ [Luc98]. The inclusion ordering $\subseteq$ on $\mathcal{P}(\mathbb{N})$ extends to an ordering $\sqsubseteq$ on M_Σ, the set of all Σ-maps: $\mu \sqsubseteq \mu'$ if for all $f \in \Sigma$, $\mu(f) \subseteq \mu'(f)$. The Boolean algebra $(\mathcal{P}(\mathbb{N}), \subseteq, \varnothing, \mathbb{N}, \cup, \cap, ^c)$ induces a Boolean algebra $(M_\Sigma, \sqsubseteq, \mu_\bot, \mu_\top, \sqcup, \sqcap, ^c)$: The minimum (maximum) element is $\mu_\bot$ ($\mu_\top$), given by $\mu_\bot(f) = \varnothing$ ($\mu_\top(f) = \{1, \ldots, ar(f)\}$) for all $f \in \Sigma$. The least upper bound (*lub*), $\sqcup$, and greatest lower bound (*glb*), $\sqcap$, are given by $(\mu \sqcup \mu')(f) = \mu(f) \cup \mu'(f)$ and $(\mu \sqcap \mu')(f) = \mu(f) \cap \mu'(f)$ for all $f \in \Sigma$. The complement μ^c of $\mu \in M_\Sigma$ is $\mu^c(f) = \{1, \ldots, ar(f)\} - \mu(f)$ for all $f \in \Sigma$. We also define the difference $\mu_1 - \mu_2$ of $\mu_1, \mu_2 \in M_\Sigma$ by $\mu_1 - \mu_2 = \mu_1 \sqcap \mu_2^c$ (or $(\mu_1 - \mu_2)(f) = \mu_1(f) - \mu_2(f)$ for all $f \in \Sigma$).

Given $\mu \in M_\Sigma$, the set of μ-*replacing* positions $\mathcal{P}os^\mu(t)$ is: $\mathcal{P}os^\mu(t) = \{\Lambda\}$, if $t \in \mathcal{X}$ and $\mathcal{P}os^\mu(t) = \{\Lambda\} \cup \bigcup_{i \in \mu(root(t))} i.\mathcal{P}os^\mu(t|_i)$, if $t \notin \mathcal{X}$. The *non-replacing* positions are $\overline{\mathcal{P}os^\mu}(t) = \mathcal{P}os(t) - \mathcal{P}os^\mu(t)$. In *context-sensitive rewriting* (*CSR* [Luc98]), we (only) rewrite redexes at *replacing* positions: t μ-rewrites to s, written $t \hookrightarrow_\mu s$, if $t \xrightarrow{p}_{\mathcal{R}} s$ and $p \in \mathcal{P}os^\mu(t)$.

Example 1. Consider the TRS [NO01]:

$$\begin{array}{lll} \texttt{2nd(x:y:z)} & \to & \texttt{y} \\ \texttt{from(x)} & \to & \texttt{x:from(s(x))} \end{array}$$

and $\mu(\texttt{:}) = \mu(\texttt{2nd}) = \mu(\texttt{from}) = \mu(\texttt{s}) = \{1\}$. Then we have:

$$\texttt{2nd(}\underline{\texttt{from(0)}}\texttt{)} \hookrightarrow_\mu \texttt{2nd(0:from(s(0)))}$$

where, since $1.2 \notin \mathcal{P}os^\mu(\texttt{2nd(0:from(s(0)))})$, μ-rewritting stops here.

A TRS $\mathcal{R}$ is μ-terminating if $\hookrightarrow_\mu$ is terminating. Given a pair $\langle \mu, \mu_D \rangle$ of replacement maps μ and μ_D, we define *on-demand rewriting* (*ODR* [Luc01]) as an extension of *CSR* (under μ), where we also permit reductions *on-demand* according to μ_D. Given $f \in \Sigma$, indices $j \in \mu_D(f)$ are aimed at enabling reductions on a subterm t_j of a function call $f(t_1, \ldots, t_j, \ldots, t_k)$ if they can eventually lead to matching a pattern of a rule defining f (i.e., $l \to r \in R$ such that $root(t) = f$). Let $\mathcal{L}azy_{\langle\mu,\mu_D\rangle}(t) = minimal_{\leq}(\overline{\mathcal{P}os^{\mu}}(t)) \cap \mathcal{P}os^{\mu \sqcup \mu_D}(t)$.

Definition 1 (On-demand rewriting [Luc01]**).** *Let $\mathcal{R} = (\Sigma, R)$ be a TRS and $\mu, \mu_D \in M_\Sigma$. Then, $t \hookrightarrow_{\langle\mu,\mu_D\rangle} s$, if $t \stackrel{p}{\to} s$ and either*

1. *$p \in \mathcal{P}os^{\mu}(t)$, or*
2. *$p \in \mathcal{P}os^{\mu \sqcup \mu_D}(t) - \mathcal{P}os^{\mu}(t)$ and there exist $e \in \mathcal{P}os^{\mu}(t)$, $r_1, \ldots, r_n, t' \in \mathcal{T}(\Sigma, \mathcal{X})$, $p_1, \ldots, p_n \in \mathcal{L}azy_{\langle\mu,\mu_D\rangle}(t)$, $l \to r \in R$, and substitution σ s.t.*
   (1) $e \leq p$, $t' = t[r_1]_{p_1} \cdots [r_n]_{p_n}$, $t'|_e = \sigma(l)$ and
   (2) for all $q \in \mathcal{P}os(l)$ s.t. $sprefix_{t|_e}(q) = sprefix_l(q)$, whenever $e.q \leq p$, we have that $l|_q \notin \mathcal{X}$.

Thus, a rewriting step $t \stackrel{p}{\to} s$ is *on-demand* (w.r.t. μ and μ_D) if either $t \stackrel{p}{\hookrightarrow}_{\mu} s$; or $t \stackrel{p}{\hookrightarrow}_{\mu \sqcup \mu_D} s$ and reducing $t|_p$ may contribute to a future μ-rewriting step at μ-replacing position e, using some rule $l \to r$. Such a contribution is approximated by checking whether changing some non-μ-replacing maximal subterms of t would eventually make the matching possible (condition 2(1) of Definition 1). Indices in μ_D determine the positions (in $\mathcal{L}azy_{\langle\mu,\mu_D\rangle}(t)$) of the subterms of t that can be refined. Note that p is always covered by some position $p_i \in \mathcal{L}azy_{\langle\mu,\mu_D\rangle}(t)$, i.e., $p_i \leq p$ (Figure 1, left).

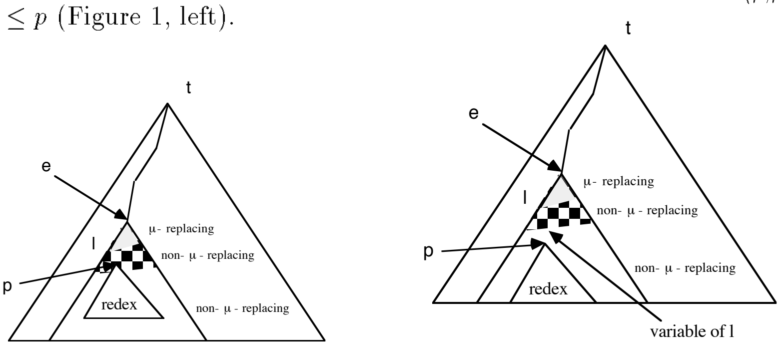

Fig. 1. Scheme of a possible/forbidden on-demand step

On the other hand, the position p on which the rewriting step is performed is constrained to having no variable position of l covering p (condition 2(2) of Definition 1); otherwise, the reduction at $t|_p$ would not improve the matching (Figure 1, right).

Example 2. Consider $\mathcal{R}$ and μ as in Example 1. Let $\mu_D(\texttt{:}) = \{2\}$. We have:

$$\texttt{2nd(}\underline{\texttt{from(0)}}\texttt{)} \hookrightarrow_{\langle\mu,\mu_D\rangle} \texttt{2nd(0:}\underline{\texttt{from(s(0))}}\texttt{)} \hookrightarrow_{\langle\mu,\mu_D\rangle} \underline{\texttt{2nd(0:s(0):from(s(s(0))))}}$$
$$\hookrightarrow_{\langle\mu,\mu_D\rangle} \texttt{s(0)}$$

but $\texttt{2nd(0:s(0):}\underline{\texttt{from(s(s(0)))}}\texttt{)} \not\hookrightarrow_{\langle\mu,\mu_D\rangle} \texttt{2nd(0:s(0):s(s(0)):from(s(s(s(0)))))}$ since $sprefix_t(1.2.2) = sprefix_l(1.2.2)$, but $l|_{1.2.2} \in \mathcal{X}$ (where $l = \texttt{2nd(x:y:z)}$).

$\hookrightarrow_{\langle\mu,\mu_D\rangle}$-normal forms are called $\langle\mu,\mu_D\rangle$-normal forms.

4 Monotony of on-demand rewriting

A simple fact about *CSR* is that it is monotone w.r.t. the ordering $\sqsubseteq$.

Proposition 1. [Luc98] *Let $\mathcal{R} = (\Sigma, R)$ be a TRS and $\mu, \mu' \in M_\Sigma$. If $\mu \sqsubseteq \mu'$, then $\hookrightarrow_\mu \subseteq \hookrightarrow_{\mu'}$.*

Monotony of *CSR* confirms what is intuitively clear: $\hookrightarrow_\mu$ provides less posibilities than $\hookrightarrow_{\mu'}$ for reduction, whenever $\mu \sqsubseteq \mu'$. Concerning monotony of *ODR*, the most natural extension of $\sqsubseteq$ on $M_\Sigma \times M_\Sigma$, i.e., the *product* order $\sqsubseteq_\times$ given by $\langle \mu, \mu_D \rangle \sqsubseteq_\times \langle \mu', \mu'_D \rangle$ if and only if $\mu \sqsubseteq \mu' \wedge \mu_D \sqsubseteq \mu'_D$, does not work.

Example 3. Consider the TRS $\mathcal{R}$:

$$\begin{array}{ll} \mathtt{c} \to \mathtt{h(c)} & \mathtt{f(a,b)} \to \mathtt{h(a)} \\ \mathtt{d} \to \mathtt{b} & \end{array}$$

and $t = \mathtt{f(c,c)}$. Let $\mu(\mathtt{f}) = \mu(\mathtt{h}) = \varnothing$, $\mu_D(\mathtt{f}) = \{1,2\}$, and $\mu_D(\mathtt{h}) = \varnothing$. Let $\mu'(\mathtt{f}) = \{1\}$, $\mu'(\mathtt{h}) = \varnothing$, and $\mu'_D = \mu_D$. Obviously, $\langle \mu, \mu_D \rangle \sqsubseteq_\times \langle \mu', \mu'_D \rangle$. Note that $1, 2 \in \mathcal{L}azy_{\langle \mu, \mu_D \rangle}(t)$. Then, since $t[\mathtt{a}]_1[\mathtt{b}]_2 = \mathtt{f(a,b)}$, we have

$$\mathtt{f(c,\underline{c})} \hookrightarrow_{\langle \mu, \mu_D \rangle} \mathtt{f(c,h(c))}$$

However, since $1 \notin \mathcal{L}azy_{\langle \mu', \mu'_D \rangle}(t)$, it follows that $\mathtt{f(c,\underline{c})} \not\hookrightarrow_{\langle \mu', \mu'_D \rangle} \mathtt{f(c,h(c))}$ since $t[\mathtt{b}]_2 = \mathtt{f(c,b)} \neq \mathtt{f(a,b)}$. Hence, $\hookrightarrow_{\langle \mu, \mu_D \rangle} \not\subseteq \hookrightarrow_{\langle \mu', \mu'_D \rangle}$.

Thus, *ODR* could exhibit the counterintuitive property that a more restrictive pair of replacement maps $\langle \mu, \mu_D \rangle \sqsubseteq_\times \langle \mu', \mu'_D \rangle$ enables reductions on $\langle \mu', \mu'_D \rangle$-normal forms.

Definition 2. *Let Σ be a signature. We define the preorder $\sqsubseteq_D$ on $M_\Sigma \times M_\Sigma$ as follows: given $\mu, \mu_D, \mu', \mu'_D \in M_\Sigma$,*

$$\langle \mu, \mu_D \rangle \sqsubseteq_D \langle \mu', \mu'_D \rangle \quad \Leftrightarrow \quad \mu \sqsubseteq \mu' \wedge (\mu_D - \mu) \sqsubseteq (\mu'_D - \mu')$$

Theorem 1. *Let $\mathcal{R} = (\Sigma, R)$ be a TRS and $\mu, \mu_D, \mu', \mu'_D \in M_\Sigma$. If $\langle \mu, \mu_D \rangle \sqsubseteq_D \langle \mu', \mu'_D \rangle$, then $\hookrightarrow_{\langle \mu, \mu_D \rangle} \subseteq \hookrightarrow_{\langle \mu', \mu'_D \rangle}$.*

Corollary 1. *Let $\mathcal{R} = (\Sigma, R)$ be a TRS and $\mu, \mu_D, \mu'_D \in M_\Sigma$. If $\mu_D \sqsubseteq \mu'_D$, then $\hookrightarrow_{\langle \mu, \mu_D \rangle} \subseteq \hookrightarrow_{\langle \mu, \mu'_D \rangle}$.*

Having a preorder instead of a partial order is not completely satisfactory. The following proposition allows us to improve this situation.

Proposition 2. *Let $\mathcal{R}$ be a TRS and $\mu, \mu_D \in M_\Sigma$. Then, $\hookrightarrow_{\langle \mu, \mu_D \rangle} = \hookrightarrow_{\langle \mu, \mu_D - \mu \rangle}$.*

Therefore, replacement maps μ and μ_D are not completely independent regarding the definition of *ODR*. We formalize this fact by only considering pairs of replacement maps μ and μ_D that are *independent* in the sense that $\mu \sqcap \mu_D = \mu_\bot$.

Definition 3. *Given a signature Σ, we let $M_\Sigma^D = \{\langle \mu, \mu_D \rangle \in M_\Sigma \times M_\Sigma \mid \mu \sqcap \mu_D = \mu_\bot\}$.*

The mapping given by $\langle \mu, \mu_D \rangle \mapsto \langle \mu, \mu_D - \mu \rangle$ for all $\langle \mu, \mu_D \rangle \in M_\Sigma \times M_\Sigma$ maps $M_\Sigma \times M_\Sigma$ onto M_Σ^D. Since Proposition 2 ensures that $\hookrightarrow_{\langle \mu, \mu_D \rangle} = \hookrightarrow_{\langle \mu, \mu_D - \mu \rangle}$, it follows that restricting the attention to M_Σ^D does not entail lack of generality.

Given an order $(A, \leq)$ and $B \subseteq A$, $\leq|_B$ denotes the order $\leq|_B = \{(a, b) \in \leq \mid a, b \in B\}$ on B. Now we have the following.

Proposition 3. *Let Σ be a signature. Then, $\sqsubseteq_D|_{M_\Sigma^D} = \sqsubseteq_\times|_{M_\Sigma^D}$.*

Thus, Proposition 3 allows us to use the *ordered* set $(M_\Sigma^D, \sqsubseteq_\times)$, where we write just $\sqsubseteq_\times$ rather than $\sqsubseteq_\times|_{M_\Sigma^D}$.

5 Lazy rewriting

In lazy *graph* rewriting [FKW00], reductions are issued on *labelled graphs*. We adapt the framework to lazy *term* rewriting on labelled terms. An alternative adaptation can be found in [Ngu01]. We believe that ours is simpler and closer to [FKW00].

Terms are built from a *lazy* signature, i.e., a signature supplied with a *laziness predicate* Λ on $\Sigma \times \mathbb{N}$ that holds for (f, i) if and only if $1 \leq i \leq ar(f)$ and the ith argument of f is lazy (Definition 3.1.1 of [FKW00]). Laziness predicate Λ can actually be identified with a replacement map μ:

$$\forall f \in \Sigma, (\ i \in \mu(f) \Leftrightarrow \neg\Lambda(f, i)\ )$$

We also label each node (or position) of a term t using e for eager *positions* or ℓ for lazy ones. Let $\mathcal{L} = \{e, \ell\}$; labelled terms are terms in $\mathcal{T}(\Sigma \times \mathcal{L}, \mathcal{X} \times \mathcal{L})$, which we denote as $\mathcal{T}(\Sigma_\mathcal{L}, \mathcal{X}_\mathcal{L})$. The labelling of a symbol $f \in \Sigma$ is denoted f^e or f^ℓ rather than $\langle f, e \rangle$ or $\langle f, \ell \rangle$. The arities are naturally extended: $ar(f^e) = ar(f^l) = ar(f)$ for all $f \in \Sigma$. Given $\mu \in M_\Sigma$, the mapping $label_\mu : \mathcal{T}(\Sigma, \mathcal{X}) \to \mathcal{T}(\Sigma_\mathcal{L}, \mathcal{X}_\mathcal{L})$ (or just *label* if no confusion arises) provides a *canonical labelling* of a term: given $t \in \mathcal{T}(\Sigma, \mathcal{X})$, the topmost position Λ of *label*(t) is always eager; given a position $p \in \mathcal{P}os(label(t))$ and $i \in \{1, \ldots, ar(root(t|_p))\}$, position $p.i$ of *label*(t) is lazy if and only $i \notin \mu(root(t|_p))$; otherwise, it is eager (Definition 3.1.2 of [FKW00]).

Example 4. Consider $\mathcal{R}$ and μ as in Example 1. Then, the labelling of term

$$s = \texttt{2nd(0:from(s(0)))} \quad \text{is} \quad t = label(s) = \texttt{2nd}^e\texttt{(0}^e\texttt{:}^e\texttt{from}^\ell\texttt{(s}^e\texttt{(0}^e\texttt{)))}.$$

It means that $\Lambda, 1, 1.1, 1.2.1$, and $1.2.1.1$ are eager positions; position 1.2 is lazy.

Given $t \in \mathcal{T}(\Sigma_\mathcal{L}, \mathcal{X}_\mathcal{L})$, $erase(t)$ is the term in $\mathcal{T}(\Sigma, \mathcal{X})$ that (obviously) corresponds to t after removing labels. Note that $erase \circ label = id_{\mathcal{T}(\Sigma,\mathcal{X})}$ but $label \circ erase \neq id_{\mathcal{T}(\Sigma_\mathcal{L},\mathcal{X}_\mathcal{L})}$.

Given $t \in \mathcal{T}(\Sigma_\mathcal{L}, \mathcal{X}_\mathcal{L})$, a position $p \in \mathcal{P}os(t)$ is eager (resp. lazy) if $root(t|_p)$ is labelled with e (resp. ℓ). The so called *active* positions of t are defined inductively as follows: Λ is an active position; if $p \in \mathcal{P}os(t)$ is active, then $p.i$ is active for all *eager* position $p.i$, $1 \leq i \leq ar(root(t|_p))$ of t (Definition 3.1.3 of [FKW00]). Active positions are always reachable from the root of the term via a path of eager positions. Eager positions do not necessarily satisfy this.

Example 5. (continuing Example 4) Positions $\Lambda, 1$, and 1.1 are active in

$$t = \texttt{2nd}^e(\texttt{0}^e \texttt{:}^e \texttt{from}^\ell(\texttt{s}^e(\texttt{0}^e)))$$

Positions 1.2.1 and 1.2.1.1 are eager but *not* active, since position 1.2 below is lazy in t.

Given $t \in \mathcal{T}(\Sigma, \mathcal{X})$ and $\mu \in M_\Sigma$, the set of active positions of *label*(t) coincides with $\mathcal{P}os^\mu(t)$. An important feature of lazy rewriting on labelled terms is that *the set of active nodes may increase as reduction of labelled terms proceeds.* Each lazy reduction step on labelled terms may have two different effects:

1. changing the status (active or not) of a given position within a term, or
2. performing a rewriting step (always on an active position).

In the following, we formally describe them as reduction of labelled terms.

5.1 Activating positions for reduction

The *activity* status of a lazy position immediately below an active position within a (labelled) term can be modified if the position is 'essential', i.e., *its contraction may lead to new redexes at active nodes* [FKW00].

Definition 4 (Matching modulo laziness [FKW00]**).** *Let $l \in \mathcal{T}(\Sigma, \mathcal{X})$ be linear, $t \in \mathcal{T}(\Sigma_\mathcal{L}, \mathcal{X}_\mathcal{L})$, and p be an active position of t. Then, l matches modulo laziness $s = t|_p$ if either $l \in \mathcal{X}$, or $l = f(l_1, \ldots, l_k)$, $s = f^e(s_1, \ldots, s_k)$ and, for all $i \in \{1, \ldots, k\}$, if $p.i$ is eager, then l_i matches modulo laziness s_i. If position $p.i$ is lazy and $l_i \notin \mathcal{X}$, then position $p.i$ is called* essential.

Example 6. Consider the TRS $\mathcal{R}$ of Example 1. Then, the *lhs* `2nd(x:y:z)` matches modulo laziness the labelled term $t = \texttt{2nd}^e(\texttt{0}^e \texttt{:}^e \texttt{from}^\ell(\texttt{s}^e(\texttt{0}^e)))$. According to Definition 4, position 1.2 of t becomes *essential.*

Note that if $l \in \mathcal{T}(\Sigma, \mathcal{X})$ matches modulo laziness an active labelled subterm $s = t|_p$ without producing essential positions, then l matches *erase*(s) in the usual sense. Changes in 'activity' of positions are formalized by the following.

Definition 5. *Let $\mathcal{R} = (\Sigma, R)$ be a left-linear TRS. The relation of* activation $\stackrel{\mathsf{A}}{\to}$ *between labelled terms is defined as follows. Let p be active in $t \in \mathcal{T}(\Sigma_\mathcal{L}, \mathcal{X}_\mathcal{L})$ and $l \to r \in R$ be such that l matches modulo laziness $t|_p$. Let q be an essential lazy position of t and $t|_q = f^\ell(s_1, \ldots, s_k)$. Then, $t \stackrel{\mathsf{A}}{\to} t[f^e(s_1, \ldots, s_k)]_q$.*

Note that $\stackrel{\mathsf{A}}{\to}$ is a terminating relation: only a finite number of relabellings (from lazy to eager) is possible for finite terms. In general, $\stackrel{\mathsf{A}}{\to}$ is not confluent.

Example 7. Consider the TRS $\mathcal{R}$:

$$\begin{array}{ll} \texttt{f(c(d,a))} & \to \texttt{a} \\ \texttt{b} & \to \texttt{f(c(b,d))} \end{array}$$

Then, we have

$$\texttt{f}^e(\texttt{c}^\ell(\texttt{b}^\ell,\texttt{d}^\ell)) \xrightarrow{\mathsf{A}} \texttt{f}^e(\texttt{c}^e(\texttt{b}^\ell,\texttt{d}^\ell)) \xrightarrow{\mathsf{A}} \texttt{f}^e(\texttt{c}^e(\texttt{b}^e,\texttt{d}^\ell))$$

and $\texttt{f}^e(\texttt{c}^e(\texttt{b}^e,\texttt{d}^\ell))$ is a $\xrightarrow{\mathsf{A}}$-normal form, since $\texttt{f(c(d,a))}$ does not match term $\texttt{f}^e(\texttt{c}^e(\texttt{b}^e,\texttt{d}^\ell))$ modulo laziness. However,

$$\texttt{f}^e(\texttt{c}^\ell(\texttt{b}^\ell,\texttt{d}^\ell)) \xrightarrow{\mathsf{A}} \texttt{f}^e(\texttt{c}^e(\texttt{b}^\ell,\texttt{d}^\ell)) \xrightarrow{\mathsf{A}} \texttt{f}^e(\texttt{c}^e(\texttt{b}^\ell,\texttt{d}^e))$$

thus leading to a different $\xrightarrow{\mathsf{A}}$-normal form.

The *canonical* replacement map $\mu^{can}_{\mathcal{R}}$ associated to a TRS $\mathcal{R} = (\Sigma, R)$, is the most restrictive replacement map (of M_Σ) that permits replacements at every non-variable position of the lhs's of R, i.e., the minimum Σ-map μ such that $\forall l \in L(\mathcal{R}), \mathcal{P}os_\Sigma(l) \subseteq \mathcal{P}os^\mu(l)$ [Luc98]. According to this, we have the following.

Proposition 4. *Let $\mathcal{R} = (\Sigma, R)$ be a a left-linear TRS and $\mu \in M_\Sigma$ be such that $\mu^{can}_{\mathcal{R}} \sqsubseteq \mu$. Then, label(t) is a $\xrightarrow{\mathsf{A}}$-normal form.*

5.2 Reducing active positions

Let $l \in \mathcal{T}(\Sigma, \mathcal{X})$ be a linear term, $t \in \mathcal{T}(\Sigma_\mathcal{L}, \mathcal{X}_\mathcal{L})$, $p \in \mathcal{P}os(t)$, and $s = t|_p$. If l matches $erase(s)$, then we let the mapping $\sigma_{l,s} : \mathcal{X} \to \mathcal{T}(\Sigma_\mathcal{L}, \mathcal{X}_\mathcal{L})$ to be

$$\sigma_{l,s}(x) = \begin{cases} s|_q & \text{if } \exists q \in \mathcal{P}os(l) \text{ such that } l|_q = x \in \mathcal{X} \\ x^e & \text{otherwise, where the labelling } {}^e \text{ for } x \text{ is arbitrary} \end{cases}$$

From $\sigma_{l,s}$, we obtain a substitution σ on labelled terms as the homomorphic extension of the following: for all $x \in \mathcal{X}$,

$$\sigma(x^e) = \begin{cases} y^e & \text{if } \sigma_{l,s}(x) = y^\lambda \in \mathcal{X}_\mathcal{L} \\ f^e(t_1,\ldots,t_k) & \text{if } \sigma_{l,s}(x) = f^\lambda(t_1,\ldots,t_k) \end{cases}$$
$$\sigma(x^\ell) = \begin{cases} y^\ell & \text{if } \sigma_{l,s}(x) = y^\lambda \in \mathcal{X}_\mathcal{L} \\ f^\ell(t_1,\ldots,t_k) & \text{if } \sigma_{l,s}(x) = f^\lambda(t_1,\ldots,t_k) \end{cases}$$

for $\lambda \in \{e, \ell\}$. This is according to Definition 3.2.3 in [FKW00]: when a substitution σ on labelled terms applies to a labelled term t, the labelling that corresponds to the symbol in position q in $\sigma(t)$ is that of q in t, for every variable position $q \in \mathcal{P}os_\mathcal{X}(t)$. Thus, we define the following.

Definition 6. *Let $\mathcal{R} = (\Sigma, R)$ be a left-linear TRS and $\mu \in M_\Sigma$. The relation of* active rewriting *$\xrightarrow{\mathsf{R}}_\mu$ between labelled terms is defined as follows. Let p be an active position of $t \in \mathcal{T}(\Sigma_\mathcal{L}, \mathcal{X}_\mathcal{L})$, $s = t|_p$ and $l \to r \in R$ be such that l matches erase(s). Let $\sigma_{l,s}$ be the corresponding mapping. Then, $t \xrightarrow{\mathsf{R}}_\mu t[\sigma(label(r))]_p$.*

Example 8. Consider the TRS

$$\texttt{f(b,x)} \to \texttt{g(x)}$$

and $\mu(\texttt{f}) = \mu(\texttt{g}) = \{1\}$. Let $t = \texttt{f}^e(\texttt{b}^e,\texttt{a}^\ell)$; notice that $label(\texttt{g(x)}) = \texttt{g}^e(\texttt{x}^e)$. Then, $\texttt{f(b,x)}$ matches $erase(t)$. We have $\sigma_{\texttt{f(b,x)},t}(\texttt{x}) = \texttt{a}^\ell$. We obtain the substitution σ given by $\sigma(\texttt{x}^e) = \texttt{a}^e$ and $\sigma(\texttt{x}^\ell) = \texttt{a}^\ell$. Then,

$$\mathtt{f}^e(\mathtt{b}^e,\mathtt{a}^\ell) \xrightarrow{\mathsf{R}}_\mu \mathtt{g}^e(\mathtt{a}^e)$$

The following connection between $\xrightarrow{\mathsf{R}}_\mu$ and *CSR* is interesting.

Proposition 5. *Let $\mathcal{R} = (\Sigma, R)$ be a left-linear TRS, $\mu \in M_\Sigma$, $t \in \mathcal{T}(\Sigma,\mathcal{X})$ and $s' \in \mathcal{T}(\Sigma_\mathcal{L},\mathcal{X}_\mathcal{L})$. Then, $label(t) \xrightarrow{\mathsf{R}}_\mu s'$ if and only if $\exists s \in \mathcal{T}(\Sigma,\mathcal{X}), t \hookrightarrow_\mu s$ and $s' = label(s)$*

Remark 1. Example 8 shows that $\xrightarrow{\mathsf{R}}_\mu$-steps can also *indirectly* activate lazy positions after contracting a (labelled) redex. For instance, we can thought of the $\xrightarrow{\mathsf{R}}_\mu$-step on $t = \mathtt{f}^e(\mathtt{b}^e,\mathtt{a}^\ell)$ as activating the lazy occurrence of $\mathtt{a}$ in t when it is reduced into $\mathtt{g}^e(\mathtt{a}^e)$.

5.3 Lazy rewriting vs. other restrictions of rewriting

The lazy *graph* rewriting as given in Definition 3.2.3 of [FKW00] corresponds to relation $\xrightarrow{\mathsf{LR}}_\mu = \xrightarrow{\mathsf{A}} \cup \xrightarrow{\mathsf{R}}_\mu$ on labelled *terms*, *LR*. Whenever *LR* is used for evaluating an *unlabelled* term $t \in \mathcal{T}(\Sigma,\mathcal{X})$, we are actually interested in $\xrightarrow{\mathsf{LR}}_\mu$-reductions issued from $label(t)$. In this way, as done in [NO01,OF00] for OBJ (like) languages (and it is implicit in [FKW00]), we can define an *evaluation* semantics, i.e., a mapping $LR\text{-}eval_\mu : \mathcal{T}(\Sigma,\mathcal{X}) \to \mathcal{P}(\mathcal{T}(\Sigma,\mathcal{X}))$ that obtains the evaluation of a given term by using *LR*:

$$LR\text{-}eval_\mu(t) = \{erase(s) \in \mathcal{T}(\Sigma,\mathcal{X}) \mid label(t) \xrightarrow{\mathsf{LR}}{}^{*}_{\mu} s \wedge s \in \mathsf{NF}_{\xrightarrow{\mathsf{LR}}_\mu}\}$$

According to this, given $\mathcal{R} = (\Sigma, R)$ and $\mu \in M_\Sigma$, we say that

> $\mathcal{R}$ is $LR(\mu)$-terminating if, for all $t \in \mathcal{T}(\Sigma,\mathcal{X})$, there is no infinite $\xrightarrow{\mathsf{LR}}_\mu$-rewrite sequence starting from $label(t)$.

We have the following.

Theorem 2. *Let $\mathcal{R} = (\Sigma, R)$ be a left-linear TRS, $\mu \in M_\Sigma$, and $t, s \in \mathcal{T}(\Sigma,\mathcal{X})$. Then, $t \hookrightarrow_\mu s$ if and only if $label(t) \xrightarrow{\mathsf{LR}}_\mu label(s)$.*

Theorem 2 expresses that *CSR* can always be seen as a restriction of *LR* that only considers 'canonically labelled' terms. We also have the following.

Theorem 3. *Let $\mathcal{R} = (\Sigma, R)$ be a left-linear TRS, $\mu \in M_\Sigma$ be such that $\mu^{can}_\mathcal{R} \sqsubseteq \mu$, $t \in \mathcal{T}(\Sigma,\mathcal{X})$, and $s' \in \mathcal{T}(\Sigma_\mathcal{L},\mathcal{X}_\mathcal{L})$. Then, $label(t) \xrightarrow{\mathsf{LR}}_\mu s'$ if and only if $\exists s \in \mathcal{T}(\Sigma,\mathcal{X}), t \hookrightarrow_\mu s$ and $s' = label(s)$*

Theorem 3 expresses than, whenever $\mu^{can}_\mathcal{R} \sqsubseteq \mu$, *LR* can be simulated by *CSR*, thus providing an alternative (simpler) evaluation mechanism: let $CSR\text{-}eval_\mu(t) = \{s \in \mathcal{T}(\Sigma,\mathcal{X}) \mid t \hookrightarrow^*_\mu s \wedge s \in \mathsf{NF}_{\hookrightarrow_\mu}\}$. We have:

Corollary 2. *Let $\mathcal{R} = (\Sigma, R)$ be a left-linear TRS and $\mu \in M_\Sigma$. If $\mu_\mathcal{R}^{can} \sqsubseteq \mu$, then,* LR-$eval_\mu$ = CSR-$eval_\mu$.

Concerning $LR(\mu)$-termination, Theorem 3 also has the following consequence.

Corollary 3. *Let $\mathcal{R} = (\Sigma, R)$ be a left-linear TRS and $\mu \in M_\Sigma$ be such that $\mu_\mathcal{R}^{can} \sqsubseteq \mu$. Then, $\mathcal{R}$ is μ-terminating if and only if $\mathcal{R}$ is* LR(μ)*-terminating.*

Example 9. Consider the following TRS $\mathcal{R}$:

```
nth(x,y:z)    → y              inf(x) → x:inf(s(x))
nth(s(x),y:z) → nth(x,z)
```

with $\mu(\texttt{:}) = \mu(\texttt{s}) = \mu(\texttt{inf}) = \{1\}$ and $\mu(\texttt{nth}) = \{1, 2\}$ [FKW00]. Replacement map μ (or the corresponding laziness predicate Λ) was selected by Fokking et al. to (hopefully) '*avoid infinite reductions* (with lazy rewriting)' ([FKW00], page 47). Since $\mu_\mathcal{R}^{can} \sqsubseteq \mu$, by Corollary 3, $LR(\mu)$-termination and μ-termination coincide. Since $\mathcal{R}$ is easily proved μ-terminating (see Example 2 in [Zan97]), Corollary 3 permits a formal proof of Fokkink et al.'s claim.

In general, *CSR* and *ODR* are related as follows.

Theorem 4. [Luc01] *Let $\mathcal{R} = (\Sigma, R)$ be a TRS and $\mu, \mu_D \in M_\Sigma$. Then, $\hookrightarrow_\mu = \hookrightarrow_{\langle\mu,\mu_\bot\rangle}$ and $\hookrightarrow_\mu \subseteq \hookrightarrow_{\langle\mu,\mu_D\rangle} \subseteq \hookrightarrow_{\mu\sqcup\mu_D}$.*

On the other hand, we also have the following.

Theorem 5. *Let $\mathcal{R} = (\Sigma, R)$ be a left-linear TRS, and $\mu, \mu_D \in M_\Sigma$ be such that $\mu_\mathcal{R}^{can} \sqsubseteq \mu$. Then, $\hookrightarrow_\mu = \hookrightarrow_{\langle\mu,\mu_D\rangle}$.*

Thus, if $\mu_\mathcal{R}^{can} \sqsubseteq \mu$, both lazy and on-demand rewriting collapse into context-sensitive rewriting. In general, *ODR* and *LR* are not (easily) comparable.

Example 10. Consider $\mathcal{R}$ and μ as in Example 3. Let $t = \texttt{f(c,b)}$. We have:

$$label(t) = \texttt{f}^e\texttt{(c}^\ell\texttt{,b}^\ell\texttt{)} \stackrel{\mathsf{A}}{\to} \texttt{f}^e\texttt{(c}^e\texttt{,b}^\ell\texttt{)} \stackrel{\mathsf{R}}{\to}_\mu \texttt{f}^e\texttt{(h}^e\texttt{(c}^\ell\texttt{),b}^\ell\texttt{)}$$

where $\texttt{f}^e\texttt{(h}^e\texttt{(c}^\ell\texttt{),b}^\ell\texttt{)}$ is a $\stackrel{\mathsf{LR}}{\to}_\mu$-normal form. However, t gives rise to an infinite $\hookrightarrow_{\langle\mu,\mu^c\rangle}$-computation:

$$\texttt{f(}\underline{\texttt{c}}\texttt{,b)} \hookrightarrow_{\langle\mu,\mu^c\rangle} \texttt{f(h(}\underline{\texttt{c}}\texttt{),b)} \hookrightarrow_{\langle\mu,\mu^c\rangle} \cdots$$

Example 11. Consider the TRS $\mathcal{R}$

```
f(c(x,a)) → a
b         → f(c(b,d))
```

and $\mu(\texttt{f}) = \varnothing$, $\mu(\texttt{c}) = \{1, 2\}$. Then, $t = \texttt{f(c(b,d))}$ is a $\langle\mu,\mu^c\rangle$-normal form, since position $1.1 \in \mathcal{P}os^{\mu\sqcup\mu^c}(t)$ corresponds to a variable of the lhs $\texttt{f(c(x,a))}$. However, $label(t)$ gives rise to an infinite $\stackrel{\mathsf{LR}}{\to}_\mu$-computation:

$$label(t) = \texttt{f}^e\texttt{(c}^\ell\texttt{(b}^e\texttt{,d}^e\texttt{))} \stackrel{\mathsf{A}}{\to} \texttt{f}^e\texttt{(c}^e\texttt{(}\underline{\texttt{b}^e}\texttt{,d}^e\texttt{))} \stackrel{\mathsf{R}}{\to}_\mu \texttt{f}^e\texttt{(c}^e\texttt{(f}^e\texttt{(c}^\ell\texttt{(b}^e\texttt{,d}^e\texttt{)),d}^e\texttt{))}$$
$$\stackrel{\mathsf{A}}{\to} \texttt{f}^e\texttt{(c}^e\texttt{(f}^e\texttt{(c}^e\texttt{(}\underline{\texttt{b}^e}\texttt{,d}^e\texttt{)),d}^e\texttt{))} \stackrel{\mathsf{R}}{\to}_\mu \cdots$$

5.4 Termination of lazy rewriting

Corollary 3 provides quite a restricted framework for proving termination of *LR*. However, it provides the basis for proving it as termination of *CSR* for a transformed TRS.

In [Ngu01], a transformation of pairs $\langle \mathcal{R}, \mu \rangle$ of TRSs and replacement maps is proposed to force that non-variable subterms of all left-hand sides of rules in $\mathcal{R}$ be μ-replacing, i.e., to achieve $\mu^{can}_{\mathcal{R}} \sqsubseteq \mu$. The transformation is as follows (see Section 6.1 of [Ngu01]): let $\mathcal{R} = (\Sigma, R)$ be a TRS and $\mu \in M_\Sigma$. Let $l \to r \in R$, $p \in \mathcal{P}os_\Sigma(l)$, $root(l|_p) = f$, and $i \notin \mu(f)$ be such that $p.i \in \mathcal{P}os_\Sigma(l)$. Then we (nondeterministically) obtain $\mathcal{R}' = (\Sigma', R')$ and $\mu' \in M_{\Sigma'}$ as follows: $\Sigma' = \Sigma \cup \{f'\}$, where f' is a new symbol of arity $ar(f') = ar(f)$ and such that $\mu'(f') = \mu(f) \cup \{i\}$, and $\mu'(g) = \mu(g)$ for all $g \in \Sigma$. On the other hand,

$$R' = R - \{l \to r\} \cup \{l' \to r, l[x]_{p.i} \to l'[x]_{p.i}\}$$

where $l' = l[f'(l|_{p.1}, \ldots, l|_{p.k})]_p$ if $ar(f) = k$, and x is a new variable.

Example 12. Consider $\mathcal{R}$ as in Example 7 and $\mu = \mu_\perp$. Then, $\mathcal{R}'$ is:

$$\begin{array}{lll} \mathtt{f_1(c(d,a))} & \to \mathtt{a} & \qquad \mathtt{b} \to \mathtt{f(c(b,d))} \\ \mathtt{f(x)} & \to \mathtt{f_1(x)} & \end{array}$$

and $\mu'(\mathtt{f}) = \mu'(\mathtt{c}) = \varnothing$, $\mu'(\mathtt{f_1}) = \{1\}$. A different possibility (among others) is the following:

$$\begin{array}{lll} \mathtt{f(c'(d,a))} & \to \mathtt{a} & \qquad \mathtt{b} \to \mathtt{f(c(b,d))} \\ \mathtt{f(c(x,a))} & \to \mathtt{f(c'(x,a))} & \end{array}$$

and $\mu'(\mathtt{f}) = \mu'(\mathtt{c}) = \varnothing$, $\mu'(\mathtt{c'}) = \{1\}$.

The transformation proceeds in this way (starting now from $\mathcal{R}'$ and μ') until obtaining $\mathcal{R}^\#$ and $\mu^\#$ such that $\mu^{can}_{\mathcal{R}^\#} \sqsubseteq \mu^\#$. In particular, if $\mu^{can}_{\mathcal{R}} \sqsubseteq \mu$, then $\mathcal{R}^\# = \mathcal{R}$ and $\mu^\# = \mu$.

Example 13. Continuing Example 12, $\mathcal{R}^\#$ is:

$$\begin{array}{llll} \mathtt{f_1(c_2(d,a))} & \to \mathtt{a} & \qquad \mathtt{f_1(c(x,y))} & \to \mathtt{f_1(c_3(x,y))} \\ \mathtt{f_1(c_1(d,x))} & \to \mathtt{f_1(c_2(d,x))} & \qquad \mathtt{f(x)} & \to \mathtt{f_1(x)} \\ \mathtt{f_1(c_3(x,a))} & \to \mathtt{f_1(c_1(x,a))} & \qquad \mathtt{b} & \to \mathtt{f(c(b,d))} \end{array}$$

and $\mu^\#$ is given by $\mu^\#(\mathtt{f}) = \mu^\#(\mathtt{c}) = \varnothing$, $\mu^\#(\mathtt{f_1}) = \mu^\#(\mathtt{c_1}) = \{1\}$, $\mu^\#(\mathtt{c_2}) = \{1,2\}$, and $\mu^\#(\mathtt{c_3}) = \{2\}$. Notice that $\mu^{can}_{\mathcal{R}^\#} \sqsubseteq \mu^\#$.

Corollary 3 suggests using such a transformation for proving $LR(\mu)$-termination of $\mathcal{R}$ as $\mu^\#$-termination of $\mathcal{R}^\#$, provided that the transformation preserves $LR(\mu)$-termination of $\mathcal{R}$. Unfortunately, this is not true.

Example 14. Consider $\mathcal{R}$ as in Example 7, $\mu = \mu_\perp$, and $\mathcal{R}^\#$ and $\mu^\#$ as in Example 13. Note that $\mathcal{R}$ is not $LR(\mu)$-terminating. For $t = \mathtt{f(c(b,d))}$, we have:

$$\begin{array}{l} label(t) = \mathtt{f}^e\mathtt{(c}^\ell\mathtt{(b}^\ell\mathtt{,d}^\ell\mathtt{))} \xrightarrow{\mathsf{A}} \mathtt{f}^e\mathtt{(c}^e\mathtt{(b}^\ell\mathtt{,d}^\ell\mathtt{))} \xrightarrow{\mathsf{A}} \mathtt{f}^e\mathtt{(c}^e\mathtt{(}\underline{\mathtt{b}^e}\mathtt{,d}^\ell\mathtt{))} \\ \quad \xrightarrow{\mathsf{R}}_\mu \mathtt{f}^e\mathtt{(c}^e\mathtt{(f}^e\mathtt{(c}^\ell\mathtt{(b}^\ell\mathtt{,d}^\ell\mathtt{)),d}^\ell\mathtt{))} \xrightarrow{\mathsf{A}}{}^+ \mathtt{f}^e\mathtt{(c}^e\mathtt{(f}^e\mathtt{(c}^e\mathtt{(}\underline{\mathtt{b}^e}\mathtt{,d}^\ell\mathtt{)),d}^\ell\mathtt{))} \xrightarrow{\mathsf{R}}_\mu \cdots \end{array}$$

However, $\mathcal{R}^{\#}$ is $LR(\mu^{\#})$-terminating. The problem is that some activations of lazy subterms are not possible now:

$$\underline{\mathtt{f}^e(\mathtt{c}^\ell(\mathtt{b}^\ell,\mathtt{d}^\ell))} \stackrel{\mathsf{R}}{\to}_{\mu^\#} \underline{\mathtt{f}_1^e(\mathtt{c}^e(\mathtt{b}^\ell,\mathtt{d}^\ell))} \stackrel{\mathsf{R}}{\to}_{\mu^\#} \mathtt{f}_1^e(\mathtt{c}_3^e(\mathtt{b}^\ell,\mathtt{d}^e))$$

The lazy subterm b^ℓ *cannot* be activated; $\mathtt{f}_1^e(\mathtt{c}_3^e(\mathtt{b}^\ell,\mathtt{d}^e))$ is a $\stackrel{\mathsf{R}}{\to}_{\mu^\#}$-normal form.

A simple modification of Nguyen's transformation provides a sound technique for proving $LR(\mu)$-termination. The trick is to include *all* possible activations of lazy (problematic) arguments for each considered symbol: given $l \to r \in R$ and $p \in \mathcal{P}os(l)$, we let

$$\mathcal{I}(l,p) = \{i \in \{1,\ldots,ar(root(l|_p))\} \mid i \notin \mu(root(l|_p)) \wedge p.i \in \mathcal{P}os_\Sigma(l)\}$$

Assume that $\mathcal{I}(l,p) = \{i_1,\ldots,i_n\}$ for some $n > 0$ (i.e., $\mathcal{I}(l,p) \neq \varnothing$) and let $f = root(l|_p)$. Then, $\mathcal{R}^\diamond = (\Sigma^\diamond, R^\diamond)$ and $\mu^\diamond \in M_{\Sigma^\diamond}$ are as follows: $\Sigma^\diamond = \Sigma \cup \{f_j \mid 1 \leq j \leq n\}$, where each f_j is a new symbol of arity $ar(f_j) = ar(f)$. We let $\mu^\diamond(f_j) = \mu(f) \cup \{i_j\}$ for $1 \leq j \leq n$, and $\mu^\diamond(g) = \mu(g)$ for all $g \in \Sigma$. On the other hand,

$$R^\diamond = R - \{l \to r\} \cup \{l'_j \to r \mid 1 \leq j \leq n\} \cup \{l[x]_{p.i_j} \to l'_j[x]_{p.i_j} \mid 1 \leq j \leq n\}$$

where $l'_j = l[f_j(l|_{p.1},\ldots,l|_{p.k})]_p$ if $ar(f) = k$, and x is a new variable.

Example 15. Consider $\mathcal{R}$ as in Example 7 and $\mu = \mu_\bot$. With the new transformation, we could obtain $\mathcal{R}^\diamond$ to be as the first $\mathcal{R}'$ obtained in Example 12. On the other hand, if symbol `c` (rather than `f`) of *lhs* `f(c(d,a))` $\to$ `a` is considered, now we obtain:

`f(c'(d,a))` $\to$ `a`	`f(c(x,a))` $\to$ `f(c'(x,a))`
`f(c''(d,a))` $\to$ `a`	`f(c(d,x))` $\to$ `f(c''(d,x))`
`b` $\to$ `f(c(b,d))`	

and $\mu^\diamond(\mathtt{f}) = \mu^\diamond(\mathtt{c}) = \varnothing$, $\mu^\diamond(\mathtt{c'}) = \{1\}$, $\mu^\diamond(\mathtt{c''}) = \{2\}$.

Again, the transformation proceeds in this way (starting now from $\mathcal{R}^\diamond$ and $\mu^\diamond$) until obtaining $\mathcal{R}^\natural = (\Sigma^\natural, R^\natural)$ and $\mu^\natural$ such that $\mu^{can}_{\mathcal{R}^\natural} \sqsubseteq \mu^\natural$. If $\mu^{can}_{\mathcal{R}} \sqsubseteq \mu$, then $\mathcal{R}^\natural = \mathcal{R}$ and $\mu^\natural = \mu$.

Example 16. Consider $\mathcal{R}$ as in Example 7 and $\mu = \mu_\bot$. Then, $\mathcal{R}^\natural$ is

$\mathtt{f}_1(\mathtt{c}'_2(\mathtt{d},\mathtt{a})) \to \mathtt{a}$	$\mathtt{f}_1(\mathtt{c}(\mathtt{x},\mathtt{y})) \to \mathtt{f}_1(\mathtt{c}_3(\mathtt{x},\mathtt{y}))$
$\mathtt{f}_1(\mathtt{c}_2(\mathtt{x},\mathtt{a})) \to \mathtt{f}_1(\mathtt{c}'_2(\mathtt{x},\mathtt{a}))$	$\mathtt{f}_1(\mathtt{c}_4(\mathtt{d},\mathtt{y})) \to \mathtt{f}_1(\mathtt{c}_2(\mathtt{d},\mathtt{y}))$
$\mathtt{f}_1(\mathtt{c}'_1(\mathtt{d},\mathtt{a})) \to \mathtt{a}$	$\mathtt{f}_1(\mathtt{c}(\mathtt{x},\mathtt{y})) \to \mathtt{f}_1(\mathtt{c}_4(\mathtt{x},\mathtt{y}))$
$\mathtt{f}_1(\mathtt{c}_1(\mathtt{d},\mathtt{x})) \to \mathtt{f}_1(\mathtt{c}'_1(\mathtt{d},\mathtt{x}))$	$\mathtt{f}(\mathtt{x}) \to \mathtt{f}_1(\mathtt{x})$
$\mathtt{f}_1(\mathtt{c}_3(\mathtt{x},\mathtt{a})) \to \mathtt{f}_1(\mathtt{c}_1(\mathtt{x},\mathtt{a}))$	$\mathtt{b} \to \mathtt{f}(\mathtt{c}(\mathtt{b},\mathtt{d}))$

and $\mu^\natural$ is given by $\mu^\natural(\mathtt{f}) = \mu^\natural(\mathtt{c}) = \varnothing$, $\mu^\natural(\mathtt{f}_1) = \mu^\natural(\mathtt{c}_1) = \mu^\natural(\mathtt{c}_4) = \{1\}$, $\mu^\natural(\mathtt{c}'_1) = \mu^\natural(\mathtt{c}'_2) = \{1,2\}$, and $\mu^\natural(\mathtt{c}_2) = \mu^\natural(\mathtt{c}_3) = \{2\}$. Notice that $\mu^{can}_{\mathcal{R}^\natural} \sqsubseteq \mu^\natural$. Consider the term t of Example 14. Now we have:

$$\begin{array}{l} label(t) = \underline{\mathtt{f}^e(\mathtt{c}^\ell(\mathtt{b}^\ell,\mathtt{d}^\ell))} \stackrel{\mathsf{R}}{\to}_{\mu^\natural} \underline{\mathtt{f}_1^e(\mathtt{c}^e(\mathtt{b}^\ell,\mathtt{d}^\ell))} \stackrel{\mathsf{R}}{\to}_{\mu^\natural} \mathtt{f}_1^e(\mathtt{c}_4^e(\underline{\mathtt{b}^e},\mathtt{d}^\ell)) \\ \quad \stackrel{\mathsf{R}}{\to}_{\mu^\natural} \mathtt{f}_1^e(\mathtt{c}_4^e(\mathtt{f}^e(\mathtt{c}^\ell(\mathtt{b}^\ell,\mathtt{d}^\ell)),\mathtt{d}^\ell)) \stackrel{\mathsf{R}}{\longrightarrow}^+_{\mu^\natural} \mathtt{f}_1^e(\mathtt{c}_4^e(\mathtt{f}_1^e(\mathtt{c}_4^e(\underline{\mathtt{b}^e},\mathtt{d}^\ell)),\mathtt{d}^\ell)) \stackrel{\mathsf{R}}{\to}_{\mu^\natural} \cdots \end{array}$$

Now, we have the following.

Theorem 6. *Let $\mathcal{R} = (\Sigma, R)$ be a left-linear TRS and $\mu \in M_\Sigma$. If $\mathcal{R}^\natural$ is $\mu^\natural$-terminating, then $\mathcal{R}$ is* LR(μ)*-terminating.*

Example 17. Consider $\mathcal{R}$ and μ as in Example 1. Then, $\mathcal{R}^\natural$ is:

$$\begin{array}{llll} \mathtt{2nd(x:'(y:z))} & \to \mathtt{y} & \mathtt{from(x)} & \to \mathtt{x:from(s(y))} \\ \mathtt{2nd(x:y)} & \to \mathtt{2nd(x:'y)} & & \end{array}$$

and $\mu^\natural$ is given by $\mu^\natural(\mathtt{2nd}) = \mu^\natural(\mathtt{:}) = \mu^\natural(\mathtt{from}) = \{1\}$ and $\mu^\natural(\mathtt{:'}) = \{1,2\}$. In fact, in this case $\mathcal{R}^\natural$ and $\mathcal{R}^\#$ coincide (see Example 21 of [Ngu01]). However, using Theorem 6, we can prove $LR(\mu)$-termination of $\mathcal{R}$, which was an open question in [Ngu01]: $\mu^\natural$-termination of $\mathcal{R}^\natural$ is proved by using Zantema's transformation [Zan97]: the TRS

$$\begin{array}{llll} \mathtt{2nd(x:'(y:z))} & \to \mathtt{y} & \mathtt{a(from'(x))} & \to \mathtt{from(x)} \\ \mathtt{2nd(x:y)} & \to \mathtt{2nd(x:'a(y))} & \mathtt{from(x)} & \to \mathtt{from'(x)} \\ \mathtt{from(x)} & \to \mathtt{x:from'(s(x))} & \mathtt{a(x)} & \to \mathtt{x} \end{array}$$

which is obtained in this way (where `a` and `from'` are new symbols) is terminating (use an *automatic* tool such as Contejean and Marché's C*i*ME 2.0 system[1]). Note that, since $\mu^{can}_{\mathcal{R}} \not\sqsubseteq \mu$, Corollary 3 does not directly apply to $\mathcal{R}$ and μ.

6 Conclusions and future work

We have characterized the set M^D_Σ of pairs of replacement maps that may produce different on-demand rewrite relations for TRSs on a signature Σ. Ordering $\sqsubseteq_\times$ between pairs of replacement maps in M^D_Σ provides a good framework for comparing on-demand rewriting computations, since monotony of *ODR* is guaranteed. We have provided an adaptation of lazy *graph* rewriting of [FKW00] to lazy *term* rewriting, *LR*. If we use replacement maps μ that are less restrictive than the canonical replacement map $\mu^{can}_{\mathcal{R}}$, then *CSR*, *LR* and *ODR* coincide for left-linear TRSs. This allows us to prove termination of *LR* by proving termination of *CSR* for a transformed rewrite system. As far as the author knows, this is the first proposal for proving termination of *LR*.

We think that our results may contribute to formally address the problem of specifying local strategies in OBJ (see [OF00,NO01]). Lazy rewriting is invoked in these works as being a kind of 'underlying' or 'inspirating' mecanism for dealing with annotations in local strategies [Ngu01,OF00]. As discussed in [Luc01], we believe that *ODR* can better achieve this, as *LR* considers a single replacement map, thus giving no explicit treatment for negative indices of local strategies. We hope that our analysis of connections between the different mechanisms helps to better understand their use in implementations. However, as noticed in [Luc01], further work should be done for describing OBJ computations as *ODR* reductions, since determinism introduced by the use of lists (rather than sets) of annotations should be expressed as some kind of *ODR* strategy.

[1] Available at `http://www.lri.fr/~demons/cime.html`.

References

[CELM96] M. Clavel, S. Eker, P. Lincoln, and J. Meseguer. Principles of Maude. In J. Meseguer, editor, *Proc. 1st International Workshop on Rewriting Logic and its Applications*, Electronic Notes in Theoretical Computer Science, volume 4, 25 pages, Elsevier Sciences, 1996.

[FGJM85] K. Futatsugi, J. Goguen, J.-P. Jouannaud, and J. Meseguer. Principles of OBJ2. In *Conference Record of the 12th Annual ACM Symposium on Principles of Programming Languages, POPL'85*, pages 52-66, ACM Press, 1985.

[FKW00] W. Fokkink, J. Kamperman, and P. Walters. Lazy Rewriting on Eager Machinery. *ACM Transactions on Programming Languages and Systems*, 22(1):45-86, 2000.

[FN97] K. Futatsugi and A. Nakagawa. An Overview of CAFE Specification Environment – An algebraic approach for creating, verifying, and maintaining formal specification over networks –. In *Proc. of 1st International Conference on Formal Engineering Methods*, 1997.

[FW76] D.P. Friedman and D.S. Wise. CONS should not evaluate its arguments. In S. Michaelson and R. Milner, editors, *Automata, Languages and Programming*, pages 257-284, Edinburgh University Press, 1976.

[GM99] J. Giesl and A. Middeldorp. Transforming Context-Sensitive Rewrite Systems. In P. Narendran and M. Rusinowitch, editors, *Proc. of 10th International Conference on Rewriting Techniques and Applications, RTA'99*, LNCS 1631:271-285, Springer-Verlag, Berlin, 1999.

[GWMFJ00] J.A. Goguen, T. Winkler, J. Meseguer, K. Futatsugi, and J.-P. Jouannaud. Introducing OBJ. In J. Goguen and G. Malcolm, editors, *Software Engineering with OBJ: algebraic specification in action*, Kluwer, 2000.

[Luc98] S. Lucas. Context-sensitive computations in functional and functional logic programs. *Journal of Functional and Logic Programming*, 1998(1):1-61, January 1998.

[Luc01] 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*, ACM Press, to appear, 2001.

[Ngu01] Q.-H. Nguyen. Compact Normalisation Trace via Lazy Rewriting. In B. Gramlich and S. Lucas, editors, *Proc. of 1st International Workshop on Reduction Strategies in Rewriting and Programming, WRS'01*, pages 79-96, volume 2359, Servicio de Publicaciones de la Universidad Politécnica de Valencia, 2001.

[NO01] M. Nakamura and K. Ogata. The evaluation strategy for head normal form with and without on-demand flags. In K. Futatsugi, editor, *Proc. of 3rd International Workshop on Rewriting Logic and its Applications, WRLA'2000, Electronic Notes in Theoretical Computer Science*, volume 36, 17 pages, 2001.

[OF00] K. Ogata and K. Futatsugi. Operational Semantics of Rewriting with the On-demand Evaluation Strategy. In *Proc of 2000 International Symposium on Applied Computing, SAC'00*, pages 756-763, ACM Press, 2000.

[Zan97] H. Zantema. Termination of Context-Sensitive Rewriting. In H. Comon, editor, *Proc. of 8th International Conference on Rewriting Techniques and Applications, RTA'97*, LNCS 1232:172-186, Springer-Verlag, Berlin, 1997.