Simple Termination of Context-Sensitive Rewriting

Bernhard Gramlich*
Institut für Computersprachen
Technische Universität Wien
Favoritenstr. 9, A-1040 Wien, Austria
gramlich@logic.at

Salvador Lucas†
DSIC
Universidad Politécnica de Valencia
Camino de Vera s/n, 46022 Valencia, Spain
slucas@dsic.upv.es

Abstract

Simple termination is the (often indirect) basis of most existing automatic techniques for proving termination of rule-based programs (e.g., Knuth-Bendix, polynomial, or recursive path orderings, but also DP-simple termination, etc.). An interesting framework for such programs is context-sensitive rewriting (CSR) that provides a bridge between the abstract world of general rewriting and the (more) applied setting of declarative specification and programming languages (e.g., OBJ*, CafeOBJ, ELAN, and Maude). In these languages, certain replacement restrictions can be specified in programs by the so-called *strategy annotations*. They may significantly improve the computational properties of programs, especially regarding their termination behaviour. Context-sensitive rewriting techniques (in particular, the methods for proving termination of CSR) have been proved useful for analyzing the properties of these programs. This entails the need to provide implementable methods for proving termination of CSR. *Simplification orderings* (corresponding to the notion of *simple termination*) which are well-known in term rewriting (and have nice properties) are, then, natural candidates for such an attempt. In this paper we introduce and investigate a corresponding notion of simple termination of CSR. We prove that our notion actually provides a unifying framework for proving termination of CSR by using standard simplification orderings via the existing (transformational) methods, and also covers CSRPO, a very recent proposal that extends the recursive path ordering (RPO) (a well-known simplification ordering) to context-sensitive terms. We also introduce polynomial orderings for dealing with (simple) termination of CSR. Finally, we also give criteria for the modularity of simple termination, for the case of disjoint unions as well as for constructor-sharing rewrite systems.

*Work partially supported by ÖAD-Programm "Acciones Integradas 2002-2003" No. 3/2002, WTZ-Büro.

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

Third ACM SIGPLAN Workshop on Rule-Based Programming (RULE'02) Pittsburgh, PA, October 5, 2002

Categories and Subject Descriptors

D1.1 [**Software**]: Applicative (Functional) Programming; D3.1 [**Programming Languages**]: Formal Definitions and Theory (D.2.1, F.3.1, F.3.2, F.4.2, F.4.3); F3.1 [**Theory of Computation**]: Specifying and Verifying and Reasoning about Programs (D.2.1, D.2.4, D.3.1, E.1); F3.2 [**Theory of Computation**]: Semantics of Programming Languages (D.3.1); F4.2 [**Theory of Computation**]: Grammars and Other Rewriting Systems (D.3.1); I.1.3 [**Computing Methodologies**]: Languages and Systems—*Evaluation strategies*

General Terms

Design, Languages, Theory, Verification

Keywords

Context-sensitive rewriting, declarative programming, evaluation strategies, automatic proofs of termination; modular program analysis and verification.

1 Introduction

Well-founded orderings (i.e., orderings allowing no infinite decreasing sequence) provide a suitable basis for proving termination in a number of programming languages and computational systems. For instance, in term rewriting systems (TRS's) a proof of termination can be achieved if we are able to find a (monotone and stable) well-founded ordering $>$ on terms (i.e., a *reduction ordering*) such that $l > r$ for every rule $l \to r$ of the rewrite system [10, 40]. In practice, if we want to implement a tool for proving termination of a TRS $\mathcal{R}$, we need to make this problem decidable. It is well-known that termination of TRSs is an undecidable problem, even for TRSs containing only one rule [7]. Hence, we can only provide effective approaches (which yield decidable termination criteria) for certain classes of systems. *Simplification orderings* are those monotone and stable orderings $>$ satisfying the following subterm property: for each term t, $t > s$ for every proper subterm s of t [8]. If termination of a TRS $\mathcal{R}$ can be proved by using a simplification ordering, then we say that $\mathcal{R}$ is simply terminating. Although simple termination is also undecidable (see [33]) it covers most usual automatizable orderings for proving termination of rewriting (e.g., recursive path orderings (*rpo* [9]), Knuth-Bendix orderings (*kbo* [23]), and polynomial orderings (*poly* [26]), see [37] for a survey on simplification orderings). Moreover, simple termination has interesting properties regarding modularity: in contrast to the general case, simple termination is modular for disjoint, constructor-sharing, and (some classes of) hierarchical unions of TRS's [36].

```
obj EXAMPLE is
  sorts Nat LNat .
  op 0     : -> Nat .
  op s     : Nat -> Nat [strat (1)] .
  op nil   : -> LNat .
  op cons  : Nat LNat -> LNat [strat (1)] .
  op from  : Nat -> LNat [strat (1 0)] .
  op sel   : Nat LNat -> Nat [strat (1 2 0)] .
  op first : Nat LNat -> LNat [strat (1 2 0)] .
  vars X Y : Nat .
  var Z : LNat .
  eq sel(s(X),cons(Y,Z)) = sel(X,Z) .
  eq sel(0,cons(X,Z)) = X .
  eq first(0,Z) = nil .
  eq first(s(X),cons(Y,Z)) = cons(Y,first(X,Z)) .
  eq from(X) = cons(X,from(s(X))) .
endo
```

Figure 1. Strategy annotations in OBJ

Context-sensitive rewriting (CSR [28]) is a restriction of rewriting which forbids reductions on selected arguments of functions. A *replacement map* $\mu : \mathcal{F} \to \mathcal{P}(\mathbb{N})$ satisfying $\mu(f) \subseteq \{1,\ldots,k\}$, for each k-ary symbol f of the signature $\mathcal{F}$ discriminates, for each symbol of the signature, the argument positions on which replacements are allowed [28]. In this way, the termination behavior of rewriting computations can be *improved*, e.g., by pruning all the infinite rewrite sequences.

In eager programming languages such as OBJ2 [14], OBJ3 [20], CafeOBJ [15], and Maude [6], it is possible to specify the so-called *strategy annotations* for controlling the execution of programs. For instance, the OBJ3 program in Figure 1 (borrowed from [2]) specifies an *explicit* strategy annotation *(1)* for the list constructor *cons* which disables replacements on the second argument (*lazy lists*, see Appendix C.5 of [20]). If we collect as $\mu(f)$ the positive indices appearing in the strategy annotation for each symbol f in a given OBJ program[1], we can use CSR to provide a framework for analyzing and ensuring essential computational properties such as termination, correctness and completeness (regarding the usual semantics: head-normalization, normalization, functional evaluation, and infinitary normalization) of programs using strategy annotations, see [1, 2, 29, 30, 31]. In particular, termination of (innermost) context-sensitive rewriting has been recently related to termination of such languages [29, 30]. For instance, we can prove termination of the OBJ3 program in Figure 1 by using the techniques for proving termination of CSR, see Examples 5 and 7 below.

Termination of CSR and its applications has been studied in a number of papers [5, 13, 16, 19, 22, 27, 32, 38, 39]. In most of these papers (e.g., [13, 16, 19, 27, 38, 39]) termination of CSR is studied in a transformational setting, namely by transforming a given context-sensitive rewrite system (CSRS i.e., a pair $(\mathcal{R},\mu)$ consisting of a TRS $\mathcal{R}$ and a replacement map μ) into an ordinary TRS such that termination of the latter implies termination of the former as CSRS. Unfortunately, such transformations typically use new symbols and rules that introduce a loss of structure and intuition, due to the encoding of the context-sensitive control in the original system by such new elements. They can unnecessarily complicate the proofs of termination and make standard techniques for easing such proofs (e.g., modular approaches) infeasible. Only recently, some work has been devoted to the *direct* analysis of termination and related properties of CSR: regarding the definition of suitable orderings for proving termination of CSR [5], and concerning the modularity of termination and related properties [22]. Moreover, the abstract properties of termination of CSR remain almost unexplored (with the notable exception of Zantema's work [39]). Indeed, all these lines of work appear to be promising and suggest to further investigate *direct* methods for achieving simpler proofs of termination of CSR.

[1] As in [20], by OBJ we mean OBJ2, OBJ3, CafeOBJ, or Maude.

In the remainder of the paper, we first introduce the necessary background on CSR. Then we introduce and discuss simple termination of CSR in Section 3. In Section 4 we show that in all cases where the transformational approaches yield simple termination, the original CSRS is simply terminating, too. Therefore, all existing transformations for proving termination of CSR can also be used for proving simple termination of CSRS's. Then, in Section 5 we deal with two direct approaches for proving termination of CSR. We show that *CSRPO-termination* [5] in fact yields simple termination, and we also cover simple termination proofs via polynomial orderings. Finally, some modularity results concerning simple termination are presented in Section 6.[2]

As a motivating example for our work consider the following.

Example 1. Consider the following one-rule TRS which is contained in the famous example of Toyama.

$$f(a,b,x) \to f(x,x,x)$$

It is well known that this TRS is not simply terminating. However, by forbidding reductions on the first and second argument, i.e., by defining $\mu(f) = \{3\}$, we obtain a 'simply terminating' behavior (regarding CSR) which can be easily proved by using a polynomial ordering for CSR, see Example 10 below.

2 Preliminaries

2.1 Basics

Subsequently we will assume in general some familiarity with the basic theory of term rewriting (cf. e.g. [3], [12]). Given a set A, $\mathcal{P}(A)$ denotes the set of all subsets of A. Given a binary relation R on a set A, we denote the transitive closure by R^+ and its reflexive and transitive closure by R^*. We say that R is *terminating* iff there is no infinite sequence $a_1\ R\ a_2\ R\ a_3 \cdots$. Throughout the paper, $\mathcal{X}$ denotes a countable set of variables and $\mathcal{F}$ denotes a signature, i.e., a set of function symbols $\{f,g,\ldots\}$, each having a fixed arity given by a mapping $ar : \mathcal{F} \to \mathbb{N}$. The set of terms built from $\mathcal{F}$ and $\mathcal{X}$ is $\mathcal{T}(\mathcal{F},\mathcal{X})$. A term is said to be linear if it has no multiple occurrences of a single variable. Terms are viewed as labelled trees in the usual way. Positions $p,q,\ldots$ are represented by chains of positive natural numbers used to address subterms of t. We denote the empty chain by Λ. The set of positions of a term t is $\mathcal{P}os(t)$. Positions of non-variable symbols in t are denoted as $\mathcal{P}os_{\mathcal{F}}(t)$, and $\mathcal{P}os_{\mathcal{X}}(t)$ are the positions of variables. The subterm at position p of t is denoted as $t|_p$ and $t[s]_p$ is the term t with the subterm at position p replaced by s. The symbol labelling the root of t is denoted as $root(t)$.

A rewrite rule is an ordered pair (l,r), written $l \to r$, with $l,r \in \mathcal{T}(\mathcal{F},\mathcal{X})$, $l \notin \mathcal{X}$ and $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$. The left-hand side (*lhs*) of the rule is l and r is the right-hand side (*rhs*). A TRS is a pair $\mathcal{R} = (\mathcal{F},R)$ where R is a set of rewrite rules. $L(\mathcal{R})$ denotes the set

[2] For the sake of readability, some proofs have been moved to the appendix.

of *lhs*'s of $\mathcal{R}$. An instance $\sigma(l)$ of a *lhs* l of a rule is a redex. The set of redex positions in t is $\mathcal{P}os_{\mathcal{R}}(t)$. A TRS $\mathcal{R}$ is left-linear if for all $l \in L(\mathcal{R})$, l is a linear term. Given TRSs $\mathcal{R} = (\mathcal{F}, R)$ and $\mathcal{R}' = (\mathcal{F}', R')$, we let $\mathcal{R} \cup \mathcal{R}'$ be the TRS $(\mathcal{F} \cup \mathcal{F}', R \cup R')$.

A term $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ rewrites to s (at position p), written $t \xrightarrow{p}_{\mathcal{R}} s$ (or just $t \to s$), if $t|_p = \sigma(l)$ and $s = t[\sigma(r)]_p$, for some rule $\rho : l \to r \in R$, $p \in \mathcal{P}os(t)$ and substitution σ. A TRS is terminating if $\to$ is terminating.

2.2 Context-Sensitive Rewriting

Given a signature $\mathcal{F}$, a mapping $\mu : \mathcal{F} \to \mathcal{P}(\mathbb{N})$ is a *replacement map* (or $\mathcal{F}$-map) if for all $f \in \mathcal{F}$, $\mu(f) \subseteq \{1, \ldots, ar(f)\}$ [28]. Let $M_{\mathcal{F}}$ (or $M_{\mathcal{R}}$ if $\mathcal{R} = (\mathcal{F}, R)$ determines the considered symbols), the set of all $\mathcal{F}$-maps. For the sake of simplicity, we will apply a replacement map $\mu \in M_{\mathcal{F}}$ on symbols $f \in \mathcal{F}'$ of any signature $\mathcal{F}'$ by assuming that $\mu(f) = \varnothing$ whenever $f \notin \mathcal{F}$. The inclusion ordering $\subseteq$ on $\mathcal{P}(\mathbb{N})$ extends to an ordering $\sqsubseteq$ on $M_{\mathcal{F}}$: $\mu \sqsubseteq \mu'$ if for all $f \in \mathcal{F}$, $\mu(f) \subseteq \mu'(f)$. Thus, $\mu \sqsubseteq \mu'$ means that μ considers less positions than μ' (for reduction). We also say that μ is more restrictive than μ'. The least upper bound (*lub*), $\sqcup$ of $\mu, \mu' \in M_{\mathcal{F}}$ are given by $(\mu \sqcup \mu')(f) = \mu(f) \cup \mu'(f)$. A replacement map μ specifies the *argument* positions which can be reduced for each *symbol* in $\mathcal{F}$. Accordingly, the set of *μ-replacing positions* $\mathcal{P}os^{\mu}(t)$ of $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ is: $\mathcal{P}os^{\mu}(t) = \{\Lambda\}$, if $t \in \mathcal{X}$ and

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

if $t \notin \mathcal{X}$. The set of positions of replacing redexes in t is $\mathcal{P}os^{\mu}_{\mathcal{R}}(t) = \mathcal{P}os_{\mathcal{R}}(t) \cap \mathcal{P}os^{\mu}(t)$. A context-sensitive rewrite system (CSRS) is a pair $(\mathcal{R}, \mu)$, where $\mathcal{R}$ is a TRS and μ is a replacement map. In *context-sensitive rewriting* (CSR [28]), we (only) contract *replacing* redexes: t μ-rewrites to s, written $t \hookrightarrow_{\mu} s$ (or just $t \hookrightarrow s$), if $t \xrightarrow{p}_{\mathcal{R}} s$ and $p \in \mathcal{P}os^{\mu}(t)$. A CSRS $(\mathcal{R}, \mu)$ is terminating if $\hookrightarrow_{\mu}$ is terminating.

3 Simple Termination of CSR

Zantema has provided an algebraic characterization of termination of CSR [39]. As for standard rewriting, he proves that termination of CSR is fully captured by the so-called *μ-reduction orderings*, i.e., well-founded, stable orderings $>$ such that, for all $p \in \mathcal{P}os^{\mu}(C)$, $C[s]_p > C[t]_p$ whenever $s > t$ (μ-monotonicity). Then, a TRS $\mathcal{R} = (\mathcal{F}, R)$ is μ-terminating if and only if there is a μ-reduction ordering $>$ which is *compatible* with the rules of $\mathcal{R}$, i.e., for all $l \to r \in R$, $l > r$ ([39, Proposition 1]). He also shows that μ-reduction ordering can be defined by means of (well-founded) μ-monotone $\mathcal{F}$-algebras. An (ordered) $\mathcal{F}$-algebra, is a triple $(A, \mathcal{F}_A, >)$, where A is a set, $\mathcal{F}_A$ is a set of mappings $f_A : A^k \to A$ for each $f \in \mathcal{F}$ where $k = ar(f)$, and $>$ is a (strict) ordering on A. Given $f \in \mathcal{F}$, we say that $f_A : A^k \to A$ is μ-monotone if for all $a, b \in A$ such that $a > b$, we have that

$$f_A(a_1, \ldots, a_{i-1}, a, \ldots, a_k) > f_A(a_1, \ldots, a_{i-1}, b, \ldots, a_k)$$

for all $a_1, \ldots, a_k \in A$ and $i \in \mu(f)$. A $\mathcal{F}$-algebra $\mathcal{A} = (A, \mathcal{F}_A, >)$ is μ-monotone if for all $f \in \mathcal{F}_A$, f is μ-monotone. A $\mathcal{F}$-algebra $\mathcal{A} = (A, \mathcal{F}_A, >)$ is well-founded if $>$ is well-founded. For a given valuation mapping $\alpha : \mathcal{X} \to A$, the evaluation mapping $[\alpha] : \mathcal{T}(\mathcal{F}, \mathcal{X}) \to A$ is inductively defined by $[\alpha](x) = \alpha(x)$ if $x \in \mathcal{X}$ and $[\alpha](f(t_1, \ldots, t_k)) = f_A([\alpha](t_1), \ldots, [\alpha](t_k))$ for $x \in \mathcal{X}$, $f \in \mathcal{F}$, $t_1, \ldots, t_k \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. We write $t >_A s$ if and only if $[\alpha](t) > [\alpha](s)$ for all $\alpha : \mathcal{X} \to A$. Then, $>_A$ is a reduction ordering on terms for every well-founded μ-monotone $\mathcal{F}$-algebra $(A, \mathcal{F}_A, >)$ ([39, Proposition 2]). We use these notions in the following.

Given a signature $\mathcal{F}$, we consider the TRS $\mathcal{E}mb(\mathcal{F}) = (\mathcal{F}, \{f(x_1, \ldots, x_k) \to x_i \mid f \in \mathcal{F}, i \in \{1, \ldots, ar(f)\}\})$. A TRS $\mathcal{R}$ is *simply terminating* if $\mathcal{R} \cup \mathcal{E}mb(\mathcal{F})$ is terminating. Regarding CSR, the most obvious extension of this notion (a CSRS $(\mathcal{R}, \mu)$ is simply terminating if $(\mathcal{R} \cup \mathcal{E}mb(\mathcal{F}), \mu)$ is terminating) is not appropriate:

Example 2. Consider the TRS $\mathcal{R}$ (where a is a constant):

$$a \to c(a)$$

with $\mu(c) = \varnothing$. The CSRS $(\mathcal{R}, \mu)$ is clearly terminating; however, $(\mathcal{R} \cup \mathcal{E}mb(\mathcal{F}), \mu)$ (where $\mathcal{E}mb(\mathcal{F})$ has a single rule $c(x) \to x$) is not: $a \hookrightarrow c(a) \hookrightarrow a \hookrightarrow \cdots$.

Formally, in contrast to TRS's where termination of $\to_{\mathcal{R}}$ implies termination of $(\to_{\mathcal{R}} \cup >_{st})$ (with $>_{st}$ being the proper subterm ordering), this property does not hold for CSRS's any more, i.e., termination of $\hookrightarrow_{\mathcal{R}}$ in general does not imply termination of $(\hookrightarrow_{\mathcal{R}} \cup >_{st})$.

The problem is that the projections in $\mathcal{E}mb(\mathcal{F})$ should not make those arguments reducible which are not reducible using the replacement map μ. Therefore, we define $\mathcal{E}mb^{\mu}(\mathcal{F}) = (\mathcal{F}, \{f(x_1, \ldots, x_k) \to x_i \mid f \in \mathcal{F}, i \in \mu(f)\})$ and propose the following.

DEFINITION 1. *A CSRS $(\mathcal{R}, \mu)$ is* simply terminating *if $(\mathcal{R} \cup \mathcal{E}mb^{\mu}(\mathcal{F}), \mu)$ is terminating.*

Obviously, if $\mathcal{E}mb^{\mu_{\top}}(\mathcal{F}) = \mathcal{E}mb(\mathcal{F})$, then simple termination as in Definition 1 and the standard notion of simple termination of TRS's coincide (as one would expect). Furthermore, any simply terminating TRS $\mathcal{R}$, viewed as CSRS $(\mathcal{R}, \mu)$ for arbitrary $\mu \in M_{\mathcal{R}}$, is also simply terminating (since $\mathcal{E}mb^{\mu}(\mathcal{F})$ is a subset of $\mathcal{E}mb(\mathcal{F})$ for all μ). Finally, we note that if $\mu \sqsubseteq \mu'$, then $\mathcal{E}mb^{\mu}(\mathcal{F})$ is included in $\mathcal{E}mb^{\mu'}(\mathcal{F})$. Hence, simple termination of $(\mathcal{R}, \mu')$ implies simple termination of $(\mathcal{R}, \mu)$ in this case.

Simplification orderings are the algebraic (indeed originating) counterpart of simple termination. A simplification ordering is a monotonic, stable ordering on terms which additionally satisfies the following 'subterm' property: for all $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and $p \in \mathcal{P}os(t) - \{\Lambda\}$, $t > t|_p$. It is well-known that simplification orderings are well-founded (for terms over finite signatures). Thus, any simplification ordering is a reduction ordering, and hence it is suitable for (trying to) prove termination of a TRS $\mathcal{R}$ by just comparing the left- and right-hand sides of its rules.

The following natural generalization of simplification orderings immediately arises.

DEFINITION 2. *Let $\mathcal{F}$ be a signature and $\mu \in M_{\mathcal{F}}$. A μ-monotonic, stable ordering $>$ is a μ-simplification ordering if, for all $t \in \mathcal{T}(\mathcal{F}, \mathcal{X}), p \in \mathcal{P}os^{\mu}(t) - \{\Lambda\}$, $t > t|_p$ (μ-subterm property).*

Unfortunately, in (sharp) contrast with the standard case, μ-simplification orderings are not well-founded any more (in general).

Example 3. Consider the signature consisting of the constant a and the unary symbol c. If we choose $\mu(c) = \varnothing$, then the binary relation $>$ consisting of pairs $\{(c^n(a), c^{n+1}(a)) \mid n \geq 0\}$ is a μ-monotone strict ordering and has the μ-subterm property. However, it admits an infinite sequence

$$a > c(a) > \cdots > c^n(a) > \cdots$$

Nevertheless, Definition 2 characterizes the notion of simple termination of Definition 1 in the following (obvious) sense.

THEOREM 1. *A CSRS $(\mathcal{R},\mu)$ is simply terminating if and only if there is a well-founded μ-simplification ordering $>$ such that for every rule $l \to r \in R$, $l > r$.*

PROOF. If $(\mathcal{R},\mu)$ is simply terminating, then the CSRS $(\mathcal{R} \cup \mathcal{E}mb^\mu(\mathcal{F}),\mu)$, where $\mathcal{R} = (\mathcal{F},R)$, is terminating and $\hookrightarrow^+$ is a μ-reduction ordering which clearly satisfies the μ-subterm property (due to the presence of the corresponding rules in $\mathcal{E}mb^\mu(\mathcal{F})$). Then, $\hookrightarrow^+$ is a μ-simplification ordering. Obviously, we have $l \hookrightarrow^+ r$ for every rule $l \to r \in R$.

On the other hand, assume that $>$ is a well-founded μ-simplification ordering such that $l > r$, for every rule $l \to r \in R$. Since it is a μ-simplification ordering, $f(x_1,\ldots,x_k) > x_i$ for all $i \in \mu(f)$ and $f \in \mathcal{F}$. Hence, $>$ is a μ-reduction ordering which is compatible with $\mathcal{R} \cup \mathcal{E}mb^\mu(\mathcal{F})$, i.e., which orients all rules from left to right. Hence, $\mathcal{R}$ is simply μ-terminating. $\square$

4 Proving Simple Termination of CSR

Termination of CSRS's $(\mathcal{R},\mu)$ is usually proved by demonstrating *termination* of a transformed TRS $\mathcal{R}^\mu_\Theta$ obtained from $\mathcal{R}$ and μ by using a transformation[3] Θ [13, 16, 17, 27, 38, 39]. A transformation Θ is

1. *correct* (regarding (simple) termination) if (simple) termination of $\mathcal{R}^\mu_\Theta$ implies (simple) termination of $(\mathcal{R},\mu)$ for all TRS $\mathcal{R}$ and replacement map $\mu \in M_\mathcal{R}$.
2. *complete* (regarding (simple) termination) if (simple) termination of $(\mathcal{R},\mu)$ implies (simple) termination of $\mathcal{R}^\mu_\Theta$ for all TRS $\mathcal{R}$ and replacement map $\mu \in M_\mathcal{R}$.

The simplest (and trivial) correct transformation for proving termination of CSRS's is the identity: if $\mathcal{R}^\mu_{ID} = \mathcal{R}$ is terminating, then $(\mathcal{R},\mu)$ is terminating for every replacement map μ.

Here, we are interested in proving simple termination of CSRS's by using existing transformations. According to this goal in mind, we review the main (non-trivial) correct transformations for proving termination of CSR regarding their suitability for proving simple termination of CSR.

4.1 The Contractive Transformation

Let $\mathcal{F}$ be a signature and $\mu \in M_\mathcal{F}$ be a replacement map. With the contractive transformation [27], the non-μ-replacing arguments of all symbols in $\mathcal{F}$ are *removed* and a new, *μ-contracted* signature $\mathcal{F}^\mu_L$ is obtained (possibly reducing the *arity* of symbols). The function $\tau_\mu : \mathcal{T}(\mathcal{F},\mathcal{X}) \to \mathcal{T}(\mathcal{F}^\mu_L,\mathcal{X})$ *drops* the non-replacing immediate subterms of a term $t \in \mathcal{T}(\mathcal{F},\mathcal{X})$ and constructs a 'μ-contracted' term by joining the (also transformed) replacing arguments below the corresponding operator of $\mathcal{F}^\mu_L$. A CSRS $(\mathcal{R},\mu)$, where $\mathcal{R} = (\mathcal{F},R)$ is μ-contracted into $\mathcal{R}^\mu_L = (\mathcal{F}^\mu_L, \{\tau_\mu(l) \to \tau_\mu(r) \mid l \to r \in R\})$. According to this definition, it is not difficult to see that $\mathcal{R}^\mu_L \cup \mathcal{E}mb(\mathcal{F}^\mu_L) = (\mathcal{R} \cup \mathcal{E}mb^\mu(\mathcal{F}))^\mu_L$. Thus, we have the following:

THEOREM 2. *Let $(\mathcal{R},\mu)$ be a CSRS. If $\mathcal{R}^\mu_L$ is simply terminating, then $(\mathcal{R},\mu)$ is simply terminating.*

PROOF. If $\mathcal{R}^\mu_L$ is simply terminating, then $\mathcal{R}^\mu_L \cup \mathcal{E}mb(\mathcal{F}^\mu_L) = (\mathcal{R} \cup \mathcal{E}mb^\mu(\mathcal{F}))^\mu_L$ is terminating. Hence, $(\mathcal{R} \cup \mathcal{E}mb^\mu(\mathcal{F}),\mu)$ is terminating, i.e., $(\mathcal{R},\mu)$ is simply terminating. $\square$

Example 4. Consider the following CSRS which can be used to obtain (as *first(n,terms(1))*) the first n terms of the series that approximates $\pi^2/6$ [31]:

$$\mathcal{R}: \quad \begin{array}{rcl} sqr(0) & \to & 0 \\ sqr(s(x)) & \to & s(sqr(x)+dbl(x)) \\ dbl(0) & \to & 0 \\ dbl(s(x)) & \to & s(s(dbl(x))) \\ 0+x & \to & x \\ s(x)+y & \to & s(x+y) \\ first(0,x) & \to & [] \\ first(s(x),y:z) & \to & y:first(x,z) \\ terms(n) & \to & recip(sqr(n)):terms(s(n)) \end{array}$$

with $\mu(:) = \{1\}$ and $\mu(f) = \{1,\ldots,k\}$ for any other k-ary symbol f. Then,

$$\mathcal{R}^\mu_L: \quad \begin{array}{rcl} sqr(0) & \to & 0 \\ sqr(s(x)) & \to & s(sqr(x)+dbl(x)) \\ dbl(0) & \to & 0 \\ dbl(s(x)) & \to & s(s(dbl(x))) \\ 0+x & \to & x \\ s(x)+y & \to & s(x+y) \\ first(0,x) & \to & [] \\ first(s(x),:(y)) & \to & :(y) \\ terms(n) & \to & :(recip(sqr(n))) \end{array}$$

is (simply) terminating: use an *rpo* based on precedence[4]

$$terms \succ_\mathcal{F} :, recip, sqr; \quad sqr \succ_\mathcal{F} dbl, + \succ_\mathcal{F} s; \quad \text{and} \quad first \succ_\mathcal{F} [].$$

Hence, $(\mathcal{R},\mu)$ is simply terminating.

The contractive transformation only works well with *μ-conservative* TRSs, i.e., satisfying that $\mathcal{V}ar^\mu(r) \subseteq \mathcal{V}ar^\mu(l)$ for every rule $l \to r$ of $\mathcal{R}$ [27]; otherwise, extra variables will appear in a rule of $\mathcal{R}^\mu_L$ which thus would become non-terminating. Let

$$CoCM_\mathcal{R} = \{\mu \in CM_\mathcal{R} \mid \text{for all } l \to r \text{ in } \mathcal{R}, \mathcal{V}ar^\mu(r) \subseteq \mathcal{V}ar^\mu(l)\}$$

That is: $CoCM_\mathcal{R}$ contains the replacement maps $\mu \in CM_\mathcal{R}$ that make $\mathcal{R}$ μ-conservative [32].

THEOREM 3. [32] *Let $\mathcal{R}$ be a left-linear TRS and $\mu \in CoCM_\mathcal{R}$. If $(\mathcal{R},\mu)$ is terminating, then $\mathcal{R}^\mu_L$ is terminating.*

Hence, we can use the contractive transformation to fully characterize simple termination of CSR in restricted cases.

THEOREM 4. *Let $\mathcal{R} = (\mathcal{F},R)$ be a left-linear TRS and $\mu \in CoCM_\mathcal{R}$. Then, $(\mathcal{R},\mu)$ is simply terminating if and only if $\mathcal{R}^\mu_L$ is simply terminating.*

---

[3] See `http://www.dsic.upv.es/users/elp/slucas/muterm` for a tool, MU-TERM 1.0, that implements these transformations.

[4] A precedence $\succeq_\mathcal{F}$ is a quasi-ordering (i.e., a reflexive and transitive relation) on the symbols of the signature $\mathcal{F}$.

PROOF. The if part is Theorem 2. For the only if part, assume that $(\mathcal{R},\mu)$ is simply terminating. Then, $(\mathcal{R} \cup \mathcal{E}mb^{\mu}(\mathcal{F}),\mu)$ is terminating and, by Theorem 3, $(\mathcal{R} \cup \mathcal{E}mb^{\mu}(\mathcal{F}))^{\mu}_{L}$ is terminating. Since $(\mathcal{R} \cup \mathcal{E}mb^{\mu}(\mathcal{F}))^{\mu}_{L} = \mathcal{R}^{\mu}_{L} \cup \mathcal{E}mb(\mathcal{F}^{\mu}_{L})$, $\mathcal{R}^{\mu}_{L} \cup \mathcal{E}mb(\mathcal{F}^{\mu}_{L})$ is terminating, hence $\mathcal{R}^{\mu}_{L}$ is simply terminating. □

4.2 Zantema's Transformation

Zantema's transformation *marks* the *non-replacing arguments* of function symbols (disregarding their positions within the term) [39]. Given $\mathcal{R} = (\mathcal{F},R)$ and $\mu \in M_{\mathcal{F}}$, $\mathcal{R}^{\mu}_{Z} = (\mathcal{F} \cup \mathcal{F}' \cup \{activate\}, R^{\mu}_{Z})$ where R^{μ}_{Z} consists of two parts. The first part results from R by replacing every function symbol f occurring in a left or right-hand side with f' (a fresh function symbol of the same arity as f which, then, is included in $\mathcal{F}'$) if it occurs in a non-replacing *argument* of the function symbol directly above it. These new function symbols are used to block further reductions at this position. In addition, if a variable x occurs in a non-replacing position in the *lhs* l of a rewrite rule $l \to r$, then all occurrences of x in r are replaced by *activate(x)*. Here, *activate* is a new unary function symbol which is used to activate blocked function symbols again.

The second part of R^{μ}_{Z} consists of rewrite rules that are needed for blocking and unblocking function symbols:

$$\begin{array}{rcl} f(x_1,\ldots,x_k) & \to & f'(x_1,\ldots,x_k) \\ activate(f'(x_1,\ldots,x_k)) & \to & f(x_1,\ldots,x_k) \end{array}$$

for every $f' \in \mathcal{F}'$, together with the rule $activate(x) \to x$ Again, we note that, since $(\mathcal{E}mb^{\mu}(\mathcal{F}))^{\mu}_{Z} = \mathcal{E}mb^{\mu}(\mathcal{F}) \cup \{activate(x) \to x\}$ and $\mathcal{E}mb^{\mu}(\mathcal{F}) \subseteq \mathcal{E}mb(\mathcal{F}^{\mu}_{Z})$, we have that $\mathcal{R}^{\mu}_{Z} \cup \mathcal{E}mb(\mathcal{F}^{\mu}_{Z}) = (\mathcal{R} \cup \mathcal{E}mb^{\mu}(\mathcal{F}))^{\mu}_{Z} \cup \mathcal{E}mb(\mathcal{F}^{\mu}_{Z})$. Thus, we have:

THEOREM 5. *Let $(\mathcal{R},\mu)$ be a CSRS. If $\mathcal{R}^{\mu}_{Z}$ is simply terminating, then $(\mathcal{R},\mu)$ is simply terminating.*

Example 5. The following CSRS

$$\mathcal{R}: \quad \begin{array}{r} sel(0,x:y) \to x \\ sel(s(x),y:z) \to sel(x,z) \\ first(0,x) \to [] \\ first(s(x),y:z) \to y:first(x,z) \\ from(x) \to x:from(s(x)) \end{array}$$

with $\mu(:) = \{1\}$ and $\mu(f) = \{1,\ldots,k\}$ for any other k-ary symbol f corresponds to the OBJ3 program in Figure 5 (we use *:* and *[]* instead of *cons* and *nil* respectively). Then,

$$\mathcal{R}^{\mu}_{Z}: \quad \begin{array}{r} sel(0,x:y) \to x \\ sel(s(x),y:z) \to sel(x,activate(z)) \\ first(0,x) \to [] \\ first(s(x),y:z) \to y:first'(x,activate(z)) \\ from(x) \to x:from'(s(x)) \\ from(x) \to from'(x) \\ activate(from'(x)) \to from(x) \\ first(x,y) \to first'(x,y) \\ activate(first'(x,y)) \to first(x,y) \\ activate(x) \to x \end{array}$$

is simply terminating: use the *rpo* which is based on precedence *sel* $\succ_{\mathcal{F}}$ *activate* $=_{\mathcal{F}}$ *first* $\succ_{\mathcal{F}}$ *from,:,first',[]* and *from* $\succ_{\mathcal{F}}$ *:,from',s*, and gives *sel* the usual lexicographic status. Hence, $(\mathcal{R},\mu)$ is simply terminating.

In [13], Ferreira and Ribeiro propose a variant of Zantema's transformation which has been proved strictly more powerful than Zantema's one (see [19]). Again, $\mathcal{R}^{\mu}_{FR}$ has two parts. The first part results from the first part of $\mathcal{R}^{\mu}_{Z}$ by marking all function symbols (except *activate*) which occur below an already marked symbol. Therefore, all function symbols of non-replacing subterms are marked. The second part consists of the rule *activate(x)* $\to x$ plus the rules:

$$\begin{array}{rcl} f(x_1,\ldots,x_k) & \to & f'(x_1,\ldots,x_k) \\ activate(f'(x_1,\ldots,x_k)) & \to & f([x_1]^{f}_{1},\ldots,[x_k]^{f}_{k}) \end{array}$$

for every $f \in \mathcal{F}$ for which f' appears in the first part of $\mathcal{R}^{\mu}_{FR}$, where $[t]^{f}_{i} = activate(t)$ if $i \in \mu(f)$ and $[t]^{f}_{i} = t$ otherwise. They also include rules

$$activate(f(x_1,\ldots,x_k)) \to f([x_1]^{f}_{1},\ldots,[x_k]^{f}_{k})$$

for k-ary symbols f where f' does *not* appear in the first part of $\mathcal{R}^{\mu}_{FR}$. However, Giesl and Middeldorp have recently shown that these rules are not necessary for obtaining a correct transformation [19]. Since we also have $\mathcal{R}^{\mu}_{FR} \cup \mathcal{E}mb(\mathcal{F}^{\mu}_{FR}) = (\mathcal{R} \cup \mathcal{E}mb^{\mu}(\mathcal{F}))^{\mu}_{FR} \cup \mathcal{E}mb(\mathcal{F}^{\mu}_{FR})$, this transformation is similar to Zantema's one regarding simple termination, i.e., we have the following:

THEOREM 6. *Let $(\mathcal{R},\mu)$ be a CSRS. If $\mathcal{R}^{\mu}_{FR}$ is simply terminating, then $(\mathcal{R},\mu)$ is simply terminating.*

4.3 Giesl and Middeldorp's Transformations

Giesl and Middeldorp introduced a transformation that explicitly *marks* the replacing positions of a term (by using a new symbol *active*). Given a TRS $\mathcal{R} = (\mathcal{F},R)$ and $\mu \in M_{\mathcal{F}}$, the TRS $\mathcal{R}^{\mu}_{GM} = (\mathcal{F} \cup \{active,mark\}, R^{\mu}_{GM})$ consists of the following rules (for all $l \to r \in R$ and $f \in \mathcal{F}$):

$$\begin{array}{rcl} active(l) & \to & mark(r) \\ mark(f(x_1,\ldots,x_k)) & \to & active(f([x_1]_f,\ldots,[x_k]_f)) \\ active(x) & \to & x \end{array}$$

where $[x_i]_f = mark(x_i)$ if $i \in \mu(f)$ and $[x_i]_f = x_i$ otherwise [16]. We have the following.

THEOREM 7. *Let $(\mathcal{R},\mu)$ be a CSRS. If $\mathcal{R}^{\mu}_{GM}$ is simply terminating, then $(\mathcal{R},\mu)$ is simply terminating.*

In [16], Giesl and Middeldorp suggest a slightly different presentation of $\mathcal{R}^{\mu}_{mGM}$ of the previous transformation. In this presentation, symbol *active* is not used anymore. However, we have proved that both transformations are equivalent regarding simple termination, i.e., $\mathcal{R}^{\mu}_{GM}$ is simply terminating if and only if $\mathcal{R}^{\mu}_{mGM}$ is [32]. Thus, Theorem 7 also holds for $\mathcal{R}^{\mu}_{mGM}$.

Giesl and Middeldorp also introduced a transformation $\mathcal{R}^{\mu}_{C}$ which is *complete*, i.e., every μ-terminating TRS is transformed into a terminating TRS $\mathcal{R}^{\mu}_{C}$ [16].

Given a TRS $\mathcal{R} = (\mathcal{F},R)$ and a replacement map μ, the TRS $\mathcal{R}^{\mu}_{C} = (\mathcal{F} \cup \{f' \mid f \in \mathcal{F} \wedge ar(f) > 0\} \cup \{active,mark,ok,proper,top\}, R^{\mu}_{C})$ consists of the following rules (see [16] for a more detailed explanation): for all $l \to r \in R$, $f \in \mathcal{F}$ such that $k = ar(f) > 0$, $i \in \mu(f)$, and constants $c \in \mathcal{F}$,

$$\begin{array}{rcl} active(l) & \to & mark(r) \\ active(f(x_1,\ldots,x_i,\ldots,x_k)) & \to & f'(x_1,\ldots,active(x_i),\ldots,x_k) \\ f'(x_1,\ldots,mark(x_i),\ldots,x_k) & \to & mark(f(x_1,\ldots,x_i,\ldots,x_k)) \\ proper(c) & \to & ok(c) \\ proper(f(x_1,\ldots,x_k)) & \to & f(proper(x_1),\ldots,proper(x_k)) \\ f(ok(x_1),\ldots,ok(x_k)) & \to & ok(f(x_1,\ldots,x_k)) \\ top(mark(x)) & \to & top(proper(x)) \\ top(ok(x)) & \to & top(active(x)) \end{array}$$

THEOREM 8. *Let $(\mathcal{R},\mu)$ be a CSRS. If $\mathcal{R}^\mu_C$ is simply terminating, then $(\mathcal{R},\mu)$ is simply terminating.*

We conclude this section by noticing that Toyama's CSRS of Example 1 cannot be proved to be simply terminating by using any of the previous transformations:

$$\mathcal{R}^\mu_L: \quad f(x) \to f(x)$$

$$\mathcal{R}^\mu_Z, \mathcal{R}^\mu_{FR}: \begin{array}{c} f(a',b',x) \to f(x,x,x) \\ a \to a' \\ b \to b' \\ activate(a') \to a \\ activate(b') \to b \\ activate(x) \to x \end{array}$$

$$\mathcal{R}^\mu_{GM}: \begin{array}{c} active(f(a,b,x)) \to mark(f(x,x,x)) \\ mark(f(x,y,z)) \to active(f(x,y,mark(z))) \\ mark(a) \to active(xa) \\ mark(b) \to active(b) \\ active(x) \to x \end{array}$$

$$\mathcal{R}^\mu_C: \begin{array}{c} active(f(a,b,x)) \to mark(f(x,x,x)) \\ active(f(x,y,z)) \to f(x,y,active(z)) \\ f(x,y,mark(z)) \to mark(f(x,y,z)) \\ proper(f(x,y,z)) \to f(proper(x),proper(y),proper(z)) \\ proper(a) \to ok(a) \\ proper(b) \to ok(b) \\ f(ok(x),ok(y),ok(z)) \to ok(f(x,y,z)) \\ top(mark(x)) \to top(proper(x)) \\ top(ok(x)) \to top(active(x)) \end{array}$$

Note that $\mathcal{R}^\mu_L$ is not even terminating; $\mathcal{R}^\mu_Z$ (and $\mathcal{R}^\mu_{FR}$ which, in this case coincides with $\mathcal{R}^\mu_Z$) is not simply terminating as it contains the non-simply terminating TRS $f(a',b',x) \to f(x,x,x)$. $\mathcal{R}^\mu_{GM}$ is not simply terminating:

$$\begin{array}{l} active(f(a,b,f(a,b,b))) \\ \quad \to_{\mathcal{R}^\mu_{GM}} mark(f(f(a,b,b),f(a,b,b),f(a,b,b))) \\ \quad \to_{\mathcal{R}^\mu_{GM}} active(f(f(a,b,b),f(a,b,b),mark(f(a,b,b)))) \\ \quad \to^+_{\mathcal{E}mb(\mathcal{F}^\mu_{GM})} active(f(a,b,f(a,b,b))) \\ \quad \to_{\mathcal{R}^\mu_{GM}} \cdots \end{array}$$

Finally, $\mathcal{R}^\mu_C$ is not simply terminating either:

$$\begin{array}{l} top(active(f(a,b,f(a,b,b)))) \\ \quad \to_{\mathcal{R}^\mu_C} top(mark(f(f(a,b,b),f(a,b,b),f(a,b,b)))) \\ \quad \to^+_{\mathcal{E}mb(\mathcal{F}^\mu_C)} top(mark(f(a,b,f(a,b,b)))) \\ \quad \to_{\mathcal{R}^\mu_C} top(proper(f(a,b,f(a,b,b)))) \\ \quad \to_{\mathcal{R}^\mu_C} top(ok(f(a,b,f(a,b,b)))) \\ \quad \to_{\mathcal{R}^\mu_C} top(active(f(a,b,f(a,b,b)))) \\ \quad \to_{\mathcal{R}^\mu_C} \cdots \end{array}$$

Therefore, these examples show that, in general, Theorems 2, 5, 6, 7, and 8 cannot be used in the 'only if' direction. In other words, this means that in certain cases simple termination cannot be proved by (the considered) transformational techniques, i.e., all of them are *incomplete* regarding simple termination of CSR. Notably, the transformation C, which is complete for proving termination of CSR, becomes incomplete for proving simple termination of CSR (if we want to use simplification orderings for proving termination of $\mathcal{R}^\mu_C$). In the following section, we consider a different class of methods which are not based on applying transformations to CSRS's; in contrast, these methods are able to directly address (simple) termination of CSRS's without transforming them.

5 Direct Approaches to SimpleTermination of CSR

5.1 The Context-Sensitive Recursive Path Ordering (CSRPO)

In [5], the recursive path ordering (RPO), a well-known technique for automatically proving (simple) termination of TRSs, has been extended to deal with termination of CSRS's. Thus, a natural question arises: are the CSRPO-terminating CSRS's simply terminating? In this section, we positively answer this question.

The definition of CSRPO is akin to that of RPO. First, we recall the definition of RPO. Given a precedence $\succeq_\mathcal{F}$ on the set of function symbols, which is the union of a well-founded ordering $\succ_\mathcal{F}$ and a compatible equality $=_\mathcal{F}$, and a status function $stat(f) \in \{lex, mul\}$ s.t. if $f =_\mathcal{F} g$ then $stat(f) = stat(g)$, RPO is defined recursively as follows:

$s = f(s_1,\ldots,s_n) \succ_{rpo} t$ iff

1. $s_i \succeq_{rpo} t$, for some $i = 1,\ldots,n$ or
2. $t = g(t_1,\ldots,t_m)$ with $f \succ_\mathcal{F} g$ and $s \succ_{rpo} t_i$ for all $i = 1\ldots n$ or
3. $t = g(t_1,\ldots,t_m)$ with $f =_\mathcal{F} g$, $stat(f) = mul$ and $\{s_1,\ldots,s_n\} \succ\succ_{rpo} \{t_1,\ldots,t_m\}$ or
4. $t = g(t_1,\ldots,t_m)$ with $f =_\mathcal{F} g$, $stat(f) = lex$, $\langle s_1,\ldots,s_n\rangle \succ^{lex}_{rpo} \langle t_1,\ldots,t_m\rangle$ and $s \succ_{rpo} t_i$ for all $i \in \{1,\ldots,m\}$.

where $\succeq_{rpo}$ is the union of $\succ_{rpo}$ and syntactic equality.

The first idea that comes in mind to extend RPO to context-sensitive rewriting (CSRPO) is *marking* the symbols which are in blocked positions and consider them smaller than the active ones. Therefore terms in blocked positions become smaller.

Example 6. Consider the rule

$$from(x) \to x : from(s(x))$$

together with $\mu(:) = \{1\}$. In order to prove that *from(x)* is greater than *x:from(s(x))*, we take into account the replacement restriction $\mu(:) = \{1\}$ thus comparing *from(x)* and $x{:}\underline{from}(s(x))$ where $\underline{from}$ is a marked version of *from* (and we set $from \succ_\mathcal{F} \underline{from}$).

However, marking all symbols in non-replacing positions can unnecessarily weaken the resulting ordering. Thus, in addition to the usual precedence $\succeq_\mathcal{F}$ on symbols, a *marking map*, denoted by $\mathfrak{m}$, which defines for every symbol and every blocked position the set of symbols that should be marked is also used. By $\underline{\mathcal{F}}$ we denote the set of marked symbols corresponding to $\mathcal{F}$. Given a symbol f in $\mathcal{F} \cup \underline{\mathcal{F}}$ and $i \in \{1,\ldots,ar(f)\}$, a marking map $\mathfrak{m}$ provides the subset of symbols in $\mathcal{F}$ that should be marked, i.e. $\mathfrak{m}(f,i) \subseteq \mathcal{F}$. Marking maps are intended to mark *only* blocked arguments, i.e., $\mathfrak{m}(f,i) = \varnothing$ if $i \in \mu(f)$ (this seems reasonable and is also technically necessary, see Definition 6 below). In this way, we mark only the necessary symbols (in blocked positions).

However, if we simply apply RPO to the terms after marking the symbols in blocked positions the resulting ordering is not stable under substitutions. The difficult/interesting point of CSRPO is the treatment of variables, since variables in blocked positions are somehow smaller than variables in active positions, which is not taken into account in RPO (see [5] for a thorough discussion of this issue). This is achieved by appropriately marking the variables of terms which are compared using CSRPO. Now, by $\underline{\mathcal{X}}$ we denote the set of labeled variables corresponding to $\mathcal{X}$. The variables in $\underline{\mathcal{X}}$ are labeled by subsets of $\mathcal{F}$, for instance $x_{\{f,g,h\}}$, and we will ambiguously use the variables of $\mathcal{X}$ to denote variables labeled by the empty set.

When using the ordering, the marking map tells us whether we have to mark the top symbol of a term every time we go to an argument of this symbol. Thus, the marking map does not apply to the whole term but only to the top symbol of the arguments in the recursive calls of the definition of CSRPO. Therefore, if we have a term $f(s_1,\ldots,s_k)$, we will access the arguments using $mt(s_i,\mathfrak{m}(f,i))$, which represents the result of marking the top symbol (of arguments s_i) when required. This conditional marking of top symbols is defined by:

$$mt(f(s_1,\ldots,s_n),W) = \begin{array}{ll} \underline{f}(s_1,\ldots,s_n) & \text{if } f \in W \\ f(s_1,\ldots,s_n) & \text{otherwise} \end{array}$$

$$mt(x,W) = x_W \quad \text{where } x \text{ is a variable}$$

We write $s \in \underline{\mathcal{T}}(\mathcal{F},\mathcal{X})$ if $s \in \underline{\mathcal{X}}$, or $s = f(s_1,\ldots,s_n)$, with $f \in \mathcal{F} \cup \underline{\mathcal{F}}$ and $s_i \in \mathcal{T}(\mathcal{F},\mathcal{X})$ for all $i \in \{1,\ldots,n\}$. Note that, as said, marked symbols can only appear at the top of a term. A ground term s is in $\underline{\mathcal{T}}(\mathcal{F})$ if it is in $\underline{\mathcal{T}}(\mathcal{F},\mathcal{X})$ and contains no variable.

As remarked above, the treatment of variables is crucial in the definition of CSRPO. This is because we need to ensure that CSRPO is a stable ordering. In order to solve this problem, given a term s, we will provide the set of labeled variables that can be considered smaller than (or equal to) s without risk of losing stability under substitutions. Note that we label the variables with the symbols that should be marked in case of applying a substitution. To ensure that some labeled variable x_W is in the set of safe (wrt. stability) labeled variables of a term s, we need x to occur in s and to be sure that for any substitution σ we have that $mt(\sigma(x),W)$ is smaller than $\sigma(s)$. Therefore, assuming that x occurs in s, the important point is what happens with the function symbols heading $\sigma(s)$. Due to this we analyze which function symbols are harmless as head symbols. In all cases, the symbols which are included in the label W of x; additionally, all function symbols which do not appear in the label when we reach some occurrence of x in s are safe. Finally, and more importantly, the symbols g that can be proved to be safe because the head symbol of s (or recursively using some subterm of s containing x) is greater than or equal to g (and in the latter case they have multiset status), and $\underline{g}$ and g have the same marking.

DEFINITION 3. *([5]) Let s be a non-variable term in $\underline{\mathcal{T}}(\mathcal{F},\mathcal{X})$ and x_W a labeled variable. Then $x_W \in Stable(s)$ if and only if $x \in \mathcal{V}ar(s)$ and $f \in Safe(s,x)$ for all $f \in \mathcal{F} \setminus W$.*

The set $Safe(s,x)$ for some variable x s.t. $x \in \mathcal{V}ar(s)$ or $s = x_V$ (for some label V) is defined as the smallest subset of $\mathcal{F}$ containing

1. *if $s = x_V$ then all symbols in $\mathcal{F} \setminus V$.*
2. *if $s = f(s_1,\ldots,s_n)$ then*
   (a) *the union of all $Safe(mt(s_i,\mathfrak{m}(f,i)),y)$ with $i \in \{1\ldots n\}$ and $y \in \mathcal{V}ar(s_i)$, and*
   (b) *all $g \in \mathcal{F}$ such that $f =_{\mathcal{F}} g$ and $stat(f) = stat(g) = mul$, and $(\mathfrak{m}(\underline{g},i) = \mathfrak{m}(g,i))$ for all $i \in \{1,\ldots,ar(g)\}$.*
   (c) *all $g \in \mathcal{F}$ such that $f \succ_{\mathcal{F}} g$ and $(\mathfrak{m}(\underline{g},i) = \mathfrak{m}(g,i))$ for all $i \in \{1,\ldots,ar(g)\}$.*

Now we can give the definition of the context-sensitive recursive path ordering. First we give the definition of the equality relation, induced by the equality on function symbols, that we will use.

DEFINITION 4. *Given two terms in $\underline{\mathcal{T}}(\mathcal{F},\mathcal{X})$, we define $=_{\mathcal{S}}$ as follows:*

- *$f(s_1,\ldots,s_k) =_{\mathcal{S}} g(t_1,\ldots,t_k)$ iff $f =_{\mathcal{F}} g$ and $mt(s_i,\mathfrak{m}(f,i)) =_{\mathcal{S}} mt(t_i,\mathfrak{m}(g,i))$ for every $i \in \{1,\ldots,k\}$.*
- *$x_W =_{\mathcal{S}} x_{W'}$ iff $W = W'$*

We can enlarge the equality relation by considering permutations of arguments of symbols with multiset status.

DEFINITION 5 (CSRPO [5]). *Let $s,t \in \underline{\mathcal{T}}(\mathcal{F},\mathcal{X})$ $s = f(s_1,\ldots,s_n) \succ_{\mathcal{S}} t$ with $t \in \underline{\mathcal{X}}$ or $t = g(t_1,\ldots,t_m)$ iff*

1. *$t = x_W \in Stable(s)$*
2. *or $mt(s_i,\mathfrak{m}(f,i)) \succeq_{\mathcal{S}} t$, for some $i \in \{1,\ldots,n\}$*
3. *or $t = g(t_1,\ldots,t_m)$ with $f \succ_{\mathcal{F}} g$ and $s \succ_{\mathcal{S}} mt(t_i,\mathfrak{m}(g,i))$ for all $i \in \{1,\ldots,m\}$*
4. *or $t = g(t_1,\ldots,t_m)$ with $f =_{\mathcal{F}} g$, $stat(f) = mul$ and $\{mt(s_1,\mathfrak{m}(f,1)),\ldots,mt(s_n,\mathfrak{m}(f,n))\}$ $\succ\succ_{\mathcal{S}} \{mt(t_1,\mathfrak{m}(g,1)),\ldots,mt(t_m,\mathfrak{m}(g,m))\}$*
5. *or $t = g(t_1,\ldots,t_m)$, $f =_{\mathcal{F}} g$, $stat(f) = lex$, $\langle mt(s_1,\mathfrak{m}(f,1)),\ldots,mt(s_n,\mathfrak{m}(f,n))\rangle$ $\succ^{lex}_{\mathcal{S}} \langle mt(t_1,\mathfrak{m}(g,1)),\ldots,mt(t_m,\mathfrak{m}(g,m))\rangle$ and $s \succ_{\mathcal{S}} mt(t_i,\mathfrak{m}(g,i))$ for all $i \in \{1,\ldots,m\}$.*

where $s \succeq_{\mathcal{S}} t$ denotes $s \succ_{\mathcal{S}} t$ or $s =_{\mathcal{S}} t$, and $\succ\succ_{\mathcal{S}}$ and $\succ^{lex}_{\mathcal{S}}$ are respectively the multiset and lexicographic extension of $\succ_{\mathcal{S}}$ wrt. $=_{\mathcal{S}}$.

The precedence $\succeq_{\mathcal{F}}$ and the marking map $\mathfrak{m}$ have to satisfy some conditions to ensure the appropriate properties of $\succeq_{\mathcal{S}}$ for proving termination of CSR.

DEFINITION 6. *([5]) Let $\succeq_{\mathcal{F}}$ be a precedence, μ a replacement map and $\mathfrak{m}$ a marking map. Then $(\succeq_{\mathcal{F}},\mathfrak{m})$ is a valid marking pair if*

1. $\mathfrak{m}(f,i) = \varnothing \quad \forall f \in \mathcal{F},\ \forall i \in \mu(f)$
2. $f \succeq_{\mathcal{F}} \underline{f} \quad \forall f \in \mathcal{F}$
3. $\mathfrak{m}(f,i) \subseteq \mathfrak{m}(\underline{f},i) \quad \forall f \in \mathcal{F},\ \forall i \in \{1,\ldots,ar(f)\}$

```
obj EXAMPLE-TR is
  sorts Nat LNat .
  ops 0 0' : -> Nat .
  ops s s' : Nat -> Nat [strat (1)] .
  ops nil nil' : -> LNat .
  op cons : Nat LNat -> LNat [strat (1)] .
  op cons' : Nat LNat -> LNat [strat (1 2)] .
  op fcons : Nat LNat -> LNat [strat (1 2 0)] .
  op from : Nat -> LNat [strat (1 0)] .
  ops sel sel' : Nat LNat -> Nat [strat (1 2 0)] .
  ops first first' : Nat LNat -> LNat [strat (1 2 0)] .
  op quote : Nat -> Nat [strat (0)] .
  op unquote : Nat -> Nat [strat (1 0)] .
  op quote' : LNat -> LNat [strat (0)] .
  op unquote' : LNat -> LNat [strat (1 0)] .
  vars X Y : Nat .
  var Z : LNat .
  eq sel(s(X),cons(Y,Z)) = sel(X,Z) .
  eq sel(0,cons(X,Z)) = X .
  eq first(0,Z) = nil .
  eq first(s(X),cons(Y,Z)) = cons(Y,first(X,Z)) .
  eq from(X) = cons(X,from(s(X))) .
  eq sel'(s(X),cons(Y,Z)) = sel'(X,Z) .
  eq sel'(0,cons(X,Z)) = quote(X) .
  eq first'(0,Z) = nil' .
  eq first'(s(X),cons(Y,Z)) = cons'(quote(Y),first'(X,Z)) .
  eq quote(0) = 0' .
  eq quote'(cons(X,Z)) = cons'(quote(X),quote'(Z)) .
  eq quote'(nil) = nil' .
  eq quote(s(X)) = s'(quote(X)) .
  eq quote(sel(X,Z)) = sel'(X,Z) .
  eq quote'(first(X,Z)) = first'(X,Z) .
  eq unquote(0') = 0 .
  eq unquote(s'(X)) = s(unquote(X)) .
  eq unquote'(nil') = nil .
  eq unquote'(cons'(X,Z)) = fcons(unquote(X),unquote'(Z)) .
  eq fcons(X,Z) = cons(X,Z) .
endo
```

Figure 2. Transformed OBJ program

When valid marking maps are used, CSRPO is a reduction μ-ordering and can be used for proving termination of CSRS's ([5]): A CSRS $(\mathcal{R},\mu)$, where $\mathcal{R} = (\mathcal{F},R)$ is CSRPO-terminating if there is a precedence $\succeq_{\mathcal{F}}$ on $\mathcal{F}$ and a valid marking map $\mathfrak{m}$ for $\mathcal{F}$ such that $l \succ_{\mathcal{S}} r$ for all $l \to r \in R$.

Example 7. Consider the CSRS $(\mathcal{R},\mu)$ of Example 5. Termination of $(\mathcal{R},\mu)$ can also be proved using CSRPO: use the marking $\mathfrak{m}(:,2) = \mathfrak{m}(\underline{:},2) = \{from\}$ and the precedence $first \succ_{\mathcal{F}} from, [\,], :$ and $sel \succ_{\mathcal{F}} from \succ_{\mathcal{F}} \{:, s, \underline{from}\}$. We use the lexicographic status for *first* and *sel* and the multiset status for all other symbols (see Example 7 of [5]).

Example 8. Using rewriting restrictions may cause that some normal forms of input expressions are unreachable by restricted computation. For instance, the evaluation of $t = first(s(0),from(0))$ using the program in Figure 1 yields[5]:

```
==========================================

obj EXAMPLE

==========================================

reduce in EXAMPLE : first(s(0),from(0))

rewrites: 2

result LNat: cons(0,first(0,from(s(0))))
```

Note that *cons(0,first(0,from(s(0))))* is *not* a normal form. However, $t \to^* cons(0,nil)$, i.e., *cons(0,nil)* is a normal form of t which cannot be obtained by using the OBJ3 interpreter.

This can be solved by using program transformation techniques. For instance, by applying the program transformation from OBJ programs to OBJ programs of [2], we obtain the program of Figure 2. In contrast to the program in Figure 1, this new program can be used to fully evaluate expressions (see [2]). Moreover, we are also able to prove termination of this new program by using CSRPO [4]: use the marking $\mathfrak{m}(cons,2) = \mathfrak{m}(\underline{cons},2) = \mathfrak{m}(quote',1) = \mathfrak{m}(\underline{quote'},1) = \{from\}$ and the precedence

$$
\begin{array}{l}
sel =_{\mathcal{F}} sel' \succ_{\mathcal{F}} from, quote\\
first =_{\mathcal{F}} first' \succ_{\mathcal{F}} cons, from, nil, cons', nil', quote\\
from \succ_{\mathcal{F}} \underline{from}, cons, s\\
quote \succ_{\mathcal{F}} 0', s'\\
quote' \succ_{\mathcal{F}} cons', nil'\\
quote' \succeq_{\mathcal{F}} quote\\
fcons \succ_{\mathcal{F}} cons\\
unquote =_{\mathcal{F}} unquote' \succ_{\mathcal{F}} 0, s, nil, fcons
\end{array}
$$

We use the lexicographic status for all symbols.

We prove that CSRPO-termination of a CSRS implies simple termination of the CSRS.

THEOREM 9. *Let $(\mathcal{R},\mu)$ be a CSRS. If $(\mathcal{R},\mu)$ is CSRPO-terminating, then it is simply terminating.*

PROOF. If $(\mathcal{R},\mu)$ is CSRPO-terminating, then there exists a precedence $\succeq_{\mathcal{F}}$ and a valid marking map $\mathfrak{m}$ with $l \succ_{\mathcal{S}} r$ for every rule $l \to r$ of $\mathcal{R}$. We only need to prove that $l \succ_{\mathcal{S}} r$ also holds for every rule $l \to r$ in $\mathcal{E}mb^{\mu}(\mathcal{F})$. Note that, if $f(x_1,\ldots,x_k) \to x_i \in \mathcal{E}mb^{\mu}(\mathcal{F})$, then $i \in \mu(f)$. Then, $f(x_1,\ldots,x_k) \succ_{\mathcal{S}} x_i$ if $x_i \in Stable(f(x_1,\ldots,x_k))$. Note that we have $x_i \in Stable(f(x_1,\ldots,x_k))$ if and only if $f \in Safe(f(x_1,\ldots,x_k),x_i)$ for all $f \in \mathcal{F}$. Now, since $i \in \mu(f)$, it follows that $\mathfrak{m}(f,i) = \varnothing$. Hence, $mt(x_i,\varnothing) = (x_i)_{\varnothing}$ and $Safe((x_i)_{\varnothing},x_i) = \mathcal{F}$. Therefore, $Safe(f(x_1,\ldots,x_k),x_i) = \mathcal{F}$ and $f \in Safe(f(x_1,\ldots,x_k),x_i)$ for all $f \in \mathcal{F}$, which means $f(x_1,\ldots,x_k) \succ_{\mathcal{S}} x_i$. $\square$

For instance, the OBJ3 program of Figure 2 (viewed as a CSRS) is simply terminating. On the other hand, it is not difficult to see that termination of the CSRS of Example 1 cannot be proved using the CSRPO: we would need to prove that the tuple $\langle a,b,x\rangle$ containing constant symbols a,b is greater than the tuple $\langle x,x,x\rangle$ that only contains variables. According to the definition of CSRPO, this is not possible.

To summarize what we have done so far, we can say that our notion of simple termination of CSR plays (almost) the same role regarding CSR as simple termination for ordinary rewriting. We have proved that simply terminating TRSs $\mathcal{R}^{\mu}_{\Theta}$ prove simple termination of $(\mathcal{R},\mu)$ for all the existing transformations Θ. We have also proved that CSRPO-terminating CSRS's are simply terminating. Furthermore, we have given an example of a simply terminating CSRS which cannot be proved to be so by using the currently developed (automatic) techniques.

[5] We use the version 2.0 of the OBJ3 interpreter (available at http://www.kindsoftware.com/products/opensource).

Next we will consider still another method of direct termination proofs of CSRS's.

5.2 Polynomial Orderings

A monomial in k variables over $\mathbb{Z}$ is a function $F : \mathbb{Z}^k \to \mathbb{Z}$ defined by $F(x_1,\ldots,x_k) = a x_1^{r_1} \cdots x_k^{r_n}$ for some integer $a \neq 0$ and some non-negative integers $r_1,\ldots,r_k$. The number a is called the coefficient of the monomial. If $r_1 = r_2 = \cdots = r_n = 0$, then the monomial is called a constant. A polynomial in k variables over $\mathbb{Z}$ is the sum of finitely many monomials in k variables over $\mathbb{Z}$.

Given a signature $\mathcal{F}$ and $\mu \in M_{\mathcal{F}}$, let $(\mathbb{N}, \mathcal{F}_{\mathbb{N}}, >)$ be a $\mathcal{F}$-algebra such that, for all $f \in \mathcal{F}$, $f_{\mathbb{N}} \in \mathcal{F}_{\mathbb{N}}$ is a polynomial in $ar(f)$ variables satisfying (1) $f_{\mathbb{N}}(x_1,\ldots,x_k) \in \mathbb{N}$ for all $x_1,\ldots,x_k \in \mathbb{N}$ (well-definedness) and (2) $f_{\mathbb{N}}$ is μ-monotone. Then, $(\mathbb{N}, \mathcal{F}_{\mathbb{N}}, >)$ is a well-founded μ-monotone algebra and the corresponding μ-reduction ordering is denoted $>^{\mu}_{poly}$ and said to be a polynomial μ-ordering. The $\mathcal{F}$-algebra A is called a *polynomial μ-interpretation* for $\mathcal{F}$.

In fact, given a polynomial μ-interpretation $(\mathbb{N}, \mathcal{F}_{\mathbb{N}}, >)$, the corresponding μ-reduction ordering $>^{\mu}_{poly}$ can equivalently be defined as follows: for $t, s \in \mathcal{T}(\mathcal{F}, \mathcal{X})$,

$$t >^{\mu}_{poly} s \;\Leftrightarrow\; \forall x_i \in \mathbb{N}, [t] > [s]$$

where $[x] = x$ if $x \in \mathcal{X}$ and $[f(t_1,\ldots,t_k)] = f_{\mathbb{N}}([t_1],\ldots,[t_k])$ (i.e., we do not need to make explicit the evaluation mapping anymore).

A positive aspect of polynomial orderings regarding other reduction orderings is that they ease the proofs of monotonicity. In unrestricted rewriting, monotonicity of polynomial interpretations is normally ensured by requiring that all coeficients of all polynomials associated to function symbols be positive (see Proposition 10 in [40] or [3, Section 5.3]). Of course, we do not want to do this, as this would actually mean that we are using a reduction ordering thus making it useless for proving termination of CSR in the 'interesting' cases, i.e., when the TRS is not terminating. Even though there is no simple way to ensure μ-monotonicity of polynomial μ-orderings by constraining the shape of coeficients of monomials, we can still use the following result (where $\partial F/\partial x_i$ means the partial derivative of function F w.r.t. its i-th argument).

THEOREM 10. *Let $F : \mathbb{Z}^k \to \mathbb{Z}$ be a polynomial over $\mathbb{Z}$. Then, $F(x_1,\ldots,x_k)$ is monotone in its i-th argument if $\partial F/\partial x_i > 0$ for all $x_1,\ldots,x_k \in \mathbb{N}$.*

PROOF. Let $x, y \in \mathbb{Z}$ be such that $y > x$. Polynomials (viewed as functions on real numbers) are obviously differentiable in all their arguments. Then, by the intermediate value theorem, there is a real number $x < z < y$ such that the value of $\frac{\partial F}{\partial x_i}$ at $(x_1,\ldots,x_{i-1},z,\ldots,x_k)$ coincides with

$$\frac{F(x_1,\ldots,x_{i-1},y,\ldots,x_k) - F(x_1,\ldots,x_{i-1},x,\ldots,x_k)}{y-x}$$

Since, by hypothesis, this is a positive number and $y > x$, we have $F(x_1,\ldots,x_{i-1},y,\ldots,x_k) > F(x_1,\ldots,x_{i-1},x,\ldots,x_k)$, hence the conclusion. $\square$

Theorem 10 does *not* provide a full characterization of μ-monotony. For instance, the polynomial $F(x) = 2x^2 - x$ is obviously monotone, but $\partial F/\partial x = 4x - 1$ is not positive for $x = 0$. However, our result can be used to ensure (full) monotony (i.e., $\mu_{\top}$-monotony) of polynomials when more standard conditions (as the aforementioned ones) do not hold. For instance, polynomial $F(x) = x^3 - x^2 + x + 1$ contains negative coefficients (which, as mentioned before, is not allowed in the usual polynomial interpretations); the monotonicity of F can be ensured using Theorem 10 since $\partial F/\partial x = 3x^2 - 2x + 1 > 0$ for all $x \in \mathbb{N}$. We can use μ-polynomial orderings for proving termination of CSRS's.

Example 9. Consider the CSRS $(\mathcal{R},\mu)$ of Example 1 and the polynomial interpretation given by $f_{\mathbb{N}}(x,y,z) = 1 + (x-y)^2 + z$, $a_{\mathbb{N}} = 2$ and $b_{\mathbb{N}} = 1$. Obviously, $f_{\mathbb{N}}(x,y,z) \in \mathbb{N}$ for all $x,y,z \in \mathbb{N}$. Note also that $f_{\mathbb{N}}$ is μ-monotone: since $\partial f_{\mathbb{N}}/\partial z = 1 > 0$, this follows by Theorem 10. We have:

$$[f(a,b,x)] \;=\; 2+x \;>\; 1+x \;=\; [f(x,x,x)]$$

Then, $(\mathcal{R},\mu)$ is terminating.

Note that the polynomial μ-interpretation used in Example 9 is *not* monotonic in the standard case: For instance $f_{\mathbb{N}}(1,1,0) = 1 < 2 = f_{\mathbb{N}}(0,1,0)$ but $1 > 0$, i.e., $f_{\mathbb{N}}$ is not monotone in its first argument. Similarly, $f_{\mathbb{N}}(1,1,0) = 1 < 2 = f_{\mathbb{N}}(1,0,0)$, i.e., $f_{\mathbb{N}}$ is not monotone in its second argument either.

As for the unrestricted case, polynomial μ-orderings are *well-founded* μ-simplification orderings if we additionally require that $f_{\mathbb{N}}(x_1,\ldots,x_k) > x_i$ for all $i \in \mu(f)$.

THEOREM 11. *Let $\mathcal{F}$ be a signature containing at least a constant symbol, $\mu \in M_{\mathcal{F}}$ and $(\mathbb{N}, \mathcal{F}_{\mathbb{N}}, >)$ be a polynomial μ-interpretation such that for all $f \in \mathcal{F}$ and $i \in \mu(f)$, $f_{\mathbb{N}}(x_1,\ldots,x_k) > x_i$ for all $x_1,\ldots,x_k \in \mathbb{N}$ Then, $>^{\mu}_{poly}$ is a well-founded μ-simplification ordering.*

Now, we have the following immediate corollary.

COROLLARY 1. *Let $(\mathcal{R},\mu)$ be a CSRS and $>^{\mu}_{poly}$ be a polynomial μ-simplification ordering. If $l >^{\mu}_{poly} r$ for all rule $l \to r$ in $\mathcal{R}$, then $(\mathcal{R},\mu)$ is simply terminating.*

Example 10. Continuing Example 9. Note that $f_{\mathbb{N}}$ in Example 9 verifies $f_{\mathbb{N}}(x,y,z) \geq z+1 > z$ for all $x,y,z \in \mathbb{N}$. Hence, $(\mathcal{R},\mu)$ is simply terminating.

6 Modularity of Simple Termination of CSRS's

We shall now investigate to what extent the notion of simple termination of CSRS's introduced behaves in a modular way, i.e., whether simple termination of two given CSRS's implies simple termination of their union. Such a kind of modularity analysis for general termination of CSRS's has recently been initiated in [22] with promising first results. Let us recall a few notions and results from [22] that we need subsequently. For simplicity, for the rest of the paper we assume that all considered CSRS's are finite. Some of the results (but not all) do also hold for arbitrary (infinite) systems.

DEFINITION 7. *We say that a property $\mathcal{P}$ of CSRS's is* modular for disjoint unions *if, whenever two disjoint CSRS's have property $\mathcal{P}$, then their (disjoint) union also does.*[6]

This notion of modularity can be generalized in a straightforward way to other (more general) classes of combinations of CSRS's.

[6]The reverse implication usually holds, too, but for simplicity we don't include it in the definition here.

DEFINITION 8. *A rule $l \to r$ in a CSRS $(\mathcal{R},\mu)$ is* non-duplicating *if for every $x \in \mathcal{V}ar(l)$ the multiset of replacing occurrences of x in r is contained in the multiset of replacing occurrences of x in l, and* duplicating *otherwise. $(\mathcal{R},\mu)$ is* non-duplicating *if all its rules are, and* duplicating, *otherwise. A rule $l \to r$ is said to be* collapsing, *if r is a variable, and* non-collapsing *otherwise. A CSRS is* collapsing *if it has a collapsing rule, and* non-collapsing, *otherwise.*

Note that for CSRS's without any replacement restrictions, collapsingness and non-duplication as above just yield the corresponding well-known notions for TRS's.

Of course, in order to sensibly combine two CSRS's, one should require some basic *compatibility* condition regarding the respective replacement restrictions.

DEFINITION 9. *Two CSRS's $(\mathcal{R}_1,\mu_1)$, $(\mathcal{R}_2,\mu_2)$ are said to be* compatible *if they have the same replacement restrictions for shared function symbols, i.e., if $\mathcal{R}_1 = (\mathcal{F}_1, R_1)$ and $\mathcal{R}_2 = (\mathcal{F}_2, R_2)$, we have $\mu_1(f) = \mu_2(f)$ for every $f \in \mathcal{F}_1 \cap \mathcal{F}_2$. The union $(\mathcal{R},\mu)$ of two compatible CSRS's $(\mathcal{R}_1,\mu_1)$, $(\mathcal{R}_2,\mu_2)$ is defined componentwise, i.e., $\mathcal{R} = (R,\mathcal{F})$ with $R = R_1 \cup R_2$, $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$, and $\mu = \mu_1 \sqcup \mu_2$.*

Disjoint CSRS's are trivially compatible.

THEOREM 12. *([22]) Let $(\mathcal{R}_1,\mu_1)$, $(\mathcal{R}_2,\mu_2)$ be two disjoint, terminating CSRS's, and let $(\mathcal{R},\mu)$ be their union. Then the following hold:*

(i) $(\mathcal{R},\mu)$ terminates, if both $(\mathcal{R}_1,\mu_1)$ and $(\mathcal{R}_2,\mu_2)$ are non-collapsing.

(ii) $(\mathcal{R},\mu)$ terminates, if both $(\mathcal{R}_1,\mu_1)$ and $(\mathcal{R}_2,\mu_2)$ are non-duplicating.

(iii) $(\mathcal{R},\mu)$ terminates, if one of the systems is both non-collapsing and non-duplicating.

DEFINITION 10. *A TRS $\mathcal{R}$ is said to be* terminating under free projections, *FP-terminating for short, if the disjoint union of $\mathcal{R}$ and the TRS $(\{G\},\{G(x,y) \to x, G(x,y) \to y\})$ is terminating. A CSRS $(\mathcal{R},\mu)$ is said to be FP-terminating, if the disjoint union of $\mathcal{R}$ and the CSRS $((\{G\},\{G(x,y) \to x, G(x,y) \to y\}),\mu)$, where $\mu(G) = \{1,2\}$, is terminating.*

THEOREM 13. *([22]) Let $(\mathcal{R}_1,\mu_1)$, $(\mathcal{R}_2,\mu_2)$ be two disjoint, terminating CSRS's, such that their union $(\mathcal{R},\mu)$ is non-terminating. Then one of the systems is not FP-terminating, and the other system is collapsing.*

As already shown for TRS's, this abstract and powerful structure result has a lot of direct and indirect consequences and corollaries. To mention only a few:

DEFINITION 11. *A CSRS is* non-deterministically collapsing *if there is a term that reduces to two distinct variables (in a finite number of context-sensitive rewrite steps).*

THEOREM 14. *Any non-deterministically collapsing, terminating CSRS is FP-terminating.*

THEOREM 15. *([22]) Termination is modular for* non-deterministically collapsing *disjoint CSRS's.*

Consequently we also get the next result.

THEOREM 16. *FP-termination is modular for disjoint CSRS's.*

In the case of TRS's it is well-known that simple termination is modular for disjoint unions ([25]). This can also be shown via the TRS version of the general Theorem 13 above (cf. [21]). In particular, we note that for TRS's we have the equivalence: $(\mathcal{F},\mathcal{R})$ is simply terminating iff $(\mathcal{F},\mathcal{R} \cup \mathcal{E}mb(\mathcal{F}))$ is terminating. Now it is obvious that if $\mathcal{F}$ contains a function symbols of arity at least 2, then $(\mathcal{R} \cup \mathcal{E}mb(\mathcal{F}),\mu)$ is non-deterministically collapsing, and, if additionally $(\mathcal{R} \cup \mathcal{E}mb(\mathcal{F}),\mu)$ is terminating, then $(\mathcal{R} \cup \mathcal{E}mb(\mathcal{F}),\mu)$ is FP-terminating. In fact, even for the case where the TRS $(\mathcal{F},\mathcal{R})$ has only functions symbols of arity 0 and 1, simple termination of $(\mathcal{F},\mathcal{R})$ implies its FP-termination as can be easily shown (e.g., by a minimal counterexample proof).[7]

With these preparations, we are ready now to tackle modularity of simple termination for CSRS's.

THEOREM 17. *Let $(\mathcal{R},\mu)$ with $\mathcal{R} = (\mathcal{F},R)$ be a CSRS with $|\mu(f)| \geq 2$ for at least one $f \in \mathcal{F}$. Then simple termination of $(\mathcal{R},\mu)$ implies FP-termination of $(\mathcal{R},\mu)$.*

PROOF. Simple termination of $(\mathcal{R},\mu)$ means termination of $(\mathcal{R} \cup \mathcal{E}mb^{\mu}(\mathcal{F}),\mu)$. By assumption, there is an $f \in \mathcal{F}$ with $|\mu(f)| \geq 2$. Thus, $f(\ldots,x_i,\ldots,x_j,\ldots)$ rewrites to both x_i and x_j (using $\mathcal{E}mb^{\mu}(\mathcal{F})$) for $i,j \in \mu(f)$, $i < j$. But this means that $(\mathcal{R} \cup \mathcal{E}mb^{\mu}(\mathcal{F}),\mu)$ is non-deterministically collapsing, hence, by Theorem 14, $(\mathcal{R} \cup \mathcal{E}mb^{\mu}(\mathcal{F}),\mu)$ and, consequently, also $(\mathcal{R},\mu)$ are FP-terminating. $\square$

By combining Theorem 17 and Theorem 13 we get now the following modularity result for simple termination.

THEOREM 18. *Let $(\mathcal{R}_1,\mu_1)$, $(\mathcal{R}_2,\mu_2)$ with $\mathcal{R}_1 = (\mathcal{F}_1,R_1)$, $\mathcal{R}_2 = (\mathcal{F}_2,R_2)$ be two disjoint CSRS's, and let $(\mathcal{R},\mu)$ be their disjoint union (with $\mathcal{R} = (\mathcal{F},R)$, $\mathcal{F} = \mathcal{F}_1 \uplus \mathcal{F}_2$, $R = R_1 \uplus R_2$, $\mu = \mu_1 \sqcup \mu_2$). Moreover suppose that there exists an $f_i \in \mathcal{F}_i$ with $|\mu(f_i)| \geq 2$, for $i = 1,2$. Then, if $(\mathcal{R}_1,\mu_1)$ and $(\mathcal{R}_2,\mu_2)$ are simply terminating, then $(\mathcal{R},\mu)$ is simply terminating, too.*

Interestingly, for the proof of this result via Theorem 17 above, we need the technical assumption that there exists an $f_i \in \mathcal{F}_i$ with $|\mu(f_i)| \geq 2$, for $i = 1,2$. Currently, we do not know whether the statement of Theorem 17 also holds without this condition. If yes, Theorem 18 would immediately generalize, too, and yield in general modularity of simple termination for CSRS's.

But note that if this condition above is not satisfied, any proof of the corresponding statement in Theorem 17 cannot work as in the TRS case any more, since now we may have arbitrarily complicated terms.

Finally, let us consider the case of (some) non-disjoint unions of CSRS's.

6.1 Extension to the Constructor-Sharing Case

Finally, let us consider the case of (some) non-disjoint unions of CSRS's.

[7] Note that in this case the terms over $\mathcal{F}$ have a very simple shape, and essentially are strings.

DEFINITION 12. *For a CSRS* $(\mathcal{R},\mu)$*, where* $\mathcal{R} = (\mathcal{F}, R)$*, the set of* defined (function) symbols *is* $\mathcal{D} = \{root(l) \mid l \to r \in R\}$*, its set of* constructors *is* $\mathcal{C} = \mathcal{F} \setminus \mathcal{D}$ *(thus* $\mathcal{F} = \mathcal{C} \uplus \mathcal{D}$*). Let* $(\mathcal{R}_1,\mu_1)$*,* $(\mathcal{R}_2,\mu_2)$ *be CSRS's with* $\mathcal{F}_1, \mathcal{F}_2$*;* $\mathcal{C}_1$*,* $\mathcal{C}_2$*, and* $\mathcal{D}_1$*,* $\mathcal{D}_2$ *denoting their respective signatures, sets of constructors, and defined function symbols. Then* $(\mathcal{R}_1,\mu_1)$ *and* $(\mathcal{R}_2,\mu_1)$ *are said to be* (at most) constructor sharing *if* $\mathcal{D}_1 \cap \mathcal{F}_2 = \mathcal{D}_2 \cap \mathcal{F}_1 = \varnothing$*. The set of* shared constructors *between them is* $\mathcal{C} = \mathcal{C}_1 \cap \mathcal{C}_2$*. A rule* $l \to r \in \mathcal{R}_i$ *is said to be* (shared) constructor lifting *if* $root(r) \in \mathcal{C}$*, for* $i = 1,2$*.* $\mathcal{R}_i$ *is said to be* (shared) constructor lifting *if it has a constructor lifting rule (*$i = 1,2$*). A rule* $l \to r \in \mathcal{R}_i$ *is said to be* shared symbol lifting *if* $root(r)$ *is a variable or a shared constructor.* $\mathcal{R}_i$ *is said to be* shared symbol lifting *if it is collapsing or has a constructor lifting rule.* $\mathcal{R}_i$ *is* layer preserving *if it is not shared symbol lifting.*

DEFINITION 13. *Let* $((\mathcal{F}, R, \mathcal{F}),\mu)$ *be a CSRS and* $f \in \mathcal{F}$*. We say that* f *is* fully replacing *if* $\mu(f) = \{1, \ldots, n\}$ *where* n *is the arity of* f*.*

THEOREM 19 ([22], EXTENDS [21, THEOREM 34]).
Let $(\mathcal{R}_1,\mu_1)$*,* $(\mathcal{R}_2,\mu_2)$ *be two constructor sharing, compatible, terminating CSRS's with all shared constructors fully replacing, such that their union* $(\mathcal{R},\mu)$ *is non-terminating. Then one of the systems, is not FP-terminating and the other system is shared symbol lifting (i.e., collapsing or constructor lifting).*[8]

PROOF. We only sketch the proof idea. Analogous to [21] one considers a minimal counterexample, i.e., a non-terminating derivation in the union with a minimal number of alternating *layers*. By using some sophisticated abstracting transfromation such a counterexample can be translated into a counterexample that uses only one of the two systems plus a disjoint system with only two rules of the form $\{G(x,y) \to x, G(x,y) \to y\}$ (that serve for "extracting relevant information of the former signature by need"). Note that for this construction to work properly, we need the assumption that the shared constructors are fully replacing. □

Without the above assumption the statement of the Theorem does not hold in general.

Example 11. Consider the CSRS's

$$\mathcal{R}_1: \quad zeros \to 0 : zeros$$

and

$$\mathcal{R}_2: \quad \begin{array}{c} length([]) \to 0 \\ length(x : y) \to s(length(y)) \end{array}$$

with $\mu(:) = \mu(length) = \mu(s) = \{1\}$, that have a shared constructor ':' which is not fully replacing. Now, both CSRS's are obviously terminating and also FP-terminating, but their union is not due to the cycle

$$length(zeros) \hookrightarrow length(0 : zeros) \hookrightarrow s(length(zeros))\ .$$

As for disjoint unions, from Theorem 19 above many results can be derived. Especially regarding simple termination, we have the following.

THEOREM 20. *Let* $(\mathcal{R}_1,\mu_1)$*,* $(\mathcal{R}_2,\mu_2)$ *with* $\mathcal{R}_1 = (\mathcal{F}_1, R_1)$*,* $\mathcal{R}_2 = (\mathcal{F}_2, R_2)$ *be constructor-sharing CSRS's with all shared constructors fully replacing, and let* $(\mathcal{R},\mu)$ *be their union (with* $\mathcal{R} = (\mathcal{F}, \mathcal{R})$*,* $\mathcal{F} = \mathcal{F}_1 \uplus \mathcal{F}_2$*,* $\mu = \mu_1 \sqcup \mu_2$*). Moreover suppose that there exists an* $f_i \in \mathcal{F}_i$ *with* $|\mu(f_i)| \geq 2$*, for* $i = 1,2$*. Then, if* $(\mathcal{R}_1,\mu_1)$ *and* $(\mathcal{R}_2,\mu_2)$ *are simply terminating, then* $(\mathcal{R},\mu)$ *is simply terminating, too.*

Note that also other "symmetric" and "asymmetric" results can be easily obtained from Theorem 19 and also from Theorem 20, anaologously to the case of disjoint unions of CSRS's (and as for TRS's).

A simple (though somewhat artificial) application example is the following variant of Example 1.

Example 12. The two CSRS's

$$\mathcal{R}_1: \quad \begin{array}{r} g(x,y) \to x \\ g(x,y) \to y \\ f(a,b,x) \to f(x,x,x) \end{array}$$

and

$$\mathcal{R}_2: \quad h(a,b) \to a$$

with $\mu(h) = \mu(g) = \{1,2\}$, $\mu(f) = \{3\}$. The systems are compatible, constructor-sharing and all shared constructors are (trivially) fully replacing. Both are simply terminating (as it is not very difficult to show; for instance, consider the polynomial interpretation of Example 9 together with $g_{\mathbb{N}}(x,y) = x+y+1$), hence the combined system is also simply terminating by Theorem 20. Note that the union includes the full version of Toyama's TRS (which is non-terminating in absence of replacement restrictions), here proved to be simply terminating regarding CSR (for the selected replacement map).

7 Conclusion

We have introduced a definition of simple termination of CSR and analyzed its suitability as a unifying notion which, similar to simple termination of rewriting, represents a class of orderings which are well-suited for automatization. We have proven that all existing transformations for proving termination of CSR can also be used for proving simple termination of CSRS's (this is in contrast to other restricted forms of termination like, e.g., innermost termination of CSR, see [18, 22]). We have also shown that *CSRPO-termination* is actually a method for proving simple termination of CSRS's. We have analyzed the use of polynomial orderings as a tool for analyzing (simple) termination of CSRS's. After this analysis, our notion of simple termination appears to be a quite natural one for the context-sensitive setting. In contrast to the context-free case where simplification orderings (over finite signatures) are automatically well-founded by Kruskal's Theorem, in the context-sensitive case termination needs to be explicitly established if such orderings are used for termination proofs. Some possible lines for doing so have also been mentioned. Finally, we have also obtained some first modularity results concerning simple termination of CSRS's. These results are quite encouraging and we expect that further similar results can be obtained, and also be extended to more general combinations of CSRS's, e.g., to composable ([34]) and to certain hierarchical ([24], [11]) systems.

Acknowledgements

We thank the anonymous referees for their helpful remarks.

8 References

[1] M. Alpuente, S. Escobar, B. Gramlich, and S. Lucas. Improving on-demand strategy annotations. In M. Baaz and A. Voronkov, editors,

[8] As for TRS's, this result holds not only for finite CSRS's, but also for finitely branching ones. But, in contrast to the disjoint union case, it doesn't hold any more for infinitely branching systems, cf. [35] for a counterexample.

Proc. 9th Int. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'02), Lecture Notes in Computer Science, Tbilisi, Georgia, Oct. 2002. Springer-Verlag. To appear.

[2] M. Alpuente, S. Escobar, and S. Lucas. Correct and complete (positive) strategy annotations for OBJ. *Electronic Notes in Theoretical Computer Science*, 71, 2002.

[3] F. Baader and T. Nipkow. *Term rewriting and All That*. Cambridge University Press, 1998.

[4] C. Borralleras. Personal communication, May 2002.

[5] C. Borralleras, S. Lucas, and A. Rubio. Recursive path orderings can be context-sensitive. In A. Voronkov, editor, *Proc. 18th International Conference on Automated Deduction (CADE'02)*, volume 2392 of *Lecture Notes in Artificial Intelligence*, pages 314–331, Copenhagen, Denmark, July 2002. Springer-Verlag.

[6] M. Clavel, S. Eker, P. Lincoln, and J. Meseguer. Principles of Maude. In J. Meseguer, editor, *Proc. 1st International Workshop on Rewriting Logic and its Applications (WRLA'96)*, volume 4 of *Electronic Notes in Theoretical Computer Science*, Pacific Grove, California, Sept. 1996. Elsevier. 25 pages.

[7] M. Dauchet. Simulation of turing machines by a regular rewrite rule. *Theoretical Computer Science*, 103(2):409–420, 1992.

[8] N. Dershowitz. A note on simplification orderings. *Information Processing Letters*, 9(5):212–215, 1979.

[9] N. Dershowitz. Orderings for term rewriting systems. *Theoretical Computer Science*, 17(3):279–301, 1982.

[10] N. Dershowitz. Termination of rewriting. *Journal of Symbolic Computation*, 3(1):69–116, 1987. Corrigendum in: Journal of Symbol Computation, 4(1):409–410, 1987.

[11] N. Dershowitz. Hierarchical termination. In N. Dershowitz and N. Lindenstrauss, editors, *Proc. 4th Int. Workshop on Conditional and Typed Rewriting Systems, Jerusalem, Israel (1994)*, volume 968 of *Lecture Notes in Computer Science*, pages 89–105. Springer-Verlag, 1995.

[12] N. Dershowitz and D. Plaisted. Rewriting. In J. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume 1, chapter 9, pages 535–610. Elsevier and MIT Press, 2001.

[13] M. Ferreira and A. Ribeiro. Context-sensitive AC-rewriting. In P. Narendran and M. Rusinowitch, editors, *Proc. 10th International Conference on Rewriting Techniques and Applications (RTA'99)*, volume 1631 of *Lecture Notes in Computer Science*, pages 286–300, Trento, Italy, July 1999. Springer-Verlag.

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

[15] K. Futatsugi and A. Nakagawa. An overview of CAFE specification environment – an algebraic approach for creating, verifying, and maintaining formal specifications over networks. In *Proc. 1st IEEE International Conference on Formal Engineering Methods (ICFEM'97)*, pages 170–182, Hiroshima, Japan, 1997. IEEE Computer Society. http://computer.org/proceedings/icfem/8002/8002toc.htm.

[16] J. Giesl and A. Middeldorp. Transformation techniques for context-sensitive rewrite systems. In P. Narendran and M. Rusinowitch, editors, *Proc. 10th International Conference on Rewriting Techniques and Applications (RTA'99)*, volume 1631 of *Lecture Notes in Computer Science*, pages 271–287, Trento, Italy, July 1999. Springer-Verlag.

[17] J. Giesl and A. Middeldorp. Transforming context-sensitive rewrite systems. In Y. Toyama, editor, *Proc. International Workshop on Rewriting in Proof and Computation (RPC'01)*, RIEC, pages 14–33, Tohoku University, Japan, 2001.

[18] J. Giesl and A. Middeldorp. Innermost termination of context-sensitive rewriting. Aachener Informatik-Berichte (AIBs) 2002-04, RWTH Aachen, 2002.

[19] J. Giesl and A. Middeldorp. Transformation techniques for context-sensitive rewrite systems. Aachener Informatik-Berichte (AIBs) 2002-02, RWTH Aachen, 2002.

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

[21] B. Gramlich. Generalized sufficient conditions for modular termination of rewriting. *Applicable Algebra in Engineering, Communication and Computing*, 5:131–158, 1994.

[22] B. Gramlich and S. Lucas. Modular termination of context-sensitive rewriting. In C. Kirchner, editor, *Proc. of 4th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming, PPDP'02*, Pittsburg, USA, 2002. ACM Press, New York. To appear.

[23] D. Knuth and P. Bendix. Simple word problems in universal algebra. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, 1970.

[24] M. Krishna Rao. Modular proofs for completeness of hierarchical term rewriting systems. *Theoretical Computer Science*, 151(2):487–512, Nov. 1995.

[25] M. Kurihara and A. Ohuchi. Modularity of simple termination of term rewriting systems with shared constructors. *Theoretical Computer Science*, 103:273–282, 1992.

[26] D. Lankford. On proving term rewriting systems are noetherian. Technical Report MTP 3, Louisiana Technical University, Ruston, 1979.

[27] S. Lucas. Termination of context-sensitive rewriting by rewriting. In F. Meyer auf der Heide and B. Monien, editors, *Proc. 23rd International Colloquium on Automata, Languages and Programming (ICALP'96)*, volume 1099 of *Lecture Notes in Computer Science*, pages 122–133, Paderborn, Germany, July 1996. Springer-Verlag.

[28] S. Lucas. Context-sensitive computations in functional and functional logic programs. *Journal of Functional and Logic Programming*, 1998(1):1–61, Jan. 1998. The MIT Press.

[29] S. Lucas. Termination of on-demand rewriting and termination of OBJ programs. In H. Sondergaard, editor, *Proc. 3rd International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming (PPDP'01)*, pages 82–93, Firenze, Italy, Sept. 2001. ACM Press, New York.

[30] S. Lucas. Termination of rewriting with strategy annotations. In R. Nieuwenhuis and A. Voronkov, editors, *Proc. 8th Int. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'01)*, volume 2250 of *Lecture Notes in Computer Science*, pages 669–684, Havana, Cuba, Dec. 2001. Springer-Verlag.

[31] S. Lucas. Context-sensitive rewriting strategies. *Information and Computation*, 2002. To appear.

[32] S. Lucas. Termination of (canonical) context-sensitive rewriting. In S. Tison, editor, *Proc. 13th International Conference on Rewriting Techniques and Applications (RTA'02)*, volume 2378 of *Lecture Notes in Computer Science*, pages 296–310, Copenhagen, Denmark, July 2002. Springer-Verlag.

[33] A. Middeldorp and B. Gramlich. Simple termination is difficult. *Applicable Algebra in Engineering, Communication and Computing*, 6(2):115–128, 1995.

[34] E. Ohlebusch. *Modular Properties of Composable Term Rewriting Systems*. PhD thesis, Universität Bielefeld, 1994. Report 94-01.

[35] E. Ohlebusch. On the modularity of termination of term rewriting systems. *Theoretical Computer Science*, 136:333–360, 1994.

[36] E. Ohlebusch. *Advanced Topics in Term Rewriting*. Springer-Verlag, Berlin, 2002.

[37] J. Steinbach. Simplification orderings: History of results. *Fundamenta Informaticae*, 24:47–88, 1995.

[38] J. Steinbach and H. Xi. Freezing – termination proofs for classical, context-sensitive and innermost rewriting. Technical report, Institut für Informatik, TU München, Jan. 1998.

[39] H. Zantema. Termination of context-sensitive rewriting. In H. Comon, editor, *Proc. 8th International Conference on Rewriting Techniques and Applications (RTA'97)*, volume 1232 of *Lecture Notes in Computer Science*, pages 172–186, Sitges, Spain, June 1997. Springer-Verlag.

[40] H. Zantema. Termination. In TeReSe, editor, *Term Rewriting Systems*, chapter 6. Cambridge University Press, 2002.

Appendix

LEMMA 1. *Let $(\mathcal{R},\mu)$ be a CSRS where $\mathcal{R} = (\mathcal{F},R)$. Then, $(\mathcal{R} \cup \mathcal{E}mb^\mu(\mathcal{F}))^\mu_{GM} = \mathcal{R}^\mu_{GM} \cup \mathcal{S}$, where the TRS $\mathcal{S}$ over the signature $\mathcal{F} \cup \{active\}$ consists of the following rules:*

$$\{active(f(x_1,\ldots,x_k)) \to mark(x_i) \mid f \in \mathcal{F}, i \in \mu(f)\}.$$

PROOF. Immediate from the definition of the transformation. □

Following Giesl and Middeldorp [19], we consider the confluent and terminating TRS $\mathcal{M}$ containing the rules

$$mark(f(x_1,\ldots,x_k)) \to active(f([x_1]_f,\ldots,[x_k]_f)$$

for each $f \in \Sigma$. Given a term t, we let $t{\downarrow_\mathcal{M}}$ to be the normal form of t according to rules of $\mathcal{M}$.

LEMMA 2. *[19, Lemma 1] Let $(\mathcal{R},\mu)$ be a CSRS over a signature $\mathcal{F}$ and $t,s \in \mathcal{T}(\mathcal{F})$. If $t \hookrightarrow s$, then $mark(t){\downarrow_\mathcal{M}} \to^+_{\mathcal{R}^\mu_{GM}} mark(s){\downarrow_\mathcal{M}}$.*

Theorem 7 *Let $(\mathcal{R},\mu)$ be a CSRS. If $\mathcal{R}^\mu_{GM}$ is simply terminating, then $(\mathcal{R},\mu)$ is simply terminating.*

PROOF. If $(\mathcal{R},\mu)$ is not simply terminating, then there exists an infinite sequence $A : t_1 \hookrightarrow t_2 \hookrightarrow \cdots \hookrightarrow t_n \hookrightarrow \cdots$ in $\mathcal{R} \cup \mathcal{E}mb^\mu(\mathcal{F})$, where $t_i \in \mathcal{T}(\mathcal{F})$ for $i \geq 1$. By Lemma 2, each step $t \hookrightarrow_{\mathcal{R}\cup\mathcal{E}mb^\mu(\mathcal{F})} s$ at position $p \in \mathcal{P}os^\mu(t)$ induces a non-empty reduction sequence $mark(t){\downarrow_\mathcal{M}} \to^+_{(\mathcal{R}\cup\mathcal{E}mb^\mu(\mathcal{F}))^\mu_{GM}} mark(s){\downarrow_\mathcal{M}}$. By Lemma 1 such a sequence makes use of rules coming either from $\mathcal{R}^\mu_{GM}$ or from

$$\mathcal{S} = \{active(f(x_1,\ldots,x_k)) \to mark(x_i) \mid f \in \mathcal{F}, i \in \mu(f)\}.$$

Borrowing the proof of Lemma 2 (see [19]), we prove by induction on p that the rewrite sequence $mark(t){\downarrow_\mathcal{M}} \to^+_{\mathcal{R}^\mu_{GM}\cup\mathcal{S}} mark(s){\downarrow_\mathcal{M}}$ can be simulated by using the rules in $\mathcal{R}^\mu_{GM} \cup \mathcal{E}mb(\mathcal{F}^\mu_{GM})$, i.e., we prove that $mark(t){\downarrow_\mathcal{M}} \to^+_{\mathcal{R}^\mu_{GM}\cup\mathcal{E}mb(\mathcal{F}^\mu_{GM})} mark(s){\downarrow_\mathcal{M}}$. If $p = \Lambda$, then $t = f(t_1,\ldots,t_k) = \sigma(l)$ for some $l \to r$ in $\mathcal{R} \cup \mathcal{E}mb^\mu(\mathcal{F})$. If $l \to r$ is a rule $f(x_1,\ldots,x_k) \to x_i \in \mathcal{E}mb^\mu(\mathcal{F})$, for some $i \in \mu(f)$ then $s = t|_i$. By Lemma 2,

$$mark(t) \to^+_\mathcal{M} mark(t){\downarrow_\mathcal{M}} \to^+_{\mathcal{R}^\mu_{GM}\cup\mathcal{S}} mark(t_i){\downarrow_\mathcal{M}}.$$

Now, since $\mathcal{F} \subseteq \mathcal{F}^\mu_{GM}$, we only need to apply the rule $f(x_1,\ldots,x_k) \to x_i \in \mathcal{E}mb(\mathcal{F}^\mu_{GM})$ to t before initiating its normalization w.r.t. $\mathcal{M}$, i.e.,

$$mark(t) \to_{\mathcal{E}mb(\mathcal{F}^\mu_{GM})} mark(t_i) \to^+_\mathcal{M} mark(t_i){\downarrow_\mathcal{M}}.$$

If $l \to r \in \mathcal{R}^\mu_{GM}$, then we only need to apply Lemma 2 to $\mathcal{R}^\mu_{GM}$ as no embedding rule is needed.

If $p = i.q$, then we have $t = f(t_1,\ldots,t_i,\ldots,t_k)$ and $s = f(t_1,\ldots,s_i,\ldots,t_k)$ with $t_i \hookrightarrow_{\mathcal{R}\cup\mathcal{E}mb^\mu(\mathcal{F})} s_i$. Note that, since $p \in \mathcal{P}os^\mu(t)$, we have $i \in \mu(f)$. For $1 \leq j \leq k$ we let $t'_i = mark(t_j){\downarrow_\mathcal{M}}$ if $j \in \mu(f)$ and $t'_j = t_j$ if $j \notin \mu(f)$. By the induction hypothesis, $t'_i = mark(t_i){\downarrow_\mathcal{M}} \to^+_{\mathcal{R}^\mu_{GM}\cup\mathcal{E}mb(\mathcal{F}^\mu_{GM})} mark(s_i){\downarrow_\mathcal{M}}$. Since $mark(t){\downarrow_\mathcal{M}} = active(f(t'_1,\ldots,t'_i,\ldots,t'_k))$ and $mark(s){\downarrow_\mathcal{M}} = active(f(s'_1,\ldots,mark(s_i){\downarrow_\mathcal{M}},\ldots,s'_k))$ we conclude the desired intermediate result.

Therefore, the infinite sequence A induces an infinite sequence

$$B : mark(t_1){\downarrow_\mathcal{M}} \to^+_{\mathcal{R}^\mu_{GM}\cup\mathcal{S}} mark(t_2){\downarrow_\mathcal{M}} \to^+_{\mathcal{R}^\mu_{GM}\cup\mathcal{S}} \cdots$$

which induces an infinite sequence

$$\begin{aligned} B' : \quad & mark(t_1){\downarrow_\mathcal{M}} \to^+_{\mathcal{R}^\mu_{GM}\cup\mathcal{E}mb(\mathcal{F}^\mu_{GM})} mark(t_2){\downarrow_\mathcal{M}} \\ & \to^+_{\mathcal{R}^\mu_{GM}\cup\mathcal{E}mb(\mathcal{F}^\mu_{GM})} mark(t_3){\downarrow_\mathcal{M}} \\ & \to^+_{\mathcal{R}^\mu_{GM}\cup\mathcal{E}mb(\mathcal{F}^\mu_{GM})} \cdots \end{aligned}$$

thus contradicting simple termination of $\mathcal{R}^\mu_{GM}$. □

LEMMA 3. *Let $(\mathcal{R},\mu)$ be a CSRS where $\mathcal{R} = (\mathcal{F},R)$. Then, $(\mathcal{R} \cup \mathcal{E}mb^\mu(\mathcal{F}))^\mu_C = \mathcal{R}^\mu_C \cup \{active(f(x_1,\ldots,x_k)) \to mark(x_i) \mid f \in \mathcal{F}, i \in \mu(f)\}$.*

PROOF. Immediate from the definition of the transformation. □

LEMMA 4. *[19, Lemma 2] Let $(\mathcal{R},\mu)$ be a CSRS over a signature $\mathcal{F}$ and $t,s \in \mathcal{T}(\mathcal{F}^\mu_C)$. We have $proper(t) \to^+_{\mathcal{R}^\mu_C} ok(s)$ if and only $t = s$ and $t \in \mathcal{T}(\mathcal{F})$.*

LEMMA 5. *[19, Lemma 3] Let $(\mathcal{R},\mu)$ be a CSRS over a signature $\mathcal{F}$ and let $t \in \mathcal{T}(\mathcal{F})$. We have $t \hookrightarrow s$ if and only if $active(t) \to^+_{\mathcal{R}^\mu_C} mark(s)$.*

Theorem 8 *Let $(\mathcal{R},\mu)$ be a CSRS. If $\mathcal{R}^\mu_C$ is simply terminating, then $(\mathcal{R},\mu)$ is simply terminating.*

PROOF. If $(\mathcal{R},\mu)$ is not simply terminating, then there exists an infinite sequence $A : t_1 \hookrightarrow t_2 \hookrightarrow \cdots \hookrightarrow t_n \hookrightarrow \cdots$ in $\mathcal{R} \cup \mathcal{E}mb^\mu(\mathcal{F})$, where $t_i \in \mathcal{T}(\mathcal{F})$ for $i \geq 1$. By Lemma 5, each step $t \hookrightarrow_{\mathcal{R}\cup\mathcal{E}mb^\mu(\mathcal{F})} s$ induces a non-empty reduction sequence $active(t) \to^+_{(\mathcal{R}\cup\mathcal{E}mb^\mu(\mathcal{F}))^\mu_C} mark(s)$. By Lemma 3 such a sequence makes use of rules from $\mathcal{R}^\mu_C$ and $\mathcal{S} = \{active(f(x_1,\ldots,x_k)) \to mark(x_i) \mid f \in \mathcal{F}, i \in \mu(f)\}$. By Lemma 4, the infinite sequence A induces an infinite sequence

$$\begin{aligned} B : \quad & top(active(t_1)) \to^+_{\mathcal{R}^\mu_C\cup\mathcal{S}} top(mark(t_2)) \\ & \to_{\mathcal{R}^\mu_C} top(proper(t_2)) \to_{\mathcal{R}^\mu_C} top(ok(t_2)) \\ & \to_{\mathcal{R}^\mu_C} top(active(t_2)) \to_{\mathcal{R}^\mu_C\cup\mathcal{S}} \cdots \end{aligned}$$

It easily follows from the proof of Lemma 5 (see [19]) that each rewrite sequence $active(t) \to^+_{(\mathcal{R}\cup\mathcal{E}mb^\mu(\mathcal{F}))^\mu_C} mark(s)$ associated to a given step $t \hookrightarrow_{\mathcal{R}\cup\mathcal{E}mb^\mu(\mathcal{F})} s$ contains exactly one application of a rule of type $active(l) \to mark(r)$ for some rule $l \to r$ in $\mathcal{R} \cup \mathcal{E}mb^\mu(\mathcal{F})$. Note that, if a rule $active(f(x_1,\ldots,x_k)) \to mark(x_i) \in \mathcal{S}$ applies, then we have $s = t[t|_{p.i}]_p$ and $root(t|_p) = f$ for some position $p \in \mathcal{P}os(t)$. Hence, we can write $top(active(t)) \to_{\mathcal{E}mb(\mathcal{F}^\mu_C)} top(active(s))$ by applying the rule $f(x_1,\ldots,x_k) \to x_i$ directly on $t|_p$. Therefore, the infinite sequence B can then be transformed into an infinite sequence B' that only uses rules in $\mathcal{R}^\mu_C \cup \mathcal{E}mb(\mathcal{F}^\mu_C)$. This means that $\mathcal{R}^\mu_C$ is not simply terminating, a contradiction. □