Transforming Outermost into Context-Sensitive Rewriting

Endrullis, J.; Hendriks, R.D.A.

published in
Logical Methods in Computer Science
2010

DOI (link to publisher)
10.2168/lmcs-6(2:5)2010

document version
Publisher's PDF, also known as Version of record

Logical Methods in Computer Science
Vol. 6 (2:??) 2010, pp. 1–39
www.lmcs-online.org

Submitted Oct. 29, 2009
Published Jun. 29, 2010

TRANSFORMING OUTERMOST INTO CONTEXT-SENSITIVE REWRITING *

JÖRG ENDRULLIS AND DIMITRI HENDRIKS

Vrije Universiteit Amsterdam, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
e-mail address: joerg@few.vu.nl, diem@cs.vu.nl

Abstract. We define two transformations from term rewriting systems (TRSs) to context-sensitive TRSs in such a way that termination of the target system implies outermost termination of the original system. In the transformation based on 'context extension', each outermost rewrite step is modeled by exactly one step in the transformed system. This transformation turns out to be complete for the class of left-linear TRSs. The second transformation is called 'dynamic labeling' and results in smaller sized context-sensitive TRSs. Here each modeled step is adjoined with a small number of auxiliary steps.

As a result state-of-the-art termination methods for context-sensitive rewriting become available for proving termination of outermost rewriting. Both transformations have been implemented in Jambox, making it the most successful tool in the category of outermost rewriting of the annual termination competition of 2008.

1. Introduction

Termination is a key aspect of program correctness, and therefore a widely studied subject in term rewriting and program verification. While termination is undecidable in general, various automated techniques have been developed for proving termination. One of the most powerful techniques is the method of dependency pairs [AG00]. In [AGL06] dependency pairs for context-sensitive rewriting have been introduced, and in [AEF+08] the dependency pair framework [GTSK04, Thi07, GL10] has been extended to context-sensitive rewriting, thereby extending the class of context-sensitive TRSs for which termination can be shown automatically. Context-sensitive rewriting [Luc98] is a restriction on term rewriting where rewriting in some fixed arguments of function symbols is disallowed. It offers a flexible paradigm to analyze properties of rewrite strategies, in particular of (lazy) evaluation strategies employed by functional programming languages.

In this paper context-sensitive rewriting is the target formalism for a transformational approach to the problem of outermost termination, that is, termination with respect to outermost rewriting. Outermost rewriting is a rewriting strategy where a redex may be contracted as long as it is not a proper subterm of another redex occurrence. The main

1998 ACM Subject Classification: D.1.1, D.3.1, F.4.1, F.4.2, I.1.1, I.1.3.
Key words and phrases: term rewriting, termination, outermost rewriting, context-sensitive rewriting.
* This is a modified and extended version of [EH09] which appeared in the proceedings of RTA 2009.

LOGICAL METHODS IN COMPUTER SCIENCE
DOI:10.2168/LMCS-6 (2:??) 2010

reason for studying outermost termination is its practical relevance: lazy functional programming languages like Miranda [Tur86], Haskell [PJ03] or Clean [PvE01], are based on outermost rewriting as an evaluation strategy, and in implementations of rewrite logic such as Maude [CELM96] and CafeOBJ [FN97], outermost rewriting is an optional strategy.

To illustrate outermost rewriting, and the transformations we propose, we consider the term rewriting system R_0 consisting of the following rules:

$$\mathsf{a} \to \mathsf{f}(\mathsf{a}) \qquad\qquad \mathsf{f}(\mathsf{f}(x)) \to \mathsf{b} \tag{R_0}$$

Clearly, this system is not terminating, as witnessed by the infinite rewrite sequence:

$$\mathsf{a} \to \mathsf{f}(\mathsf{a}) \to \mathsf{f}(\mathsf{f}(\mathsf{a})) \to \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))) \to \dots$$

However, R_0 is outermost terminating. Indeed, the third step in the rewrite sequence above is not an outermost step, since the contraction takes place inside another redex. The only (maximal) outermost rewrite sequence the term a admits is:

$$\mathsf{a} \overset{out}{\to} \mathsf{f}(\mathsf{a}) \overset{out}{\to} \mathsf{f}(\mathsf{f}(\mathsf{a})) \overset{out}{\to} \mathsf{b} \tag{1.1}$$

The contribution of the present paper consists of two transformations of arbitrary TRSs into context-sensitive TRSs (henceforth also called 'μTRSs') in such a way that rewriting in the μTRS corresponds to outermost rewriting in the original TRS. As a result, advanced termination techniques for μTRSs become available for proving outermost termination. Automated termination provers for μTRSs can directly (without modification, only preprocessing) be used for proving outermost termination. One of the transformations turns out to be complete for the class of quasi-left-linear TRSs, a generalized form of left-linear TRSs, see [RZ09]. In other words, termination of the resulting μTRS is equivalent to outermost termination of the original system.

The transformations are comprised of a variant of semantic labeling [Zan95]. In semantic labeling the function symbols in a term are labeled by the interpretation of their arguments (or a label depending on these values) according to some given semantics. We employ semantic labeling to mark symbols at redex positions, and then obtain a μTRS by defining a replacement map that disallows rewriting inside arguments of marked symbols.

We illustrate our use of semantic labeling by means of the TRS R_0 given above. We choose an algebra with values 0 and 1, indicating the presence of the symbol f:

$$\mathcal{A}_0 = \langle \{0,1\}, [\![\cdot]\!] \rangle \qquad [\![\mathsf{a}]\!] = [\![\mathsf{b}]\!] = 0 \qquad [\![\mathsf{f}]\!](x) = 1 \quad \text{for } x \in \{0,1\} \tag{$\mathcal{A}_0$}$$

We write $\mathsf{f}^\star$, and say that 'f is marked', if the value of its argument is 1, and just f if the value is 0. The symbol a is a redex, and hence it is always marked, while b never is. If f is marked it corresponds to a redex position with respect to the rule $\mathsf{f}(\mathsf{f}(x)) \to \mathsf{b}$. For example the term $\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a})))$ is labeled as $\mathsf{f}^\star(\mathsf{f}^\star(\mathsf{f}(\mathsf{a}^\star)))$. We obtain a μTRS by forbidding rewriting inside the argument of the symbol $\mathsf{f}^\star$. Since $\mathsf{a}^\star$ is a constant, there is nothing to be forbidden. Hence for correctly labeled terms, rewriting inside redex occurrences is disallowed, and this corresponds to the strategy of outermost rewriting.

In order to rewrite labeled terms we have to label the rules of a TRS. Simply labeling both sides of a rule does not always work. For example, when we label the rules of R_0 using the algebra $\mathcal{A}_0$, we obtain the following μTRS:

$$\mathsf{a}^\star \to \mathsf{f}(\mathsf{a}^\star) \qquad\qquad \mathsf{f}^\star(\mathsf{f}(x)) \to \mathsf{b} \qquad\qquad \mathsf{f}^\star(\mathsf{f}^\star(x)) \to \mathsf{b} \tag{$\overline{R}_0$}$$

This system has two instances of the second rule, one for each possible value assigned to the variable x. Now, despite of the fact that the original TRS is outermost terminating,

the labeled system $\overline{R}_0$ still admits an infinite rewrite sequence:

$$\mathsf{a}^\star \to_\mu \mathsf{f}(\mathsf{a}^\star) \to_\mu \mathsf{f}(\mathsf{f}(\mathsf{a}^\star)) \to_\mu \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}^\star))) \to_\mu \dots \tag{1.2}$$

The reason is that the term $\mathsf{f}(\mathsf{f}(\mathsf{a}^\star))$ is not correctly labeled, as its root symbol f should have been marked. In [Zan95] this problem is avoided by allowing labeling only with *models*. Roughly speaking, an algebra is a model for a TRS R if left and right-hand sides of all rewrite rules of R have equal interpretations. However, this requirement is too strict for the purpose of marking redexes, because contraction of a redex at a position p may create a redex above p in the term tree, as exemplified by (1.2). In fact, for R_0 there exists *no* model which is able to distinguish between redex and non-redex positions. Let us explain. The rewrite step $\mathsf{f}(\mathsf{a}) \to \mathsf{f}(\mathsf{f}(\mathsf{a}))$ creates a redex at the top. The term $\mathsf{f}(\mathsf{a})$ is not a redex, and therefore its root symbol f should not be marked. On the other hand $\mathsf{f}(\mathsf{f}(\mathsf{a}))$ is a redex and so the outermost f has to be marked. The change of the labeling of a context (here $\mathsf{f}(\Box)$) implies that the interpretation of its arguments a and $\mathsf{f}(\mathsf{a})$ cannot be the same. Therefore we cannot require the rule $\mathsf{a} \to \mathsf{f}(\mathsf{a})$ to preserve the interpretation.

To that end, we generalize this notion of model and relax the condition $[\![\ell]\!] = [\![r]\!]$ to:

$$\exists n.\, [\![C[\ell]]\!] = [\![C[r]]\!] \quad \text{for all contexts } C \text{ of depth} \geq n \tag{1.3}$$

Thus rules are allowed to change the interpretation as long as the effect is limited to contexts of a bounded depth. We call this depth *the C-depth of* $\ell \to r$ and denote it by $\delta_{\mathcal{A}}(\ell \to r)$. As it turns out, algebras satisfying this weaker requirement (1.3), are strong enough to recognize redex positions. Such algebras we will call *C-models*.

The algebra $\mathcal{A}_0$ given above is a C-model for the TRS R_0. As opposed to models, for C-models it is no longer sufficient to simply label the rules. This is demonstrated by the rewrite sequence (1.2) in the μTRS $\overline{R}_0$: an application of the rule $\mathsf{a}^\star \to \mathsf{f}(\mathsf{a}^\star)$ in the term $\mathsf{f}(\mathsf{a}^\star)$ creates the incorrectly labeled term $\mathsf{f}(\mathsf{f}(\mathsf{a}^\star))$.

Therefore, in order to preserve correct labeling, labels in the context of the original rewrite step sometimes have to be updated. We present two solutions to this problem: the transformation of *context extension* and the transformation of *dynamic labeling*.

In the transformation based on context extension [EH09], worked out in Section 5, the update of semantic labels is established by prefixing appropriate contexts to both sides of a rewrite rule. The depth of these contexts is bounded by the C-depth of the rule. Thus, the update of the labels is coded within the context, and no additional rewrite steps are needed. As a result of that, every outermost rewrite step in the original system is modeled by exactly one rewrite step in the transformed μTRS. A disadvantage of the transformation, however, is that the resulting μTRS can have a large number of rules arising from the prepending of contexts in combination with semantic labeling.

An alternative solution (and new with respect to [EH09]) is dynamic labeling, described in Section 6: instead of extending rules with contexts we now use rewriting to propagate the changed information upward in the term tree. With respect to context extension, this approach results in a smaller number of rules of the transformed system. On the other hand, the property of the context extension of a one–to–one correspondence of the rewrite steps, is now weakened to a one–to–m correspondence where $m \leq 1 + \delta_{\mathcal{A}}(\ell \to r)$. This means that an outermost rewrite step is modeled by one step in the transformed system plus a number of auxiliary steps necessary for updating the labels, and this number is bounded by the C-depth of the corresponding rule. In most practical cases this value is typically small (≤ 2). This is shown in Section 10 where we evaluate the implemented transformations.

We illustrate the two transformations by means of our running example, the TRS R_0 together with the algebra $\mathcal{A}_0$ which forms a C-model for R_0. The algorithm based on context extension transforms R_0 into the following μTRS $\triangle^\pi R_0$, which truthfully simulates outermost rewriting in R_0:

$$\begin{aligned} \mathsf{f}(\mathsf{a}^\star) &\to \mathsf{f}^\star(\mathsf{f}(\mathsf{a}^\star)) & \mathsf{top}(\mathsf{f}^\star(\mathsf{f}(x))) &\to \mathsf{top}(\mathsf{b}) \\ \mathsf{top}(\mathsf{a}^\star) &\to \mathsf{top}(\mathsf{f}(\mathsf{a}^\star)) & \mathsf{top}(\mathsf{f}^\star(\mathsf{f}^\star(x))) &\to \mathsf{top}(\mathsf{b}) \end{aligned} \tag{$\triangle^\pi R_0$}$$

The rule $\mathsf{f}(\mathsf{a}^\star) \to \mathsf{f}^\star(\mathsf{f}(\mathsf{a}^\star))$ is obtained from prepending the context $\mathsf{f}(\Box)$ to $\mathsf{a} \to \mathsf{f}(\mathsf{a})$. This enables correct updating of the labeling of the context during rewriting. Because we still have to allow rewrite steps $\mathsf{a} \to \mathsf{f}(\mathsf{a})$ of the original TRS at the top of a term, we extend the signature with a unary function symbol top which represents the top of a term. Thus when prepending contexts we include $\mathsf{top}(\Box)$, giving rise to the rule $\mathsf{top}(\mathsf{a}^\star) \to \mathsf{top}(\mathsf{f}(\mathsf{a}^\star))$. The necessity of the symbol top becomes apparent especially when we consider the rule $\mathsf{f}(\mathsf{f}(x)) \to \mathsf{b}$. Here prepending the context $\mathsf{f}(\Box)$ is not even an option since $\mathsf{f}(\mathsf{f}(\mathsf{f}(x))) \to \mathsf{f}(\mathsf{b})$ is not an outermost rewrite step; this rule can only be applied at the top of a term. Hence we get the two rules displayed on the right, one for each possible interpretation of the variable x.

The second algorithm we define, that of dynamic labeling, transforms R_0 (using $\mathcal{A}_0$) into the following μTRS , which we denote by $\uparrow^\pi R_0$:

$$\begin{aligned} \mathsf{a}^\star &\to \mathsf{relabel}^{0,1}(\mathsf{f}(\mathsf{a}^\star)) & \mathsf{f}(\mathsf{relabel}^{0,1}(x)) &\to \mathsf{f}^\star(x) \\ \mathsf{f}^\star(\mathsf{f}(x)) &\to \mathsf{relabel}^{1,0}(\mathsf{b}) & \mathsf{top}(\mathsf{relabel}^{0,1}(x)) &\to \mathsf{top}(x) \\ \mathsf{f}^\star(\mathsf{f}^\star(x)) &\to \mathsf{relabel}^{1,0}(\mathsf{b}) & \mathsf{top}(\mathsf{relabel}^{1,0}(x)) &\to \mathsf{top}(x) \end{aligned} \tag{$\uparrow^\pi R_0$}$$

where rewriting beneath the redex symbol $\mathsf{f}^\star$ and the symbols $\mathsf{relabel}^{0,1}$ and $\mathsf{relabel}^{1,0}$ is disallowed. Displayed on the left, we recognize the original rules. Since left and right-hand side of the original rule $\mathsf{a} \to \mathsf{f}(\mathsf{a})$ have distinct interpretations (0 and 1), in $\uparrow^\pi R_0$ the right-hand side is prefixed with the symbol $\mathsf{relabel}^{0,1}$. By application of the relabeling rules (displayed on the right), this symbol moves upward to take care of the update of labels in the context of the original rule application. Likewise, the rule $\mathsf{f}(\mathsf{f}(x)) \to \mathsf{b}$ (of which there are two versions in $\uparrow^\pi R_0$, one for each value x can be assigned to) means a change of interpretation, and relabeling the context is necessary. In this example, each original step is accompanied by exactly one relabel step. The $\mathsf{relabel}$ symbols dissolve after one such step. The only (maximal) rewrite sequence from the term $\mathsf{top}(\mathsf{a}^\star)$ in $\uparrow^\pi R_0$ is:

$$\begin{aligned} \mathsf{top}(\mathsf{a}^\star) &\to_\mu \mathsf{top}(\mathsf{relabel}^{0,1}(\mathsf{f}(\mathsf{a}^\star))) \to_\mu \mathsf{top}(\mathsf{f}(\mathsf{a}^\star)) \to_\mu \mathsf{top}(\mathsf{f}(\mathsf{relabel}^{0,1}(\mathsf{f}(\mathsf{a}^\star)))) \\ &\to_\mu \mathsf{top}(\mathsf{f}^\star(\mathsf{f}(\mathsf{a}^\star))) \to_\mu \mathsf{top}(\mathsf{relabel}^{1,0}(\mathsf{b})) \to_\mu \mathsf{top}(\mathsf{b}) \end{aligned}$$

Notice the correspondence with the outermost rewrite sequence (1.1).

Clearly, semantic labeling increases the number of rules and the number of symbols of a TRS. This results in a larger search space for finding termination proofs, and hence may lead to exhaustion of time or memory resources. On the other hand, one can say that semantic labeling does not complicate termination proofs, in the sense that proofs for the unlabeled system carry over to the labeled one: whenever R' is a labeling of a TRS R and $\mathcal{A} = \langle A, [\![\cdot]\!], \succ, \sqsupseteq \rangle$ is a monotone Σ-algebra [EWZ08] which proves termination of R, then the extension of $[\![\cdot]\!]$ to the labeled signature Σ' by defining $[\![\mathsf{f}^\lambda]\!] = [\![\mathsf{f}]\!]$ for every $\mathsf{f} \in \Sigma$ and label λ, yields a monotone Σ'-algebra witnessing termination of R'. Apart from this, the labeled systems often allow for simpler proofs, because the enriched signature provides for

more freedom in the choice of interpretations, see [Zan95]. As the transformations presented here are based on a variant of semantic labeling, they inherit these properties from semantic labeling.

The two transformations have been implemented by the first author in the termination prover Jambox [End09]. Notwithstanding the increased number of rules by semantic labeling and context extension, Jambox performs efficiently on the set of examples from the Termination Problem Database (TPDB [Ter08]), and was best in proving termination in the category of outermost rewriting of the termination competition of 2008 [Ter08], see Section 10.

Related work. The first tool for proving outermost termination was Cariboo [FGK02, GK09]. Cariboo is a stand-alone tool, and its method is based on induction.

For the idea of a transformational approach to outermost termination in order to make use of the power of termination provers we were inspired by [RZ09], which in turn is based on ideas in [GM04]. In [RZ09] the signature is enriched with unary symbols top, up, and down and the TRS is extended with 'anti-matching' rules such that $\mathsf{down}(t)$ is a redex if and only if t is not a redex with respect to the original TRS. The idea is that the symbol down is moved down in the term tree as long as no redex is encountered. Once a redex is encountered, a rewrite step is performed, and the symbol down is replaced by up, which then moves upwards again to the top of the term, marked by top. This transformation is implemented in the tool TrafO [RZ09], participant in the termination competion of 2008 [Ter08].

Based on a similarly elegant idea, Thiemann [Thi09] defines a complete transformation from outermost to innermost rewriting, which is implemented in AProVE. For traversal to the redex positions, rules of the form $\mathsf{down}(\mathsf{isRedex}(\mathsf{f}(\ldots))) \to \mathsf{f}(\ldots,\mathsf{down}(\mathsf{isRedex}(\ldots)),\ldots)$ are used. In order to simulate outermost rewriting and to prevent from moving inside redexes, rules $\mathsf{isRedex}(\ell) \to \mathsf{up}(r)$ are added for every rule $\ell \to r$ of the original TRS. Then, by the innermost rewriting strategy, the latter rules have priority over the traversal rules, whenever an original redex is encountered.

The simplicity of both approaches is attractive, but the yo-yoing effect in the resulting TRSs makes that the original outermost rewrite steps are 'hidden' among a vast amount of auxiliary steps. This increases derivational complexity, and makes it hard for automated termination provers to find proofs for the transformed systems.

The present paper is a modified and extended version of [EH09]. In particular, we introduce a novel approach for proving outermost termination: dynamic labeling (Section 6). We stress that the number of extra relabeling steps introduced in the dynamic labeling of a system is typically small and bounded by the C-depth of the applied rewrite rule.

2. Preliminaries

For a general introduction to term rewriting and to context-sensitive rewriting, we refer to [Ter03] and [Luc98], respectively. Here we repeat some of the main definitions, for the sake of completeness, and to fix notations.

A *signature* Σ is a non-empty set of symbols each having a fixed *arity*, given by a mapping $\sharp : \Sigma \to \mathbb{N}$. We write $\sharp\mathsf{f}$ for the arity of $\mathsf{f} \in \Sigma$, and we define $\Sigma_n = \{\mathsf{f} \in \Sigma \mid \sharp\mathsf{f} = n\}$. Given Σ and a set $\mathcal{X}$ of variables, the *set* $\mathcal{T}(\Sigma,\mathcal{X})$ *of terms over* Σ is the smallest set satisfying: $\mathcal{X} \subseteq \mathcal{T}(\Sigma,\mathcal{X})$, and $\mathsf{f}(t_1,\ldots,t_n) \in \mathcal{T}(\Sigma,\mathcal{X})$ if $\mathsf{f} \in \Sigma$ of arity n and $t_i \in \mathcal{T}(\Sigma,\mathcal{X})$

for all $1 \leq i \leq n$. We use $x, y, z, \ldots$ to range over variables, and write $Var(t)$ for the set of variables occurring in a term t.

The set of *positions* $Pos(t) \subseteq \mathbb{N}^*$ of a term $t \in \mathcal{T}(\Sigma, \mathcal{X})$ is defined as follows: $Pos(x) = \{\epsilon\}$ for variables $x \in \mathcal{X}$ and $Pos(\mathsf{f}(t_1, \ldots, t_n)) = \{\epsilon\} \cup \{ip \mid 1 \leq i \leq n, p \in Pos(t_i)\}$ for symbols $\mathsf{f} \in \Sigma_n$. We write $root(t)$ to denote the root symbol (or variable) of t, and $t|_p$ for the subterm of t rooted at position p. Then $root(t|_p)$ is the symbol at position p in t.

A *substitution* σ is a map $\sigma : \mathcal{X} \to \mathcal{T}(\Sigma, \mathcal{X})$ from variables to terms. For terms $t \in \mathcal{T}(\Sigma, \mathcal{X})$ and substitutions σ, $t\sigma$ is inductively defined by $x\sigma = \sigma(x)$ for $x \in \mathcal{X}$, and $\mathsf{f}(t_1, \ldots, t_n)\sigma = \mathsf{f}(t_1\sigma, \ldots, t_n\sigma)$ for $\mathsf{f} \in \Sigma_n$, and $t_1, \ldots, t_n \in \mathcal{T}(\Sigma, \mathcal{X})$.

Let $\Box$ be a fresh symbol, i.e., $\Box \notin \Sigma \cup \mathcal{X}$. A *context* C is a term from $\mathcal{T}(\Sigma, \mathcal{X} \cup \{\Box\})$ containing precisely one occurrence of $\Box$. By $C[s]$ we denote the term $C\sigma$ where $\sigma(\Box) = s$ and $\sigma(x) = x$ for all $x \in \mathcal{X}$. We use $\mathcal{C}(\Sigma, \mathcal{X})$ to denote the set of contexts over Σ and $\mathcal{X}$. We write $Var(C)$ with $C \in \mathcal{C}(\Sigma, \mathcal{X})$ to denote the set of variables of C excluding $\Box$. The *depth* of a context C is defined as the length $|p|$ of the position p at which $\Box$ resides, that is, the position p such that $root(C|_p) = \Box$.

A *term rewriting system (TRS)* over Σ is a finite set of pairs $\langle \ell, r\rangle \in \mathcal{T}(\Sigma, \mathcal{X}) \times \mathcal{T}(\Sigma, \mathcal{X})$, called *rewrite rules* and written as $\ell \to r$, for which the *left-hand side* ℓ is not a variable ($\ell \notin \mathcal{X}$) and all variables in the *right-hand side* r occur in ℓ: $Var(r) \subseteq Var(\ell)$. For a TRS R we define $\to_R$, the *rewrite relation* induced by R as follows: For terms $s, t \in \mathcal{T}(\Sigma, \mathcal{X})$ we write $s \to_R t$, or just $s \to t$ if R is clear from the context, if there exists a rule $\ell \to r \in R$, a substitution σ and a context $C \in \mathcal{C}(\Sigma, \mathcal{X})$ such that $s = C[\ell\sigma]$ and $t = C[r\sigma]$; we sometimes write $s \to_{R,p} r$ to explicitly indicate the rewrite position p, i.e., when $root(C|_p) = \Box$. A term of the form $\ell\sigma$, for some rule $\ell \to r \in R$, and a substitution σ, is called a *redex*.

For terms s and t, we say that s *outermost rewrites to* t at a position $p \in Pos(s)$, denoted by $s \xrightarrow{out}_{R,p} t$, if $s \to_{R,p} t$ and for all positions p' strictly above p (i.e., p' a proper prefix of p) we have that $s|_{p'}$ is not a redex with respect to R.

A binary relation $\succ \subseteq A \times A$ over a set A is called *well-founded* if no infinite decreasing sequence $a_1 \succ a_2 \succ a_3 \succ \ldots$ exists. A TRS R is called *terminating* or *strongly normalizing*, denoted by $\mathsf{SN}(R)$, if $\to_R$ is well-founded.

A mapping $\mu : \Sigma \to \mathbf{2}^{\mathbb{N}}$ is called a *replacement map (for Σ)* if for all symbols $\mathsf{f} \in \Sigma$ we have $\mu(\mathsf{f}) \subseteq \{1, \ldots, \sharp\mathsf{f}\}$. When we define a replacement map μ, the case for constants $\mathsf{a} \in \Sigma$ is left implicit, as we always have $\mu(\mathsf{a}) = \varnothing$. A *context-sensitive term rewriting system (μTRS)* is a pair $\langle R, \mu\rangle$ consisting of a TRS R and a replacement map μ. The set of *μ-replacing positions* $Pos^\mu(t)$ of a term $t \in \mathcal{T}(\Sigma, \mathcal{X})$ is defined by $Pos^\mu(x) = \{\epsilon\}$ for $x \in \mathcal{X}$ and $Pos^\mu(\mathsf{f}(t_1, \ldots, t_n)) = \{\epsilon\} \cup \{ip \mid i \in \mu(\mathsf{f}), p \in Pos^\mu(t_i)\}$ for $\mathsf{f} \in \Sigma_n$ and $t_1, \ldots, t_n \in \mathcal{T}(\Sigma, \mathcal{X})$.

In context-sensitive term rewriting only redexes at μ-replacing positions are contracted: we say s *μ-rewrites to* t, and denote it by $s \to_{R,\mu} t$ whenever $s \to_{R,p} t$ with $p \in Pos^\mu(s)$. For instance, consider the system R consisting of the single rule:

$$\mathsf{a} \to \mathsf{cons}(\mathsf{b}, \mathsf{a})$$

and let μ be given by $\mu(\mathsf{cons}) = \{1\}$. Then, obviously R is non-terminating. On the other hand, the context-sensitive TRS $\langle R, \mu\rangle$ is terminating, because the replacement map of the symbol cons allows rewriting only in its first argument.

We conclude this section by defining some non-standard notions.

Definition 2.1. A *thin context* is a context that has at every depth at most one symbol from $\Sigma \cup \{\Box\}$; all other symbols are variables.

Example 2.2. $\mathsf{f}(\mathsf{f}(x, \mathsf{g}(\Box)), y)$ is a thin context, whereas $\mathsf{f}(\mathsf{g}(\Box), \mathsf{h}(x))$ is not, since g and h are at the same depth.

Definition 2.3 ([SM08])**.** A *flat context* is a context $C \in \mathcal{C}(\Sigma, \mathcal{X})$ of the form:

$$C = \mathsf{f}(x_1, \ldots, x_{j-1}, \Box, x_{j+1}, \ldots, x_n)$$

where $\mathsf{f} \in \Sigma_n$ with $n > 0$ and $x_1, x_2, \ldots \in \mathcal{X}$ are pairwise distinct variables. For a term $t \in \mathcal{T}(\Sigma, \mathcal{X})$ we say that C is *fresh for* t if $Var(C) \cap Var(t) = \varnothing$. We denote the set of flat contexts fresh for t by $\mathcal{C}^\flat_t(\Sigma, \mathcal{X})$.

Definition 2.4 ([RZ09])**.** A TRS R is called *quasi-left-linear* if every non-linear left-hand side of a rule in R is an instance of a linear left-hand side from R.

Example 2.5. The following TRS is quasi-left-linear (and outermost terminating):

$$\mathsf{g}(\mathsf{f}(x), x) \to \mathsf{g}(\mathsf{g}(x, x), x) \qquad\qquad \mathsf{g}(x, y) \to y$$

3. Generalizing Models to C-models

In outermost rewriting the only redexes which are allowed to be rewritten are those which are not nested within any other redex occurrence. We model this strategy by context-sensitive rewriting with the use of semantic labeling: we mark the symbols which are the root of a redex in order to disallow rewriting within that redex. We first recall the definition of semantic labeling and of models from [Zan95], and then generalize these to fit our purpose.

Definition 3.1. A Σ-*algebra* $\mathcal{A} = \langle A, [\![\cdot]\!] \rangle$ consists of a non-empty set A, called the *domain of* $\mathcal{A}$, and for each n-ary symbol $\mathsf{f} \in \Sigma$ a function $[\![\mathsf{f}]\!] : A^n \to A$, called the *interpretation of* f. Given an *assignment* $\alpha : \mathcal{X} \to A$ of the variables to A, the *interpretation of a term* $t \in \mathcal{T}(\Sigma, \mathcal{X})$ *with respect to* α is denoted by $[\![t, \alpha]\!]$ and inductively defined by:

$$[\![x, \alpha]\!] = \alpha(x) \qquad\qquad [\![\mathsf{f}(t_1, \ldots, t_n), \alpha]\!] = [\![\mathsf{f}]\!]([\![t_1, \alpha]\!], \ldots, [\![t_n, \alpha]\!])$$

where $x \in \mathcal{X}$, $\mathsf{f} \in \Sigma_n$, and $t_i \in \mathcal{T}(\Sigma, \mathcal{X})$ for $1 \leq i \leq n$. For substitutions $\sigma : \mathcal{X} \to \mathcal{T}(\Sigma, \mathcal{X})$, we write $[\![\sigma, \alpha]\!]$ for the function $\lambda x.[\![\sigma(x), \alpha]\!]$. For ground terms $t \in \mathcal{T}(\Sigma, \varnothing)$ and ground substitutions $\sigma : \mathcal{X} \to \mathcal{T}(\Sigma, \varnothing)$ we write $[\![t]\!]$ and $[\![\sigma]\!]$ for short. We usually write $\mathcal{A}$ for both the algebra and its domain, and we use $[\![\cdot]\!]$ to denote the interpretation function of $\mathcal{A}$.

Lemma 3.2. *Let $\mathcal{A}$ be a Σ-algebra, $\alpha : \mathcal{X} \to \mathcal{A}$ an assignment, and $\sigma : \mathcal{X} \to \mathcal{T}(\Sigma, \mathcal{X})$ a substitution. Then, for all terms $t \in \mathcal{T}(\Sigma, \mathcal{X})$: $[\![t\sigma, \alpha]\!] = [\![t, [\![\sigma, \alpha]\!]]\!]$.* □

For completeness of the transformation of context extension (Theorem 5.13), it is important that there exist no 'junk' elements in the Σ-algebra, that is, elements a for which there are no ground terms t such that $[\![t]\!] = a$. See Example 5.14. For that reason we restrict Σ-algebras to 'core' Σ-algebras whose domain equals the set of all interpretations of ground terms over Σ.

Definition 3.3. The *core of a Σ-algebra* $\mathcal{A}$ is the Σ-algebra $\mathcal{A}_c = \langle A_c, [\![\cdot]\!]_c \rangle$ where A_c is the least set such that $[\![\mathsf{f}]\!](a_1, \ldots, a_n) \in A_c$ whenever $\mathsf{f} \in \Sigma_n$ and $a_1, \ldots, a_n \in A_c$, and where $[\![\cdot]\!]_c$ is the restriction of $[\![\cdot]\!]$ to A_c. We say that $\mathcal{A}$ *is core* whenever $\mathcal{A} = \mathcal{A}_c$.

By construction of the core of a Σ-algebra we then obtain:

Lemma 3.4. *For every element $a \in \mathcal{A}_c$ of the core of a Σ-algebra $\mathcal{A}$ there exists a ground term $t \in \mathcal{T}(\Sigma, \varnothing)$ with $[\![t]\!] = a$.* □

Definition 3.5 ([Zan95])**.** A *semantic labeling* $\langle \mathcal{A}, \pi \rangle$ consists of a Σ-algebra $\mathcal{A}$ and a family $\pi = \{\pi_{\mathsf{f}}\}_{\mathsf{f} \in \Sigma}$ of labeling functions $\pi_{\mathsf{f}} : \mathcal{A}^{\sharp \mathsf{f}} \to \Lambda_{\mathsf{f}}$ where Λ_{f} is a finite and non-empty set of labels for each symbol $\mathsf{f} \in \Sigma$. For a term $t \in \mathcal{T}(\Sigma, \mathcal{X})$ and an assignment $\alpha : \mathcal{X} \to \mathcal{A}$, we define $lab(t, \alpha)$, *the labeling of* t *with respect to* α, inductively as follows:

$$lab(x, \alpha) = x$$

$$lab(\mathsf{f}(t_1, \dots, t_n), \alpha) = \mathsf{f}^{\lambda}(lab(t_1, \alpha), \dots, lab(t_n, \alpha))$$

where $x \in \mathcal{X}$, $\mathsf{f} \in \Sigma_n$, $t_1, \dots, t_n \in \mathcal{T}(\Sigma, \mathcal{X})$ and $\lambda = \pi_{\mathsf{f}}([\![t_1, \alpha]\!], \dots, [\![t_n, \alpha]\!])$.

Let R be a TRS over Σ. The *semantic labeling of* R is the TRS $lab(R)$ over the labeled signature $lab(\Sigma) = \{\mathsf{f}^{\lambda} \mid \mathsf{f} \in \Sigma,\ \lambda \in \Lambda_{\mathsf{f}}\}$, defined by:

$$lab(R) = \{lab(\ell, \alpha) \to lab(r, \alpha) \mid \ell \to r \in R\,,\ \alpha : Var(\ell) \to \mathcal{A}\}$$

For a substitution $\sigma : \mathcal{X} \to \mathcal{T}(\Sigma, \mathcal{X})$, and an assignment $\alpha : \mathcal{X} \to \mathcal{A}$, we write $lab(\sigma, \alpha)$ for the function $\lambda x.lab(\sigma(x), \alpha)$. For ground terms $t \in \mathcal{T}(\Sigma, \varnothing)$ and ground substitutions $\sigma : \mathcal{X} \to \mathcal{T}(\Sigma, \varnothing)$ we write $lab(t)$ and $lab(\sigma)$ for short.

Term labeling satisfies the following useful property:

Lemma 3.6 ([Zan95])**.** *Let $\mathcal{A}$ be a Σ-algebra, let $\alpha : \mathcal{X} \to \mathcal{A}$ be an assignment, and let $\sigma : \mathcal{X} \to \mathcal{T}(\Sigma, \mathcal{X})$ be a substitution. Then, for all terms $t \in \mathcal{T}(\Sigma, \mathcal{X})$ it holds that:*

$$lab(t\sigma, \alpha) = lab(t, [\![\sigma, \alpha]\!])\, lab(\sigma, \alpha)$$

Proof. Direct by induction on the term structure, and Lemma 3.2. □

The Σ-algebra of a semantic labeling has to satisfy certain constraints in order to obtain that a TRS is terminating if and only if its labeled version is. In [Zan95] the algebra has to be a 'model':

Definition 3.7. A Σ-algebra $\mathcal{A}$ is called a *model* for a TRS R if for all rules $\ell \to r \in R$ and assignments of variables in the left-hand side $\alpha : Var(\ell) \to \mathcal{A}$ we have that $[\![\ell, \alpha]\!] = [\![r, \alpha]\!]$.

In the introduction we argued why this notion of model is too restrictive for the purpose at hand. In order to be able to distinguish between redex and non-redex positions we introduce C-models, a generalization of models.

Definition 3.8. A *C-model* for a TRS R over Σ is a Σ-algebra $\mathcal{A}$ where for each rule $\ell \to r \in R$ there exists an $n \in \mathbb{N}$ such that for each context C of depth n and assignment $\alpha : \mathcal{X} \to \mathcal{A}$ we have $[\![C[\ell], \alpha]\!] = [\![C[r], \alpha]\!]$. When $n \in \mathbb{N}$ is minimal for a rule $\ell \to r$ with respect to this property, we call n the *C-depth for $\ell \to r$*, and denote it by $\delta_{\mathcal{A}}(\ell \to r)$. The *$C$-depth for R with respect to $\mathcal{A}$*, denoted by $\delta_{\mathcal{A}}(R)$, is defined as the maximal C-depth of the rules of R: $\delta_{\mathcal{A}}(R) = \max\{\delta_{\mathcal{A}}(\ell \to r) \mid \ell \to r \in R\}$.

Example 3.9. Let R_1 be the following TRS over $\Sigma = \{\mathsf{c}, \mathsf{f}, \mathsf{g}\}$ (where c is a constant):

$$\mathsf{f}(\mathsf{g}(x)) \to \mathsf{f}(\mathsf{f}(\mathsf{g}(x))) \qquad\qquad \mathsf{f}(\mathsf{f}(\mathsf{f}(x))) \to x \qquad (R_1)$$

The algebra $\mathcal{A}_1 = \{\bot, f, ff, g\}$ with the interpretation function defined, for all $x \in \mathcal{A}_1$, by:

$$[\![\mathsf{c}]\!] = \bot \qquad [\![\mathsf{f}]\!](\bot) = [\![\mathsf{f}]\!](g) = f \qquad [\![\mathsf{f}]\!](f) = [\![\mathsf{f}]\!](ff) = ff \qquad [\![\mathsf{g}]\!](x) = g \qquad (\mathcal{A}_1)$$

forms a C-model for R_1. The C-depth of the rule $\mathsf{f}(\mathsf{g}(x)) \to \mathsf{f}(\mathsf{f}(\mathsf{g}(x)))$ is 1; both contexts $\mathsf{f}(\Box)$ and $\mathsf{g}(\Box)$ make that left and right-hand side of the rule have equal interpretations, respectively ff and g, regardless of the value we assign to the variable x. For the other rule

the C-depth is 2: If the interpretation of x is $\bot$ or g then $\mathsf{f}(\Box)$ is not yet enough to interpret both sides by the same element of the algebra; an additional context $\mathsf{f}(\Box)$ or $\mathsf{g}(\Box)$ has to be wrapped around. Thus the C-depth of the TRS is $\delta_{\mathcal{A}_1}(R_1) = 2$.

We now define semantic labelings based on C-models, to which we refer as 'C-labelings'. We define a C-labeling over an extended signature $\Sigma_{\mathsf{top}} = \Sigma \cup \{\mathsf{top}\}$. The symbol top represents the top of a term, and we assume top to be fresh for Σ, i.e., $\mathsf{top} \notin \Sigma$. Moreover, a C-labeling includes a set $\Sigma^{red} \subseteq lab(\Sigma)$ of 'redex symbols', the set of symbols below which rewriting should be forbidden. For example, for a sound transformation from outermost to context-sensitive rewriting it has to be guaranteed that in a well-labeled term the symbols from Σ^{red} occur at redex positions only.

Definition 3.10. Let R be a TRS over Σ. A *C-labeling for R* is a tuple $\langle \mathcal{A}, \pi, \Sigma^{red} \rangle$ where $\mathcal{A}$ is a C-model for R, $\langle \mathcal{A}, \pi \rangle$ is a semantic labeling over the signature $\Sigma_{\mathsf{top}} = \Sigma \cup \{\mathsf{top}\}$, and $\Sigma^{red} \subseteq lab(\Sigma)$ is a subset of the labeled signature. We fix the interpretation $[\![\mathsf{top}]\!]$ of top to be an arbitrary constant function $\lambda x.a$ for some $a \in \mathcal{A}$.

A C-labeling $\langle \mathcal{A}, \pi, \Sigma^{red} \rangle$ for R is called:

(i) *sound* if $root(lab(t)) \in \Sigma^{red}$ implies that t is a redex with respect to R, for all ground terms $t \in \mathcal{T}(\Sigma, \varnothing)$;
(ii) *complete* if $root(lab(t)) \in \Sigma^{red}$ whenever t is a redex with respect to R, for all ground terms $t \in \mathcal{T}(\Sigma, \varnothing)$;
(iii) *maximal* if $\pi_{\mathsf{f}}(a_1, \ldots, a_n) = \langle a_1, \ldots, a_n \rangle$, for all symbols $\mathsf{f} \in \Sigma_n$ and all values $a_1, \ldots, a_n \in \mathcal{A}$;
(iv) *core* if the Σ-algebra $\mathcal{A}$ is core.

Remark 3.11. From Definition 3.10 it follows that a C-labeling $\langle \mathcal{A}, \pi, \Sigma^{red} \rangle$ for R is:
- sound if and only if $root(lab(t)|_p) \in \Sigma^{red}$ implies that $t|_p$ is a redex with respect to R, for all ground terms $t \in \mathcal{T}(\Sigma, \varnothing)$ and all positions $p \in Pos(t)$, and
- complete if and only if $root(lab(t)|_p) \in \Sigma^{red}$ whenever $t|_p$ is a redex with respect to R, for all ground terms $t \in \mathcal{T}(\Sigma, \varnothing)$ and all positions $p \in Pos(t)$.

Example 3.12. We continue with Example 3.9 where we defined a C-model $\mathcal{A}_1$ for the TRS R_1. We let $\langle \mathcal{A}_1, \pi \rangle$ be the semantic labeling where π labels each symbol with the interpretation of its arguments. The set of redex symbols is defined by $\Sigma^{red} = \{\mathsf{f}^{g}, \mathsf{f}^{ff}\}$. These symbols correspond to redex positions with respect to the first and the second rule of R_1. Then $\langle \mathcal{A}_1, \pi, \Sigma^{red} \rangle$ forms a sound, complete, maximal, core C-labeling for R_1.

We explain why we fix the interpretation $[\![\mathsf{top}]\!]$ to be a constant function. First note that if $[\![\mathsf{top}]\!]$ is a constant function, then the extension of the signature with top does not interfere with the property of $\mathcal{A}$ being a C-model for R. Second, the transformation given in Section 5 extends the rules with contexts until the interpretations of the left and right-hand side are equal. If $[\![\mathsf{top}]\!]$ is constant, then the extension halts at the symbol top, that is, no further contexts are prefixed to top. This corresponds to the intuition of top representing the top of the term. Moreover, it is not important which constant function $\lambda x.a$ we choose for $[\![\mathsf{top}]\!]$: as no symbols will be prefixed to top, no symbol will be labeled with its value.

Remark 3.13. The transformations defined in Sections 5 and 6 are sound whenever we use a sound C-labeling. This means that termination of the target system implies outermost termination of the original system. On the other hand, using a complete C-labeling does not guarantee completeness of either transformations. More precisely, using a complete

C-labeling does not imply that the transformed system is terminating whenever the original system is outermost terminating. A complete C-labeling guarantees that in a correctly labeled term all redex positions are marked, so that only outermost steps are possible. But, for the transformation to be complete we need two more properties. First, rewriting needs to preserve correct labeling of terms. Secondly, for the labeled system, global termination of all terms (including the not correctly labeled ones) should be equivalent with local termination [EdVW09] of the well-labeled terms. This point is of practical importance because the state of the art of automated analysis for global termination is far more advanced than for local termination. Both properties do in general not hold for complete C-labelings.

Remark 3.14. Maximal C-labelings can be defined in a more general fashion by requiring $\pi_{\mathsf{f}}(\vec{a}) \neq \pi_{\mathsf{f}}(\vec{b})$ for all $\mathsf{f} \in \Sigma_n$ and all $\vec{a}, \vec{b} \in \mathcal{A}^n$ with $\vec{a} \neq \vec{b}$. The important point is that the value of all arguments can be inferred from the label. For the sake of a simple presentation we stick to the definition where labels are tuples of argument values.

4. Static Context Extension

In this section we describe a naive approach for semantic labeling with C-models. This serves both as an introduction and as a motivation for the transformations that we present in Sections 5 and 6. Input for these transformations is a TRS together with a C-model for this TRS. In Sections 7–9 we explain how C-models are constructed.

As can be inferred from Definition 3.8, it is possible to transform a TRS R by prepending contexts to its rules in such a way that its C-model $\mathcal{A}$ becomes a model for the transformed system $\tilde{R}$, and then apply the usual semantic labeling to $\tilde{R}$. We call this transformation 'static context extension', as opposed to the transformation of 'dynamic context extension' presented in the next section. In the dynamic version, contexts are prepended only when needed and dependent on the values assigned to the variables in the rules.

Since every context is an instance of a thin context (Definition 2.1) with the same depth, rules are prefixed by thin contexts, in both versions of context extension.

Definition 4.1. Let R be a TRS and $\langle \mathcal{A}, \pi, \Sigma^{red} \rangle$ a C-labeling for R. The *static context extension of R with respect to* $\langle \mathcal{A}, \pi, \Sigma^{red} \rangle$, is the μTRS $\langle \mathbb{A}^{\pi} R, \mu \rangle$ where $\mathbb{A}^{\pi} R$ is the TRS resulting from the steps listed below and where the replacement map μ is defined by $\mu(\mathsf{f}) = \varnothing$ if $\mathsf{f} \in \Sigma^{red}$, and $\mu(\mathsf{f}) = \{1, \dots, \sharp\mathsf{f}\}$ otherwise, for all $\mathsf{f} \in \mathit{lab}(\Sigma_{\mathsf{top}})$.

(i) Replace each rule $\ell \to r$ with C-depth n by the set of rules:
  (a) $C[\ell] \to C[r]$ for each thin context C of depth n, and
  (b) $\mathsf{top}(C[\ell]) \to \mathsf{top}(C[r])$ for each thin context C of depth $< n$.
  We let $\tilde{R}$ denote the union of these sets.
(ii) Apply semantic labeling to $\tilde{R}$ using the C-labeling $\langle \mathcal{A}, \pi \rangle$. We obtain $\mathit{lab}(\tilde{R})$.
(iii) Remove from $\mathit{lab}(\tilde{R})$ all labeled rules that contain a redex symbol from Σ^{red} in the prepended context. The TRS thus obtained is denoted by $\mathbb{A}^{\pi} R$.

Note that the rules from item i(b) of Definition 4.1 model the application of an original rule at a depth smaller than its C-depth, that is, 'near' the top of the term. The created rules are hence wrapped into $\mathsf{top}(\Box)$.

Example 4.2. We illustrate the transformation by static context extension on the TRS R_1:

$$\mathsf{f}(\mathsf{g}(x)) \to \mathsf{f}(\mathsf{f}(\mathsf{g}(x))) \qquad \mathsf{f}(\mathsf{f}(\mathsf{f}(x))) \to x \qquad (R_1)$$

together with the C-model $\mathcal{A}_1 = \{\bot, f, ff, g\}$ introduced in Example 3.9 and the C-labeling $\langle \mathcal{A}_1, \pi, \Sigma^{red} \rangle$ defined in Example 3.12.

The first step of the transformation yields $\tilde{R}_1$ which consists of the following 10 rules:

$$C[\mathsf{f}(\mathsf{g}(x))] \to C[\mathsf{f}(\mathsf{f}(\mathsf{g}(x)))] \qquad D[\mathsf{f}(\mathsf{f}(\mathsf{f}(x)))] \to D[x] \qquad (\tilde{R}_1)$$

where $C \in \{\mathsf{top}(\Box), \mathsf{f}(\Box), \mathsf{g}(\Box)\}$, and $D \in \{\mathsf{top}(\Box), \mathsf{top}(\mathsf{f}(\Box)), \mathsf{top}(\mathsf{g}(\Box)), \mathsf{f}(\mathsf{f}(\Box)), \mathsf{f}(\mathsf{g}(\Box)), \mathsf{g}(\mathsf{f}(\Box)), \mathsf{g}(\mathsf{g}(\Box))\}$. Note that the algebra $\mathcal{A}_1$ extended with the symbol top and interpretation $[\![\mathsf{top}]\!](x) = \bot$ for all $x \in \mathcal{A}_1$, is a model for the TRS $\tilde{R}_1$ (in fact, any value for the interpretation of top will do).

The second step is to label the TRS $\tilde{R}_1$. This yields $lab(\tilde{R}_1)$ consisting of $4 \cdot (3+7) = 40$ rules, four instances for each of the ten rules, one for each value that can be assigned to x.

The final step is to remove from $lab(\tilde{R}_1)$ each rule which contains a redex symbol within the context that was prepended in the construction of $\tilde{R}_1$. Such a rule would enable a rewrite step which is not outermost. Of the 40 rules of above, 12 have to be thrown out. This concerns the labelings of the second rule of R_1 where $\mathsf{f}(\mathsf{f}(\Box))$, $\mathsf{top}(\mathsf{f}(\Box))$ or $\mathsf{g}(\mathsf{f}(\Box))$ has been prepended. They contain the redex symbol f^{ff} in the prepended context. For instance, if we prepend the thin context $\mathsf{f}(\mathsf{f}(\Box))$ to the rewrite rule $\mathsf{f}(\mathsf{g}(x)) \to \mathsf{f}(\mathsf{f}(\mathsf{g}(x)))$ and then perform semantic labeling, we obtain the following rule in $lab(\tilde{R}_1)$:

$$\mathsf{f}^{ff}(\mathsf{f}^{f}(\mathsf{f}^{g}(\mathsf{g}^{\bot}(x)))) \to \mathsf{f}^{ff}(\mathsf{f}^{ff}(\mathsf{f}^{f}(\mathsf{f}^{g}(\mathsf{g}^{\bot}(x))))) \qquad (4.1)$$

This rule (with the redex symbol f^{ff} in the prepended context) has to be discarded as it would admit the following infinite rewrite sequence, altough R_1 is outermost terminating:

$$\begin{aligned}\mathsf{top}^{f}(\mathsf{f}^{g}(\mathsf{g}^{\bot}(\mathsf{c}))) &\to_{\mu} \mathsf{top}^{ff}(\mathsf{f}^{f}(\mathsf{f}^{g}(\mathsf{g}^{\bot}(\mathsf{c})))) \to_{\mu} \mathsf{top}^{ff}(\mathsf{f}^{ff}(\mathsf{f}^{f}(\mathsf{f}^{g}(\mathsf{g}^{\bot}(\mathsf{c})))))\\ &\overset{(4.1)}{\to_{\mu}} \mathsf{top}^{ff}(\mathsf{f}^{ff}(\mathsf{f}^{ff}(\mathsf{f}^{f}(\mathsf{f}^{g}(\mathsf{g}^{\bot}(\mathsf{c})))))) \to_{\mu} \mathsf{top}^{f}(\mathsf{f}^{g}(\mathsf{g}^{\bot}(\mathsf{c}))) \to_{\mu} \dots\end{aligned}$$

We note in advance that the *dynamic* context extension $\triangle^{\pi} R_1$ of R_1, worked out in Example 5.1, consists of 19 rewrite rules, whereas the static version $\triangleq^{\pi} R_1$ above consists of 28 rules.

In general, the presence of a redex symbol may depend on the interpretation of the variables. This is better demonstrated by the following example.

Example 4.3. Consider the TRS over the signature $\{\mathsf{c}, \mathsf{f}, \mathsf{g}\}$ (with c a constant):

$$\mathsf{g}(\mathsf{f}(\mathsf{g}(x))) \to \mathsf{f}(\mathsf{g}(\mathsf{g}(\mathsf{f}(x)))) \qquad \mathsf{f}(x) \to x \qquad (R_2)$$

We use the C-model $\mathcal{A}_2 = \{\bot, g, fg\}$ with the interpretation of the symbols defined by:

$$[\![\mathsf{c}]\!] = \bot \qquad [\![\mathsf{g}]\!](x) = g \qquad [\![\mathsf{f}]\!](g) = fg \qquad [\![\mathsf{f}]\!](\bot) = [\![\mathsf{f}]\!](fg) = \bot \qquad (\mathcal{A}_2)$$

for all $x \in \mathcal{A}_2$. Again we use maximal labeling so that the symbols g^{fg}, $\mathsf{f}^{\bot}$, f^{g} and f^{fg} correspond to redex positions. The C-depth of the rule $\mathsf{f}(x) \to x$ is 2 and its static context extension contains the rule $\mathsf{g}(\mathsf{g}(\mathsf{f}(x))) \to \mathsf{g}(\mathsf{g}(x))$. From this we obtain three labeled rules:

$$\begin{aligned}\mathsf{g}^{g}(\mathsf{g}^{\bot}(\mathsf{f}^{\bot}(x))) &\to \mathsf{g}^{g}(\mathsf{g}^{\bot}(x)) && \text{for } \alpha(x) = \bot\\ \mathsf{g}^{g}(\mathsf{g}^{fg}(\mathsf{f}^{g}(x))) &\to \mathsf{g}^{g}(\mathsf{g}^{g}(x)) && \text{for } \alpha(x) = g\\ \mathsf{g}^{g}(\mathsf{g}^{\bot}(\mathsf{f}^{fg}(x))) &\to \mathsf{g}^{g}(\mathsf{g}^{fg}(x)) && \text{for } \alpha(x) = fg\end{aligned}$$

The second rule should not be allowed, as it would enable a rewrite step that is not outermost. This is witnessed by the symbol g^{fg} in the prepended context.

5. Dynamic Context Extension

We present an approach for semantic labeling with C-models, called 'dynamic context extension',[1] where we stepwise extend rules by contexts, only when needed and dependent on the variable interpretation used for the semantic labeling. For different interpretations of the variables usually different context depths are necessary for achieving equal interpretations of left and right-hand side. In each extension step we check whether a candidate symbol is a redex symbol, and, if it is, this symbol is excluded from prepending. Here, by a redex symbol we mean a labeled symbol which indicates the presence of a redex in the original system. Dynamic context extension is more efficient in the sense that both the number and the size of the rules of the resulting μTRS are smaller than in the static version defined in the previous version.

The transformation starts with constructing pairs $\langle \ell \to r, \alpha \rangle$ of rules and variable assignments. Then these rules are extended with flat contexts until the interpretations of left and right-hand side are equal. Finally, each obtained rule is labeled using the corresponding interpretation. More precisely, we implement this process as follows.

We iteratively construct sets $P_0, P_1, \ldots$, until $P_{i+1} = P_i$ for some i. The initial set P_0 consists of pairs $\langle \ell \to r, \alpha \rangle$ for each rule $\ell \to r$, and each interpretation $\alpha : \mathit{Var}(\ell) \to \mathcal{A}$ of the variables. Then, in each step, P_{i+1} is obtained from P_i by replacing every pair $\langle \ell \to r, \alpha \rangle$ of P_i for which the interpretation of the left-hand side differs from the right-hand side ($[\![\ell, \alpha]\!] \neq [\![r, \alpha]\!]$), by the pairs $\langle C[\ell] \to C[r], \alpha' \rangle$ for every flat context C (Definition 2.3) and every extension $\alpha' : \mathit{Var}(C[\ell]) \to \mathcal{A}$ of α, such that the root of the labeled, extended left-hand side $\mathit{lab}(C[\ell], \alpha')$ is not a redex symbol. Among the flat contexts to be prepended we include $\mathsf{top}(\Box)$ to cater for the case that the rule is applied at the top of the term.

Example 5.1. We reconsider from Examples 3.9 and 3.12 the term rewriting system R_1:

$$\mathsf{f}(\mathsf{g}(x)) \to \mathsf{f}(\mathsf{f}(\mathsf{g}(x))) \qquad\qquad \mathsf{f}(\mathsf{f}(\mathsf{f}(x))) \to x \tag{R_1}$$

together with the C-labeling $\langle \mathcal{A}_1, \pi, \Sigma^{red} \rangle$, where $\mathcal{A}_1 = \{\bot, f, \mathit{ff}, g\}$, π labels symbols with their arguments and $\Sigma^{red} = \{\mathsf{f}^{g}, \mathsf{f}^{\mathit{ff}}\}$. The initial set P_0 of pairs $\langle \text{rule}, \text{assignment} \rangle$ is:

$$P_0 = \left\{ \langle \mathsf{f}(\mathsf{g}(x)) \to \mathsf{f}(\mathsf{f}(\mathsf{g}(x))), \lambda x.a \rangle, \langle \mathsf{f}(\mathsf{f}(\mathsf{f}(x))) \to x, \lambda x.a \rangle \mid a \in \mathcal{A}_1 \right\}$$

The only element $\langle \ell \to r, \alpha \rangle$ of P_0 such that $[\![\ell, \alpha]\!] = [\![r, \alpha]\!]$ is $\langle \mathsf{f}(\mathsf{f}(\mathsf{f}(x))) \to x, \lambda x.\mathit{ff} \rangle$. For this pair no context needs to be prepended. The other pairs have to be replaced by their context extensions and thus P_1 consists of the following ($4 \cdot 3 + 1 + 3 \cdot 2 = 19$) pairs:

$$\begin{aligned}
\langle C[\mathsf{f}(\mathsf{g}(x))] \to C[\mathsf{f}(\mathsf{f}(\mathsf{g}(x)))], \lambda x.a \rangle \qquad & \text{for all } a \in \mathcal{A}_1,\ C \in \{\mathsf{top}(\Box), \mathsf{f}(\Box), \mathsf{g}(\Box)\} \\
\langle \mathsf{f}(\mathsf{f}(\mathsf{f}(x))) \to x, \lambda x.\mathit{ff} \rangle & \\
\langle C[\mathsf{f}(\mathsf{f}(\mathsf{f}(x)))] \to C[x], \lambda x.a \rangle \qquad & \text{for all } a \neq \mathit{ff},\ C \in \{\mathsf{top}(\Box), \mathsf{g}(\Box)\}
\end{aligned}$$

In the last line the context $\mathsf{f}(\Box)$ is excluded, because the labeled left-hand side of the rule would contain the redex symbol f^{ff} within the prepended context, and thus the step would not be outermost. Because of the outermost strategy, the original rule is only applicable in a context $C[\mathsf{g}(\Box)]$ (where C does not contain any redexes) or at the top of a term. Now for all rules in P_1 the left and right-hand side have equal interpretations, and hence the iterative construction is finished.

[1] In [EH09] we used the term 'dynamic labeling' for what we here call 'dynamic context extension'. The term 'dynamic labeling' is now reserved for the transformation that we define in Section 6.

Secondly, the obtained set P_1 is labeled using the family π of labeling functions. The desired context-sensitive TRS $\triangle^\pi R_1$ then consists of the rules $lab(\ell,\alpha) \to lab(r,\alpha)$ for every $\langle \ell \to r, \alpha\rangle \in P_1$, with the replacement map μ defined by $\mu(\mathsf{h}) = \varnothing$ if $\mathsf{h} \in \{\mathsf{f}^g, \mathsf{f}^{ff}\}$, and $\mu(\mathsf{h}) = \{1,\ldots,\sharp\mathsf{h}\}$ otherwise, for all $\mathsf{h} \in lab(\Sigma)$. Thus the dynamic context extension of R_1 consists of 19 rules. Recall from Example 4.2 that the static context extension of R_1 had 28 rules.

We now formalize this transformation.

Definition 5.2. Let R be a TRS over Σ, and $\langle \mathcal{A}, \pi, \Sigma^{red}\rangle$ a C-labeling for R. We define $P^\pi(R)$ as the least fixed point of the following construction of sets $P_0, P_1, \ldots$, that is, $P^\pi(R) = P_i$ as soon as $P_{i+1} = P_i$ for some i. The initial set P_0 is defined by:

$$P_0 = \left\{ \langle \ell \to r, \alpha\rangle \mid \ell \to r \in R,\ \alpha : Var(\ell) \to \mathcal{A} \right\}$$

and for $i = 0, 1, \ldots$ the set P_{i+1} is obtained from P_i by replacing every pair $\langle \ell \to r, \alpha\rangle$ such that $[\![\ell,\alpha]\!] \neq [\![r,\alpha]\!]$, or $r \in \mathcal{X}$, by all pairs in $\triangle(\ell \to r, \alpha)$ where we define:

$$\triangle(\ell \to r, \alpha) = \left\{ \langle C[\ell] \to C[r], \alpha + \beta\rangle \;\middle|\; C \in \mathcal{C}^\flat_\ell(\Sigma_{\mathsf{top}}, \mathcal{X}),\ \beta : Var(C) \to \mathcal{A},\ root(lab(C[\ell], \alpha+\beta)) \notin \Sigma^{red} \right\}$$

Here, for partial functions f and g with disjoint domains, we write $f + g$ for the function defined by $(f+g)(x) = f(x)$ if $x \in dom(f)$, and $(f+g)(x) = g(x)$ if $x \in dom(g)$.

The construction of $P^\pi(R)$ is guaranteed to terminate because of the assumption that $\mathcal{A}$ is a C-model for R.

Definition 5.3 (Dynamic context extension)**.** Let R be a TRS over Σ, and $\langle \mathcal{A}, \pi, \Sigma^{red}\rangle$ a C-labeling for R. The *dynamic context extension of R with respect to* $\langle \mathcal{A}, \pi, \Sigma^{red}\rangle$ is the μTRS $\langle \triangle^\pi R, \mu\rangle$ consisting of:

$$\triangle^\pi R = \left\{ lab(\ell,\alpha) \to lab(r,\alpha) \mid \langle \ell \to r, \alpha\rangle \in P^\pi(R) \right\}$$

and the replacement map μ, defined by $\mu(\mathsf{f}) = \varnothing$ if $\mathsf{f} \in \Sigma^{red}$, and $\mu(\mathsf{f}) = \{1,\ldots,\sharp\mathsf{f}\}$ otherwise, for all $\mathsf{f} \in lab(\Sigma_{\mathsf{top}})$. Whenever the set Σ^{red}, which determines the replacement map, is clear from the context, we write $\triangle^\pi R$ as a shorthand for $\langle \triangle^\pi R, \mu\rangle$.

Remark 5.4. In the transformation given in Definition 5.3 collapsing rules are always prepended by at least one flat context. Consequently, $\langle \triangle^\pi R, \mu\rangle$ contains no collapsing rules. This is used in the proof of Theorem 5.13 in order to apply Theorem 5.12. Without this elimination of collapsing rules, the transformation is still sound (Theorem 5.8). Note that in the TRS R_1 worked out before, we did not eliminate the collapsing rule.

Let us work out another example.

Example 5.5. We consider problem `zantema08/dupl_rhs.trs` from the TPDB [Ter08]:

$$\begin{array}{ll} \mathsf{f}(\mathsf{h}(x),\mathsf{c}) \to \mathsf{f}(\mathsf{i}(x),\mathsf{s}(x)) & \mathsf{i}(x) \to \mathsf{h}(x) \\ \mathsf{f}(\mathsf{i}(x),y) \to x & \mathsf{h}(x) \to \mathsf{f}(\mathsf{h}(x),\mathsf{c}) \end{array} \qquad (R_3)$$

We denote this TRS by R_3, and take the algebra $\mathcal{A}_3 = \langle \{\bot, c, h, i\}, [\![\cdot]\!]\rangle$ with $[\![\cdot]\!]$ defined by:

$$[\![\mathsf{c}]\!] = c \qquad [\![\mathsf{h}]\!](x) = h \qquad [\![\mathsf{i}]\!](x) = i \qquad [\![\mathsf{f}]\!](x,y) = [\![\mathsf{s}]\!](x) = \bot$$

for all $x, y \in \mathcal{A}_3$. Furthermore, we employ minimal labeling; only the function symbols that are at the root of a redex occurrence are marked. Thus the symbols h, i are always marked:

$\pi_{\mathsf{h}}(x) = \pi_{\mathsf{i}}(x) = \star$. We let $\pi_{\mathsf{f}}(i, x) = \pi_{\mathsf{f}}(h, c) = \star$, and leave f unmarked otherwise. Also, the symbols s and c are never marked.

The dynamic context extension $\triangle^\pi R_3$ is then formed by the rules:

$$
\begin{aligned}
\mathsf{f}^\star(\mathsf{h}^\star(x), \mathsf{c}) &\to \mathsf{f}^\star(\mathsf{i}^\star(x), s(x)) & \mathsf{s}(\mathsf{i}^\star(x)) &\to \mathsf{s}(\mathsf{h}^\star(x)) \\
& & \mathsf{f}(y, \mathsf{i}^\star(x)) &\to \mathsf{f}(y, \mathsf{h}^\star(x)) \\
\mathsf{s}(\mathsf{f}^\star(\mathsf{i}^\star(x), y)) &\to \mathsf{s}(x) & \mathsf{top}(\mathsf{i}^\star(x)) &\to \mathsf{top}(\mathsf{h}^\star(x)) \\
\mathsf{f}(z, \mathsf{f}^\star(\mathsf{i}^\star(x), y)) &\to \mathsf{f}(z, x) & & \qquad (\triangle^\pi R_3) \\
\mathsf{f}(z, \mathsf{f}^\star(\mathsf{i}^\star(x), y)) &\to \mathsf{f}^\star(z, x) & \mathsf{s}(\mathsf{h}^\star(x)) &\to \mathsf{s}(\mathsf{f}^\star(\mathsf{h}^\star(x), \mathsf{c})) \\
\mathsf{f}(\mathsf{f}^\star(\mathsf{i}^\star(x), y), z) &\to \mathsf{f}(x, z) & \mathsf{f}(y, \mathsf{h}^\star(x)) &\to \mathsf{f}(y, \mathsf{f}^\star(\mathsf{h}^\star(x), \mathsf{c})) \\
\mathsf{f}(\mathsf{f}^\star(\mathsf{i}^\star(x), y), z) &\to \mathsf{f}^\star(x, z) & \mathsf{f}(\mathsf{h}^\star(x), y) &\to \mathsf{f}(\mathsf{f}^\star(\mathsf{h}^\star(x), \mathsf{c}), y) \\
\mathsf{top}(\mathsf{f}^\star(\mathsf{i}^\star(x), y)) &\to \mathsf{top}(x) & \mathsf{top}(\mathsf{h}^\star(x)) &\to \mathsf{top}(\mathsf{f}^\star(\mathsf{h}^\star(x), \mathsf{c}))
\end{aligned}
$$

We now work towards the first main theorem, stating that outermost ground termination of R is implied by termination of the transformed system $\triangle^\pi R$.

Lemma 5.6. *Let R be a TRS over Σ, and let $\langle \mathcal{A}, \pi, \Sigma^{red} \rangle$ be a sound C-labeling for R. Moreover, let $s, t \in \mathcal{T}(\Sigma, \varnothing)$ be ground terms and $p \in Pos(s)$ such that $s \xrightarrow{out}_{R,p} t$. Then for all proper prefixes q of $1p$ we have $root(lab(\mathsf{top}(s))|_q) \notin \Sigma^{red}$.*

Proof. For $q = \epsilon$ this follows from $\mathsf{top}^\lambda \notin \Sigma^{red}$ for any label λ. Otherwise we have that $root(lab(\mathsf{top}(s))|_q) = root(lab(s)|_{q'})$ with q' a proper prefix of p, and if $root(lab(s)|_{q'}) \in \Sigma^{red}$, then, by definition of sound C-labeling, s contains a redex at position q', quod non. □

The following lemma states that any outermost ground rewrite step in R can be transformed into a rewrite step in $\triangle^\pi R$.

Lemma 5.7. *Let R be a TRS over Σ, and let $\langle \mathcal{A}, \pi, \Sigma^{red} \rangle$ be a sound C-labeling for R. Let $s, t \in \mathcal{T}(\Sigma, \varnothing)$ be ground terms such that $s \xrightarrow{out}_R t$. Then:*

$$lab(\mathsf{top}(s)) \to_{\triangle^\pi R, \mu} lab(\mathsf{top}(t))$$

Proof. Assume $s \xrightarrow{out}_{R,p} t$ for some position $p \in Pos(s)$. Then there exists a rule $\ell \to r \in R$, a context C with $root(C|_p) = \Box$ and a ground substitution σ such that $s = C[\ell\sigma]$ and $t = C[r\sigma]$. We consider the construction of the dynamic context extension from Definition 5.3, and prove by induction that for all $i = 0, 1, \ldots$ there exists a context C_i which is a prefix of $\mathsf{top}(C)$, a ground substitution σ_i, and terms ℓ_i, r_i such that $\mathsf{top}(s) = C_i[\ell_i\sigma_i]$, $\mathsf{top}(t) = C_i[r_i\sigma_i]$ and $\langle \ell_i \to r_i, [\![\sigma_i]\!] \rangle \in P_i$. For the base case we have $\langle \ell_0 \to r_0, [\![\sigma_0]\!] \rangle \in P_0$ with $\ell_0 = \ell$, $r_0 = r$, $\sigma_0 = \sigma$, and $C_0 = \mathsf{top}(C)$. For the induction step we assume the existence of C_i, σ_i, and $\langle \ell_i \to r_i, [\![\sigma_i]\!] \rangle \in P_i$ with the above properties. If $[\![\ell_i, [\![\sigma_i]\!]]\!] = [\![r_i, [\![\sigma_i]\!]]\!]$ and $r_i \notin \mathcal{X}$ then by definition $\langle \ell_i \to r_i, [\![\sigma_i]\!] \rangle \in P_{i+1}$, and so we are done. For the remaining cases $[\![\ell_i, [\![\sigma_i]\!]]\!] \neq [\![r_i, [\![\sigma_i]\!]]\!]$ and $r_i \in \mathcal{X}$, we first show that $C_i \neq \Box$. If $[\![\ell_i, [\![\sigma_i]\!]]\!] \neq [\![r_i, [\![\sigma_i]\!]]\!]$ and $C_i = \Box$, then $\ell_i\sigma_i = \mathsf{top}(s)$ and $r_i\sigma_i = \mathsf{top}(t)$, and hence $root(\ell_i) = root(r_i) = \mathsf{top}$, contradicting $[\![\ell_i, [\![\sigma_i]\!]]\!] \neq [\![r_i, [\![\sigma_i]\!]]\!]$ (recall that the interpretation of top is constant). Furthermore, we have $r_i \in \mathcal{X}$ only if $i = 0$, and then $C_i = \mathsf{top}(C) \neq \Box$. Thus we have $C_i = D[D'\sigma']$ for some context D, flat context $D' \in \mathcal{C}^\flat_{\ell_i}$ and substitution σ'. We choose $C_{i+1} = D$, $\ell_{i+1} = D'[\ell_i]$, $r_{i+1} = D'[r_i]$, and $\sigma_{i+1} = \sigma_i + \sigma'$. It remains to be shown that $\langle \ell_{i+1} \to r_{i+1}, [\![\sigma_{i+1}]\!] \rangle \in P_{i+1}$. For this it suffices to prove that $root(lab(\ell_{i+1}, [\![\sigma_{i+1}]\!])) \notin \Sigma^{red}$. We have $C_{i+1}[\ell_{i+1}\sigma_{i+1}] = \mathsf{top}(s)$. Let q be the position such that $root(C_{i+1}|_q) = \Box$. Then, by Lemma 3.6 we obtain

$root(lab(\ell_{i+1}, [\![\sigma_{i+1}]\!])) = root(lab(\ell_{i+1}\sigma_{i+1})) = root(\mathsf{top}(lab(s))|_q)$. Note that q is a proper prefix of $1p$. By Lemma 5.6 we have $root(lab(\mathsf{top}(s))|_q) \notin \Sigma^{red}$.

Let i be such that $P_{i+1} = P_i$. By the result above we have $\langle \ell_i \to r_i, [\![\sigma_i]\!] \rangle \in P$ with $[\![\ell_i\sigma_i]\!] = [\![r_i\sigma_i]\!]$, and $lab(\ell_i, [\![\sigma_i]\!]) \to lab(r_i, [\![\sigma_i]\!]) \in \triangle^\pi R$ by definition. Let τ and υ be defined by $\tau(\Box) = \ell_i\sigma_i$, $\upsilon(\Box) = r_i\sigma_i$, and $\tau(x) = \upsilon(x) = x$ for $x \in \mathcal{X}$. Then we have that $lab(C_i, [\![\tau]\!]) = lab(C_i, [\![\upsilon]\!])$ since $[\![\tau]\!] = [\![\upsilon]\!]$. Let $E = lab(C_i, [\![\tau]\!])$. We get $lab(\mathsf{top}(s)) = lab(C_i[\ell_i\sigma_i]) = lab(C_i\tau) = Elab(\tau) = E[lab(\ell_i\sigma_i)] = E[lab(\ell_i, [\![\sigma_i]\!])lab(\sigma_i)]$ and $lab(\mathsf{top}(t)) = \ldots = E[lab(r_i, [\![\sigma_i]\!])lab(\sigma_i)]$, by Lemma 3.6. By Lemma 5.6 all symbols above position $1p$ in the term $lab(\mathsf{top}(s))$ are not in Σ^{red} and hence we have a μ-rewrite step: $lab(\mathsf{top}(s)) \to_{\triangle^\pi R,\mu} lab(\mathsf{top}(t))$. □

Theorem 5.8. *Let R be a TRS over Σ, and $\langle \mathcal{A}, \pi, \Sigma^{red} \rangle$ a sound C-labeling for R. Then R is outermost ground terminating if $\triangle^\pi R$ is terminating.*

Proof. Assume that R admits an infinite outermost rewrite sequence:

$$t_1 \overset{out}{\to}_R t_2 \overset{out}{\to}_R t_3 \overset{out}{\to}_R \ldots$$

Then from Lemma 5.7 it follows that $\triangle^\pi R$ admits an infinite rewrite sequence:

$$lab(\mathsf{top}(t_1)) \to_{\triangle^\pi R,\mu} lab(\mathsf{top}(t_2)) \to_{\triangle^\pi R,\mu} lab(\mathsf{top}(t_3)) \to_{\triangle^\pi R,\mu} \ldots$$ □

The following three examples illustrate why our method is sound, but not complete when applied to non-left-linear (and non-quasi-left-linear) TRSs. The first example can be handled by our approach employing the C-labeling constructed in Section 9. The second example fails using the C-labeling from Section 9, but can successfully be treated using a manually constructed C-labeling. For the third example, we show that there exists no C-labeling that can be employed for proving outermost ground termination; this example is out of reach for the approach proposed in this paper.

Example 5.9. We consider the non-left-linear TRS R_4 with three rules:

$$\mathsf{g}(x,x) \to \mathsf{f}(\mathsf{f}(x,x),x) \qquad \mathsf{f}(x,x) \to \mathsf{g}(x,x) \qquad \mathsf{f}(x,y) \to y \qquad (R_4)$$

over the signature $\Sigma = \{\mathsf{f}, \mathsf{g}, \mathsf{a}\}$ where a is a constant (necessary for the existence of ground terms). We choose the algebra $\mathcal{A}_4 = \{\bot\}$ with $[\![\mathsf{a}]\!] = \bot$, $[\![\mathsf{f}]\!](\bot,\bot) = \bot$, and $[\![\mathsf{g}]\!](\bot,\bot) = \bot$. We label the symbols with the interpretations of their arguments, and define $\Sigma^{red} = \{\mathsf{f}^{\bot,\bot}\}$.

Note that Σ^{red} does not contain $\mathsf{g}^{\bot,\bot}$. The reason is that using a finite algebra we can (in general) not recognize redex positions with respect to non-left-linear rules. By excluding $\mathsf{g}^{\bot,\bot}$ from Σ^{red} we allow rewriting below g even when g is the root of a redex. This is sound for proving outermost termination as it does not restrict the possible rewrite steps, but allows only additional steps. The symbol $\mathsf{f}^{\bot,\bot}$ is part of Σ^{red}; due to the rule $\mathsf{f}(x,y) \to y$ each occurrence of f is a redex position.

The dynamic labeling $\triangle^\pi R_4$ is then formed by:

$$\begin{aligned} \mathsf{g}^{\bot,\bot}(x,x) &\to \mathsf{f}^{\bot,\bot}(\mathsf{f}^{\bot,\bot}(x,x),x) \\ \mathsf{f}^{\bot,\bot}(x,x) &\to \mathsf{g}^{\bot,\bot}(x,x) \\ \mathsf{f}^{\bot,\bot}(x,y) &\to y \end{aligned} \qquad (\triangle^\pi R_4)$$

where $\mu(\mathsf{f}^{\bot,\bot}) = \varnothing$ and $\mu(\mathsf{g}^{\bot,\bot}) = \{1,2\}$. This system is terminating which can be seen as follows. After an application of the first rule:

$$C[\mathsf{g}^{\bot,\bot}(t,t)] \to_{\triangle^\pi R_4,\mu} C[\mathsf{f}^{\bot,\bot}(\mathsf{f}^{\bot,\bot}(t,t),t)]$$

the replacement map μ prevents us from reducing the inner $\mathsf{f}^{\bot,\bot}$. Moreover, the second rule cannot be applied to the outer $\mathsf{f}^{\bot,\bot}$ since the left and the right subterm are not equal. Thus the only rule applicable to the displayed subterm is $\mathsf{f}^{\bot,\bot}(x,y) \to y$ which reduces the size of the term, and we can conclude termination by induction.

Hence we conclude outermost ground termination of R_4 by Theorem 5.8. Actually the same C-labeling allows also to infer outermost termination, see Lemma 5.15 (we simply add a fresh constant $\mathsf{0}$ and a unary symbol s with interpretations $[\![\mathsf{0}]\!] = \bot$ and $[\![\mathsf{s}]\!](\bot) = \bot$).

Example 5.10. We consider the non-left-linear TRS R_5 over the signature $\Sigma_5 = \{\mathsf{g}, \mathsf{a}, \mathsf{b}\}$:

$$\mathsf{a} \to \mathsf{g}(\mathsf{a},\mathsf{a}) \qquad\qquad \mathsf{g}(x,x) \to \mathsf{b} \tag{R_5}$$

This TRS is outermost terminating. However, there exists no C-labeling that recognizes redex positions with respect to the non-left-linear rule $\mathsf{g}(x,x) \to \mathsf{b}$. A finite algebra cannot be used to check whether two arbitrary subterms t_1 and t_2 of $\mathsf{g}(t_1,t_2)$ are equal. Thus it appears that, in order to have a sound transformation, we cannot include any symbol g^λ in the set Σ^{red} of redex symbols. But then rewriting below g is allowed, and the rule $\mathsf{a} \to \mathsf{g}(\mathsf{a},\mathsf{a})$ would lead to non-termination of the dynamic labeling $\triangle^\pi R_5$.

Nonetheless, in this particular example, the problem can be solved. If some element e of the algebra is the interpretation of precisely one ground term t, then, of course, $[\![t_1]\!] = [\![t_2]\!] = e$ implies that $t_1 = t_2$. Let us take the algebra $\mathcal{A}_5 = \{\bot, a\}$ with $[\![\mathsf{a}]\!] = a$, $[\![\mathsf{b}]\!] = \bot$, and $[\![\mathsf{g}]\!](x,y) = \bot$ for all $x, y \in \mathcal{A}_5$. We use maximal labeling and define $\Sigma^{red} = \{\mathsf{g}^{a,a}\}$. That is, we mark redex positions $\mathsf{g}(t,t)$ only for the special case $t = \mathsf{a}$. This C-labeling is sound since only redex positions are marked, but it is not complete; not all redex positions are marked. Nevertheless, this labeling can be used to prove outermost ground termination of R_5. The dynamic labeling $\triangle^\pi R_5$ of R_5 consists of:

$$\begin{aligned}
\mathsf{g}^{a,\bot}(\mathsf{a},x) &\to \mathsf{g}^{\bot,\bot}(\mathsf{g}^{a,a}(\mathsf{a},\mathsf{a}),x) & \mathsf{g}^{a,a}(x,x) &\to \mathsf{b} \\
\mathsf{g}^{\bot,a}(x,\mathsf{a}) &\to \mathsf{g}^{\bot,\bot}(x,\mathsf{g}^{a,a}(\mathsf{a},\mathsf{a})) & \mathsf{g}^{\bot,\bot}(x,x) &\to \mathsf{b} \qquad (\triangle^\pi R_5) \\
\mathsf{top}^{\mathsf{a}}(\mathsf{a}) &\to \mathsf{top}^{\bot}(\mathsf{g}^{a,a}(\mathsf{a},\mathsf{a}))
\end{aligned}$$

The employed C-labeling is not complete, and so the μTRS $\triangle^\pi R_5$ admits rewrite sequences (starting from correctly labeled terms) that do not correspond to outermost rewriting, e.g:

$$\begin{aligned}
\mathsf{top}^{\bot}(\mathsf{g}^{\bot,\bot}(\mathsf{g}^{a,a}(\mathsf{a},\mathsf{a}),\mathsf{g}^{a,a}(\mathsf{a},\mathsf{a}))) &\to_{\triangle^\pi R_5,\mu} \mathsf{top}^{\bot}(\mathsf{g}^{\bot,\bot}(\mathsf{b},\mathsf{g}^{a,a}(\mathsf{a},\mathsf{a}))) \\
&\to_{\triangle^\pi R_5,\mu} \mathsf{top}^{\bot}(\mathsf{g}^{\bot,\bot}(\mathsf{b},\mathsf{b})) \\
&\to_{\triangle^\pi R_5,\mu} \mathsf{top}^{\bot}(\mathsf{b})
\end{aligned}$$

Despite of this, the μTRS can be shown to be terminating, and since the C-labeling was sound, we conclude outermost ground termination of R_5 by Theorem 5.8.

Example 5.11. In Examples 5.9 and 5.10 we have seen how our method can be applied to prove outermost termination of non-quasi-left-linear TRSs. We now consider an example which shows that not every non-left-linear TRS can be handled by our method:

$$\mathsf{f}(x) \to \mathsf{g}(\mathsf{f}(x),\mathsf{f}(x)) \qquad\qquad \mathsf{g}(x,x) \to \mathsf{b} \tag{R_6}$$

This TRS is outermost terminating. Now the trick used in Example 5.10 does not work. In order to construct a terminating μTRS $\triangle^\pi R_6$ we need to forbid rewriting in all terms of the form $\mathsf{g}(\mathsf{f}(t),\mathsf{f}(t))$. This is impossible using a finite algebra.

We need the following adaptation of [Ohl02, Proposition 5.5.24] for μTRSs; the proof proceeds along the same lines.

Theorem 5.12. *Let $\langle R, \mu\rangle$ be a terminating many-sorted μTRS. If the μTRS obtained from $\langle R, \mu\rangle$ by dropping sorts admits an infinite rewrite sequence, then $\langle R, \mu\rangle$ is collapsing and duplicating.* □

While for soundness of the transformation (Theorem 5.8) a sound labeling suffices, for a complete transformation we need the C-labeling to be complete, maximal and core:

Theorem 5.13. *Let R be a TRS over Σ, and $\langle \mathcal{A}, \pi, \Sigma^{red}\rangle$ a complete, maximal, and core C-labeling for R. Then $\triangle^\pi R$ is terminating if R is outermost ground terminating.*

Proof. Assume that $\triangle^\pi R$ is not terminating. We turn $\triangle^\pi R$ into a sorted TRS. The sorts are chosen from the set $\mathcal{A} \cup \{\mathsf{top}\}$. Since the C-labeling is maximal, for each n-ary symbol $\mathsf{f}^\lambda \in lab(\Sigma_{\mathsf{top}})$ we have $\lambda = \langle a_1, \ldots, a_n\rangle$. We let f^λ have input sort λ and output sort $[\![\mathsf{f}]\!](a_1, \ldots, a_n)$. The only exception is the output sort of the symbols top^λ which we fix to be the sort top. Then by Theorem 5.12 together with non-collapsingness of $\triangle^\pi R$ yields the existence of a well-sorted infinite rewrite sequence τ in $\triangle^\pi R$. Since the C-labeling is core, by Lemma 3.4 there exists a ground term for every sort in $\mathcal{A}$. Thus by applying a ground substitution to τ we obtain a well-sorted infinite ground term rewrite sequence τ'.

Well-sortedness implies correct labeling: for each well-sorted term $t \in \mathcal{T}(lab(\Sigma_{\mathsf{top}}), \varnothing)$ there exists a term $t' \in \mathcal{T}(\Sigma_{\mathsf{top}}, \varnothing)$ such that $t = lab(t')$. Moreover, a symbol top^λ can only occur at the top of a term. Without loss of generality we assume that every term in τ' has top^λ (for some $\lambda \in \mathcal{A}$) as root (as rewriting below top^λ is allowed and context-sensitive rewriting is closed under μ-replacing contexts). Hence it suffices to show that for all terms $s, t \in \mathcal{T}(\Sigma, \varnothing)$ with $lab(\mathsf{top}(s)) \to_{\triangle^\pi R, \mu} lab(\mathsf{top}(t))$ we have $s \xrightarrow{out}_R t$. By construction, each rule in $\triangle^\pi R$ is the result of prepending contexts to, and labeling of, a rule in R. Let $\rho : s \to_R t$ be the step corresponding to $lab(\mathsf{top}(s)) \to_{\triangle^\pi R, \mu} lab(\mathsf{top}(t))$. We show that ρ is an outermost step. Assume there would be a redex u above the rewrite position. Then by completeness of the C-labeling we get $root(lab(u)) \in \Sigma^{red}$. But then this symbol must be in $lab(\mathsf{top}(s))$, either above the applied rule from $\triangle^\pi R$ or within the prepended context. Both cases yield a contradiction: the former since $\mu(root(lab(u))) = \varnothing$ would prohibit the μ-step, and the latter because we do not prepend symbols from Σ^{red}. □

Let us consider the three conditions of Theorem 5.13 on C-labelings: complete, maximal and core. To see that completeness and maximality are necessary, we refer to Examples 5.11, and 9.4, respectively. The following example shows the need to restrict to core algebras:

Example 5.14. Let R_7 be the following term rewriting system:

$$\mathsf{f}(x) \to \mathsf{g}(x, \mathsf{f}(x)) \qquad \mathsf{g}(\mathsf{a}, x) \to \mathsf{a} \qquad \mathsf{g}(\mathsf{f}(x), y) \to \mathsf{a} \qquad \mathsf{g}(\mathsf{g}(x, y), z) \to \mathsf{a} \qquad (R_7)$$

This TRS is outermost ground terminating: First note that without the first rule R_7 is terminating. So consider a rewrite step $\mathsf{f}(t) \to \mathsf{g}(t, \mathsf{f}(t))$ for $t \in \mathcal{T}(\{\mathsf{f}, \mathsf{g}, \mathsf{a}\}, \varnothing)$. Then one of the three g-rules matches $\mathsf{g}(t, \mathsf{f}(t))$ and blocks all inner rules by the outermost strategy.

We take the C-model $\mathcal{A}_7 = \{0, 1\}$ with $[\![\mathsf{a}]\!] = [\![\mathsf{f}]\!](x) = [\![\mathsf{g}]\!](x, y) = 0$, for all $x, y \in \mathcal{A}_7$. We let π be the maximal labeling and define $\Sigma^{red} = \{\mathsf{f}^0, \mathsf{g}^{0,0}\}$. Then the dynamic context extension $\triangle^\pi R_7$ contains, amongst others, the following two rules:

$$\mathsf{f}^0(x) \to \mathsf{g}^{0,0}(x, \mathsf{f}^0(x)) \qquad\qquad \mathsf{f}^1(x) \to \mathsf{g}^{1,0}(x, \mathsf{f}^1(x))$$

where $\mu(\mathsf{g}^{0,0}) = \varnothing$ and $\mu(\mathsf{g}^{1,0}) = \{1, 2\}$. Consequently, the second rule is not terminating, although the original TRS is outermost ground terminating. The C-labeling $\langle \mathcal{A}_7, \pi, \Sigma^{red} \rangle$ is complete for R and maximal, but not core. Note that there exists no ground term which has the interpretation 1, and hence the label 1 should never occur.

Theorems 5.8 and 5.13 are about outermost ground termination. This is not a severe restriction, as by adding a fresh constant $\mathsf{0}$ and a fresh unary symbol s outermost ground termination implies (and for quasi-left-linear TRSs coincides with) outermost termination:

Lemma 5.15. *A TRS R over Σ is outermost terminating if R over $\Sigma \cup \{\mathsf{s}, \mathsf{0}\}$ is outermost ground terminating. If R is also quasi-left-linear, the converse direction holds as well.*

Proof. Let $\mathcal{X}$ be countably infinite, and $\phi : \mathcal{X} \to \mathbb{N}$ a bijection. We define a substitution σ by $\sigma(x) = \mathsf{s}^{\phi(x)}(\mathsf{0})$. Then, we have $s\sigma \overset{out}{\to} t\sigma$ whenever $s \overset{out}{\to} t$ with $s, t \in \mathcal{T}(\Sigma, \mathcal{X})$, since the symbols s and $\mathsf{0}$ do not occur in any pattern of a rule, and for all $p, q \in Pos(s)$ we have $s|_p = s|_q \Leftrightarrow s\sigma|_p = s\sigma|_q$. This concludes the proof of the first part of the theorem.

For the converse direction, let R be a quasi-left-linear TRS such that R over $\Sigma \cup \{\mathsf{s}, \mathsf{0}\}$ is not outermost ground terminating. Let $t \in \mathcal{T}(\Sigma \cup \{\mathsf{s}, \mathsf{0}\}, \varnothing)$ be a ground term of minimal size admitting an infinite rewrite sequence $t = t_1 \overset{out}{\to} t_2 \overset{out}{\to} t_3 \overset{out}{\to} \ldots$. By minimality, infinitely many of these steps must be in the prefix of t not containing s and $\mathsf{0}$. Let $t' \in \mathcal{T}(\Sigma, \mathcal{X})$ be obtained from t by replacing all subterms with root symbol s or $\mathsf{0}$ in t by a (arbitrary, but fixed) variable x. Then t' admits an infinite outermost rewrite sequence as well. Note that by replacing the subterms no redex in the $\{\mathsf{s}, \mathsf{0}\}$-free prefix of t is destroyed since the symbols s and $\mathsf{0}$ do not occur any rule pattern. Fresh redexes with respect to non-left-linear rules may be created (but not with respect to left-linear rules). By quasi-left-linearity, at each position where a redex is created, there is also redex with respect to a left-linear rule. Hence, no additional redexes get blocked by outermost strategy. □

The following example shows that extending the signature with a single fresh constant $\mathsf{0}$ only is not enough for the implication: *R over the extended signature is outermost ground terminating $\Rightarrow$ R is outermost terminating.*

Example 5.16. Consider the following term rewriting system R:

$$\begin{array}{rrr} \mathsf{f}(x,y) \to \mathsf{a}(\mathsf{f}(x,y)) & \mathsf{a}(\mathsf{f}(\mathsf{b},x)) \to \mathsf{b} & \mathsf{a}(\mathsf{f}(x,\mathsf{b})) \to \mathsf{b} \\ \mathsf{a}(\mathsf{f}(x,x)) \to \mathsf{b} & \mathsf{a}(\mathsf{f}(\mathsf{a}(x),y)) \to \mathsf{b} & \mathsf{a}(\mathsf{f}(x,\mathsf{a}(y))) \to \mathsf{b} \\ & \mathsf{a}(\mathsf{f}(\mathsf{f}(x,y),z)) \to \mathsf{b} & \mathsf{a}(\mathsf{f}(x,\mathsf{f}(y,z))) \to \mathsf{b} \end{array}$$

Because of the first rule R is not outermost terminating:

$$\mathsf{f}(x,y) \overset{out}{\to} \mathsf{a}(\mathsf{f}(x,y)) \overset{out}{\to} \mathsf{a}(\mathsf{a}(\mathsf{f}(x,y))) \overset{out}{\to} \ldots$$

but the TRS over the extended signature $\Sigma' = \{\mathsf{a}, \mathsf{f}, \mathsf{b}, \mathsf{0}\}$ *is* outermost ground terminating: Consider a step $\mathsf{f}(s,t) \to \mathsf{a}(\mathsf{f}(s,t))$ with $s, t \in \mathcal{T}(\Sigma', \varnothing)$. If $s \neq \mathsf{0}$ one of the rules in the second column applies, and if $t \neq \mathsf{0}$ then one of the rules in the third column is applicable. However if $s = t = \mathsf{0}$ then the rule $\mathsf{a}(\mathsf{f}(x,x)) \to \mathsf{b}$ matches.

Note that for the second part of Lemma 5.15 we require quasi-left-linearity. This requirement was erroneously missing from [EH09], but is necessary as the following example illustrates.

Example 5.17. We consider the following term rewriting system R:

$$\begin{aligned}
\mathsf{f}(x,y,y) &\to \mathsf{a}(\mathsf{f}(x,x,y)) & \mathsf{a}(\mathsf{f}(x,\mathsf{a}(y),\mathsf{a}(z))) &\to \bot\\
\mathsf{b} &\to \mathsf{a}(\mathsf{b}) & \mathsf{a}(\mathsf{f}(x,\mathsf{b},\mathsf{b})) &\to \bot\\
\mathsf{a}(\mathsf{b}) &\to \bot & \mathsf{a}(\mathsf{f}(x,\mathsf{f}(y_1,y_2,y_3),\mathsf{f}(z_1,z_2,z_3))) &\to \bot\\
\mathsf{a}(\mathsf{f}(x,x,x)) &\to \bot & \mathsf{a}(\mathsf{f}(x,\bot,\bot)) &\to \bot
\end{aligned}$$

and explain why this system is outermost terminating. Without the rule $\mathsf{f}(x,y,y) \to \mathsf{a}(\mathsf{f}(x,x,y))$ outermost termination of R is obvious; Hence, in an infinite rewrite sequence this rule must be applied infinitely often. Let us consider a rewrite step $C[\mathsf{f}(t,u,u)] \overset{out}{\to} C[\mathsf{a}(\mathsf{f}(t,t,u))]$. If $t \in \mathcal{X}$, then $u = t$ since no non-variable term rewrites to a variable; then $\mathsf{a}(\mathsf{f}(x,x,x)) \to \bot$ is applicable and has priority (by outermost strategy) over all inner rewrite steps (and we terminate). If $t \notin \mathcal{X}$, then the second argument u and third argument t in $\mathsf{a}(\mathsf{f}(t,t,u))$ have to rewrite to a common non-variable reduct (in order to make the first rule applicable again). However, as soon as the common reduct is reached, one of the rules displayed on the right would be applicable and have priority by outermost rewriting strategy.

Nevertheless, R over the signature $\Sigma \cup \{\mathsf{s},\mathsf{0}\}$ is not outermost ground terminating:

$$\begin{aligned}
\mathsf{a}(\mathsf{f}(\mathsf{s}(\mathsf{b}),\mathsf{s}(\mathsf{a}(\mathsf{b})),\mathsf{s}(\mathsf{a}(\mathsf{b})))) &\overset{out}{\to} \mathsf{a}(\mathsf{a}(\mathsf{f}(\mathsf{s}(\mathsf{b}),\mathsf{s}(\mathsf{b}),\mathsf{s}(\mathsf{a}(\mathsf{b})))))\\
&\overset{out}{\to} \mathsf{a}(\mathsf{a}(\mathsf{f}(\mathsf{s}(\mathsf{b}),\mathsf{s}(\mathsf{a}(\mathsf{b})),\mathsf{s}(\mathsf{a}(\mathsf{b})))))\\
&\overset{out}{\to} \dots
\end{aligned}$$

6. Dynamic Labeling

This paper is about employing context-sensitive rewriting to model outermost rewriting. We do so by marking redexes, and forbid rewriting below them. As we have seen, contracting a redex may create another redex higher up in the term tree. Hence it may be necessary to update some labels during a rewrite step. In Section 5 we defined a transformation where this updating was accounted for by extending rules with contexts. Here we give an alternative transformation from TRSs to context-sensitive TRSs. We call this tranformation 'dynamic labeling'. Instead of extending rules with contexts, we now employ rewriting to propagate the changed information upward in the term tree, and set the labels in the surrounding context right, step by step. Again the C-depth (Definition 3.8) serves as a bound: here on the number of successive ancestor nodes that have to be relabeled. Each original rewrite step will give rise to a corresponding step and a bounded number ($\leq$ the C-depth) of auxiliary steps in the transformed system. Thus, although the derivational complexity (the length of rewrite sequences) is changed, this is only by a constant factor. We prove that dynamic labeling is sound for arbitrary TRSs. Moreover, for left-linear TRSs, the method is complete in a weakened sense, see Theorem 6.12. In Section 10, we compare the performance of this method to the one of dynamic context extension described in Section 5.

We begin with an analysis for evaluating which value changes can occur by rewriting and need to be propagated upward. As we will see, this restricts the number of auxiliary 'relabel symbols', and, in particular, the number of 'relabeling rules'.

Definition 6.1. Let R be a TRS over Σ, and let $\langle \mathcal{A}, \pi, \Sigma^{red} \rangle$ be a C-labeling for R. For $i = 0, 1, 2, \ldots$, we define the set $L_i \subseteq \mathcal{A} \times \mathcal{A}$ inductively by:

$$L_0 = \{\, \langle [\![l, \alpha]\!], [\![r, \alpha]\!] \rangle \mid \ell \to r \in R, \alpha : \mathit{Var}(\ell) \to \mathcal{A},\ [\![l, \alpha]\!] \neq [\![r, \alpha]\!] \,\}$$
$$L_{i+1} = L_i \cup \{\, \langle [\![\mathsf{f}]\!](\vec{a}, b, \vec{c}), [\![\mathsf{f}]\!](\vec{a}, b', \vec{c}) \rangle \mid \mathsf{f} \in \Sigma,\ \vec{a} \cdot b \cdot \vec{c} \in \mathcal{A}^{\sharp\mathsf{f}},\ \langle b, b' \rangle \in L_i,\ \mathsf{f}^{\pi_\mathsf{f}(\vec{a}, b, \vec{c})} \in \mathit{lab}(\Sigma_{\mathsf{top}}) \setminus \Sigma^{red} \,\}$$

Then we define the set $L^\pi(R)$ of *value-change pairs* by:

$$L^\pi(R) = L_i \setminus \{\langle a, a \rangle\}_{a \in \mathcal{A}}$$

with i the least number such that $L_{i+1} = L_i$.

The 'dynamic labeling' $\uparrow^\pi R$ of a TRS R is partitioned into two sets of rules. The first set is denoted by $\uparrow^\pi_1 R$ and consists of a semantic labeling of the original rules, where, additionally, a right-hand side is prefixed by a symbol $\mathsf{relabel}^{a,a'}$ whenever application of the rule causes a change of interpretation from a to a'. The second set, $\uparrow^\pi_2 R$, is a set of rules for relabeling the context of the rule application. A symbol $\mathsf{relabel}^{a,a'}$, with $\langle a, a' \rangle \in L^\pi(R)$, indicates that the value of its subterm has changed from a to a', and the rules in $\uparrow^\pi_2 R$ take care of propagating this change of value upward in the term.

Definition 6.2 (Dynamic labeling)**.** Let R be a TRS over Σ, and let $\langle \mathcal{A}, \pi, \Sigma^{red} \rangle$ be a C-labeling for R. The TRS $\uparrow^\pi R$ over the signature $\mathit{lab}(\Sigma_{\mathsf{top}}) \cup \{\mathsf{relabel}^{a,a'} \mid \langle a, a' \rangle \in L^\pi(R)\}$ is defined by $\uparrow^\pi R = \uparrow^\pi_1 R \cup \uparrow^\pi_2 R$. Here the set $\uparrow^\pi_1 R$ of *labeled rules* contains, for each rule $\ell \to r \in R$ and assignment $\alpha : \mathit{Var}(\ell) \to \mathcal{A}$, one of the rules:

$$\mathit{lab}(\ell, \alpha) \to \begin{cases} \mathit{lab}(r, \alpha) & \text{if } [\![\ell, \alpha]\!] = [\![r, \alpha]\!] \\ \mathsf{relabel}^{[\![\ell,\alpha]\!],[\![r,\alpha]\!]}(\mathit{lab}(r, \alpha)) & \text{otherwise} \end{cases}$$

Secondly, the set $\uparrow^\pi_2 R$ of *relabeling rules* contains, for each n-ary $\mathsf{f} \in \Sigma$, $\langle b, b' \rangle \in L^\pi(R)$, and $\langle \vec{a}, b, \vec{c} \rangle \in \mathcal{A}^n$ such that $\mathsf{f}^\lambda \in \mathit{lab}(\Sigma_{\mathsf{top}}) \setminus \Sigma^{red}$ with $\lambda = \pi_\mathsf{f}(\vec{a}, b, \vec{c})$, one of the rules:

$$\mathsf{f}^\lambda(\vec{x}, \mathsf{relabel}^{b,b'}(y), \vec{z}) \to \begin{cases} \mathsf{f}^{\lambda'}(\vec{x}, y, \vec{z}) & \text{if } d = d' \\ \mathsf{relabel}^{d,d'}(\mathsf{f}^{\lambda'}(\vec{x}, y, \vec{z})) & \text{otherwise} \end{cases}$$

where $\lambda' = \pi_\mathsf{f}(\vec{a}, b', \vec{c})$, $d = [\![\mathsf{f}]\!](\vec{a}, b, \vec{c})$, $d' = [\![\mathsf{f}]\!](\vec{a}, b', \vec{c})$, $|\vec{x}| = |\vec{a}|$, and $|\vec{z}| = |\vec{c}|$.

The *dynamic labeling of R (with respect to the C-labeling $\langle \mathcal{A}, \pi, \Sigma^{red} \rangle$)* is the context-sensitive TRS $\langle \uparrow^\pi R, \mu \rangle$, where the replacement map μ is defined by $\mu(\mathsf{relabel}^{a,a'}) = \varnothing$ for all $\langle a, a' \rangle \in L^\pi(R)$, $\mu(\mathsf{f}) = \varnothing$ if $\mathsf{f} \in \Sigma^{red}$, and $\mu(\mathsf{f}) = \{1, \ldots, \sharp\mathsf{f}\}$ otherwise, for all $\mathsf{f} \in \mathit{lab}(\Sigma_{\mathsf{top}})$. Whenever Σ^{red} is clear from the context, we leave μ implicit, and overload the notation $\uparrow^\pi R$ to denote $\langle \uparrow^\pi R, \mu \rangle$.

Example 6.3. We revisit the TRS R_1 from Example 3.9 for which we worked out the static and dynamic context extensions in Examples 4.2 and 5.1. We repeat its definition and the C-labeling from Example 3.12: R_1 is the TRS over $\Sigma = \{\mathsf{a}, \mathsf{f}, \mathsf{g}\}$ consisting of the rules:

$$\mathsf{f}(\mathsf{g}(x)) \to \mathsf{f}(\mathsf{f}(\mathsf{g}(x))) \qquad \mathsf{f}(\mathsf{f}(\mathsf{f}(x))) \to x \tag{R_1}$$

A C-model for R_1 is formed by the Σ-algebra $\mathcal{A}_1 = \{\bot, f, \mathit{ff}, g\}$ with interpretation:

$$[\![\mathsf{c}]\!] = \bot \qquad [\![\mathsf{f}]\!](\bot) = [\![\mathsf{f}]\!](g) = f \qquad [\![\mathsf{f}]\!](f) = [\![\mathsf{f}]\!](\mathit{ff}) = \mathit{ff} \qquad [\![\mathsf{g}]\!](x) = g \tag{$\mathcal{A}_1$}$$

for all $x \in \mathcal{A}_1$. Furthermore, $\langle \mathcal{A}_1, \pi \rangle$ denotes the maximal labeling for R_1, and $\Sigma^{red} = \{\mathsf{f}^g, \mathsf{f}^{\mathit{ff}}\}$. Then $\langle \mathcal{A}_1, \pi, \Sigma^{red} \rangle$ forms a sound and complete C-labeling of R_1. Also note that

$\mathcal{A}_1$ forms a core algebra; for each value $e \in \mathcal{A}_1$ there is a ground term t such that $[\![t]\!] = e$. We first compute the set $L^\pi(R_1)$ of value-change pairs. For the initial set L_0, note that the rule $\mathsf{f}(\mathsf{g}(x)) \to \mathsf{f}(\mathsf{f}(\mathsf{g}(x)))$ changes the interpretation from f to ff, regardless of the value assigned to x. The other rule creates three value-change pairs; one for each of the values $g, \bot, f$ assigned to x. If the interpretation of x is ff there is no change. Hence we get:

$$L_0 = \{\langle f, \mathit{ff}\rangle, \langle \mathit{ff}, \bot\rangle, \langle \mathit{ff}, f\rangle, \langle \mathit{ff}, g\rangle\}$$

All symbols $\mathsf{relabel}^{e,e'}$ with $\langle e, e'\rangle \in L_0$ will disappear in one relabeling step, whence $L^\pi(R_1) = L_0$. The dynamic labeling of R_1 then is $\uparrow^\pi R_1 = \uparrow^\pi_1 R_1 \cup \uparrow^\pi_2 R_1$ where $\uparrow^\pi_1 R_1$ consists of the rules:

$$\begin{aligned}
\mathsf{f}^g(\mathsf{g}^e(x)) &\to \mathsf{relabel}^{f,\mathit{ff}}(\mathsf{f}^f(\mathsf{f}^g(\mathsf{g}^e(x)))) && \text{for all } e \in \mathcal{A}_1\\
\mathsf{f}^{\mathit{ff}}(\mathsf{f}^{\mathit{ff}}(\mathsf{f}^{e'}(x))) &\to \mathsf{relabel}^{e,e'}(x) && \text{for all } \langle e, e'\rangle \in \{\langle \mathit{ff}, \bot\rangle, \langle \mathit{ff}, f\rangle, \langle \mathit{ff}, g\rangle\}\\
\mathsf{f}^{\mathit{ff}}(\mathsf{f}^{\mathit{ff}}(\mathsf{f}^{\mathit{ff}}(x))) &\to x
\end{aligned}$$

and where $\uparrow^\pi_2 R_1$ is formed by:

$$\begin{aligned}
\mathsf{g}^e(\mathsf{relabel}^{e,e'}(x)) &\to \mathsf{g}^{e'}(x) && \text{for all } \langle e, e'\rangle \in L^\pi(R_1)\\
\mathsf{f}^f(\mathsf{relabel}^{f,\mathit{ff}}(x)) &\to \mathsf{f}^{\mathit{ff}}(x)
\end{aligned}$$

In total the dynamic labeling of R_1 has 13 rules. Had we not restricted the construction of the set of the relabeling rules to the 'reachable' symbols $\mathsf{relabel}^{e,e'}$ (by the requirement $\langle e, e'\rangle \in L^\pi(R_1)$ in Definition 6.2), we would have come up with 18 instead of 5 relabeling rules.

Example 6.4. We reconsider the term rewrite system R_3 from Example 5.5:

$$\begin{aligned}
\mathsf{f}(\mathsf{h}(x), \mathsf{c}) &\to \mathsf{f}(\mathsf{i}(x), \mathsf{s}(x)) & \mathsf{i}(x) &\to \mathsf{h}(x)\\
\mathsf{f}(\mathsf{i}(x), y) &\to x & \mathsf{h}(x) &\to \mathsf{f}(\mathsf{h}(x), \mathsf{c})
\end{aligned}$$

and the algebra $\mathcal{A}_3 = \langle\{\bot, c, h, i\}, [\![\cdot]\!]\rangle$ with $[\![\cdot]\!]$ defined, for all $x, y \in \mathcal{A}_3$, as follows:

$$[\![\mathsf{c}]\!] = c \qquad [\![\mathsf{h}]\!](x) = h \qquad [\![\mathsf{i}]\!](x) = i \qquad [\![\mathsf{f}]\!](x, y) = [\![\mathsf{s}]\!](x) = \bot$$

Moreover, we employ minimal labeling again; see Example 5.5.

The set of change-value pairs is:

$$L^\pi(R_3) = \{\langle \bot, c\rangle, \langle \bot, h\rangle, \langle \bot, i\rangle, \langle i, h\rangle, \langle h, \bot\rangle\}$$

The set $\uparrow^\pi_1 R_3$ of labeled rules is constructed thus:

$$\begin{aligned}
\mathsf{f}^\star(\mathsf{h}^\star(x), \mathsf{c}) &\to \mathsf{f}^\star(\mathsf{i}^\star(x), \mathsf{s}(x))\\
\mathsf{f}^\star(\mathsf{i}^\star(x), y) &\to x\\
\mathsf{f}^\star(\mathsf{i}^\star(x), y) &\to \mathsf{relabel}^{e,e'}(x) && \langle e, e'\rangle \in \{\langle \bot, c\rangle, \langle \bot, h\rangle, \langle \bot, i\rangle\}\\
\mathsf{i}^\star(x) &\to \mathsf{relabel}^{i,h}(\mathsf{h}^\star(x))\\
\mathsf{h}^\star(x) &\to \mathsf{relabel}^{h,\bot}(\mathsf{f}^\star(\mathsf{h}^\star(x), \mathsf{c}))
\end{aligned}$$

There are four rules with left-hand side $\ell = \mathsf{f}^\star(\mathsf{i}^\star(x), y)$, one for each value assigned to x. In case $\alpha(x) = \bot$ there is no change of interpretation, for we have that $[\![\ell, \alpha]\!] = \bot$ for all $\alpha : \{x, y\} \to \mathcal{A}_3$ and so no $\mathsf{relabel}$ symbol is inserted. But if, for instance, $[\![\sigma(x)]\!] = c$ for some substitution σ, then some labels in the context C of a rewrite step $C[\ell\sigma] \to C[\sigma(x)]$

have to be updated, since the value of $\Box$ has changed from $\bot$ to c, whence the insertion of $\mathsf{relabel}^{\bot,c}$ to the right-hand side x.

The set $\uparrow_2^\pi R_3$ of relabeling rules is formed by:

$$\begin{aligned}
\mathsf{f}(\mathsf{relabel}^{e,e'}(x),y) &\to \mathsf{f}(x,y) && \langle e,e'\rangle \in \{\langle\bot,c\rangle,\langle\bot,h\rangle,\langle h,\bot\rangle\}\\
\mathsf{f}(\mathsf{relabel}^{e,e'}(x),y) &\to \mathsf{f}^\star(x,y) && \langle e,e'\rangle \in \{\langle\bot,h\rangle,\langle\bot,i\rangle\}\\
\mathsf{f}(x,\mathsf{relabel}^{\bot,c}(y)) &\to \mathsf{f}^\star(x,y)\\
\mathsf{f}(x,\mathsf{relabel}^{\bot,c}(y)) &\to \mathsf{f}(x,y) && \langle e,e'\rangle \in L^\pi(R_3)\\
\mathsf{s}(\mathsf{relabel}^{e,e'}(x)) &\to \mathsf{s}(x) && \langle e,e'\rangle \in L^\pi(R_3)\\
\mathsf{top}(\mathsf{relabel}^{e,e'}(x)) &\to \mathsf{top}(x) && \langle e,e'\rangle \in L^\pi(R_3)
\end{aligned}$$

Some remarks for clarification: First, note that all $\mathsf{relabel}$ symbols disappear upon one relabeling step. Secondly, observe the overlap in, for example, the rules with left-hand side $\mathsf{f}(\mathsf{relabel}^{\bot,h}(x),y)$. If the value assigned to y is c, then a redex is created; this is witnessed by the marked symbol $\mathsf{f}^\star$ on the right. For other values for y, this is not the case. Also note that there is no rule for $t = \mathsf{f}(\mathsf{relabel}^{i,h}(x),y)$. This is because when the left argument of f is interpreted as i, then t forms a redex, and so f should be marked. Definition 6.2 does not allow $\mathsf{relabel}$ symbols to commute with redex symbols. Intuitively, a $\mathsf{relabel}$ symbol is a witness of a rewrite step which we do not want to occur inside other redexes, as we want to model outermost terminination. However, more technically, sometimes illegal (i.e., non-outermost) $\mathsf{relabel}$ steps are allowed. This is illustrated in Example 6.11. The point is that by preventing $\mathsf{relabel}$ symbols to commute with redex symbols, for local completeness (Theorem 6.12) it is as if illegal steps never happened.

Remark 6.5. We elaborate on the role of the element a in $\mathsf{relabel}^{a,a'}$. Whenever the application of a rule $C[\ell\sigma] \to C[r\sigma]$ changes the interpretation, i.e., $[\![\ell\sigma]\!] \neq [\![r\sigma]\!]$, then a symbol $\mathsf{relabel}^{[\![\ell\sigma]\!],[\![r\sigma]\!]}$ is inserted. A term of the form $\mathsf{relabel}^{a,a'}(t')$ can be thought of as a witness of a rewrite step $t \to t'$ causing a change of interpretation from $a = [\![t]\!]$ to $a' = [\![t']\!]$. This change of the value then needs to be propagated upward to update the labels accordingly, using the relabeling rules from $\uparrow_2^\pi R$. At first sight, the value a in $\mathsf{relabel}^{a,a'}(t)$ seems redundant for relabeling: why would we store the previous value? However, the label a is important in order to restrict the number of applicable rules, and to have a bound on the number of relabeling steps. To see this, consider the system R_8 with single rewrite rule:

$$\mathsf{f}(\mathsf{g}(\mathsf{f}(x))) \to \mathsf{d} \tag{R_8}$$

and the algebra $\mathcal{A}_8 = \{\bot, f, gf\}$ with $[\![\mathsf{f}]\!](x) = f$ for all $x \in \mathcal{A}_8$, $[\![\mathsf{g}]\!](f) = gf$, $[\![\mathsf{g}]\!](x) = \bot$ for all $x \neq f$, and $[\![\mathsf{d}]\!] = \bot$. We employ minimal labeling, that is, only $\pi_\mathsf{f}(gf) = \star$, and all the other symbols are unlabeled.

The dynamic labeling $\uparrow^\pi R_8$ gives rise to two labelings of the original rule:

$$\mathsf{f}^\star(\mathsf{g}(\mathsf{f}(x))) \to \mathsf{relabel}^{f,\bot}(\mathsf{d}) \tag{6.1}$$

$$\mathsf{f}^\star(\mathsf{g}(\mathsf{f}^\star(x))) \to \mathsf{relabel}^{f,\bot}(\mathsf{d}) \tag{6.2}$$

And, among the fourteen rules in $\uparrow^\pi R_8$ for updating labels, we find the following two:

$$\mathsf{g}(\mathsf{relabel}^{f,\bot}(x)) \to \mathsf{relabel}^{gf,\bot}(\mathsf{g}(x)) \tag{6.3}$$

$$\mathsf{g}(\mathsf{relabel}^{gf,\bot}(x)) \to \mathsf{g}(x) \tag{6.4}$$

Now consider the term $t = \mathsf{top}(\mathsf{g}(\cdots(\mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{f}^\star(\mathsf{g}(\mathsf{f}(\mathsf{d}))))))))))$, and the rewrite sequence:

$$\begin{aligned} t &\to_{6.1,\mu} \mathsf{top}(\mathsf{g}(\cdots(\mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{relabel}^{f,\bot}(\mathsf{d})))))))\\ &\to_{6.3,\mu} \mathsf{top}(\mathsf{g}(\cdots(\mathsf{g}(\mathsf{g}(\mathsf{relabel}^{gf,\bot}(\mathsf{g}(\mathsf{d})))))))\\ &\to_{6.4,\mu} \mathsf{top}(\mathsf{g}(\cdots(\mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{d})))))) \end{aligned}$$

After an application of (6.1), relabeling takes two steps, resulting in a correctly labeled term.

In the alternative, let us say 'forgetful' version of dynamic labeling, where the 'from' value a in symbols $\mathsf{relabel}^{a,b}$ is omitted, the rules (6.1)–(6.4) look like this:

$$\mathsf{f}^\star(\mathsf{g}(\mathsf{f}(x))) \to \mathsf{relabel}^{\bot}(\mathsf{d}) \tag{6.1′}$$

$$\mathsf{f}^\star(\mathsf{g}(\mathsf{f}^\star(x))) \to \mathsf{relabel}^{\bot}(\mathsf{d}) \tag{6.2′}$$

$$\mathsf{g}(\mathsf{relabel}^{\bot}(x)) \to \mathsf{relabel}^{\bot}(\mathsf{g}(x)) \tag{6.3′}$$

$$\mathsf{g}(\mathsf{relabel}^{\bot}(x)) \to \mathsf{g}(x) \tag{6.4′}$$

Due to the overlap in rules (6.3′) and (6.4′), the resulting μTRS has a rewrite sequence from t where the symbol $\mathsf{relabel}^{\bot}$ goes up all the way to the top:

$$\begin{aligned} t &\to_{6.1',\mu} \mathsf{top}(\mathsf{g}(\cdots(\mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{relabel}^{\bot}(\mathsf{d})))))))\\ &\to_{6.3',\mu} \mathsf{top}(\mathsf{g}(\cdots(\mathsf{g}(\mathsf{g}(\mathsf{relabel}^{\bot}(\mathsf{g}(\mathsf{d})))))))\\ &\to_{6.3',\mu} \mathsf{top}(\mathsf{g}(\cdots(\mathsf{g}(\mathsf{relabel}^{\bot}(\mathsf{g}(\mathsf{g}(\mathsf{d})))))))\\ &\to_{6.3',\mu} \cdots \end{aligned}$$

From the following lemma it follows that every $\mathsf{relabel}$ symbol can be rewritten at most $\delta_{\mathcal{A}}(R)$ times (before it vanishes). By rewriting a '$\mathsf{relabel}$ symbol' we refer to a notion of residuals that extends the usual definition of orthogonal projection [Ter03] with a concept suggested by the definition of $\uparrow_2^\pi R$: Whenever we have a rule of the form:

$$\mathsf{f}^\lambda(\vec{x}, \mathsf{relabel}^{a,a'}(lab(t)), \vec{z}) \to \mathsf{relabel}^{b,b'}(\mathsf{f}^{\lambda'}(\vec{x}, lab(t), \vec{z}))$$

then we call $\mathsf{relabel}^{b,b'}$ in the right-hand side a residual of $\mathsf{relabel}^{a,a'}$ in the left-hand side.

Lemma 6.6. *Let R be a TRS over Σ, and let $\langle \mathcal{A}, \pi, \Sigma^{red}\rangle$ be a C-labeling for R. We define the relation $\rightsquigarrow \subseteq L^\pi(R) \times L^\pi(R)$ by:*

$$\langle b, b'\rangle \rightsquigarrow \langle [\![\mathsf{f}]\!](\vec{a}, b, \vec{c}), [\![\mathsf{f}]\!](\vec{a}, b', \vec{c})\rangle \qquad \text{for every } \mathsf{f} \in \Sigma,\ \langle b, b'\rangle \in L^\pi(R),\ \vec{a} \cdot b \cdot \vec{c} \in \mathcal{A}^{\sharp\mathsf{f}},\ \mathsf{f}^{\pi_{\mathsf{f}}(\vec{a},b,\vec{c})} \in lab(\Sigma_{\mathsf{top}}) \setminus \Sigma^{red}$$

Then $\rightsquigarrow$ is well-founded and every $\rightsquigarrow$ path has length $\leq \delta_{\mathcal{A}}(R)$.

Proof. By definition of value-change pairs we have that for every pair $\langle b, b'\rangle \in L^\pi(R)$ there exists a rule $\ell \to r \in R$ and assignment $\alpha : Var(\ell) \to \mathcal{A}$ such that $\langle [\![\ell, \alpha]\!], [\![r, \alpha]\!]\rangle \rightsquigarrow^* \langle b, b'\rangle$.

Assume, to arrive at a contradiction, there exists a sequence

$$\langle [\![\ell, \alpha]\!], [\![r, \alpha]\!]\rangle = \langle b_0, b_0'\rangle \rightsquigarrow \langle b_1, b_1'\rangle \rightsquigarrow \ldots \rightsquigarrow \langle b_m, b_m'\rangle$$

with $m > \delta_{\mathcal{A}}(R)$. For $i = 0, 1, \ldots, m$ we construct thin contexts D_i and assignments $\alpha_i : Var(D_i) \to \mathcal{A}$ such that $b_i = [\![D_i[\ell], \alpha_i]\!]$ and $b_i' = [\![D_i[r], \alpha_i]\!]$. We begin with $D_0 = \Box$ and $\alpha_0 = \alpha$. Then we have $b_0 = [\![\ell, \alpha]\!]$ and $b_0' = [\![r, \alpha]\!]$. For $i = 1, \ldots, m$ there exist $\mathsf{f}_i \in \Sigma$, and $\vec{a_i} \cdot b_{i-1} \cdot \vec{c_i} \in \mathcal{A}^{\sharp\mathsf{f}_i}$ such that $b_i = [\![\mathsf{f}_i]\!](\vec{a_i}, b_{i-1}, \vec{c_i})$ and $b_i' = [\![\mathsf{f}_i]\!](\vec{a_i}, b_{i-1}', \vec{c_i})$. We pick

fresh variables $\vec{x_i}$ and $\vec{z_i}$ with $|\vec{x_i}| = |\vec{a_i}|$ and $|\vec{z_i}| = |\vec{c_i}|$, and define $D_i = \mathsf{f}_i(\vec{x_i}, D_{i-1}, \vec{z_i})$, and α_i is α_{i-1} extended by mapping variables $\vec{x_i}$ to the corresponding $\vec{a_i}$ and $\vec{z_i}$ to $\vec{c_i}$. It follows that $b_i = [\![D_i[\ell], \alpha_i]\!]$ and $b'_i = [\![D_i[r], \alpha_i]\!]$. But then $[\![D[\ell], \alpha]\!] = b_m \neq b'_m = [\![D[r], \alpha]\!]$ which contradicts that $\delta_{\mathcal{A}}(\ell \to r)$ is the C-depth of $\ell \to r$. □

Corollary 6.7. *Every* relabel *symbol disappears at latest after having applied $\delta_{\mathcal{A}}(R)$ many relabeling rules (to this symbol).*

Proof. For every rule in $\uparrow_2^\pi R$ of the form:

$$\mathsf{f}^\lambda(\vec{x}, \mathsf{relabel}^{a,a'}(lab(t)), \vec{z}) \to \mathsf{relabel}^{b,b'}(\mathsf{f}^{\lambda'}(\vec{x}, lab(t), \vec{z}))$$

we have that $\langle a, a' \rangle \rightsquigarrow \langle b, b' \rangle$. □

For the dynamic context extension, the 'intended' terms in $\mathcal{T}(lab(\Sigma), \varnothing)$ are those terms that can be obtained by correctly labeling terms in $\mathcal{T}(\Sigma, \varnothing)$. For the purpose of adapting this definition to dynamic labeling, we enrich the (unlabeled) signature Σ to Σ_+:

$$\Sigma_+ = \Sigma \cup \{\mathsf{relabel}^a \mid \langle a, a' \rangle \in L^\pi(R)\}$$

and extend the C-labeling to Σ_+ by:

$$\pi_{\mathsf{relabel}^b}(b') = b' \qquad\qquad [\![\mathsf{relabel}^b]\!] = b$$

for all $b, b' \in \mathcal{A}$ such that $\langle b, c \rangle \in L^\pi(R)$ for some $c \in \mathcal{A}$. Then labeled symbols are identified by $(\mathsf{relabel}^a)^{a'} = \mathsf{relabel}^{a,a'}$.

We obtain the following lemma:

Lemma 6.8. *Let R be a TRS over Σ, and let $\langle \mathcal{A}, \pi, \Sigma^{red} \rangle$ be a C-labeling for R. Whenever we have a ground term s of the form:*

$$s = lab(C[\mathsf{f}(s_1, \ldots, \mathsf{relabel}^a(t), \ldots, s_n)])$$

with $\langle a, a' \rangle \in L^\pi(R)$, $a' = [\![t]\!]$, and where the displayed relabel *symbol is at a μ-replacing position, then one of the following steps applies:*

$$s \to_{\uparrow_2^\pi R, \mu} lab(C[\mathsf{relabel}^b(\mathsf{f}(s_1, \ldots, t, \ldots, s_n))]) \tag{6.5}$$

$$s \to_{\uparrow_2^\pi R, \mu} lab(C[\mathsf{f}(s_1, \ldots, t, \ldots, s_n)]) \tag{6.6}$$

where $b = [\![\mathsf{f}(s_1, \ldots, \Box, \ldots, s_n), \Box \mapsto a]\!]$.

Proof. Let $b' = [\![\mathsf{f}(s_1, \ldots, t, \ldots, s_n)]\!]$. Note that $b' = [\![\mathsf{f}(s_1, \ldots, \Box, \ldots, s_n), \Box \mapsto a']\!]$. Then:

$$lab(\mathsf{f}(s_1, \ldots, \mathsf{relabel}^a(t), \ldots, s_n)) = \mathsf{f}^\lambda(lab(s_1), \ldots, \mathsf{relabel}^{a,a'}(lab(t)), \ldots, lab(s_n))$$

$$lab(\mathsf{relabel}^b(\mathsf{f}(s_1, \ldots, t, \ldots, s_n))) = \mathsf{relabel}^{b,b'}(\mathsf{f}^{\lambda'}(lab(s_1), \ldots, lab(t), \ldots, lab(s_n)))$$

where $\lambda = \pi_{\mathsf{f}}([\![s_1]\!], \ldots, a, \ldots, [\![s_n]\!])$ and $\lambda' = \pi_{\mathsf{f}}([\![s_1]\!], \ldots, a', \ldots, [\![s_n]\!])$.

By Definition 6.2 the dynamic labeling $\uparrow^\pi R$ contains a rule of the form:

$$\mathsf{f}^\lambda(\vec{x}, \mathsf{relabel}^{a,a'}(lab(t)), \vec{z}) \to C[\mathsf{f}^{\lambda'}(\vec{x}, lab(t), \vec{z})]$$

with $C = \Box$ or $C = \mathsf{relabel}^{b,b'}(\Box)$. Consequently we have a step of the form:

$$lab(\mathsf{f}(s_1, \ldots, \mathsf{relabel}^a(t), \ldots, s_n)) \to_{\uparrow^\pi R, \mu} lab(D[\mathsf{f}(s_1, \ldots, t, \ldots, s_n)])$$

with $D = \Box$ or $D = \mathsf{relabel}^b(\Box)$.

Now the claim follows since $s = lab(C, \Box \mapsto b)[lab(\mathsf{f}(s_1, \ldots, \mathsf{relabel}^a(t), \ldots, s_n))]$. □

Lemma 6.9. *Let R be a TRS over Σ, and let $\langle \mathcal{A}, \pi, \Sigma^{red} \rangle$ be a sound C-labeling for R. Let $s, t \in \mathcal{T}(\Sigma, \varnothing)$ be ground terms such that $s \xrightarrow{out}_R t$. Then, for some $m \leq \delta_{\mathcal{A}}(R)$:*

$$lab(\mathsf{top}(s)) \to_{\uparrow_1^\pi R,\mu} \cdot \to^m_{\uparrow_2^\pi R,\mu} lab(\mathsf{top}(t))$$

Proof. Assume $s \xrightarrow{out}_{R,p} t$ for some position $p \in Pos(s)$. Then there exists a rule $\ell \to r \in R$, a context C with $root(C|_p) = \Box$ and a ground substitution $\sigma : \mathcal{X} \to \mathcal{T}(\Sigma, \varnothing)$ such that $s = C[\ell\sigma]$ and $t = C[r\sigma]$. Let $\overline{C}_a = lab(\mathsf{top}(C), \Box \mapsto a)$ and $\overline{\sigma} = lab(\sigma)$, then by Lemma 3.6 we obtain:

$$lab(\mathsf{top}(s)) = lab(\mathsf{top}(C[\ell\sigma])) = \overline{C}_{[\![\ell\sigma]\!]}[lab(\ell\sigma)] = \overline{C}_{[\![\ell\sigma]\!]}[lab(\ell, [\![\sigma]\!])\,\overline{\sigma}] \tag{6.7}$$

and, likewise:

$$lab(\mathsf{top}(t)) = \overline{C}_{[\![r\sigma]\!]}[lab(r, [\![\sigma]\!])\,\overline{\sigma}] \tag{6.8}$$

By definition of dynamic labeling, one of the following rules is in $\uparrow_1^\pi R$:

$$lab(\ell, [\![\sigma]\!]) \to lab(r, [\![\sigma]\!]) \qquad \text{if } [\![\ell\sigma]\!] = [\![r\sigma]\!] \tag{6.9}$$

$$lab(\ell, [\![\sigma]\!]) \to \mathsf{relabel}^{[\![\ell\sigma]\!],[\![r\sigma]\!]}(lab(r, [\![\sigma]\!])) \qquad \text{if } [\![\ell\sigma]\!] \neq [\![r\sigma]\!] \tag{6.10}$$

(Note that $[\![\ell, [\![\sigma]\!]]\!] = [\![\ell\sigma]\!]$, and $[\![r, [\![\sigma]\!]]\!] = [\![r\sigma]\!]$ by Lemma 3.2.)

In case $[\![\ell\sigma]\!] = [\![r\sigma]\!]$, no relabeling is needed and we take $m = 0$:

$$lab(\mathsf{top}(s)) \overset{(6.7)}{=} \overline{C}_{[\![\ell\sigma]\!]}[lab(\ell, [\![\sigma]\!])\,\overline{\sigma}] \to^{(6.9)}_{\uparrow_1^\pi R,\mu} \overline{C}_{[\![\ell\sigma]\!]}[lab(r, [\![\sigma]\!])\,\overline{\sigma}] \overset{6.8}{=} lab(\mathsf{top}(t))$$

If $[\![\ell\sigma]\!] \neq [\![r\sigma]\!]$, we get:

$$\begin{aligned} lab(\mathsf{top}(s)) &\overset{(6.7)}{=} \overline{C}_{[\![\ell\sigma]\!]}[lab(\ell, [\![\sigma]\!])\,\overline{\sigma}] \\ &\to^{(6.10)}_{\uparrow_1^\pi R,\mu} \overline{C}_{[\![\ell\sigma]\!]}[\mathsf{relabel}^{[\![\ell\sigma]\!],[\![r\sigma]\!]}(lab(r, [\![\sigma]\!]))\,\overline{\sigma}] = lab(\mathsf{top}(C[\mathsf{relabel}^{[\![\ell\sigma]\!]}(r\sigma)])) \end{aligned}$$

By Lemma 6.8 the $\mathsf{relabel}$ symbol can 'walk' upward until it disappears, and at the latest it vanishes when it meets top. Hence we have:

$$lab(\mathsf{top}(s)) \to_{\uparrow_1^\pi R,\mu} lab(\mathsf{top}(C[\mathsf{relabel}^{[\![\ell\sigma]\!]}(r\sigma)])) \to^m_{\uparrow_2^\pi R,\mu} lab(\mathsf{top}(C[r\sigma]))$$

for some $m \leq \delta_{\mathcal{A}}(R)$ by Lemma 6.6. □

Theorem 6.10. *Let R be a TRS over Σ, and $\langle \mathcal{A}, \pi, \Sigma^{red} \rangle$ a sound C-labeling for R. Then R is outermost ground terminating if $\uparrow^\pi R$ is terminating.*

Proof. Assume that R admits an infinite outermost rewrite sequence of ground terms:

$$t_1 \xrightarrow{out}_R t_2 \xrightarrow{out}_R t_3 \xrightarrow{out}_R \dots$$

Then from Lemma 6.9 it follows that $\uparrow^\pi R$ admits an infinite rewrite sequence:

$$lab(\mathsf{top}(t_1)) \to^+_{\uparrow^\pi R,\mu} lab(\mathsf{top}(t_2)) \to^+_{\uparrow^\pi R,\mu} lab(\mathsf{top}(t_3)) \to^+_{\uparrow^\pi R,\mu} \dots$$ □

We note that Theorem 6.10 can be strengthened by weakening the termination of $\uparrow^\pi R$ to local termination of $\uparrow^\pi R$ on the set of correctly labeled ground terms without $\mathsf{relabel}$ symbols. Let us denote this set by $lab(\mathcal{T}(\Sigma, \varnothing))$:

$$lab(\mathcal{T}(\Sigma, \varnothing)) = \{lab(t) \mid t \in \mathcal{T}(\Sigma, \varnothing)\}$$

Theorem 6.12 below states that dynamic labeling is complete with respect to local termination on $lab(\mathcal{T}(\Sigma, \varnothing))$. More precisely, outermost ground termination of R implies termination of $\uparrow^\pi R$ on $lab(\mathcal{T}(\Sigma, \varnothing))$. The following example helps to understand the proof

of that theorem; it illustrates that even when starting from terms in $lab(\mathcal{T}(\Sigma,\varnothing))$, not every rewrite step in $\uparrow^\pi R$ corresponds to an outermost step in R.

Example 6.11. Let R consist of the following rules:

$$\mathsf{f}(\mathsf{b},x)\to\mathsf{a} \qquad \mathsf{f}(x,\mathsf{b})\to\mathsf{a}$$
$$\mathsf{f}(\mathsf{b},\mathsf{b})\to\mathsf{f}(\mathsf{a},\mathsf{a}) \qquad \mathsf{a}\to\mathsf{b}$$

Moreover, let $\mathcal{A}=\{\bot,b\}$ with $[\![\mathsf{a}]\!]=\bot$, $[\![\mathsf{b}]\!]=b$, and $[\![\mathsf{f}]\!](x,y)=\bot$ for all $x,y\in\mathcal{A}$. Labeling symbols with the value of their arguments, we obtain for $\uparrow_1^\pi R$:

$$\mathsf{f}^{b,\bot}(\mathsf{b},x)\to\mathsf{a} \qquad \mathsf{f}^{\bot,b}(x,\mathsf{b})\to\mathsf{a}$$
$$\mathsf{f}^{b,b}(\mathsf{b},x)\to\mathsf{a} \qquad \mathsf{f}^{b,b}(x,\mathsf{b})\to\mathsf{a}$$
$$\mathsf{f}^{b,b}(\mathsf{b},\mathsf{b})\to\mathsf{f}^{\bot,\bot}(\mathsf{a},\mathsf{a}) \qquad \mathsf{a}\to\mathsf{relabel}^{\bot,b}(\mathsf{b})$$

and for $\uparrow_2^\pi R$:

$$\mathsf{f}^{\bot,\bot}(\mathsf{relabel}^{\bot,b}(x),y)\to\mathsf{f}^{b,\bot}(x,y) \qquad \mathsf{f}^{\bot,\bot}(x,\mathsf{relabel}^{\bot,b}(y))\to\mathsf{f}^{\bot,b}(x,y)$$

where $\Sigma^{red}=\{\mathsf{a},\mathsf{f}^{b,\bot},\mathsf{f}^{\bot,b},\mathsf{f}^{b,b}\}$. Then we obtain the following rewrite sequence in $\uparrow^\pi R$:

$$\begin{aligned} lab(\mathsf{f}(\mathsf{a},\mathsf{a})) &= \mathsf{f}^{\bot,\bot}(\mathsf{a},\mathsf{a})\\ &\to_{\uparrow_1^\pi R,\mu} \mathsf{f}^{\bot,\bot}(\mathsf{relabel}^{\bot,b}(\mathsf{b}),\mathsf{a})\\ &\to_{\uparrow_1^\pi R,\mu} \mathsf{f}^{\bot,\bot}(\mathsf{relabel}^{\bot,b}(\mathsf{b}),\mathsf{relabel}^{\bot,b}(\mathsf{b}))\\ &\to_{\uparrow_2^\pi R,\mu} \mathsf{f}^{b,\bot}(\mathsf{b},\mathsf{relabel}^{\bot,b}(\mathsf{b})) \end{aligned}$$

The second step in this rewrite sequence does not correspond to an outermost step. Nevertheless, Theorem 6.12 states that such 'illegal' steps do not harm completeness of the transformation. The reason is that if the relabeling rules create a redex above some $\mathsf{relabel}$ symbol, then this $\mathsf{relabel}$ symbol is prevented from further propagating its information upward (until it becomes μ-replacing again). The crucial point is that above $\mathsf{relabel}$ symbols the labels are unchanged, thus as if the step would not have taken place. Moreover, it is essential that $\uparrow^\pi R$ prohibits $\mathsf{relabel}$ to propagate over symbols from Σ^{red}. For instance, in the above example $\uparrow^\pi R$ does not contain a rule of the form:

$$\mathsf{f}^{b,\bot}(x,\mathsf{relabel}^{\bot,b}(y))\to\mathsf{f}^{b,b}(x,y) \tag{6.11}$$

This rule would cause non-termination:

$$\begin{aligned} \mathsf{f}^{\bot,\bot}(\mathsf{a},\mathsf{a}) &\to_\mu{}^2 \mathsf{f}^{\bot,\bot}(\mathsf{relabel}^{\bot,b}(\mathsf{b}),\mathsf{relabel}^{\bot,b}(\mathsf{b}))\\ &\to_\mu \mathsf{f}^{b,\bot}(\mathsf{b},\mathsf{relabel}^{\bot,b}(\mathsf{b}))\\ &\overset{(6.11)}{\to}_\mu \mathsf{f}^{b,b}(\mathsf{b},\mathsf{b})\\ &\to_\mu \mathsf{f}^{\bot,\bot}(\mathsf{a},\mathsf{a}) \end{aligned}$$

Theorem 6.12. *Let R be a left-linear TRS over Σ, and $\langle\mathcal{A},\pi,\Sigma^{red}\rangle$ a complete, maximal, and core C-labeling for R. Then $\uparrow^\pi R$ is terminating on the set of terms $lab(\mathcal{T}(\Sigma,\varnothing))$ if R is outermost ground terminating.*

Proof. Define $T=lab(\mathcal{T}(\Sigma,\varnothing))$, and $\hookrightarrow=(\to_{\uparrow_1^\pi R,\mu}\cdot\to^*_{\uparrow_2^\pi R,\mu})\cap(T\times T)$. Note that the relation $\hookrightarrow$ is restricted to terms which contain no $\mathsf{relabel}$ symbols. Hence always the maximal number of relabeling rules is applied. It is clear that each $\hookrightarrow$ rewrite step corresponds to

an outermost rewrite step in the original TRS R. Therefore it suffices to show that any infinite rewrite sequence $t = t_0 \to_{\uparrow^\pi R,\mu} t_1 \to_{\uparrow^\pi R,\mu} \dots$ gives rise to an infinite rewrite sequence $t = s_0 \hookrightarrow s_1 \hookrightarrow \dots$. We prove the claim by a kind of standardization of reductions. We first classify the rules from $\uparrow^\pi R$:

$$lab(\ell, \alpha) \to lab(r, \alpha) \tag{$c1$}$$

$$lab(\ell, \alpha) \to \mathsf{relabel}^{[\![\ell,\alpha]\!],[\![r,\alpha]\!]}(lab(r, \alpha)) \tag{$c2$}$$

$$\mathsf{f}^{\pi_\mathsf{f}(\vec{a},b,\vec{c})}(\vec{x}, \mathsf{relabel}^{b,b'}(y), \vec{z}) \to \mathsf{f}^{\pi_\mathsf{f}(\vec{a},b',\vec{c})}(\vec{x}, y, \vec{z}) \tag{$c3$}$$

$$\mathsf{f}^{\pi_\mathsf{f}(\vec{a},b,\vec{c})}(\vec{x}, \mathsf{relabel}^{b,b'}(y), \vec{z}) \to \mathsf{relabel}^{d,d'}(\mathsf{f}^{\pi_\mathsf{f}(\vec{a},b',\vec{c})}(\vec{x}, y, \vec{z})) \tag{$c4$}$$

For $i = 0, 1, \dots$, we analyse the steps $t_i \to_{\uparrow^\pi R,\mu} t_{i+1}$ and construct $s_0 \hookrightarrow s_1 \hookrightarrow \dots \hookrightarrow s_j$ in such a way that $s_j \xrightarrow{\neg\mu}{}^*_{c2,c4} t_{i+1}$ where we use $\xrightarrow{\neg\mu}_{c2,c4}$ to denote standard term rewriting ignoring the replacement map μ, and using rules from $(c2)$ and $(c4)$ only. Observe that then the maximal prefix C_{i+1} of t_{i+1} not containing $\mathsf{relabel}$ symbols is also a prefix of s_j (since everything changed by $(c2)$ and $(c4)$ is 'hidden' inside a $\mathsf{relabel}$ symbol). We begin with $t = s_0$, and $i = j = 0$. For $i = 0, 1, \dots$, we consider the step $\tau_i : t_i \to_{\uparrow^\pi R,\mu} t_{i+1}$.

If τ_i is a step with respect to a rule from:

- $(c2)$ or $(c4)$, then we append τ_i to the rewrite sequence $s_j \xrightarrow{\neg\mu}{}^*_{c2,c4} t_i$ yielding $s_j \xrightarrow{\neg\mu}{}^*_{c2,c4} t_{i+1}$. Note that this leaves the $\hookrightarrow$-rewrite sequence $s_0 \hookrightarrow^* s_j$ untouched.
- $(c1)$, then the pattern of τ_i lies entirely in C_i which is also prefix of s_j. Then we append τ_i to $s_0 \hookrightarrow^* s_j$ (using left-linearity of R) yielding $s_0 \hookrightarrow^* s_j \overset{\tau_i}{\hookrightarrow} s_{j+1}$. We have $s_{j+1} \xrightarrow{\neg\mu}{}^*_{c2,c4} t_{i+1}$ by orthogonal projection of the steps $s_j \xrightarrow{\neg\mu}{}^*_{c2,c4} t_i$ over $s_j \overset{\tau_i}{\hookrightarrow} s_{j+1}$ (all steps in $s_j \xrightarrow{\neg\mu}{}^*_{c2,c4} t_i$ are below the prefix C_i).
- $(c3)$, then a $\mathsf{relabel}$ symbol 'disappears'. We can trace this symbol back to a sequence of steps $\sigma_i : s_j \xrightarrow{\neg\mu}_{c2} \cdot \xrightarrow{\neg\mu}{}^+_{c4} s'_j$, that is, it must have been created in s_j by a $(c2)$ step, followed by a number of $(c4)$ steps. We combine σ_i and τ_i to a $\hookrightarrow$ step, yielding $s_0 \hookrightarrow^* s_j \overset{\sigma_i \cdot \tau_i}{\hookrightarrow} s_{j+1}$. Then $s_{j+1} \xrightarrow{\neg\mu}{}^*_{c2,c4} t_{i+1}$ as the remaining steps from $s_j \xrightarrow{\neg\mu}{}^*_{c2,c4} t_i$ are not harmed by the permutation (performing σ_i first).

It remains to be shown that the constructed sequence $s_0 \hookrightarrow s_1 \hookrightarrow s_2 \hookrightarrow \dots$ is infinite. This follows from the fact that an infinite number of steps in $t_0 \to_{\uparrow^\pi R,\mu} t_1 \to_{\uparrow^\pi R,\mu} \dots$ must be of type $(c1)$ or $(c3)$. This is a direct consequence of the fact that $\to_{c2,c4}$ is terminating (with every step the prefix in which rewriting is allowed gets smaller). □

The following example demonstrates why the completeness result for dynamic labeling (Theorem 6.12) is restricted to the set $lab(\mathcal{T}(\Sigma, \varnothing))$ of correctly labeled terms which do not contain $\mathsf{relabel}$ symbols. The point is that, although the original TRS is outermost terminating the transformed system may in general be non-terminating due to the existence of 'non-reachable' terms.

Example 6.13. We consider the following term rewriting system R:

$$\mathsf{a} \to \mathsf{b} \qquad \mathsf{f}(\mathsf{b}, y) \to \mathsf{b} \qquad \mathsf{f}(\mathsf{c}, y) \to \mathsf{h}(\mathsf{f}(y, y))$$
$$\mathsf{h}(\mathsf{f}(x, \mathsf{b})) \to \mathsf{b} \qquad \mathsf{h}(\mathsf{f}(x, \mathsf{c})) \to \mathsf{b}$$

We explain why this TRS is outermost ground terminating. Without the rule $\rho : \mathsf{f}(\mathsf{c}, y) \to \mathsf{h}(\mathsf{f}(y, y))$, the system would even be terminating. Now note that the rule ρ can only be

applied once to each occurrence of $\mathsf{f}(\mathsf{c},\Box)$ since $\mathsf{h}(\mathsf{f}(t,t)) \to^* \mathsf{h}(\mathsf{f}(\mathsf{c},t'))$ implies that $t = \mathsf{c}$, and then the rule $\mathsf{h}(\mathsf{f}(x,\mathsf{c})) \to \mathsf{b}$ has priority by the strategy of outermost rewriting.

We define a maximal, complete, core C-labeling $\langle \mathcal{A}, \pi, \Sigma^{red}\rangle$ for R (isomorphic to the result of the construction given in the next section) where the algebra $\mathcal{A} = \{bc, fbc, \bot\}$ with the interpretation function defined by:

$$\llbracket \mathsf{f} \rrbracket(x, bc) = fbc \qquad \llbracket \mathsf{b} \rrbracket = \llbracket \mathsf{c} \rrbracket = bc$$
$$\llbracket \mathsf{f} \rrbracket(x, y) = \bot \qquad \llbracket \mathsf{a} \rrbracket = \llbracket \mathsf{h} \rrbracket(x) = \bot$$

for all $x, y \in \mathcal{A}$ with $y \neq bc$, and with $\Sigma^{red} = \{\mathsf{a}, \mathsf{f}^{bc}, \mathsf{h}^{fbc}\}$.

The dynamic labeling $\uparrow^\pi R$ of R with respect to this C-labeling then includes the rules:

$$\mathsf{a} \to \mathsf{relabel}^{\bot,bc}(\mathsf{b})$$
$$\mathsf{f}^{bc,\bot}(\mathsf{c}, y) \to \mathsf{h}^{\bot}(\mathsf{f}^{\bot,\bot}(y, y))$$
$$\mathsf{f}^{\bot,\bot}(\mathsf{relabel}^{\bot,bc}(x), y) \to \mathsf{f}^{bc,\bot}(x, y)$$

Now the context-sensitive TRS $\uparrow^\pi R$ admits the following infinite rewrite sequence:

$$\begin{aligned}\mathsf{f}^{bc,\bot}(\mathsf{c}, \mathsf{relabel}^{\bot,bc}(\mathsf{c})) &\to_{\uparrow^\pi R,\mu} \mathsf{h}^{\bot}(\mathsf{f}^{\bot,\bot}(\mathsf{relabel}^{\bot,bc}(\mathsf{c}), \mathsf{relabel}^{\bot,bc}(\mathsf{c})))\\ &\to_{\uparrow^\pi R,\mu} \mathsf{h}^{\bot}(\mathsf{f}^{bc,\bot}(\mathsf{c}, \mathsf{relabel}^{\bot,bc}(\mathsf{c})))\\ &\to_{\uparrow^\pi R,\mu} \cdots\end{aligned}$$

Observe that this anomaly is caused by the subterm $\mathsf{relabel}^{\bot,bc}(c)$, which is not reachable from any term in $lab(\mathcal{T}(\Sigma,\varnothing))$.

Remark 6.14. Theorem 6.12 states completeness of dynamic labeling with respect to local termination on the set of terms $lab(\mathcal{T}(\Sigma,\varnothing))$. We briefly indicate how the theorem can be generalized to termination on $\mathcal{T}(lab(\Sigma),\varnothing)$ by altering the definition of $\uparrow^\pi R$. Note that $lab(\mathcal{T}(\Sigma,\varnothing)) \subsetneq \mathcal{T}(lab(\Sigma),\varnothing)$. In particular, the set $\mathcal{T}(lab(\Sigma),\varnothing)$ includes terms that are not correctly labeled. The necessary modification of the definition of dynamic labeling concerns the elimination of collapsing rules $\ell \to x$. This can be achieved by wrapping the right-hand side into $\mathsf{relabel}^{a,a}(\Box)$ even when the interpretations of the left and right-hand side are equal. Additionaly, we let the symbols $\mathsf{relabel}^{a,a}$ disappear after one relabeling step. By an application of Theorem 5.12 it then follows that termination on $lab(\mathcal{T}(\Sigma,\varnothing))$ coincides with termination on $\mathcal{T}(lab(\Sigma),\varnothing)$.

7. Constructing Suitable Algebras

We construct C-models which are able to recognize redex positions with respect to left-linear rules. The construction of C-models is similar to the construction of a deterministic tree automaton (DTA, [CDG+07]) for recognizing left-linear redexes [Com00]. A DTA is a Σ-algebra $\langle A, \llbracket\cdot\rrbracket\rangle$ with a distinguished set $A_F \subseteq A$ of final states. A term t is accepted by the automaton whenever $\llbracket t\rrbracket \in A_F$. A difference with the construction of a DTA is that for the construction of a C-model we do not distinguish final and non-final states, but instead have a family of functions $isRedex_\mathsf{f} : A^{\sharp\mathsf{f}} \to Bool$ for indicating the presence of a redex.

Definition 7.1 (Redex-algebra)**.** A *redex-algebra* $\langle \mathcal{A}, isRedex \rangle$ consists of a Σ-algebra $\mathcal{A}$ together with a family $\{isRedex_{\mathsf{f}}\}_{\mathsf{f}\in\Sigma}$ of functions $isRedex_{\mathsf{f}} : \mathcal{A}^{\sharp\mathsf{f}} \to Bool$. The *language of* $\mathcal{A}$ is the set:

$$\mathcal{L}(\mathcal{A}) = \{\mathsf{f}(t_1,\ldots,t_n) \in \mathcal{T}(\Sigma,\varnothing) \mid isRedex_{\mathsf{f}}([\![t_1]\!],\ldots,[\![t_n]\!]) = true\}$$

Let R be a TRS. A redex-algebra $\mathcal{A}$ is called *sound for* R whenever $t \in \mathcal{L}(\mathcal{A})$ implies that t is a redex, and $\mathcal{A}$ is called *complete for* R if for all redexes $t \in \mathcal{T}(\Sigma,\varnothing)$ we have $t \in \mathcal{L}(\mathcal{A})$.

Intuitively, a redex-algebra needs to 'remember' only the subterms $t_1,\ldots,t_n$ and not the term $\mathsf{f}(t_1,\ldots,t_n)$ itself. To see this, consider the one-rule system:

$$\mathsf{f}(\mathsf{g}(x)) \to \mathsf{a}$$

A tree automaton which can recognize redex positions for this TRS needs at least three states: one for indicating a redex $\mathsf{f}(\mathsf{g}(\ldots))$, one for $\mathsf{g}(\ldots)$, and one garbage state. For redex-algebras two states suffice: one state for $\mathsf{g}(\ldots)$ and one for garbage, and then we use $isRedex_{\mathsf{f}}(\mathsf{g}(\ldots)) = true$ and *false*, otherwise.

We now describe a syntactical construction of redex-algebras. The algebras we construct are C-models: the C-depth of a rule $\ell \to r$ is the maximal pattern depth of a left-hand side (minus 1), since for recognizing the subterms of left-hand sides we may 'forget' all information that lies below the patterns. We first define some auxiliary functions, introduced with different notations in [KM91, HL91].

Definition 7.2. Let Σ be a signature and $\mathcal{X}$ a set of variables. We let $\bot$ be a new symbol, $\bot \notin \Sigma$, and we define $\mathcal{T}_\bot = \mathcal{T}(\Sigma \cup \{\bot\},\varnothing)$. The function $cut : \mathcal{T}(\Sigma,\mathcal{X}) \to \mathcal{T}_\bot$ is defined such that $cut(t)$ is the result of replacing all variables in a term $t \in \mathcal{T}(\Sigma,\mathcal{X})$ by $\bot$:

$$cut(x) = \bot \qquad\qquad cut(\mathsf{f}(t_1,\ldots,t_n)) = \mathsf{f}(cut(t_1),\ldots,cut(t_n))$$

We define the function $match : \mathcal{T}_\bot \times \mathcal{T}_\bot \to Bool$ such that $match(s,t) = true$ if s can be obtained from t by replacing subterms of t by $\bot$, and we let $match(s,t) = false$, otherwise. Further, we let $merge(s,t)$ be the 'most general common instance' of s and t, that is, $merge : \mathcal{T}_\bot \times \mathcal{T}_\bot \rightharpoonup \mathcal{T}_\bot$ is the partial function defined by:

$$merge(\bot,t) = merge(t,\bot) = t$$
$$merge(\mathsf{f}(s_1,\ldots,s_n),\mathsf{f}(t_1,\ldots,t_n)) = \mathsf{f}(merge(s_1,t_1),\ldots,merge(s_n,t_n))$$

Hence $merge(s,t)$ is undefined whenever there exists a position $p \in Pos(s)$ such that $root(s|_p) \in \Sigma$, $root(t|_p) \in \Sigma$, and $root(s|_p) \neq root(t|_p)$. For a term $s \in \mathcal{T}_\bot$ and a set $T \subseteq \mathcal{T}_\bot$ we define the term $shrink(s,T)$ as the largest $t \in T$ (with respect to the number of symbols) such that $match(t,s) = true$. Note that $shrink(s,T)$ is well-defined whenever T is closed under *merge* and $\bot \in T$: whenever two terms $t_1 \neq t_2$ of equal size match s then $merge(t_1,t_2)$ is larger and matches s.

Definition 7.3 (Construction of redex-algebra)**.** We define a mapping F which constructs a redex-algebra for a given TRS R. Let $F(R) = \langle \mathcal{A}, isRedex \rangle$ where $\mathcal{A}$ is the smallest set such that:

- $\bot \in \mathcal{A}$,
- $t \in \mathcal{A}$ for every proper subterm t of $cut(\ell)$ with ℓ a left-hand side of a rule in R,
- $merge(s,t) \in \mathcal{A}$ whenever $s,t \in \mathcal{A}$ and $merge(s,t)$ is defined.

The interpretation function $[\![\cdot]\!]$ of $\mathcal{A}$, and the functions *isRedex* are defined by:

$$[\![\mathsf{f}]\!](t_1,\dots,t_n) = \mathit{shrink}(\mathsf{f}(t_1,\dots,t_n),\mathcal{A})$$

$$\mathit{isRedex}_{\mathsf{f}}(t_1,\dots,t_n) = \begin{cases} \mathit{true} & \text{if } \mathit{match}(\mathit{cut}(\ell),\mathsf{f}(t_1,\dots,t_n)) = \mathit{true} \text{ for some } \ell \to r \in R \\ \mathit{false} & \text{otherwise} \end{cases}$$

for all function symbols $\mathsf{f} \in \Sigma$ with arity n, and terms $t_1,\dots,t_n$.

The core of the algebra $F(R)$, which we denote by $F(R)_c$, is called the *full redex-algebra for* R. Furthermore, let $S \subseteq R$ be the set consisting of all left-linear rules of R. Then $F(S)_c$ is called the *left-linear redex-algebra for* R.

Of course, if R is a left-linear TRS, then a left-linear redex-algebra for R also is a full redex-algebra for R. Moreover, if R is a quasi-left-linear TRS, then the minimized left-linear and full redex-algebras for R are isomorphic. Minimization of redex-algebras is introduced in the next section. We now consider two examples which illustrate that $F(R)_c$ indeed can be a proper subalgebra of $F(R)$.

Example 7.4. We consider the term rewriting system R which consists of the rules:

$$\mathsf{a}(x) \to \mathsf{f}(\mathsf{a}(x),x) \qquad \mathsf{f}(x,\mathsf{a}(y)) \to \mathsf{b} \qquad \mathsf{f}(x,\mathsf{f}(y,z)) \to \mathsf{b} \qquad \mathsf{f}(x,\mathsf{b}) \to \mathsf{b}$$

This TRS is outermost ground terminating, but it is not outermost terminating. We construct the redex-algebra $F(R) = \langle \mathcal{A}, \mathit{isRedex}\rangle$ where $\mathcal{A} = \{\bot, \mathsf{a}(\bot), \mathsf{f}(\bot,\bot), \mathsf{b}\}$ with $[\![\mathsf{a}]\!](x) = \mathsf{a}(\bot)$, $[\![\mathsf{f}]\!](x,y) = \mathsf{f}(\bot,\bot)$ and $[\![\mathsf{b}]\!] = \mathsf{b}$ for all $x,y \in \mathcal{A}$. But note that $\bot$ is not part of the core, and hence the left-linear (full) redex-algebra $F(R)_c$ contains only the elements $\{\mathsf{a}(\bot), \mathsf{f}(\bot,\bot), \mathsf{b}\}$.

Example 7.5. Consider the term rewriting system R consisting of the rules:

$$\mathsf{h}(\mathsf{h}(\mathsf{h}(x))) \to \mathsf{a} \qquad\qquad \mathsf{h}(\mathsf{h}(\mathsf{a})) \to \mathsf{h}(\mathsf{h}(\mathsf{h}(\mathsf{h}(\mathsf{a}))))$$

The domain of the redex-algebra $F(R)$ is $\{\mathsf{h}(\mathsf{h}(\bot)), \mathsf{h}(\bot), \bot, \mathsf{h}(\mathsf{a}), \mathsf{a}\}$ with the interpretation of the symbols defined by:

$$[\![\mathsf{a}]\!] = \mathsf{a} \qquad [\![\mathsf{h}]\!](\mathsf{a}) = \mathsf{h}(\mathsf{a}) \qquad [\![\mathsf{h}]\!](\mathsf{h}(\mathsf{a})) = \mathsf{h}(\mathsf{h}(\bot)) \qquad [\![\mathsf{h}]\!](\mathsf{h}(\mathsf{h}(\bot))) = \mathsf{h}(\mathsf{h}(\bot))$$

The values $\bot$ and $\mathsf{h}(\bot)$ are not part of the core, and hence the domain of the left-linear (full) redex-algebra $F(R)_c$ is $\{\mathsf{h}(\mathsf{h}(\bot)), \mathsf{h}(\mathsf{a}), \mathsf{a}\}$ with $\mathit{isRedex}_{\mathsf{h}}(\mathsf{h}(\mathsf{a})) = \mathit{isRedex}_{\mathsf{h}}(\mathsf{h}(\mathsf{h}(\bot))) = \mathit{true}$, and *false* otherwise.

The next example illustrates the use of the function *merge* in the construction of a redex-algebra.

Example 7.6. We construct the left-linear (full) redex-algebra for the TRS:

$$\mathsf{f}(x,y) \to \mathsf{a}(\mathsf{f}(\mathsf{c}(x),y)) \qquad\qquad \mathsf{a}(\mathsf{f}(\mathsf{c}(\mathsf{c}(x)),y)) \to \mathsf{e}$$
$$\mathsf{f}(x,y) \to \mathsf{b}(\mathsf{f}(x,\mathsf{c}(y))) \qquad\qquad \mathsf{b}(\mathsf{f}(x,\mathsf{c}(\mathsf{c}(y)))) \to \mathsf{e}$$

The subterms of $\mathit{cut}(\ell)$ of linear left-hand sides ℓ are:

$$S = \{\bot,\ \mathsf{f}(\mathsf{c}(\mathsf{c}(\bot)),\bot),\ \mathsf{f}(\bot,\mathsf{c}(\mathsf{c}(\bot))),\ \mathsf{c}(\mathsf{c}(\bot)),\ \mathsf{c}(\bot)\}$$

and closure of S under *merge* yields the algebra $\mathcal{A} = S \cup \{\mathsf{f}(\mathsf{c}(\mathsf{c}(\bot)), \mathsf{c}(\mathsf{c}(\bot)))\}$ with the interpretation function defined by:

$$\begin{aligned}
[\![\mathsf{a}]\!](x) = [\![\mathsf{b}]\!](x) = [\![\mathsf{e}]\!] &= \bot && \text{for all } x \in A\\
[\![\mathsf{c}]\!](\mathsf{c}(\bot)) &= \mathsf{c}(\mathsf{c}(\bot))\\
[\![\mathsf{c}]\!](\mathsf{c}(\mathsf{c}(\bot))) &= \mathsf{c}(\mathsf{c}(\bot))\\
[\![\mathsf{c}]\!](x) &= \mathsf{c}(\bot) && \text{for all } x \notin \{\mathsf{c}(\bot), \mathsf{c}(\mathsf{c}(\bot))\}\\
[\![\mathsf{f}]\!](\mathsf{c}(\mathsf{c}(\bot)), \mathsf{c}(\mathsf{c}(\bot))) &= \mathsf{f}(\mathsf{c}(\mathsf{c}(\bot)), \mathsf{c}(\mathsf{c}(\bot)))\\
[\![\mathsf{f}]\!](\mathsf{c}(\mathsf{c}(\bot)), x) &= \mathsf{f}(\mathsf{c}(\mathsf{c}(\bot)), \bot) && \text{for all } x \neq \mathsf{c}(\mathsf{c}(\bot))\\
[\![\mathsf{f}]\!](x, \mathsf{c}(\mathsf{c}(\bot))) &= \mathsf{f}(\bot, \mathsf{c}(\mathsf{c}(\bot))) && \text{for all } x \neq \mathsf{c}(\mathsf{c}(\bot))\\
[\![\mathsf{f}]\!](x, y) &= \bot && \text{otherwise}
\end{aligned}$$

For the family of *isRedex* functions we obtain:

$$\begin{aligned}
\mathit{isRedex}_{\mathsf{f}}(x, y) &= \mathit{true} && \text{for all } x, y \in A\\
\mathit{isRedex}_{\mathsf{a}}(\mathsf{f}(\mathsf{c}(\mathsf{c}(\bot)), \bot)) &= \mathit{true}\\
\mathit{isRedex}_{\mathsf{a}}(\mathsf{f}(\mathsf{c}(\mathsf{c}(\bot)), \mathsf{c}(\mathsf{c}(\bot)))) &= \mathit{true}\\
\mathit{isRedex}_{\mathsf{b}}(\mathsf{f}(\bot, \mathsf{c}(\mathsf{c}(\bot)))) &= \mathit{true}\\
\mathit{isRedex}_{\mathsf{b}}(\mathsf{f}(\mathsf{c}(\mathsf{c}(\bot)), \mathsf{c}(\mathsf{c}(\bot)))) &= \mathit{true}
\end{aligned}$$

and *isRedex* returns *false* in all remaining cases. Finally, note that the core of the constructed algebra is the algebra itself, i.e., $F(R)_c = F(R)$, because $\mathsf{e} \in \Sigma$ with $[\![\mathsf{e}]\!] = \bot$.

The following theorem states that left-linear (full) redex-algebras recognize only redex positions (at least all redex positions).

Theorem 7.7. *Let R be a TRS over Σ. The following properties hold:*

(i) *The left-linear redex-algebra for R is sound.*
(ii) *The full redex-algebra for R is complete.*
(iii) *Let $\mathcal{A}$ be the left-linear redex-algebra for R. For all $t \in \mathcal{T}(\Sigma, \varnothing)$ we have $t \in \mathcal{L}(\mathcal{A})$ if and only if t is a redex with respect to a left-linear rule in R. Hence, if R is quasi-left-linear, then the left-linear redex-algebra for R is sound and complete.*

Proof. We prove (ii) and leave (i) and (iii) to the reader.

Let $\langle \mathcal{A}, \mathit{isRedex} \rangle = F(R)_c$ with F the mapping defined in Definition 7.3. Let $t \in \mathcal{T}(\Sigma, \varnothing)$ be a redex with respect to a rule $\ell \to r \in R$.

We show $\mathit{match}(\mathit{cut}(\ell|_p), [\![t|_p]\!]) = \mathit{true}$ for all positions $p \in \mathit{Pos}(\ell) \setminus \{\epsilon\}$ by induction on the structure of ℓ. If $\ell|_p$ is a variable then $\mathit{cut}(\ell|_p) = \bot$, and we have $\mathit{match}(\bot, a) = \mathit{true}$ for all $a \in \mathcal{A}$. If $\ell|_p = \mathsf{f}(s_1, \ldots, s_n)$, then $t|_p = \mathsf{f}(t_1, \ldots, t_n)$ and by induction hypothesis we have $\mathit{match}(\mathit{cut}(s_i), [\![t_i]\!]) = \mathit{true}$. Hence $\mathit{match}(\mathit{cut}(\mathsf{f}(s_1, \ldots, s_n)), \mathsf{f}([\![t_1]\!], \ldots, [\![t_n]\!])) = \mathit{true}$ by definition of *cut*. Moreover, $\mathit{match}(a, t) = \mathit{true}$ implies $\mathit{match}(a, \mathit{shrink}(t, \mathcal{A})) = \mathit{true}$ for all $a \in \mathcal{A}$. By definition we have $[\![\mathsf{f}(t_1, \ldots, t_n)]\!] = \mathit{shrink}(\mathsf{f}([\![t_1]\!], \ldots, [\![t_n]\!]), \mathcal{A})$, and hence $\mathit{match}(\mathit{cut}(\mathsf{f}(s_1, \ldots, s_n)), [\![\mathsf{f}(t_1, \ldots, t_n)]\!]) = \mathit{true}$.

Let $t = \mathsf{g}(u_1, \ldots, u_m)$ and $\ell = \mathsf{g}(w_1, \ldots, w_m)$. Then we know $\mathit{match}(\mathit{cut}(w_i), [\![u_i]\!]) = \mathit{true}$. Hence $\mathit{match}(\mathit{cut}(\ell), \mathsf{g}([\![u_1]\!], \ldots, [\![u_m]\!])) = \mathit{true}$ and $t \in \mathcal{L}(\mathcal{A})$. □

8. Minimizing Algebras

In this section we are concerned with the minimization of redex-algebras. The algorithm is similar to the minimization of deterministic tree automata, see [CDG+07]. For the set of 291 TRSs of the outermost termination competition of 2008 [Ter08], the redex-algebras constructed according to Definition 7.3 have an average size of 4.6 elements. After an application of the minimization algorithm described here, the average size falls to 3.4, a reduction of 27%. This reduction has a polynomial influence on the number of rules of the transformed system.

Definition 8.1. Two core redex-algebras $\mathcal{A}_1$, $\mathcal{A}_2$ are called *equivalent* if $\mathcal{L}(\mathcal{A}_1) = \mathcal{L}(\mathcal{A}_2)$.

Lemma 8.2. *Let $\mathcal{A}_1$, $\mathcal{A}_2$ be equivalent, core redex-algebras. Then $\mathcal{A}_1$ is sound or complete if and only if $\mathcal{A}_2$ has the respective property.* □

For a given core redex-algebra we now construct a minimal equivalent algebra. The difference to the minimization of tree automata from [CDG+07] lies in the initial equivalence E_0. For tree automata this initial equivalence consists of two partitions, the final and the non-final states. In our setting two states are initially equivalent if they cannot be distinguished using the *isRedex* functions, that is, $\mathit{isRedex}_{\mathsf{f}}(\vec{x}, a, \vec{y}) = \mathit{isRedex}_{\mathsf{f}}(\vec{x}, b, \vec{y})$ for each symbol $\mathsf{f} \in \Sigma$ and each assignment of $\vec{x}$ and $\vec{y}$. This can yield any number of partitions between 1 and $|A|$.

Definition 8.3 (Minimization of redex-algebra)**.** Let $\langle \mathcal{A}, \mathit{isRedex} \rangle$ be a core redex-algebra over Σ. We define equivalence relations E_i for $i \in \mathbb{N}$ on the elements of $\mathcal{A}$. Initially two elements $a, b \in \mathcal{A}$ are equivalent, $a\ E_0\ b$, if:

$$\mathit{isRedex}_{\mathsf{f}}(\vec{x}, a, \vec{y}) = \mathit{isRedex}_{\mathsf{f}}(\vec{x}, b, \vec{y})$$

for all symbols $\mathsf{f} \in \Sigma_n$, $j \in \{1, \dots, n\}$, $\vec{x} \in \mathcal{A}^{j-1}$, and $\vec{y} \in \mathcal{A}^{n-j}$. Then for $i = 0, 1, \dots$ and $a, b \in \mathcal{A}$ we define $a\ E_{i+1}\ b$ if $a\ E_i\ b$ holds and:

$$[\![\mathsf{f}]\!](\vec{x}, a, \vec{y})\ E_i\ [\![\mathsf{f}]\!](\vec{x}, b, \vec{y})$$

for all n-ary symbols $\mathsf{f} \in \Sigma$, $j \in \{1, \dots, n\}$, $\vec{x} \in \mathcal{A}^{j-1}$, and $\vec{y} \in \mathcal{A}^{n-j}$. The process halts when $E_{i+1} = E_i$ for some $i \in \mathbb{N}$, and then we define $E = E_i$. Let $[a]$ denote the equivalence class of $a \in \mathcal{A}$ with respect to E. The *minimized redex-algebra of* $\mathcal{A}$, denoted $\mathcal{A}^{\mathit{min}}$, is defined as $\mathcal{A}^{\mathit{min}} = \langle E, [\![\cdot]\!]^E, \mathit{isRedex}^E \rangle$ where:

$$[\![\mathsf{f}]\!]^E([a_1], \dots, [a_n]) = [\mathsf{f}(a_1, \dots, a_n)]$$
$$\mathit{isRedex}^E_{\mathsf{f}}([a_1], \dots, [a_n]) = \mathit{isRedex}_{\mathsf{f}}(a_1, \dots, a_n)$$

for each symbol $\mathsf{f} \in \Sigma$ with arity n.

Lemma 8.4. *Let $\mathcal{A}$ be a core redex-algebra, then $\mathcal{A}$ is equivalent to $\mathcal{A}^{\mathit{min}}$.* □

Example 8.5. We consider the TRS R consisting of the following three rules:

$$\mathsf{f}(\mathsf{i}(\mathsf{a})) \to \mathsf{a} \qquad \mathsf{f}(\mathsf{j}(\mathsf{a})) \to \mathsf{a} \qquad \mathsf{f}(\mathsf{a}) \to \mathsf{a}$$

The left-linear (full) redex-algebra for R is $\mathcal{A} = \{\mathsf{a}, \mathsf{i}(\mathsf{a}), \mathsf{j}(\mathsf{a}), \bot\}$ with the interpretation $[\![\mathsf{a}]\!] = \mathsf{a}$, $[\![\mathsf{i}]\!](\mathsf{a}) = \mathsf{i}(\mathsf{a})$, $[\![\mathsf{j}]\!](\mathsf{a}) = \mathsf{j}(\mathsf{a})$, and the interpretation is $\bot$ in all non-listed cases; $\mathit{isRedex}_{\mathsf{f}}(x) = \mathit{true}$ for all $x \neq \bot$, and *false*, otherwise.

The minimization algorithm starts with $E_0 = \{\{\mathsf{a}, \mathsf{i}(\mathsf{a}), \mathsf{j}(\mathsf{a})\}, \{\bot\}\}$ as initial equivalence, since $\bot$ can be distinguished from the other elements ($\mathit{isRedex}_{\mathsf{f}}(\bot) = \mathit{false}$). The

first iteration of the algorithm yields $E_1 = \{\{\mathsf{a}\}, \{\mathsf{i}(\mathsf{a}), \mathsf{j}(\mathsf{a})\}, \{\bot\}\}$ as $[\![\mathsf{i}]\!](\mathsf{a}) = \mathsf{i}(\mathsf{a})$ whereas $[\![\mathsf{i}]\!](\mathsf{i}(\mathsf{a})) = [\![\mathsf{i}]\!](\mathsf{j}(\mathsf{a})) = \bot$. The elements $\mathsf{i}(\mathsf{a})$ and $\mathsf{j}(\mathsf{a})$ are indistinguishable, and so in the second iteration we obtain $E_2 = E_1$. Thus the elements $\mathsf{i}(\mathsf{a})$ and $\mathsf{j}(\mathsf{a})$ are identified and we obtain an algebra that has one element less than the algebra we started with.

9. Constructing Minimal and Maximal C-labelings

In the previous sections we have constructed and minimized redex-algebras for recognizing redex positions. For completing the transformation we still have to explain how to construct C-labelings from the redex-algebras.

In minimal labeling symbols are marked with a $\star$ if they correspond to redex positions and stay unlabeled otherwise. This labeling creates a small signature and thereby results in a small number of rules of the transformed system.

Definition 9.1. Let R be a TRS over Σ, and $\mathcal{A}$ a redex-algebra. The *minimal labeling with respect to* $\mathcal{A}$ is the C-labeling $\langle \mathcal{A}, \pi, \Sigma^{red} \rangle$ defined for each n-ary symbol $\mathsf{f} \in \Sigma_{\mathsf{top}}$ by:

$$\pi_{\mathsf{f}}(a_1, \ldots, a_n) = \begin{cases} \star & \text{if } \mathit{isRedex}_{\mathsf{f}}(a_1, \ldots, a_n) = \mathit{true} \\ \epsilon & \text{otherwise} \end{cases}$$

The set of redex symbols is defined by $\Sigma^{red} = \{\mathsf{f}^\star \mid \mathsf{f} \in \Sigma\}$.

Theorem 9.2. *Let R be a TRS, and $\mathcal{A}$ a sound redex-algebra for R. The minimal labeling with respect to $\mathcal{A}$ is a sound C-labeling for R.*

Proof. Let $t = \mathsf{f}(t_1, \ldots, t_n) \in \mathcal{T}(\Sigma, \varnothing)$ such that $\mathit{root}(\mathit{lab}(t)) \in \Sigma^{red}$. Then by definition of minimal labeling we have $\mathit{isRedex}_{\mathsf{f}}([\![t_1]\!], \ldots, [\![t_n]\!]) = \mathit{true}$. Hence $t \in \mathcal{L}(\mathcal{A})$ and thus t is a redex by definition of sound redex-algebra. □

The construction and minimization of redex-algebras (Definitions 7.3 and 8.3) give rise to sound minimal C-labelings (Theorem 7.7, Lemmas 8.4 and 8.2, and Theorem 9.2). In combination with Theorems 5.8 and 6.10 this provides us with sound transformations for proving outermost termination:

Corollary 9.3. *Let R be a TRS, and let $\langle \mathcal{A}, \pi, \Sigma^{red} \rangle$ be the minimal labeling with respect to the minimized left-linear redex-algebra for R. Then R is outermost ground terminating whenever the dynamic context extension $\triangle^\pi R$ or the dynamic labeling $\uparrow^\pi R$ is terminating.*

Minimal labeling is sound and efficient, but it is not complete (not even for left-linear TRSs where the left-linear redex-algebra is complete):

Example 9.4. Let R be the term rewriting system consisting of the rules:

$$\mathsf{inf}(x) \to \mathsf{cons}(x, \mathsf{inf}(\mathsf{s}(x))) \qquad\qquad \mathsf{cons}(\mathsf{s}(x), y) \to \mathsf{nil}$$

Obviously, R is outermost terminating. The minimized left-linear redex-algebra for R is:

$$\mathcal{A} = \{\mathsf{s}, \bot\} \qquad [\![\mathsf{s}]\!](x) = \mathsf{s} \qquad [\![\mathsf{nil}]\!] = [\![\mathsf{inf}]\!](x) = [\![\mathsf{cons}]\!](x, y) = \bot$$

for all $x, y \in \mathcal{A}$. The C-depth of both rules (with respect to $\mathcal{A}$) is 0. Using minimal labeling we obtain $\pi_{\mathsf{cons}}(\mathsf{s}, x) = \star$ and $\pi_{\mathsf{inf}}(x) = \star$ for all $x \in A$, and other symbols are left unmarked (ϵ). Thus the set of redex symbols is $\Sigma^{red} = \{\mathsf{inf}^\star, \mathsf{cons}^\star\}$.

The dynamic context extension $\triangle^\pi R$ of R with respect to the C-labeling $\langle \mathcal{A}, \pi, \Sigma^{red} \rangle$ consists of the following rules, the first two of which arise from the inf-rule, with the values $\bot$ and s assigned to x respectively:

$$\begin{aligned} \mathsf{inf}^\star(x) &\to \mathsf{cons}(x, \mathsf{inf}^\star(\mathsf{s}(x))) \\ \mathsf{inf}^\star(x) &\to \mathsf{cons}^\star(x, \mathsf{inf}^\star(\mathsf{s}(x))) \\ \mathsf{cons}^\star(\mathsf{s}(x), y) &\to \mathsf{nil} \end{aligned}$$

The replacement map is defined by $\mu(\mathsf{inf}^\star) = \mu(\mathsf{cons}^\star) = \varnothing$.

Now $\triangle^\pi R$ admits an infinite derivation:

$$\mathsf{inf}^\star(x) \to_{\triangle^\pi R, \mu} \mathsf{cons}(x, \mathsf{inf}^\star(\mathsf{s}(x))) \to_{\triangle^\pi R, \mu} \mathsf{cons}(x, \mathsf{cons}(\mathsf{s}(x), \mathsf{inf}^\star(\mathsf{s}(\mathsf{s}(x))))) \to \dots$$

The third term is labeled incorrectly, as the inner occurrence of cons should be marked. The reason is that in the second step, instead of the first $\mathsf{inf}^\star$-rule, the second should have been applied; however, the left-hand side $\mathsf{inf}^\star(x)$ contains too little information to 'decide' what the labeling of the right-hand side should be.

This motivates the use of maximal labeling for which correct labeling is preserved under rewriting. Function symbols are labeled with the interpretation of their arguments:

Definition 9.5. Let R be a TRS over Σ, and let $\mathcal{A}$ be a redex-algebra for R. The *maximal labeling with respect to* $\mathcal{A}$ is the C-labeling $\langle \mathcal{A}, \pi, \Sigma^{red} \rangle$ defined for each n-ary $\mathsf{f} \in \Sigma_{\mathsf{top}}$ by:

$$\pi_{\mathsf{f}}(a_1, \dots, a_n) = \langle a_1, \dots, a_n \rangle$$

The set of redex symbols is defined by: $\Sigma^{red} = \{\mathsf{f}^{\langle a_1, \dots, a_n \rangle} \mid \mathit{isRedex}_{\mathsf{f}}(a_1, \dots, a_n) = \mathit{true}\}$.

Theorem 9.6. *Let R be a TRS, and let $\mathcal{A}$ be a redex-algebra for R. The maximal labeling with respect to $\mathcal{A}$ is a maximal C-labeling for R, and it is sound, complete, and core whenever $\mathcal{A}$ has the respective property.*

Proof. Maximality of the maximal labeling is immediate from the definition. Let $\mathcal{A}$ be a complete redex-algebra. Let $t = \mathsf{f}(t_1, \dots, t_n) \in \mathcal{T}(\Sigma, \varnothing)$ be a redex. Then by definition of complete redex-algebra $\mathit{isRedex}_{\mathsf{f}}([\![t_1]\!], \dots, [\![t_n]\!]) = \mathit{true}$, and it follows that $\mathit{root}(\mathit{lab}(t)) = \mathsf{f}^{\langle [\![t_1]\!], \dots, [\![t_n]\!] \rangle} \in \Sigma^{red}$. Hence the C-labeling is complete. Analogous to the proof of Theorem 9.2, we obtain that maximal labeling is sound whenever the redex-algebra $\mathcal{A}$ is sound. Note that the remaining claim concerning coreness is immediate by definition. □

The construction and minimization of redex-algebras (Definitions 7.3 and 8.3) give rise to sound and complete maximal C-labelings (Theorem 7.7, Lemmas 8.4 and 8.2, and Theorem 9.6). In combination with Theorems 5.8, 5.13 and 6.10 this provides us with sound transformations for proving outermost termination for arbitrary TRSs. For quasi-left-linear TRSs dynamic context extension is both sound and complete.

Corollary 9.7. *Let R be a TRS, and let $\langle \mathcal{A}, \pi, \Sigma^{red} \rangle$ be the maximal labeling for the minimized left-linear redex-algebra for R. Then R is outermost ground terminating whenever the dynamic context extension $\triangle^\pi R$ or the dynamic labeling $\uparrow^\pi R$ is terminating. Moreover, if R is quasi-left-linear, then R is outermost ground terminating if and only if the dynamic context extension $\triangle^\pi R$ terminates.*

As a consequence, the full redex-algebra for an arbitrary TRS can be used to disprove outermost ground termination:

Corollary 9.8. *Let R be a TRS, and let $\langle \mathcal{A}, \pi, \Sigma^{red} \rangle$ be the maximal labeling for the minimized full redex-algebra for R. Then the dynamic context extension $\triangle^{\pi} R$ is terminating whenever R is outermost ground terminating.*

Example 9.9. We revisit Example 9.4, but this time we give the dynamic context extension with respect to *maximal* labeling:

$$\mathsf{inf}^{\perp}(x) \to \mathsf{cons}^{\perp,\perp}(x,\mathsf{inf}^{\mathsf{s}}(\mathsf{s}^{\perp}(x))) \qquad \mathsf{inf}^{\mathsf{s}}(x) \to \mathsf{cons}^{\mathsf{s},\perp}(x,\mathsf{inf}^{\mathsf{s}}(\mathsf{s}^{\mathsf{s}}(x)))$$
$$\mathsf{cons}^{\mathsf{s},\perp}(\mathsf{s}^{\perp}(x),y) \to \mathsf{nil} \qquad \mathsf{cons}^{\mathsf{s},\perp}(\mathsf{s}^{\mathsf{s}}(x),y) \to \mathsf{nil}$$
$$\mathsf{cons}^{\mathsf{s},\mathsf{s}}(\mathsf{s}^{\perp}(x),y) \to \mathsf{nil} \qquad \mathsf{cons}^{\mathsf{s},\mathsf{s}}(\mathsf{s}^{\mathsf{s}}(x),y) \to \mathsf{nil}$$

with $\mu(\mathsf{inf}^{\perp}) = \mu(\mathsf{inf}^{\mathsf{s}}) = \mu(\mathsf{cons}^{\mathsf{s},\perp}) = \mu(\mathsf{cons}^{\mathsf{s},\mathsf{s}}) = \varnothing$. This context-sensitive TRS is terminating as opposed to the one constructed in Example 9.4. To prove termination we give a strictly decreasing polynomial interpretation over the natural numbers:

$$[\![\mathsf{nil}]\!] = 0 \qquad [\![\mathsf{cons}^{\perp,\perp}]\!](x,y) = x + y \qquad [\![\mathsf{inf}^{\perp}]\!](x) = x + 3 \qquad [\![\mathsf{s}^{\perp}]\!](x) = x$$
$$[\![\mathsf{cons}^{\mathsf{s},\perp}]\!](x,y) = 1 \qquad [\![\mathsf{inf}^{\mathsf{s}}]\!](x) = 2 \qquad [\![\mathsf{s}^{\mathsf{s}}]\!](x) = x$$

10. Evaluation

With the implementation of the transformation by dynamic context extension, described in Section 5, the termination prover Jambox [End09] gained first place in the category of outermost rewriting of the termination competition of 2008 [Ter08], see Table 1. With an average time of 4.1 seconds per termination proof, Jambox was also faster than the

	score	average time
Jambox [End09]	72 (93.5%)	4.1s
TrafO [RZ09]	46 (59.7%)	8.1s
AProVE [GSKT06]	27 (35.0%)	10.8s

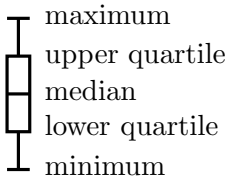
Table 1: Results of proving outermost termination in the competition of 2008 [Ter08].

other participants, providing empirical evidence for the efficiency of the transformation of dynamic context extension. Not listed in Table 1 is TTT2 [KSZM09], which did not prove outermost termination, but performed best in *dis*proving outermost termination.

The percentages listed in Table 1 are relative to the total number of term rewriting systems which were proven to be outermost terminating by some participating tool. The TPDB 2008 contained 291 TRSs in the category of outermost rewriting of which 77 were proven outermost terminating, 161 not outermost terminating, and 53 remained unsolved in the competition. We note that around 50 systems in the database are, strictly speaking, not term rewriting systems, as they contain variables in the right-hand sides that do not occur in the left-hand sides.

In the termination competition 2008, Jambox used exclusively the approach of dynamic context extension (Section 5). If we additionally use dynamic labeling, as defined in Section 6, the score of Jambox improves by 4, thus proving 76 systems to be outermost terminating.

The secret behind the efficiency of Jambox is threefold: First, we construct and minimize the algebras employed for marking redex positions, see Sections 7 and 8. Secondly, we try two labeling strategies: minimal and maximal, see Section 9. Minimal labeling is very efficient and contributes to 75% of the success of Jambox. In order to have a complete transformation we also employ maximal labeling. Thirdly, dynamic context extension is the combination of labeling and context extension, where the prefixing of contexts to rules depends on the interpretation of the variables. All these optimizations minimize the number of rules and their size in the transformed systems, which is important to keep a manageable search space.



Next, we compare the performance of dynamic context extension and dynamic labeling (Section 6). Figure 1 shows the size of the transformed systems in relation to the size of the input system, as measured on the TPDB [Ter08]. For each input size we display the minimum, the lower quartile (25th percentile), the median, the upper quartile (75th percentile), and the maximum size of the transformed systems. From Figure 1 it can be inferred that for larger input systems the dynamic labeling usually is a factor 5 or 10 smaller than the dynamic context extension. For systems with more than 10 rules there are only a few examples available in the database, which explains why some of the quartiles fall together.

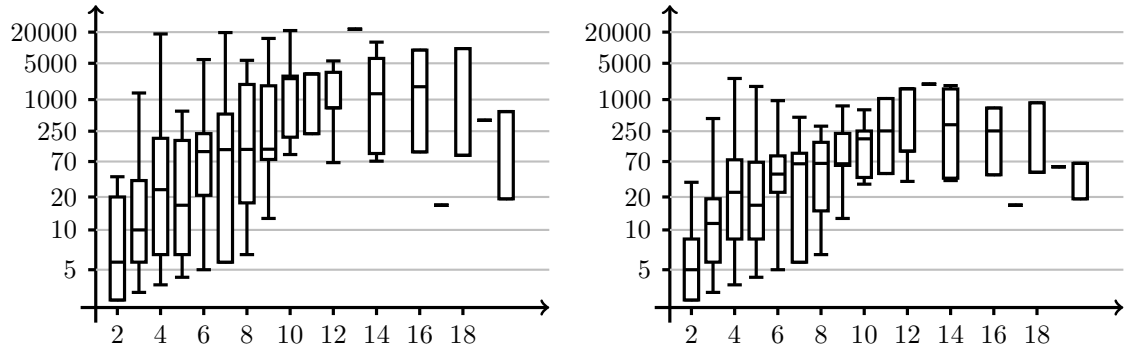


Figure 1: Size of the transformed systems (y-axis) in relation to the size of the input TRS (x-axis) using dynamic context extension (left), and dynamic labeling (right), both with maximal C-labelings.

Table 2 shows a comparison of our different methods (dynamic context extension and dynamic labeling, combined with minimal or maximal labeling). Each row lists the total score of one method with the number of systems it can solve that cannot be solved by the method corresponding to the column. For example, the value 3 in Table 2 means that three systems can be solved by dynamic maximal labeling, but not by dynamic context extension in combination with maximal labeling. The table shows that dynamic context extension

Method	Total Score	$\neg\triangle^\pi R$, max	$\neg\uparrow^\pi R$, max	$\neg\triangle^\pi R$, min	$\neg\uparrow^\pi R$, min
$\triangle^\pi R$, max	71	0	5	16	25
$\uparrow^\pi R$, max	69	3	0	16	21
$\triangle^\pi R$, min	57	2	4	0	9
$\uparrow^\pi R$, min	50	4	2	2	0

Table 2: Comparison of the proposed methods on the TPDB 2008.

and dynamic labeling are roughly equal in strength, and that maximal labeling gives the best results.

Table 3 illustrates that the C-depth of rules is typically small (employing left-linear redex-algebras): it is 0 or 1 in 94.5% of the cases. Note that 54% of the rules have C-depth 0, but this does not mean that the same percentage of the TRSs could be handled by a model (Definition 3.7). Only 14% of the TRSs have C-depth 0.

C-depth	0	1	2	3	4	5	6	7	8	9	10
#rules	54%	40.5%	3.3%	1.3%	0.5%	0.2%	0%	0%	0%	0.1%	0%

Table 3: Ratio of rules having a certain C-depth in the TPDB 2008.

11. Discussion

For arbitrary TRSs the transformation based on dynamic context extension (including the construction of C-labelings) is sound, and for quasi-left-linear TRSs it is sound and complete. The sound redex-algebra we construct recognizes redexes with respect to left-linear rules. As a consequence, in the μTRS $\triangle^\pi R$ rewriting is forbidden only inside such redex positions. This corresponds to a weakening of the outermost rewriting strategy: contraction of a redex is disallowed only if it is contained within a redex with respect to left-linear rule. Let us call this the 'left-linear outermost' rewriting strategy. Dynamic context extension combined with maximal labeling is sound and complete for termination with respect to this rewriting strategy for all TRSs.

In a similar way the transformation of [RZ09] can be generalized from quasi-left-linear TRSs to arbitrary TRSs. For soundness the anti-matching rules do not need to exactly match the non-redex terms, as long as at least all non-redex terms are matched. Then the symbol down can be moved inside redexes with respect to rules which are not left-linear. This enables only additional rewrite steps but does not harm soundness. More precisely, using this generalization the transformation of [RZ09] becomes complete with respect to left-linear outermost termination. Thereby the score of TrafO in the termination competition of 2008 [Ter08] could possibly have been improved by around 20%, resulting in a score of 57 instead of 47.

We have shown that the transformation of dynamic labeling is complete on the set of correctly labeled terms $lab(\mathcal{T}(\Sigma, \varnothing))$ without the auxiliary relabel symbols (Theorem 6.12). The non-completeness with respect to termination on all terms arises from 'illegally placed' relabel symbols in combination with duplicating rules, see Example 6.13. The duplicating rules can multiply the illegal symbols and make them reusable over and over again. To

prevent this, one can introduce an extra symbol block with $\mu(\mathsf{block}) = \varnothing$ for disallowing $\mathsf{relabel}$ symbols beneath the rule application. For this purpose, we wrap each duplicated variable in the right-hand side of a labeled rule into a context $\mathsf{block}(\Box)$, and we extend the dynamic labeling with rules of the form $\mathsf{block}(\mathsf{f}(x_1, \dots, x_n)) \to \mathsf{f}(\mathsf{block}(x_1), \dots, \mathsf{block}(x_n))$ for each symbol $\mathsf{f} \in \mathit{lab}(\Sigma_n)$ (excluding $\mathsf{relabel}$ symbols!). Note that this implies that block symbols disappear when meeting a constant. For instance, reconsider the TRS from Example 6.13, which had among others the following rule in its dynamic labeling:

$$\mathsf{f}^{bc,\bot}(\mathsf{c}, y) \to \mathsf{h}^{\bot}(\mathsf{f}^{\bot,\bot}(y, y))$$

This rule would be modified to:

$$\mathsf{f}^{bc,\bot}(\mathsf{c}, y) \to \mathsf{h}^{\bot}(\mathsf{f}^{\bot,\bot}(\mathsf{block}(y), \mathsf{block}(y)))$$

In this way we 'block' each duplicated variable.

Another question is whether there are interesting labelings between minimal and maximal. In particular, are there more efficient complete labelings? Here efficiency is measured in the size of the signature and the number of rules of the transformed system. In Example 9.9 it would have been sufficient to label cons with the interpretation of the left argument, saving two symbols and two rules of the transformed system.

References

[AEF+08] B. Alarcón, F. Emmes, C. Fuhs, J. Giesl, R. Gutiérrez, S. Lucas, P. Schneider-Kamp, and R. Thiemann. Improving Context-Sensitive Dependency Pairs. In *Proc. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2008)*, volume 5330 of *LNCS*, pages 636–651. Springer, 2008.

[AG00] T. Arts and J. Giesl. Termination of Term Rewriting Using Dependency Pairs. *Theoretical Computer Science*, 236:133–178, 2000.

[AGL06] B. Alarcón, R. Gutiérrez, and S. Lucas. Context-Sensitive Dependency Pairs. In *Proc. Conf. on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2006)*, volume 4337 of *LNCS*, pages 297–308. Springer, 2006.

[CDG+07] H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree Automata Techniques and Applications. `http://www.grappa.univ-lille3.fr/tata`, 2007.

[CELM96] M. Clavel, S. Eker, P. Lincoln, and J. Meseguer. Principles of Maude. *Electronic Notes in Theoretical Computer Science*, 4, 1996.

[Com00] H. Comon. Sequentiality, Monadic Second-Order Logic and Tree Automata. *Information and Computation*, 157(1–2):25–51, 2000.

[EdVW09] J. Endrullis, R. C. de Vrijer, and J. Waldmann. Local Termination. In *Proc. Conf. on Rewriting Techniques and Applications (RTA 2009)*, volume 5595 of *LNCS*, pages 270–284. Springer, 2009.

[EH09] J. Endrullis and D. Hendriks. From Outermost to Context-Sensitive Rewriting. In *Proc. Conf. on Rewriting Techniques and Applications (RTA 2009)*, volume 5595 of *LNCS*, pages 305–319. Springer, 2009.

[End09] J. Endrullis. Jambox, 2009. Available at `http://joerg.endrullis.de`.

[EWZ08] J. Endrullis, J. Waldmann, and H. Zantema. Matrix Interpretations for Proving Termination of Term Rewriting. *Journal of Automated Reasoning*, 40(2-3):195–220, 2008.

[FGK02] O. Fissore, I. Gnaedig, and H. Kirchner. System Presentation – CARIBOO: An Induction Based Proof Tool for Termination with Strategies. In *Proc. Conf. on Principles and Practice of Declarative Programming (PPDP 02)*, pages 62–73. ACM, 2002.

[FN97] K. Futatsugi and A. T. Nakagawa. An Overview of CAFE Specification Environment — An Algebraic Approach for Creating, Verifying, and Maintaining Formal Specifications over Networks. In *Proc. Conf. on Formal Engineering Methods (ICFEM 1997)*, pages 170–181, 1997.

[GK09] I. Gnaedig and H. Kirchner. Termination of Rewriting under Strategies. *ACM Transactions on Computational Logic*, 10(2), 2009.

[GL10] R. Gutiérrez and S. Lucas. Proving Termination in the Context-Sensitive Dependency Pairs Framework. In *Proc. Workshop on Rewriting Logic and its Applications (WRLA 2010)*, LNCS, 2010. To appear.

[GM04] J. Giesl and A. Middeldorp. Transformation Techniques for Context-Sensitive Rewrite Systems. *Journal of Functional Programming*, 14(4):379–427, 2004.

[GSKT06] J. Giesl, P. Schneider-Kamp, and R. Thiemann. AProVE 1.2: Automatic Termination Proofs in the Dependency Pair Framework. In *Proc. Int. Joint Conf. on Automated Reasoning (IJCAR 2006*, volume 4130 of *LNAI*, pages 281–286. Springer, 2006.

[GTSK04] J. Giesl, R. Thiemann, and P. Schneider-Kamp. The Dependency Pair Framework: Combining Techniques for Automated Termination Proofs. In *Proc. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2004)*, volume 3452 of *Lecture Notes in Computer Science*, pages 301–331, 2004.

[HL91] G. P. Huet and J. J. Lévy. Computations in Orthogonal Rewriting Systems, parts I and II. In *Computational Logic — Essays in Honor of Alan Robinson*, pages 395–443, 1991.

[KM91] J. W. Klop and A. Middeldorp. Sequentiality in Orthogonal Term Rewriting Systems. *Journal of Symbolic Computation*, 12(2):161–196, 1991.

[KSZM09] M. Korp, C. Sternagel, H. Zankl, and A. Middeldorp. Tyrolean Termination Tool 2. In *Proc. Conf. on Rewriting Techniques and Applications (RTA 2009)*, volume 5595 of *LNCS*, pages 295–304. Springer, 2009.

[Luc98] S. Lucas. Context-Sensitive Computations in Functional and Functional Logic Programs. *Journal of Functional and Logic Programming*, 1998(1), 1998.

[Ohl02] E. Ohlebusch. *Advanced Topics in Term Rewriting*. Springer, New York, 2002.

[PJ03] S. Peyton-Jones. *Haskell 98 Language and Libraries, The Revised Report*. Cambridge University Press, 2003.

[PvE01] M. J. Plasmeijer and M. van Eekelen. The Concurrent Clean Language Report (version 2.0). Technical report, University of Nijmegen, 2001.

[RZ09] M. Raffelsieper and H. Zantema. A Transformational Approach to Prove Outermost Termination Automatically. *Electronic Notes in Theoretical Computer Science*, 237:3–21, 2009.

[SM08] C. Sternagel and A. Middeldorp. Root-Labeling. In *Proc. Conf. on Rewriting Techniques and Applications (RTA 2008)*, volume 5117 of *LNCS*, pages 336–350. Springer, 2008.

[Ter03] Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

[Ter08] Termination Portal. `http://www.termination-portal.org/`, 2008. Termination Competition and Termination Problems Data Base (TPDB).

[Thi07] R. Thiemann. *The DP Framework for Proving Termination of Term Rewriting*. PhD thesis, RWTH Aachen, 2007. Available as technical report AIB-2007-17.

[Thi09] R. Thiemann. From Outermost Termination to Innermost Termination. In *Proc. Conf. on Theory and Practice of Computer Science (SOFSEM 2009)*, volume 5404, pages 533–545. Springer, 2009.

[Tur86] D. A. Turner. An Overview of Miranda. *SIGPLAN Notices*, 21(12):158–166, 1986.

[Zan95] H. Zantema. Termination of Term Rewriting by Semantic Labeling. *Fundamenta Informaticae*, 24:89–105, 1995.