Termination of Lazy Rewriting Revisited

Felix Schernhammer and Bernhard Gramlich

TU Wien, Austria, email: {felixs,gramlich}@logic.at

Abstract

Lazy rewriting is a proper restriction of term rewriting that dynamically restricts the reduction of certain arguments of functions in order to obtain termination. In contrast to context-sensitive rewriting reductions at such argument positions are not completely forbidden but delayed. Based on the observation that the only existing (non-trivial) approach to prove termination of such lazy rewrite systems is flawed, we develop a modified approach for transforming lazy rewrite systems into context-sensitive ones that is sound and complete with respect to termination.

1 Introduction

In functional programming languages, evaluations are often carried out in a lazy fashion. This means that in the evaluation of an expression, the result of certain subexpressions is not computed until it is known that the particular result is actually needed. A very similar idea is used in lazy rewriting ([FKW00]) where the reduction of certain subterms is delayed as long as possible.

Initially, lazy rewriting was introduced by [FKW00] in a graph rewriting setting (although the basic underlying idea is much older, cf. e.g. [FW76], [Str89], [Pv93]). However, for the termination analysis of lazy rewrite systems it is favorable to consider term rewriting instead of graph rewriting. Therefore, we will use the notion of lazy rewriting introduced in [Luc02b].

Restrictions of term rewriting have been studied over the last decades and have been used both for practical implementations of specification languages (cf. e.g. strategy annotations in Maude [CDE+03]) and for theoretical results (cf. e.g. [Luc98,Luc06], [GL06]). Some of the most sophisticated approaches like *on-demand rewriting* ([Luc01]) and *rewriting with on-demand strategy annotations* ([AEGL03]) incorporate lazy evaluation features. Thus, a better understanding of lazy rewriting may contribute to an improved understanding and analysis of these more recent approaches.

In [Luc02b] a transformation from lazy rewrite systems into context-sensitive ones was proposed, which was supposed to preserve non-termination and conjectured to be complete w.r.t. termination. Unfortunately, a counterexample (see Example 3.3 below) proves that this transformation is unsound w.r.t. termination. In this paper we repair the transformation and prove both soundness and completeness of the new transformation w.r.t. termination.

In Section 2 of this paper we will give basic definitions and introduce basic notations of lazy rewriting. In Section 3 we introduce the transformation of [Luc02b] and give a counterexample to its soundness w.r.t. termination. We propose a modified

version of the transformation which is proved to be sound and complete w.r.t. termination. Section 4 contains a discussion of the presented results and concludes. [1]

2 Preliminaries

We assume familiarity with the basic concepts and notations in term rewriting as well as context-sensitive term rewriting as provided for instance in [BN98,Luc98].

As in [FKW00] and [Luc02b] we are concerned with left-linear lazy rewrite systems in this work.

General assumption: Throughout the paper we assume that all lazy rewrite systems are *left-linear* [2] and *finite*.

Lazy rewriting operates on labelled terms. Each function and variable symbol of a term has either an *eager* label e or a *lazy* label l which we will write as superscripts. So given a signature $\Sigma = \{f_1, ..., f_n\}$, we consider a new signature $\Sigma' = \{f_1^e, f_1^l, ..., f_n^e, f_n^l\}$. We denote by V' the set of labelled variables, so $\mathcal{T}(\Sigma', V')$ is the set of labelled terms of a labelled signature Σ'. Following [Luc02b] we use a replacement map μ to specify for each function $f \in \Sigma$ which arguments should be evaluated eagerly. Given a replacement map μ we define the *canonical labelling* of terms as a mapping $label_\mu : \mathcal{T}(\Sigma, V) \to \mathcal{T}(\Sigma', V')$, where Σ' is the labelled signature and V' are the labelled variables [Luc02b]:

$$\begin{aligned} label_\mu(t) &= label^e_\mu(t) \\ label^\alpha_\mu(x) &= x^\alpha (\alpha \in \{e, l\}) \\ label^\alpha_\mu(f(t_1, ..., t_n)) &= f^\alpha(label^{\alpha_1}_\mu(t_1), ..., label^{\alpha_n}_\mu(t_n)) \\ &\quad \text{where } \alpha_i = e \text{ if } i \in \mu(f), \ l \text{ otherwise, and } \alpha \in \{e, l\} \end{aligned}$$

Given a labelled term t, the unlabelled term $erase(t)$ is constructed from t by omitting all labels. A position p of a term t is said to be *eager* (resp. *lazy*), if the symbol at the root of the subterm starting at position p of t has an *eager* (resp. *lazy*) label. Note that the *lazy* positions of a term are not the same as the non-replacing positions in context-sensitive rewriting. The reason is that in lazy rewriting eager positions may occur below lazy ones whereas in context-sensitive rewriting all positions which are below a non-replacing position are non-replacing.

However, rewrite steps may only be performed at so-called *active* positions. A position p is called *active* if all positions on the path from the root to p are eager.

Definition 2.1 ([Luc02b], [FKW00]) *The* active *positions of a labelled term t (denoted $Act(t)$) are recursively defined as follows.*

- *the root position ϵ of t is* active
- *if p is an* active *position and position $p.i$ is* eager*, then position $p.i$ is* active.

[1] Due to lack of space, the proofs of some auxiliary results (Propositions 3.7, 3.8 and 3.9 and Lemmata 3.13, 3.14, 3.19, 3.20 and 3.21 have been omitted here. They can be found in the full version of the paper at `http://www.logic.at/people/schernhammer/papers/wrs07-long.pdf`.

[2] Nevertheless, for clarity we will mention this assumption in the main results.

Note that given an unlabelled term t and a replacement map μ, the active positions of $label_\mu(t)$ are exactly the replacing positions w.r.t. context-sensitive rewriting.

Definition 2.2 ([Luc02b], [FKW00]) *Let $l \in \mathcal{T}(\Sigma, V)$ be linear, $t \in \mathcal{T}(\Sigma', V')$ be a labelled term and let p be an* active *position of t. Then l matches $t|_p$ modulo laziness if either*

- *$l \in V$ or*
- *If $l = f(l_1, ..., l_n)$ and $t|_p = f^e(t_1^\alpha, ..., t_n^\alpha)$ $(\alpha \in \{e, l\})$, then for all* eager *subterms t_i^e, l_i matches modulo laziness t_i^e.*

If t_i^l at positions $p.i$ is a lazy subterm and $l_i \notin V$, then position $p.i$ is called essential.

When writing t^e (resp. t^l) we mean that the term t has an eager (resp. lazy) root label. Informally, a matching modulo laziness is a partial matching ignoring (possible) clashes at lazy positions. Positions where such clashes occur may be activated (i.e., their label may be changed from lazy to eager).

Definition 2.3 ([Luc02b]) *Let $\mathcal{R} = (\Sigma, R)$ be a (left-linear) TRS. Let t be a labelled term and let l be the left-hand side of a rule of R. If l matches modulo laziness $t|_p$, and this matching gives rise to an essential position $p.i$ $(t|_{p.i} = f^l(t_1, ..., t_n))$, then $t \xrightarrow{A} t[f^e(t_1, ..., t_n)]_{p.i}$. The relation $\xrightarrow{A}$ is called* activation relation.

Definition 2.4 ([Luc02b]) *Let l be the (linear) left-hand side of a rewrite rule and let t be a labelled term. If l matches $erase(t)$, then the mapping $\sigma_{l,t} : Var(l) \to \mathcal{T}(\Sigma', V')$ is defined as follows. For all $x \in V$, with $l|_q = x$: $\sigma_{l,t}(x) = t|_q$.*

Informally, $\sigma_{l,t}$ is the matcher when matching l against t, where one adds the appropriate labels of t.

This substitution is modified to operate on labelled terms in the following way, yielding the mapping $\sigma\colon V' \to \mathcal{T}(\Sigma', V')$ [Luc02b]:

$$\sigma(x^e) = \begin{cases} y^e & \text{if } \sigma_{l,u}(x) = y^\alpha \in V' \\ f^e(t_1, ..., t_n) & \text{if } \sigma_{l,u}(x) = f^\alpha(t_1, ..., t_n) \end{cases}$$

$$\sigma(x^l) = \begin{cases} y^l & \text{if } \sigma_{l,u}(x) = y^\alpha \in V' \\ f^l(t_1, ..., t_n) & \text{if } \sigma_{l,u}(x) = f^\alpha(t_1, ..., t_n) \end{cases}$$

σ is homeomorphically extended to a mapping $\mathcal{T}(\Sigma', V') \to \mathcal{T}(\Sigma', V')$ as usual.

Definition 2.5 ([Luc02b]) *Let $\mathcal{R} = (\Sigma, R)$ be a (left-linear) TRS with replacement map μ. The* active rewrite *relation $\xrightarrow{R}_\mu$: $\mathcal{T}(\Sigma', V') \times \mathcal{T}(\Sigma', V')$ is defined as follows: Let t be a labelled term such that the left-hand side of a rewrite rule $l \to r$ matches $erase(t|_p)$ with $\sigma_{l,t|_p}$ and let $p \in Act(t)$. Then $t \xrightarrow{R}_\mu t[\sigma(label_\mu(r))]_p$.*

Informally, the active rewrite relation $\xrightarrow{R}_\mu$ performs rewrite steps according to rewrite rules as usual at active positions, where labels are considered. The lazy rewrite relation $\xrightarrow{LR}_\mu$ is the union of the activation relation and the active rewrite relation.

Definition 2.6 ([Luc02b]) *Let $\mathcal{R}$ be a (left-linear) TRS and let μ be a replacement map for $\mathcal{R}$. The lazy rewrite relation $\overset{LR}{\to}_\mu$ induced by $(\mathcal{R}, \mu)$ is the union of the two relations $\overset{A}{\to}$ and $\overset{R}{\to}_\mu$ ($\overset{LR}{\to}_\mu = \overset{A}{\to} \cup \overset{R}{\to}_\mu$).*

Definition 2.7 *Let $\mathcal{R}$ be a TRS with a replacement map μ. Then $\mathcal{R}$ is $LR(\mu)$-terminating if there is no infinite $\overset{LR}{\to}_\mu$-sequence starting from a term t, whose labelling is canonical or more liberal (i.e., whenever $label_\mu(erase(t))|_p$ is eager, then $t|_p$ is eager as well).*

Informally, we call a labelled term t more liberal than its canonically labelled version $label_\mu(erase(t))$ if it has strictly more eager labels. The reason for considering terms with canonical or more liberal labelling in the definition of $LR(\mu)$-termination, is that only such terms appear in lazy reduction sequences starting from canonically labelled terms, in which we are actually interested.

Note that $LR(\mu)$-termination and well-foundedness of $\overset{LR}{\to}_\mu$ do not coincide in general.

Example 2.8 *Consider the TRS $g(f(a), c) \to a$, $h(x, f(b)) \to g(x, h(x, x))$ with a replacement map $\mu(f) = \mu(g) = \{1\}$ and $\mu(h) = \{1, 2\}$. This system is $LR(\mu)$-terminating. This can be shown with the transformation of Definition 3.6 and Theorem 3.22. However, $\overset{LR}{\to}_\mu$ is not well-founded:*

$$\underline{g^e(f^e(b^l), h^l(f^e(b^l), f^e(b^l)))} \overset{LR}{\to}_\mu g^e(f^e(b^l), h^e(f^e(b^l), f^e(b^l))) \overset{LR}{\to}_\mu g^e(f^e(b^l), \underline{g^e(f^e(b^l), h^l(f^e(b^l), f^e(b^l)))}) \overset{LR}{\to}_\mu \dots .$$

3 Transforming Lazy Rewrite Systems

We start with the definition of the transformation of [Luc02b], because it provides the basic ideas for our new one. The main idea of the transformation is to explicitly mimic activation steps of lazy rewriting through special activation rules in the transformed system which basically exchange function symbols to make them more eager (this goes back to [Ngu01]). Activations in lazy rewriting are possible at positions which correspond to a non-variable position of the left-hand side of some rule in a partial matching. This is why in the transformation we are concerned with non-variable lazy positions of left-hand sides of rules.

The transformation is iterative. In each iteration new rules are created until a fixpoint is reached. The following definition identifies for a rule $l \to r$ and a position p the positions $p.i$ which are lazy in $label_\mu(l)$. These positions are dealt with in parallel in one step of the transformation.

Definition 3.1 ([Luc02b]) *Let $l \to r$ be a rewrite rule and p a non-variable position of l, then*

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

Definition 3.2 ([Luc02b]) *Let $\mathcal{R} = (\Sigma, R)$ be a TRS with replacement map μ and let $\mathcal{I}(l, p) = \{i_1, \dots, i_n\} \neq \emptyset$ for some rule $l \to r \in R$ and $p \in Pos_\Sigma(l)$ where $root(l|_p) = f$. The transformed system $\mathcal{R}^\diamond = (\Sigma^\diamond, R^\diamond)$ and $\mu^\diamond$ are defined as follows:*

- $\Sigma^\diamond = \Sigma \cup \{f_j \mid 1 \leq j \leq n\}$
- $\mu^\diamond(f_j) = \mu(f) \cup \{i_j\}$ *for all* $1 \leq j \leq n$ *and* $\mu^\diamond(g) = \mu(g)$ *for all* $g \in \Sigma$
- $R' = R - \{l \to r\} \cup \{l'_j \to r \mid 1 \leq j \leq n\} \cup \{l[x]_{p.i_j} \to l'_j[x]_{p.i_j} \mid 1 \leq j \leq n\}$

where $l'_j = l[f_j(l|_{p.1}, ..., l|_{p.m})]_p$ *if* $ar(f) = m$, x *is a fresh variable and* f_j *are new function symbols of arity* $ar(f_j) = ar(f)$.

The transformation of Definition 3.2 is iterated until arriving at a system $\mathcal{R}^\natural = (\Sigma^\natural, R^\natural)$ and $\mu^\natural$ such that $\mathcal{I}(l, p) = \emptyset$ for every rule $l \to r \in R^\natural$ and every position $p \in Pos_\Sigma(l)$.

In [Luc02b] it remains unspecified how the pair l, p is selected in one step of the transformation. However, the order in which those pairs are considered can be essential.

Example 3.3 *Consider the TRS*

$$f(g(a), a) \to a \qquad b \to f(g(c), b)$$

with a replacement map $\mu(f) = \{1\}$ *and* $\mu(g) = \emptyset$. *This system is not* $LR(\mu)$*-terminating:*

$$b^e \overset{LR}{\to}_\mu f^e(g^e(c^l), b^l) \overset{LR}{\to}_\mu f^e(g^e(c^l), b^e) \overset{LR}{\to}_\mu f^e(g^e(c^l), f^e(g^e(c^l), b^l)) \overset{LR}{\to}_\mu \dots$$

However, if we start the transformation with the first rule and position $p = \epsilon$, *and consider position* 1 *of the first rule in the second step of the transformation, then we arrive at the context-sensitive system*

$$\begin{aligned} f_2(g_1(a), a) &\to a & f(g'_1(a), x) &\to f_2(g(a), x) \\ f_2(g(x), a) &\to f_2(g_1(x), a) & f(g(x), y) &\to f(g'_1(x), y) \\ b &\to f(g(c), b) \end{aligned}$$

with $\mu(f) = \mu(g_1) = \mu(g'_1) = \{1\}$ *and* $\mu(f_2) = \{1, 2\}$. *This system is* μ*-terminating (proved with AProVE [GTSK06]). The lazy reduction sequence starting from* b *cannot be mimicked anymore, because due to the two transformation steps first the argument of* g *has to be activated which prevents the activation of the* b *in the second argument of* f.

In Lucas' transformation, positions that are dealt with last during the transformation must be activated first in rewrite sequences of the transformed system. This can be seen in Example 3.3 where $\mathcal{I}(f(g(a), a), \epsilon)$ is considered in the first step of the transformation, but position 2 must be activated after position 1.1 (whose activation is enabled by a later transformation step considering $\mathcal{I}(f(g(a), x), 1)$).

Thus, the order in which lazy positions of rules are dealt with during the transformation is the reversed order in which they may be activated in the resulting transformed system. Hence, as we want to be able to activate more outer positions before more inner ones, we consider more inner lazy positions first in our new transformation.

Despite considering more inner positions first in the transformation, we do not

want to prioritize any lazy positions. Thus, we define $\mathcal{I}(l)$ which identifies the innermost lazy positions in a term with respect to a given replacement map μ.

Definition 3.4

$$\mathcal{I}(l) = \{p \in Pos_\Sigma(l) \mid p \text{ is lazy in } label_\mu(l) \wedge \\ \wedge (\nexists q \in Pos_\Sigma(l) : q \text{ lazy in } label_\mu(l) \wedge q > p)\}$$

Since all new function symbols, which are introduced by the transformation, are substituted for function symbols of the original signature, we define the mapping *orig* from the signature of the transformed system into the original signature which identifies for each new function symbol the original one for which it was substituted.

Definition 3.5 *Let $\mathcal{R} = (\Sigma, R)$ be a TRS with replacement map μ. If in one step of the transformation $f \in \Sigma$ is replaced by a new function symbol f', then $orig(f') = f$. Furthermore, if f' is substituted for a function symbol $g \notin \Sigma$, then $orig(f') = orig(g)$. For function symbols $h \in \Sigma$, we set $orig(h) = h$ and for variables we have $orig(x) = x$.*

The actual transformation proceeds in 3 stages. First, a set of initial activation rules is created. These rules allow the activation of one *innermost* position of a left-hand side of the original rules of the lazy TRS. As already indicated, by a rule activating position $p.i$ we mean a rule $l \to r$ where l and r differ only in the function symbol at position p and $p.i$ is replacing in r but non-replacing in l.

In the second stage one rule $l \to r$ (activating a position p) created in stage 1 (or stage 2) is replaced by a set of rules, such that each lazy *innermost* position q of l may be activated by rules where p is non-replacing in both sides, and another set of rules which activate p where q is replacing in both sides (thus such a positions q must be activated before p). This construction is repeated until the rules obtained do not have any lazy (non-variable) positions. We would like to point out that as we consider *innermost* positions of terms in stage one and one step of stage two in our transformation, the outermost lazy positions of the initial rules of the lazy system are dealt with last. So these are the positions which may be activated first in reduction sequences of the transformed system.

In the third phase of the transformation for each rule of the original lazy system one active rewrite rule is created which uses the new extended signature.

Definition 3.6 *Let $\mathcal{R} = (\Sigma, R)$ be a TRS with replacement map μ. The transformed system $\widetilde{\mathcal{R}} = (\widetilde{\Sigma}, \widetilde{R})$ with $\widetilde{\mu}$ is constructed in the following three stages.*

1 **Generation of Initial Activation Rules.** *The transformed signature $\overline{\Sigma} \supseteq \Sigma$ and the set $A(l)$ for every rule $l \to r \in R$ are defined as the least sets satisfying*

$$l[x]_{p.i} \to l'[x]_{p.i} \in A(l) \text{ if } p.i \in \mathcal{I}(l) \text{ and } l' = l[f_i(l|_{p.1}, ..., l|_{p.n})]_p \qquad (1)$$
$$\wedge\ f_i \in \widetilde{\Sigma}$$
$$\wedge\ orig(g) = orig(h) \wedge \widetilde{\mu}(g) = \widetilde{\mu}(h) \Rightarrow g = h \text{ for all } g, h \in \widetilde{\Sigma}$$

where $\widetilde{\mu}$ is defined by $\widetilde{\mu}(f) = \mu(f)$ for all $f \in \Sigma$ and $\widetilde{\mu}(f_i) = \mu(orig(f_i)) \cup \{i\}$ if f_i was introduced in (1). Then we have $\widetilde{R} := \bigcup_{l \to r \in R} A(l)$.

2 **Saturation of Activation Rules.**

2.a **Processing one Activation Rule.** *Let* $\widetilde{R} = A(l_1) \cup ... \cup A(l_n)$ *and let* $l \to r \in A(l_i)$ *for some* $i \in \{1, ..., n\}$ *such that* $\mathcal{I}(l)$ *is not empty. Then we modify the set* $A(l_i)$ *in the following way*

$$A(l_i) = A(l_i) - \{l \to r\} \cup \{l[x]_{p.i} \to l'[x]_{p.i}\} \cup \{l' \to r'\}$$

for all $p.i \in \mathcal{I}(l)$ *where* $l' = l[f_i(l|_{p.1}, ..., l|_{p.n})]_p$ *and* $r' = r[f_i'(r|_{p.1}, ..., r|_{p.n})]_p$. *If there is no* $g \in \widetilde{\Sigma}$ *with* $orig(g) = orig(f_i)$ *and* $\widetilde{\mu}(g) = \widetilde{\mu}(root(l|_p)) \cup \{i\}$, *then* $\widetilde{\Sigma} = \widetilde{\Sigma} \cup \{f_i\}$ *and* $\widetilde{\mu}(f) = \widetilde{\mu}(root(l|_p)) \cup \{i\}$, *otherwise* $f_i = g$. *Analogously, if there is no* $g \in \widetilde{\Sigma}$ *with* $orig(g) = orig(f_i')$ *and* $\widetilde{\mu}(g) = \widetilde{\mu}(root(r|_p)) \cup \{i\}$, *then* $\widetilde{\Sigma} = \widetilde{\Sigma} \cup \{f_i'\}$ *and* $\widetilde{\mu}(f_i') = \widetilde{\mu}(root(r|_p)) \cup \{i\}$, *otherwise* $f_i' = g$.

$$\widetilde{R} := \bigcup_{l \to r \in R} A(l)$$

2.b **Iteration.** *Step 2.a is iterated until for all rules* $l \to r$ *of* $\widetilde{R}$ *we have that* $\mathcal{I}(l) = \emptyset$.

3 **Generation of Active Rewrite Rules.** *For each rule* $l \to r \in R$ *we add one active rewrite rule to* $\widetilde{R}$ *as follows. For every position* $p \in Pos_\Sigma(l)$, *we consider the set*

$$Symb(p, l) = \{root(r'|_p) \mid l' \to r' \in A(l) \wedge p \in Pos_\Sigma(r')\}.$$

The function symbol which is least restrictive in this set (i.e. the maximal element of $\widetilde{\mu}(f)$ *w.r.t. the subset relation of all* $f \in Symb(p, l)$*) is unique (cf. Proposition 3.9). We write* $maxSymb(p, l)$. *Then, we set*

$$\widetilde{R} := \widetilde{R} \cup \bigcup_{l \to r \in R} l'' \to r$$

where l'' *is given by* $Pos(l) = Pos(l'')$, $root(l''|_p) = maxSymb(p, l)$ *for all* $p \in Pos_\Sigma(l)$ *and* $root(l''|_p) = root(l|_p)$ *for all* $p \in Pos_V(l)$. *The signature of the transformed system is not altered in this stage.*

We have the following important properties of the transformation.

Proposition 3.7 *Let* $\mathcal{R}$ *be a TRS with replacement map* μ *and let* $\widetilde{\mathcal{R}} = (\widetilde{\Sigma}, \widetilde{R})$ *be the transformed system with replacement map* $\widetilde{\mu}$. *For* $f, g \in \widetilde{\Sigma}$

$$orig(f) = orig(g) \wedge \widetilde{\mu}(f) = \widetilde{\mu}(g) \Rightarrow f = g$$

Proposition 3.8 *The transformation of Definition 3.6 terminates and yields a finite transformed system for every TRS* $\mathcal{R}$ *and every replacement map* μ.

Proposition 3.9 *Let* $\mathcal{R}$ *be a TRS with replacement map* μ. *Let* $\widetilde{\mathcal{R}}$ *and* $\widetilde{\mu}$ *be the TRS (resp. replacement map) obtained after stages* 1 *and* 2 *of the transformation of Definition 3.6. Then the symbol* $maxSymb(p, l)$ *is unique for every rule* $l \to r \in R$ *and every* $p \in Pos_\Sigma(l)$.

Example 3.10 *Consider the TRS from Example 3.3*

$$f(g(a), a) \to a \qquad b \to f(g(c), b)$$

with a replacement map μ, s.t. $\mu(f) = \{1\}$ and $\mu(g) = \emptyset$. In the first stage of the transformation we have $\mathcal{I}(l_1) = \{1.1, 2\}$ and the following two initial activation *rules are added.*

$$f(g(x), a) \to f(g_1(x), a) \qquad f(g(a), x) \to f_2(g(a), x)$$

with $\widetilde{\mu}(g_1) = \{1\}$ and $\widetilde{\mu}(f_2) = \{1, 2\}$. In step 2.a, the first of these rules is replaced by

$$f(g(x), y) \to f_2(g(x), y) \qquad f_2(g(x), a) \to f_2(g_1(x), a)$$

and in the second iteration the second rule is replaced by

$$f(g(x), y) \to f(g_1(x), y) \qquad f(g_1(a), x) \to f_2(g_1(a), x).$$

Finally, the following active rewrite rules *are added:*

$$f_2(g_1(a), a) \to a \qquad b \to f(g(c), b).$$

Hence, the system $\widetilde{\mathcal{R}}$ is

$$\begin{array}{ll} f(g(x), y) \to f_2(g(x), y) & f_2(g(x), a) \to f_2(g_1(x), a) \\ f(g(x), y) \to f(g_1(x), y) & f(g_1(a), x) \to f_2(g_1(a), x) \\ f_2(g_1(a), a) \to a & b \to f(g(c), b) \end{array}$$

with $\widetilde{\mu}(f) = \widetilde{\mu}(g_1) = \{1\}$, $\widetilde{\mu}(f_2) = \{1, 2\}$ and $\widetilde{\mu}(g) = \emptyset$. $\widetilde{\mathcal{R}}$ is not $\widetilde{\mu}$-terminating:

$$\underline{b} \to_{\widetilde{\mu}} \underline{f(g(c), b)} \to_{\widetilde{\mu}} f_2(g(c), \underline{b}) \to_{\widetilde{\mu}} f_2(g(c), \underline{f(g(c), b)}) \to_{\widetilde{\mu}} \dots$$

The rest of the paper is concerned with the proof of soundness and completeness of the transformation of Definition 3.6 w.r.t. termination. First, we will deal with the simpler case of completeness.

Theorem 3.11 *Let $\mathcal{R} = (\Sigma, R)$ be a left-linear TRS with replacement map μ; and let $\widetilde{\mathcal{R}} = (\widetilde{\Sigma}, \widetilde{R})$, $\widetilde{\mu}$ be the transformed system (resp. replacement map) according to Definition 3.6. If $\mathcal{R}$ is $LR(\mu)$-terminating, then $\widetilde{\mathcal{R}}$ is $\widetilde{\mu}$-terminating.*

Proof. We will prove the result indirectly by showing that every infinite $\widetilde{\mathcal{R}}_{\widetilde{\mu}}$-derivation implies the existence of an infinite lazy $\mathcal{R}$-derivation. Assume there is an infinite $\widetilde{\mathcal{R}}_{\widetilde{\mu}}$-sequence starting from a term t. Then we construct an infinite lazy reduction sequence starting from the labelled term t' defined by

$$Pos(t) = Pos(t') \wedge \forall p \in Pos(t) : (orig(root(t|_p)) = root(erase(t'|_p)) \wedge t'|_p \text{ is eager iff } label_{\widetilde{\mu}}(t)|_p \text{ is eager}).$$

In this case we write $t' \approx t$. Note that t' is labelled canonically or more liberally as $\mu(orig(f)) \subseteq \mu(f)$ for all $f \in \widetilde{\Sigma}$. Now consider a $\widetilde{\mu}$-step $t \to_{\widetilde{\mu}} s$ and a labelled term t' with $t' \approx t$. We will prove that there is a labelled term s', such that $t' \overset{LR}{\to}_{\mu} s'$ and $s' \approx s$. We make a case distinction on the type of $\widetilde{\mu}$-step.

(i) First assume the step is an activation step. Then there is an activation rule $l' \to l''$ in $\widetilde{R}$ which can be applied to t. This activation rule stems from a rule

$l \to r \in R$, and we have that $orig(root(l'|_p)) = root(l|_p)$ for all non-variable positions p of l'. Furthermore, all variable positions of l' which are non-variable in l are lazy in $label_{\widetilde{\mu}}(l')$ and thus in t'. Hence, l matches modulo laziness t' and the same position as in t can be activated yielding s' with $s' \approx s$ (note that the active positions of t' are exactly the replacing positions of t).

(ii) If the step $t \to_{\widetilde{\mu}} s$ is an active rewrite step, a rule $l' \to r$ matches (a subterm of) t. This rule is the transformed version of a rule $l \to r \in R$ with $orig(root(l'|_p)) = root(l|_p)$ for all $p \in Pos(l) = Pos(l')$. Thus, l matches $erase(t')$ and the rule can be applied to t' yielding s' with $s' \approx s$. The reason is that $orig(root(s'|_p)) = root(s|_p)$ for all position of s (note that the right hand-sides of the rules applied to t and t' are identical). Regarding the labels of s' assume that the rewrite steps were performed at a position q (in t and t'). For all positions $o \in Pos(t)$ with $o||q \vee o < q$ we have $s'|_o$ is eager if and only if $label_{\widetilde{\mu}}(s)|_o$ is eager because this has already been the case in t' and t. Furthermore, positions $q.o$ where $o \in Pos(r)$ are eager in s' if and only if they are eager in $label_{\widetilde{\mu}}(s)$ because of the canonical labelling of r inside s'. Finally, positions $p.o$ where $o \notin Pos(r)$ are eager in s' if and only if they are eager in $label_{\widetilde{\mu}}(s)$, because a proper superterm of each term $s'|_{p.o}$ occurred already in t' and thus, if an eager position of s' had not been eager in $label_{\widetilde{\mu}}(s)$ (or vice versa), then this would be a contradiction to $t' \approx t$. □

The soundness proof is similar to the completeness proof in the sense that given an infinite lazy reduction sequence, we are going to construct a corresponding infinite reduction sequence in the transformed system. So assume there is an infinite lazy reduction sequence in a TRS $\mathcal{R}$ with replacement map μ. The first observation is that every lazy reduction sequence naturally corresponds to a context-free $\to_{\mathcal{R}}$ sequence, which performs the active rewrite steps of the lazy reduction sequence.

We will construct a $\to_{\widetilde{\mu}}$-reduction sequence in the transformed system $\widetilde{\mathcal{R}}$, that corresponds to a context-free $\to_{\mathcal{R}}$-sequence. Terms in the context-free $\to_{\mathcal{R}}$-sequence and terms in the corresponding $\widetilde{\mu}$-sequence are in a special relationship.

Definition 3.12 *Let $\mathcal{R} = (\Sigma, R)$ be a TRS, μ a replacement map and let $s, t \in \mathcal{T}(\Sigma, V)$ be two terms. Abusing notation we write $s \to^*_{\mu^c} t$ if and only if*

(i) for all positions $p \in Pos^\mu(t)$ we have $root(t|_p) = root(s|_p)$, and

(ii) for all minimal positions $q \in Pos(t) \setminus Pos^\mu(t)$ we have $s|_q \to^*_\mu s'$ and $s' \to^*_{\mu^c} t|_q$.

The idea behind $\to^*_{\mu^c}$ is that context-free reduction steps which occur at positions that are in the replacing part of the simulating term should be simulated, thus the replacing parts of two terms s and t with $s \to^*_{\mu^c} t$ must be entirely equal. On the other hand, context-free steps that occur at positions which are forbidden in the simulating term are ignored. Yet, if the forbidden subterm in which they occur eventually gets activated, then these steps may still be simulated.

As minimal non-replacing positions in a term are always strictly below the root, the recursive description of $\to^*_{\mu^c}$ in Definition 3.12 is well-defined.

We have $s = t \Rightarrow s \to^*_{\mu^c} t$. Figure 1 illustrates the correspondence between a lazy

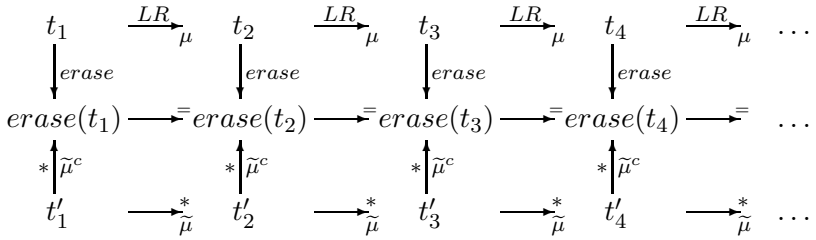

Fig. 1. Relation between the various rewrite sequences occurring in the soundness proof.

reduction sequence, the corresponding context-free one, and the $\to_{\widetilde{\mu}}$-sequence. Note that if the lazy reduction sequence is infinite, then there are infinitely many non-empty steps in the context-free reduction sequence, as every labelled term admits only finitely many activation steps.

In the first part of the soundness proof we show the existence of a $\to_{\widetilde{\mu}}$-sequence of the shape as in Figure 1.

The next lemma provides a criterion for the existence of an activation rule in the transformed system, that is able to activate a certain position in a term t over the new signature.

Lemma 3.13 *Let $(\mathcal{R} = (\Sigma, R), \mu)$ be a TRS with replacement map and let $(\widetilde{\mathcal{R}} = (\widetilde{\Sigma}, \widetilde{R}), \widetilde{\mu})$ be the system obtained by the transformation of Definition 3.6. Let $t \in \mathcal{T}(\widetilde{\Sigma}, V)$ be a term and $\alpha\colon l \to r \in R$ a rewrite rule of the original TRS, such that the following preconditions hold.*

(i) For all replacing *positions p in t with $p \in Pos_\Sigma(l)$: $orig(root(t|_p)) = root(l|_p)$.*

(ii) For all positions $p.i$ that are variable positions in l we have that $t|_q \in \mathcal{T}(\Sigma, V)$ for some $q \leq p$.

Then, every position q, which is minimal non-replacing in t and non-variable in l, can be activated (i.e. we have $t \to_{\widetilde{\mu}} t'$ such that q is $\widetilde{\mu}$-replacing in t' and $orig(root(t|_p)) = orig(root(t'|_p))$ for all $p \in Pos(t)$).

The next lemma establishes the relationship between a context-free reduction sequence and a corresponding $\to_{\widetilde{\mu}}$ reduction of Figure 1.

Lemma 3.14 *Let $(\mathcal{R} = (\Sigma, R), \mu)$ be a TRS with replacement map and let $(\widetilde{\mathcal{R}} = (\widetilde{\Sigma}, \widetilde{R}), \widetilde{\mu})$ be the system obtained by the transformation of Definition 3.6. Let s and t be terms from $\mathcal{T}(\Sigma, V)$, such that $s \to^*_{\widetilde{\mu}^c} t$. If $t \xrightarrow{p} t^*$ (with a rule $l \to r$) and $p \in Pos^{\widetilde{\mu}}(s)$, then $s \to^+_{\widetilde{\mu}} s^*$ and $s^* \to^*_{\widetilde{\mu}^c} t^*$. Otherwise, if $t \xrightarrow{p} t^*$ and $p \notin Pos^{\widetilde{\mu}}(s)$, then $s \to^*_{\widetilde{\mu}} s^*$ and $s^* \to^*_{\widetilde{\mu}^c} t^*$.*

Unfortunately, the last lemma and the correspondence of lazy, context-free and $\to_{\widetilde{\mu}}$-reduction sequences of Figure 1 is not sufficient to prove the existence of an infinite $\to_{\widetilde{\mu}}$-sequence in the presence of an infinite lazy reduction sequence, since there may be only finitely many non-empty $\to_{\widetilde{\mu}}$-reductions in the simulating sequence.

Example 3.15 *Consider the TRS $\mathcal{R}$*

$$a \to f(a) \qquad f(b) \to b$$

with a replacement map $\mu(f) = \emptyset$. *The transformed system* $\widetilde{\mathcal{R}}$ *is*

$$a \to f(a) \qquad f(x) \to f_1(x) \qquad f_1(b) \to b$$

with $\widetilde{\mu}(f) = \emptyset$, $\widetilde{\mu}(f_1) = \{1\}$. *We have the following lazy reduction sequence*

$$\underline{a^e} \overset{LR}{\to}_\mu f^e(\underline{a^l}) \overset{LR}{\to}_\mu f^e(\underline{a^e}) \overset{LR}{\to}_\mu \dots$$

which corresponds to the context-free sequence

$$a \to f(a) \to f(f(a)) \to \dots.$$

Consider a corresponding sequence in the system $\widetilde{\mathcal{R}}$,

$$a \to_{\widetilde{\mu}} f(a).$$

Then we could activate a *in* $f(a)$ *according to rule* 2 *of the transformed system. However, it is a priori not clear whether such a step should be performed when trying to simulate an infinite reduction sequence. The following example illustrates the potential problems.*

Example 3.16 *Consider the non-terminating TRS* $\mathcal{R}$

$$f(g(x)) \to f(g(x)) \qquad g(a) \to g(b) \qquad a \to c$$

with a replacement map $\mu(f) = \{1\}$ *and* $\mu(g) = \emptyset$. *The transformed system* $\widetilde{\mathcal{R}}$ *is*

$$f(g(x)) \to f(g(x)) \quad g(x) \to g_1(x)$$
$$g_1(a) \to g(b) \qquad a \to c$$

with $\widetilde{\mu}(f) = \widetilde{\mu}(g_1) = \{1\}$ *and* $\widetilde{\mu}(g) = \emptyset$. *Consider the following context-free reduction sequence.*

$$f(g(a)) \to f(g(c)) \to f(g(c)) \to \dots$$

If we activate position 1.1 *in* $f(g(a))$ *in the simulating* $\to_{\widetilde{\mu}}$*-sequence, we cannot further simulate the sequence, i.e. we get*

$$f(g(a)) \to_{\widetilde{\mu}} f(g_1(a)) \to_{\widetilde{\mu}} f(g_1(c)),$$

but the term $f(g_1(c))$ *is a* $\to_{\widetilde{\mu}}$*-normal form.*

The crucial difference why the activation of a subterm is essential in Example 3.15 and unnecessary in Example 3.16 is that in the former example the activated subterm itself initiates an infinite lazy reduction sequence. This observation will be used in the second part of the soundness proof (cf. Theorem 3.22).

With the following definition we intend to identify labelled terms in an infinite lazy reduction sequence with non-terminating subterms that have possibly been activated. For such terms t, the predicate $mininf(t)$ does not hold.

Definition 3.17 *Let* Σ *be a signature and* μ *be a replacement map for* Σ. *A labelled term* t *is said to be* minimal non-terminating *if it admits an infinite lazy*

reduction sequence and for each eager labelled proper subterm $t|_p$ of t, either $t|_p$ does not initiate an infinite lazy rewrite sequence, or position p is eager in the term $label_\mu(erase(t))$. We write $mininf(t)$ if t has this property.

Definition 3.18 *Let $\mathcal{R} = (\Sigma, R)$ be a TRS with replacement map μ. Let t be a labelled term t and $t \overset{LR}{\to}_\mu s$ be an activation step. This activation step is called* inf-activating *(thus it is an* inf-activation step*) if and only if $mininf(t)$ but* not *$mininf(s)$.*

It is easy to see that whenever $mininf(t)$ holds for a labelled term t, there is no active position $p \in Act(t)$ which is non-active in $label_\mu(erase(t))$, such that $t|_p$ initiates an infinite lazy reduction sequence.

In the second part of the soundness proof we will show that each infinite lazy reduction sequence contains either an inf-activation step or an active rewrite step $s \overset{LR}{\to}_\mu t$ at position p, such that position p is μ-replacing in $erase(s)$. Furthermore, such steps result in non-empty simulations by the $\to_{\widetilde{\mu}}$-sequence.

Lemma 3.19 *Let $\mathcal{R} = (\Sigma, R)$ be a TRS with replacement map μ. Let t be a labelled term satisfying $mininf(t)$. Then we have:*

(i) If $t \overset{LR}{\to}_\mu s$ with an inf-activation step at position q_1 activating position q_2 and $q_1 < p \leq q_2$ is the maximal (w.r.t. $\leq$) eager position in s which does initiate an infinite reduction sequence s.t. $t|_p$ does not, then we have $mininf(s|_p)$.

(ii) If $t \overset{LR}{\to}_\mu s$ with any other step than in (i) (i.e. activation steps which are not inf-activating, or active rewrite steps), then $mininf(s)$.

Lemma 3.20 *Let $\mathcal{R} = (\Sigma, R)$ be a TRS with a replacement map μ. Let $t \in \mathcal{T}(\Sigma, V)$ be an unlabelled term. If t initiates an infinite context-free reduction sequence with infinitely many root reduction steps, then a labelled term t' initiates an infinite lazy reduction sequence if $erase(t') = t$ and t' has an eager root label.*

The next lemma characterizes infinite lazy reduction sequences starting from minimal non-terminating labelled terms. It states that in such an infinite lazy reduction sequence after finitely many steps there is either an active rewrite step $s_i \overset{LR}{\to}_\mu s_{i+1}$ at some position p which is active in $label_\mu(erase(s_i))$ or there is an inf-activation step. We already proved in Lemma 3.14 that active rewrite steps at such positions can be simulated by a non-empty sequence in the transformed system (remember that the active rewrite steps of a lazy reduction sequence correspond to a context-free derivation). In Theorem 3.22 we will prove that the same is true for inf-activation steps.

Lemma 3.21 *Let $\mathcal{R} = (\Sigma, R)$ be a TRS with a replacement map μ. Let t_0 be a labelled term with the property $mininf(t_0)$. Let $P : t_0 \overset{LR}{\to}_\mu t_1 \overset{LR}{\to}_\mu \ldots \overset{LR}{\to}_\mu t_n \overset{LR}{\to}_\mu \ldots$ be an infinite lazy reduction sequence starting from t_0. Then, either there is an active rewrite step $t_i \overset{LR}{\to}_\mu t_{i+1}$ at position p, where p is active in $label_\mu(erase(t_i))$, or there is an* inf-activation *step in P.*

Theorem 3.22 *Let $(\mathcal{R} = (\Sigma, R), \mu)$ be a left-linear TRS with replacement map and*

let $(\widetilde{\mathcal{R}} = (\widetilde{\Sigma}, \widetilde{R}), \widetilde{\mu})$ *the system obtained by the transformation of Definition 3.6. If* $\widetilde{\mathcal{R}}$ *is* $\widetilde{\mu}$*-terminating, then* $\mathcal{R}$ *is* $LR(\mu)$*-terminating.*

Proof. We will show that the existence of an infinite lazy reduction sequence $P : t_0 \overset{LR}{\to}_\mu t_1 \overset{LR}{\to}_\mu \dots$ (where t_0 is canonically or more liberally labelled) implies the existence of an infinite reduction sequence in the transformed system. The following invariant will be maintained for every labelled term t_i on an infinite reduction sequence P. Let $s_0 \to_{\widetilde{\mu}} s_1 \to_{\widetilde{\mu}} \dots$ be the simulating reduction sequence we are going to construct:

There is a s_j such that

$$s_j|_o \to^*_{\widetilde{\mu}^c} erase(t_i|_o) \wedge mininf(t_i|_o)$$

and position o is $\widetilde{\mu}$-replacing in s_j and active in t_i. Furthermore, $t_i|_o$ is at least as eager as its canonically labelled version (i.e. whenever $label_\mu(erase(t_i|_o))$ has an eager label at some position q, then the label of $t_i|_o$ is eager at that position, too). Note that the latter condition is trivially fullfilled by all terms t_i in P, and thus by all subterms as no "deactivations" are possible in lazy rewriting and active rewrite steps only introduce canonically labelled terms.

We show that a finite subsequence of P implies the existence of a non-empty reduction sequence in the transformed system which preserves the invariant. As each term $t_i|_o$ itself initiates an infinite lazy reduction sequence this suffices to show that there is an infinite reduction sequence in the transformed system.

In order to apply Lemma 3.21 we assume $mininf(t_0)$. This minimality constraint can be satisfied, as w.l.o.g. we can find a t_0 such that *each* proper subterm of t_0 with an eager label does not initiate an infinite lazy reduction sequence.

The infinite reduction sequence we are going to construct in the transformed system starts with the term $s_0 = erase(t_0)$. We have $s_0 \to^*_{\widetilde{\mu}^c} erase(t_0)$.

Lemma 3.21 states that in the lazy reduction sequence starting from t_0 there is either an active rewrite step $t_i \overset{LR}{\to}_\mu t_{i+1}$ at position p such that p is active in $label_\mu(erase(t_i))$, or there is an inf-activation step. Let $t_j \overset{LR}{\to}_\mu t_{j+1}$ be the first step, which is of one of the two kinds.

The goal is to show that the reduction sequence $t_0 \overset{LR}{\to}^*_\mu t_{j+1}$ can be simulated by a sequence $s_0 \to^+_{\widetilde{\mu}} s_i$ such that $s_i|_o \to^*_{\widetilde{\mu}^c} erase(t_{j+1|_o})$ and $mininf(t_{j+1}|_o)$ holds for some position o which is active in t_{j+1} and replacing in s_i. We make a case distinction on whether the step $t_j \overset{LR}{\to}_\mu t_{j+1}$ is an active rewrite step or an *inf-activation* step.

(i) Assume the step $t_j \overset{LR}{\to}_\mu t_{j+1}$ is an active rewrite step at position p such that p is active in $label_\mu(erase(t_j))$. We have (according to Lemma 3.14) $s_0 \to^*_{\widetilde{\mu}} s_i$ and $s_i \to^*_{\widetilde{\mu}^c} erase(t_j)$. If position p is active in $label_\mu(erase(t_j))$ then p is replacing in s_i (note that $s_i \in \mathcal{T}(\Sigma, V)$). Thus, with Lemma 3.14 we have $s_0 \to^*_{\widetilde{\mu}} s_i \to^+_{\widetilde{\mu}} s_{i+1}$ and $s_{i+1} \to^*_{\widetilde{\mu}^c} erase(t_{j+1})$. Furthermore, we have $mininf(t_{j+1})$ according to Lemma 3.19.

(ii) Assume the step $t_j \overset{LR}{\to}_\mu t_{j+1}$ is an inf-activation step. Again by Lemma 3.14

we have $s_0 \to_{\widetilde{\mu}}^* s_i$ and $s_i \to_{\widetilde{\mu}^c}^* erase(t_j)$ ($s_i \in \mathcal{T}(\Sigma, V)$). The matching modulo laziness of a rule with t_j was established at a position q_{inf} which is active in $label_\mu(erase(t_j))$. The reason is that otherwise in t_j there were a non-terminating active subterm which is non-active in $label_\mu(erase(t_j))$. Thus, $mininf(t_j)$ would not hold, which contradicts Lemma 3.19.

The fact that the activation step from t_j to t_{j+1} is inf-activating implies that there is a unique maximal active subterm $t_{j+1}|_p$ of t_{j+1} which is non-active in $label_\mu(erase(t_{j+1}))$ and initiates an infinite lazy reduction sequence. For this position p we have $p \leq q$ where q is the position that is activated in the inf-activation step: If we had $p > q$ or $p||q$, then $t_j|_p = t_{j+1}|_p$. Furthermore, as $label_\mu(erase(t_j)) = label_\mu(erase(t_{j+1}))$, this would contradict $mininf(t_j)$.

Note that, since the position q_{inf} where the matching modulo laziness was established in t_j is active in $label_\mu(erase(t_j))$, we have that $q_{inf} < p \leq q$.

In the simulating sequence we will activate position p in the term s_i. We note that p is non-replacing in s_i (as it is non-active in $label_\mu(erase(t_j))$), but it is not necessarily minimal non-replacing. Thus, in order to activate position p in s_i, we possibly need to activate positions $o < p$ in s_i first.

Let $o < p$ be the minimal non-replacing position in s_i. According to Lemma 3.13 we can activate o in s_i yielding s_i'. Note that as t_j is at least as eager as $label_\mu(erase(t_j))$, we have $orig(root(s_i|_{q_{inf}.q'})) = root(erase(t_j|_{q_{inf}.q'}))$ for every replacing position $q_{inf}.q'$ of s_i and $root(erase(t_j|_{q_{inf}.q'})) = l|_{q'}$ for some rule $l \to r$. Then, as $s_i \to_{\widetilde{\mu}^c}^* erase(t_j)$, we have $s_i' \overset{\geq o}{\to}_{\widetilde{\mu}}^* s_i''$ such that $s_i''|_o \to_{\widetilde{\mu}^c}^* erase(t_j|_o)$ according to Definition 3.12. Position o is replacing in s_i''. If there is still a non-replacing position $o' < p$ in s_i'', it is again activated and s_i'' is reduced to a term s_i''' such that $s_i'''|_{o'} \to_{\widetilde{\mu}^c}^* erase(t_j|_{o'})$. This construction is repeated until position p is replacing in a term s_i^* and we have $s_i^*|_p \to_{\widetilde{\mu}^c}^* erase(t_j|_p)$.

Note that $s_i^*|_p$ is a term over the original signature Σ, so it does not contain any function symbols introduced by the transformation. Clearly, we have that $s_i^*|_p \to_{\widetilde{\mu}^c}^* erase(t_{j+1}|_p)$, since $erase(t_j) = erase(t_{j+1})$. Finally, according to Lemma 3.19 we have $mininf(t_{j+1}|_p)$.

Now given an infinite lazy reduction sequence P starting from a labelled term t_0 and a term s_0 with $s_0 \to_{\widetilde{\mu}^c}^* erase(t_0)$, we have shown that a *finite* subsequence $t_0 \overset{LR}{\to}_\mu^+ t_i$ of a special shape implies the existence of a nonempty sequence $s_0 \to_{\widetilde{\mu}}^+ s_j$ such that $s_j|_p \to_{\widetilde{\mu}^c}^* erase(t_i|_p)$, where p is active in t_i and replacing in s_j, $mininf(t_i|_p)$ holds and $t_i|_p$ initiates an infinite lazy reduction sequence. Thus, again the infinite lazy sequence starting at $t_i|_p$ has a finite subsequence, that can be simulated by a non-empty reduction sequence in the transformed system. By repeating this construction, we get an infinite reduction sequence in the transformed system starting at s_0. □

4 Discussion

In the proof of Theorem 3.22 we saw that the CSRS obtained by our transformation cannot simulate lazy reduction sequences in a one-to-one fashion. When simulating

infinite lazy reduction sequences, after every inf-activation step in the lazy reduction an entirely new infinite lazy sequence was considered, namely the one initiated by the activated subterm. Thus, the question arises whether we can define a transformation from lazy rewrite systems into context-sensitive ones, such that the transformed system is able to fully simulate the lazy reduction system. We conjecture that this is indeed possible ([Sch07]).

Regarding the size of the transformed system, we have that the number of rules created by our transformation is in general exponentially higher than the number of lazy non-variable subterms in left-hand sides of rules of the lazy system. This leads to the question whether our transformation is practical in proofs of termination of real world lazy TRSs. Currently, it is unclear if termination of the resulting systems can easily be inferred by automated termination tools. However, this would be most desirable in practice. Experiments to answer this question have yet to be performed.

References

[AEGL03] M. Alpuente, S. Escobar, B. Gramlich, and S. Lucas. On-demand strategy annotations revisited. Technical Report DSIC–II/18/03, UPV, Valencia, Spain, July 2003.

[BN98] F. Baader and T. Nipkow. *Term rewriting and all that.* Cambridge University Press, New York, NY, USA, 1998.

[CDE+03] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and C. Talcott. The Maude 2.0 system. In R. Nieuwenhuis, ed., *Proc. RTA'03*, LNCS 2706, pp. 76–87. Springer, 2003.

[FKW00] W. Fokkink, J. Kamperman, and P. Walters. Lazy rewriting on eager machinery. *ACM Trans. Program. Lang. Syst.*, 22(1):45–86, 2000.

[FW76] D. P. Friedman and D. S. Wise. CONS should not evaluate its arguments. In S. Michaelson and R. Milner, eds., *Proc. 3rd ICALP*, pp. 257–284. Edinburgh University Press, 1976.

[GL06] B. Gramlich and S. Lucas. Generalizing Newman's Lemma for left-linear rewrite systems. In F. Pfenning, ed., *Proc. RTA'06,*, LNCS 4098, pp. 66–80. Springer, 2006.

[GTSK06] J. Giesl, R. Thiemann, and P. Schneider-Kamp. AProVE 1.2: Automatic termination proofs in the dependency pair framework. In Ulrich Furbach and Natarajan Shankar eds., *Proc. IJCAR'06,* LNCS 4130, pp. 281–286, 2006.

[HMJ76] P. Henderson and J. H. Morris Jr. A lazy evaluator. In *Conference Record of the Third ACM Symp. on Principles of Programming Languages, Atlanta, Georgia, Jan. 1976*, pp. 95–103, 1976.

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

[Luc01] S. Lucas. Termination of on-demand rewriting and termination of OBJ programs. In H. Sondergaard, ed., *Proc. PPDP'01*, pp. 82–93, September 2001. ACM Press, New York.

[Luc02a] S. Lucas. Context-sensitive rewriting strategies. *Inform. and Comput.*, 178(1):294–343, 2002.

[Luc02b] S. Lucas. Lazy rewriting and context-sensitive rewriting. *In M. Hanus, ed., Proc. WFLP'01,* ENTCS 64, Elsevier, 2002.

[Luc06] S. Lucas. Proving termination of context-sensitive rewriting by transformation. *Information and Computation*, 204(1):1782–1846, January 2006.

[Ngu01] Q. H. Nguyen. Compact normalisation trace via lazy rewriting. *Proc. WRS'01,* ENTCS 57, 2001.

[Pv93] M. J. Plasmeijer and M. C. J. D. van Eekelen. *Functional Programming and Parallel Graph Rewriting.* Addison-Wesley, 1993.

[Str89] R. Strandh. Classes of equational programs that compile into efficient machine code. In N. Dershowitz, ed., *Proc. RTA'89*, LNCS 355, pp. 449–461. Springer, April 1989.

[Sch07] F. Schernhammer. On context-sensitive term rewriting. Master's thesis, Vienna University of Technology, Feb. 2007.