

Using Context-Sensitive Rewriting for Proving Innermost Termination of Rewriting

Beatriz Alarcón and Salvador Lucas^{1,2}

*Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Valencia, Spain*

Abstract

Computational systems based on reducing expressions usually have a predefined reduction strategy to break down the nondeterminism which is inherent to reduction relations. The innermost strategy corresponds to call by value or *eager* computation, that is, the computational mechanism of several programming languages like Maude, OBJ, etc. where the arguments of a function call are always *evaluated* before calling the function. This strategy usually fails to terminate when nonterminating computations are possible in the programs and many eager programming languages also admit the *explicit* specification of a particular class of *strategy annotations* to (try to) avoid them. Context-Sensitive Rewriting provides an abstract model to describe and analyze the operational behavior of such programs. This paper aims at contributing to the development of appropriate techniques and tools for the verification of *program termination* in the aforementioned programming languages, so we focus on termination of innermost (context-sensitive) rewriting. We adapt the notion of usable argument introduced by Fernández to prove innermost termination by proving termination of context-sensitive rewriting. Thanks to our recent developments for proving termination of (innermost) context-sensitive rewriting using dependency pairs, now we can also relax monotonicity requirements for proving innermost termination of (context-sensitive) rewriting. We have implemented these new improvements in the termination tool MU-TERM and evaluated the results with some benchmarks.

Keywords: Dependency pairs, innermost rewriting, context-sensitive rewriting program analysis, termination.

1 Introduction

Most computational systems whose operational principle is based on reducing expressions (e.g., functional, algebraic, and equational programming languages as well as theorem provers based on rewriting techniques) incorporate a predefined reduction strategy which is used to break down the nondeterminism which is inherent to reduction relations. Thus, every program will be executed according to that strategy. One of the most commonly used is the *innermost* strategy, in which only

¹ This work has been partially supported by the EU (FEDER) and the Spanish MICINN, under grant TIN2007-68093-C02-02 and HA 2006-2007, the Generalitat Valenciana under grant GVPRE/2008/113. Also it was partially supported by the Spanish MICINN under FPU grant AP2005-3399.

² Email: {balarcon,slucas}@dsic.upv.es

innermost redexes are reduced. Here, by an innermost redex we mean a redex containing no other redex. The innermost strategy corresponds to call by value or *eager* computation, that is, the computational mechanism of several programming languages where the arguments of a function are always evaluated before the application of the function which use them. It is well-known, however, that programs written in eager programming languages frequently run into a nonterminating behavior if the programs have not carefully been written to avoid such problems. For this reason, the designers of such eager programming languages have also developed some features and language constructs aimed at giving the user more flexible control of the program execution. For instance, *syntactic annotations* (which are associated to the arguments of the function symbols) have been used in eager programming languages such as Maude [7], OBJ2 [11], OBJ3 [16], and CafeOBJ [12] to introduce *replacement restrictions* which are able to (hopefully) avoid nontermination. Such languages admit the *explicit* specification of a particular class of *strategy annotations*, which (basically) are lists of integers associated to function symbols which specify the ordering in which the arguments are (eventually) evaluated in function calls. This very simple strategy language provides quite a powerful way to control the program execution. Due to its simplicity, such strategy annotations also provide a simple interface for understanding and eventually modifying the execution of programs. Context-sensitive rewriting (*CSR* [18,21]) provides an abstract model to describe and analyze the operational behavior of such programs, thus providing an appropriate basis for the development of program verification tools [8,19,20]. In *CSR*, a *replacement map* (i.e., a mapping $\mu : \mathcal{F} \rightarrow \mathcal{P}(\mathbb{N})$ satisfying $\mu(f) \subseteq \{1, \dots, k\}$, for each k -ary symbol f of a signature \mathcal{F}) associates a subset of its argument indices to each function symbol. We use a replacement map to indicate the argument positions on which rewriting steps are allowed. In this way, we can *achieve* a terminating behavior by pruning (all) infinite rewrite sequences. A Term Rewriting System (TRS) \mathcal{R} together with a replacement map μ is often called a CS-TRS (written (\mathcal{R}, μ)). The research in this paper aims at contributing to the development of appropriate techniques and tools for the verification of *program termination* in the aforementioned programming languages. Our focus is on termination of *innermost context-sensitive rewriting* (i.e., the variant of *CSR* where only the deepest μ -replacing redexes are contracted). Techniques for proving termination of innermost *CSR* were first investigated in [13,19]. These papers, though, only consider *transformational* techniques, where the original CS-TRS (\mathcal{R}, μ) is transformed into a TRS \mathcal{R}_Θ^μ (where Θ represents the transformation which has been used) whose *innermost* termination implies the innermost termination of *CSR* for (\mathcal{R}, μ) . In [5], we have extended the context-sensitive dependency pairs approach in [2,3] for proving termination of innermost *CSR*. Roughly speaking the (context-sensitive) dependency pairs associated to a (CS-)TRS are of a set of rewrite rules which are used together with the original ones to obtain an often easier proof of termination due to the possibility of applying a number of new auxiliary techniques, see, e.g., [15,17] for recent state-of-the-art accounts. As shown in [3], proofs of termination using context-sensitive dependency pairs (CS-DPs) are much more powerful and faster than any other technique for proving termination of *CSR*.

Dealing with innermost *CSR*, we have a similar situation, see [5].

The main topic we are going to develop in this work is the analysis and extension of Fernández’s work [9]. In her paper, she noticed that, when dealing with proofs of innermost termination, requiring monotonicity of the orderings w.r.t. all arguments of function symbols is not always necessary. According to this, she showed that innermost termination of rewriting can be rephrased as a context-sensitive rewriting termination problem. She introduced the notion of *usable arguments*, which can be thought of as the argument positions on which innermost reductions take place. Then, Fernández showed that innermost termination of a TRS \mathcal{R} can be proved by proving termination of the CS-TRS (\mathcal{R}, μ) which is obtained when $\mu(f)$ collects the *usable arguments* of f for each symbol f in the signature. We have implemented her techniques for the first time, and then we have investigated the practical use of her results. We have adapted Fernández ideas to deal with proofs of *innermost termination of CSR*; this was left as an open problem in [9].

After some preliminaries in Section 2, in Section 3 we summarize the last developments concerning context-sensitive rewriting and innermost context-sensitive rewriting using dependency pairs. In Section 4 we show how to adapt Fernández’s criterion to relax monotonicity requirements when proving innermost termination of *CSR*. In Section 5 we apply this criterion to be used with dependency pairs in proofs of termination of innermost (context-sensitive) rewriting. Section 6 provides an experimental evaluation of our techniques for proving innermost termination of (context-sensitive) rewriting automatically. Finally, we conclude and comment on some future work.

2 Preliminaries

Relations.

A (strict) partial ordering $>$ is an irreflexive and transitive relation. We say that $>$ is well-founded if there is no infinite decreasing sequence with $>$. A quasi-ordering \gtrsim is a transitive and reflexive relation.

Terms.

Throughout the paper, \mathcal{X} denotes a countable set of variables and \mathcal{F} denotes a signature, i.e., a set of function symbols $\{f, g, \dots\}$, each having a fixed arity given by a mapping $ar : \mathcal{F} \rightarrow \mathbb{N}$. The set of terms built from \mathcal{F} and \mathcal{X} is $\mathcal{T}(\mathcal{F}, \mathcal{X})$. Positions p, q, \dots are represented by chains of positive natural numbers used to address subterms of t . Given positions p, q , we denote their concatenation as $p.q$. Positions are ordered by the standard prefix ordering \leq . If p is a position, and Q is a set of positions, then $p.Q = \{p.q \mid q \in Q\}$. We denote the topmost position by Λ . The set of positions of a term t is $\mathcal{Pos}(t)$. Positions of nonvariable symbols in t are denoted as $\mathcal{Pos}_{\mathcal{F}}(t)$ while $\mathcal{Pos}_{\mathcal{X}}(t)$ 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 $root(t)$.

Term rewriting.

A rewrite rule is an ordered pair (l, r) , written $l \rightarrow r$, with $l, r \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, $l \notin \mathcal{X}$ and $\text{Var}(r) \subseteq \text{Var}(l)$. The left-hand side (*lhs*) of the rule is l and r is the right-hand side (*rhs*). A rule is collapsing if $r \in \mathcal{X}$. An instance σ of a left-hand side of a rewrite rule $l \rightarrow r$, written $\sigma(l)$, is a redex (**r**educible **e**xpression). A TRS is a pair $\mathcal{R} = (\mathcal{F}, R)$ where R is a set of rewrite rules. Given $\mathcal{R} = (\mathcal{F}, R)$, we consider \mathcal{F} as the disjoint union $\mathcal{F} = \mathcal{C} \uplus \mathcal{D}$ of symbols $c \in \mathcal{C}$, called *constructors* and symbols $f \in \mathcal{D}$, called *defined functions*, where $\mathcal{D} = \{\text{root}(l) \mid l \rightarrow r \in R\}$ and $\mathcal{C} = \mathcal{F} - \mathcal{D}$.

Innermost rewriting.

A term is a normal form if it contains no redex. A substitution σ is normalized if $\sigma(x)$ is a normal form for all $x \in \text{Dom}(\sigma)$. A term $f(t_1, \dots, t_k)$ is argument normalized if t_i is a normal form for all $1 \leq i \leq k$. An innermost redex is an argument normalized redex. A term s rewrites innermost to t , written $s \rightarrow_i t$, if $s \rightarrow t$ at position p and $s|_p$ is an innermost redex. Let \mathcal{R} be a TRS. For any symbol f let $\text{Rules}(\mathcal{R}, f)$ be the set of rules $l \rightarrow r$ defining f and such that the left-hand sides l are argument normalized. For any term t the set of usable rules $\mathbf{U}(\mathcal{R}, t)$ is as follows:

$$\mathbf{U}(\mathcal{R}, x) = \emptyset$$

$$\mathbf{U}(\mathcal{R}, f(t_1, \dots, t_n)) = \text{Rules}(\mathcal{R}, f) \cup \bigcup_{1 \leq i \leq \text{ar}(f)} \mathbf{U}(\mathcal{R}', t_i) \cup \bigcup_{l \rightarrow r \in \text{Rules}(\mathcal{R}, f)} \mathbf{U}(\mathcal{R}', r)$$

where $\mathcal{R}' = \mathcal{R} - \text{Rules}(\mathcal{R}, f)$.

Context-sensitive rewriting.

A mapping $\mu : \mathcal{F} \rightarrow \mathcal{P}(\mathbb{N})$ is a *replacement map* (or \mathcal{F} -map) if $\forall f \in \mathcal{F}$, $\mu(f) \subseteq \{1, \dots, \text{ar}(f)\}$ [18]. Let $M_{\mathcal{F}}$ be the set of all \mathcal{F} -maps (or $M_{\mathcal{R}}$ for the \mathcal{F} -maps of a TRS (\mathcal{F}, R)). A binary relation R on terms is μ -monotonic if $t R s$ implies $f(t_1, \dots, t_{i-1}, t, \dots, t_k) R f(t_1, \dots, t_{i-1}, s, \dots, t_k)$ for all $f \in \mathcal{F}$, $i \in \mu(f)$ and $t, s, t_1, \dots, t_k \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. The set of μ -replacing positions $\text{Pos}^\mu(t)$ of $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ is: $\text{Pos}^\mu(t) = \{\Lambda\}$, if $t \in \mathcal{X}$ and $\text{Pos}^\mu(t) = \{\Lambda\} \cup \bigcup_{i \in \mu(\text{root}(t))} i.\text{Pos}^\mu(t|_i)$, if $t \notin \mathcal{X}$. The set of μ -replacing variables of t is $\text{Var}^\mu(t) = \{x \in \text{Var}(t) \mid \exists p \in \text{Pos}^\mu(t), t|_p = x\}$. A rule $l \rightarrow r$ is μ -conservative if $\text{Var}^\mu(r) \subseteq \text{Var}^\mu(l)$. The μ -replacing subterm relation \sqsupseteq_μ is given by $t \sqsupseteq_\mu s$ if there is $p \in \text{Pos}^\mu(t)$ such that $s = t|_p$. We write $t \triangleright_\mu s$ if $t \sqsupseteq_\mu s$ and $t \neq s$ and say that s is a strict μ -replacing subterm of t . We write $t \triangleright_\mu s$ to denote that s is a non- μ -replacing (hence strict) subterm of t : $t \triangleright_\mu s$ if there is $p \in \text{Pos}(t) - \text{Pos}^\mu(t)$ such that $s = t|_p$. In *context-sensitive rewriting* (*CSR* [18]), we (only) contract μ -replacing redexes: s μ -rewrites to t , written $s \hookrightarrow_\mu t$ (or $s \hookrightarrow_{\mathcal{R}, \mu} t$ and even $s \hookrightarrow t$, if \mathcal{R} and μ are clear from the context), if $s \xrightarrow{p}_{\mathcal{R}} t$ and $p \in \text{Pos}^\mu(s)$. A TRS \mathcal{R} is μ -terminating if \hookrightarrow_μ is terminating. Termination of *CSR* is fully captured by the so-called μ -reduction orderings, i.e., well-founded, stable orderings \sqsubset which are μ -monotonic. A term t is μ -terminating if there is no infinite μ -rewrite sequence $t = t_1 \hookrightarrow_\mu t_2 \hookrightarrow_\mu \dots \hookrightarrow_\mu t_n \hookrightarrow_\mu \dots$ starting from t . A term t μ -narrows to a term s (written $t \rightsquigarrow_{\mathcal{R}, \mu, \theta} s$), if there is a nonvariable μ -replacing position $p \in \text{Pos}_{\mathcal{F}}^\mu(t)$ and a rule $l \rightarrow r$ in \mathcal{R} (sharing no variable with t) such that $t|_p$ and l unify with most general unifier θ and $s = \theta(t[r]_p)$. Then, we say that

t is μ -narrowable. A μ -normal form is a term which cannot be μ -rewritten. Let $\text{NF}_\mu(\mathcal{R})$ (or just NF_μ if no confusion arises) be the set of μ -normal forms of a TRS \mathcal{R} . A substitution σ is μ -normalized if $\sigma(x)$ is a μ -normal form for all $x \in \text{Dom}(\sigma)$. A term $t = f(t_1, \dots, t_k)$ is *argument μ -normalized* if t_i is a μ -normal form for all $i \in \mu(f)$. A μ -innermost redex is an argument μ -normalized redex.

A term s innermost μ -rewrites to t , written $s \hookrightarrow_i t$, if $s \xrightarrow{\mathcal{R}} t$, $p \in \text{Pos}^\mu(s)$, and $s|_p$ is a μ -innermost redex. A TRS \mathcal{R} is innermost μ -terminating if $\hookrightarrow_{\mu,i}$ is terminating. We write $s \hookrightarrow_{\mathcal{R},\mu,i}^! t$ if $s \xrightarrow{*}_{\mathcal{R},\mu,i} t$ and $t \in \text{NF}_\mu$. A pair (\mathcal{R}, μ) where \mathcal{R} is a TRS and $\mu \in M_{\mathcal{R}}$ is often called a CS-TRS.

Dependency pairs.

Given a TRS $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ a new TRS $\text{DP}(\mathcal{R}) = (\mathcal{F}^\sharp, D(R))$ of *dependency pairs* for \mathcal{R} is given as follows: if $f(t_1, \dots, t_m) \rightarrow r \in R$ and $r = C[g(s_1, \dots, s_n)]$ for some defined symbol $g \in \mathcal{D}$ and $s_1, \dots, s_n \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, then $f^\sharp(t_1, \dots, t_m) \rightarrow g^\sharp(s_1, \dots, s_n) \in D(R)$, where f^\sharp and g^\sharp are new fresh symbols (called *tuple symbols*) associated to defined symbols f and g respectively [6]. Let \mathcal{D}^\sharp be the set of tuple symbols associated to symbols in \mathcal{D} and $\mathcal{F}^\sharp = \mathcal{F} \cup \mathcal{D}^\sharp$. As usual, for $t = f(t_1, \dots, t_k) \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, we write t^\sharp to denote the *marked* term $f^\sharp(t_1, \dots, t_k)$. Given $T \subseteq \mathcal{T}(\mathcal{F}, \mathcal{X})$, T^\sharp denotes $\\{t^\sharp \mid t \in T\}$. For the sake of readability, capital letters denote marked symbols in examples.

Reduction pairs.

Given a signature \mathcal{F} , a reduction pair (\succeq, \sqsubset) for terms in $\mathcal{T}(\mathcal{F}, \mathcal{X})$ consists of a stable and monotonic quasi-ordering \succeq on terms, and a stable and well-founded ordering \sqsubset satisfying either $\succeq \circ \sqsubset \subseteq \sqsubset$ or $\sqsubset \circ \succeq \subseteq \sqsubset$. Note that *monotonicity is not required* for \sqsubset . A μ -reduction pair is a reduction pair (\succeq, \sqsubset) where the quasi-ordering \succeq is μ -monotonic (instead of monotonic).

3 Context-sensitive dependency pairs

In the following, we write $\text{NARR}^\mu(t)$ to indicate that t is μ -narrowable (w.r.t. the intended TRS \mathcal{R}). We consider a function REN^μ which *independently* renames all *occurrences* of μ -replacing variables within a term t by using new fresh variables which are not in $\text{Var}(t)$:

- $\text{REN}^\mu(x) = y$ if x is a variable, where y is intended to be a fresh new variable which has not yet been used (we could think of y as the ‘next’ variable in an infinite list of variables); and
- $\text{REN}^\mu(f(t_1, \dots, t_k)) = f([t_1]_1^f, \dots, [t_k]_k^f)$ for every k -ary symbol f , where given a term $s \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, $[s]_i^f = \text{REN}^\mu(s)$ if $i \in \mu(f)$ and $[s]_i^f = s$ if $i \notin \mu(f)$.

Let $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ be a TRS and $\mu \in M_{\mathcal{F}}$. We define $\text{iDP}(\mathcal{R}, \mu) = \text{iDP}_{\mathcal{F}}(\mathcal{R}, \mu) \cup \text{iDP}_{\mathcal{X}}(\mathcal{R}, \mu)$ to be the set of *innermost context-sensitive* dependency pairs (ICS-DPs) where:

$$\begin{aligned} \text{iDP}_{\mathcal{F}}(\mathcal{R}, \mu) &= \{l^\sharp \rightarrow s^\sharp \mid l \rightarrow r \in R, l^\sharp \in \text{NF}_\mu(\mathcal{R}), r \succeq_\mu s, \text{root}(s) \in \mathcal{D}, l \not\succeq_\mu s, \text{NARR}^\mu(\text{REN}^\mu(s))\} \\ \text{iDP}_{\mathcal{X}}(\mathcal{R}, \mu) &= \{l^\sharp \rightarrow x \mid l \rightarrow r \in R, l^\sharp \in \text{NF}_\mu(\mathcal{R}), x \in \text{Var}^\mu(r) - \text{Var}^\mu(l)\} \end{aligned}$$

We extend $\mu \in M_{\mathcal{F}}$ into $\mu^\sharp \in M_{\mathcal{F} \cup \mathcal{D}^\sharp}$ by $\mu^\sharp(f) = \mu(f)$ if $f \in \mathcal{F}$, and $\mu^\sharp(f^\sharp) = \mu(f)$ if $f \in \mathcal{D}$. Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $\mu \in M_{\mathcal{F}}$. We say that $t \in \mathcal{T}(\mathcal{F}, \mathcal{X}) - \mathcal{X}$ is a *hidden term* if there is a rule $l \rightarrow r \in R$ such that $r \triangleright_\mu t$. Let $\mathcal{HT}(\mathcal{R}, \mu)$ (or just \mathcal{HT} , if \mathcal{R} and μ are clear for the context) be the set of all hidden terms in (\mathcal{R}, μ) . We use $\mathcal{DHT} = \{t \in \mathcal{HT} \mid \text{root}(t) \in \mathcal{D}\}$ for the set of hidden terms which are rooted by a *defined* symbol. We also let $\mathcal{NHT}(\mathcal{R}, \mu) = \{t \in \mathcal{DHT} \mid \text{NARR}^\mu(\text{REN}^\mu(t))\}$ be the set of *hidden terms* which are rooted by a *defined* symbol, and that, after applying REN^μ , become μ -*narrowable* (see [4] for further motivation and explanations about these definitions).

Let $\mathcal{R} = (\mathcal{F}, R)$ and $\mathcal{P} = (\mathcal{G}, P)$ be TRSs and $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$. An innermost $(\mathcal{P}, \mathcal{R}, \mu)$ -chain is a finite or infinite sequence of pairs $u_i \rightarrow v_i \in \mathcal{P}$, together with a substitution σ satisfying that, for all $i \geq 1$, $\sigma(u_i) \in \text{NF}_\mu(\mathcal{R})$ and :

- (i) if $v_i \notin \text{Var}(u_i) - \text{Var}^\mu(u_i)$, then $\sigma(v_i) \hookrightarrow_{\mathcal{R}, \mu, i}^! \sigma(u_{i+1})$, and
- (ii) if $v_i \in \text{Var}(u_i) - \text{Var}^\mu(u_i)$, then there is $s_i \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ such that $\sigma(v_i) \triangleright_\mu s_i$ and $s_i^\sharp \hookrightarrow_{\mathcal{R}, \mu, i}^! \sigma(u_{i+1})$.

As usual, we assume that different occurrences of dependency pairs do not share any variable (renaming substitutions are used if necessary). An innermost $(\mathcal{P}, \mathcal{R}, \mu)$ -chain is called *minimal* if for all $i \geq 1$,

- (i) if $v_i \notin \text{Var}(u_i) - \text{Var}^\mu(u_i)$, then $\sigma(v_i)$ is innermost (\mathcal{R}, μ) -terminating, and
- (ii) if $v_i \in \text{Var}(u_i) - \text{Var}^\mu(u_i)$, then s_i^\sharp is innermost (\mathcal{R}, μ) -terminating and $\exists \bar{s}_i \in \mathcal{NHT}(\mathcal{R}, \mu)$ such that $s_i = \sigma(\bar{s}_i)$.

This more abstract notion of chain can be particularized to be used with ICS-DPs, by just taking $\mathcal{P} = \text{iDP}(\mathcal{R}, \mu)$. In the following, the pairs in a CS-TRS (\mathcal{P}, μ) , where $\mathcal{P} = (\mathcal{G}, P)$, are partitioned according to its role in the previous Definition as follows:

$$P_{\mathcal{X}} = \{u \rightarrow v \in P \mid v \in \text{Var}(u) - \text{Var}^\mu(u)\} \text{ and } P_{\mathcal{G}} = P - P_{\mathcal{X}}$$

Innermost Context-Sensitive Dependency Graph.

Let \mathcal{R} be a TRS and $\mu \in M_{\mathcal{R}}$. The innermost context-sensitive dependency graph consists of the set $\text{iDP}(\mathcal{R}, \mu)$ of innermost context-sensitive dependency pairs and arcs which connect them as follows:

- (i) There is an arc from $u \rightarrow v \in \mathcal{P}_{\mathcal{G}}$ to $u' \rightarrow v' \in \mathcal{P}$ if there are substitutions θ and θ' such that $\theta(v) \hookrightarrow_{\mathcal{R}, \mu, i}^! \theta'(u')$ and $\theta(u), \theta'(u') \in \text{NF}_\mu(\mathcal{R})$.
- (ii) There is an arc from $u \rightarrow v \in \mathcal{P}_{\mathcal{X}}$ to $u' \rightarrow v' \in \mathcal{P}$ if there is $t \in \mathcal{NHT}(\mathcal{R}, \mu)$ and substitutions θ and θ' such that $\theta(t^\sharp) \hookrightarrow_{\mathcal{R}, \mu, i}^! \theta'(u')$ and $\theta'(u') \in \text{NF}_\mu(\mathcal{R})$.

In order to approximate the ICS-DG, we have also adapted functions REN and CAP (used in standard rewriting) to the innermost context-sensitive setting (see [4,5] for details).

Basic usable CS-rules.

Let \mathcal{R} be a TRS and $\mu \in M_{\mathcal{R}}$. For any symbol f , let $Rules(\mathcal{R}, \mu, f)$ be the set of rules $l \rightarrow r$ defining f and such that the left-hand sides l are argument μ -normalized: $Rules(\mathcal{R}, \mu, f) = \{f(l_1, \dots, l_k) \rightarrow r \in \mathcal{R} \mid \forall i \in \mu(f), l_i \in NF_{\mu}(\mathcal{R})\}$. For any term t , the set of *basic* usable CS-rules $U_0(\mathcal{R}, \mu, t)$ is as follows:

$$\begin{aligned} U_0(\mathcal{R}, \mu, x) &= \emptyset \\ U_0(\mathcal{R}, \mu, f(t_1, \dots, t_n)) &= Rules(\mathcal{R}, \mu, f) \cup \bigcup_{i \in \mu(f)} U_0(\mathcal{R}', \mu, t_i) \cup \bigcup_{l \rightarrow r \in Rules(\mathcal{R}, \mu, f)} U_0(\mathcal{R}', \mu, r) \end{aligned}$$

where $\mathcal{R}' = \mathcal{R} - Rules(\mathcal{R}, \mu, f)$.

If \mathcal{P} is a TRS, then $U_0(\mathcal{R}, \mu, \mathcal{P}) = \bigcup_{l \rightarrow r \in \mathcal{P}} U_0(\mathcal{R}, \mu, r)$.

4 Simplifying monotonicity requirements for innermost μ -termination

In the innermost setting, matching substitutions are always normalized. For this reason, in an innermost sequence $t_1 \xrightarrow{p_1}_i t_2 \xrightarrow{p_2}_i \dots \xrightarrow{p_n}_i t_{n+1}$ starting at root position (i.e., $p_1 = \Lambda$), every redex $t_j|_{p_j}$ for $j > 1$ comes from a defined symbol introduced after applying a rule $l_k \rightarrow r_k$ in a previous step $k < j$. Hence the set of arguments which are reduced can be handled by looking for defined symbols in right-hand sides of the involved rules $l \rightarrow r$.

In [6], Arts and Giesl already noticed that in the treatment of innermost chains, monotonicity requirements for the reduction pairs can be weaker. In [9] Fernández defines the notion of *usable arguments* for a function symbol when proving innermost termination. The idea is that, in innermost sequences, some arguments are not relevant for proving termination.

Example 4.1 Consider the following TRS \mathcal{R} :

$$\begin{aligned} f(s(0), s(0)) &\rightarrow f(x, g(x)) & g(s(x)) &\rightarrow g(x) \end{aligned}$$

No innermost sequence starting at root position takes into account the first argument of f nor the argument of g . The reason is that (instances of) innermost redexes are argument normalized. That means that all variables (e.g. x) introduced by the applied rule are normalized and cannot be reduced. Only the second argument $g(x)$ of f in the right-hand side of the first rule could be innermost reduced after applying it.

Roughly speaking, the usable arguments of a symbol f with respect to a TRS \mathcal{R} are those arguments with a subterm rooted by a defined symbol in some right-hand side of a dependency pair or usable rule.

Definition 4.2 [Usable arguments] [9, Definition 3] Let $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ be a TRS and \mathcal{P} a set of pairs of terms s.t. for all $u \rightarrow v \in \mathcal{P}$, u is argument normalized with respect to \mathcal{R} . The set of usable arguments for a function symbol $f \in \mathcal{F}$ with respect to \mathcal{R} and \mathcal{P} is defined as $\mathbf{UA}(f, \mathcal{R}, \mathcal{P}) = \{1 \leq k \leq ar(f) \mid \exists u \rightarrow v \in \mathcal{P} \cup \mathbf{U}(\mathcal{R}, \mathcal{P}), \exists p, p' \in \text{Pos}(v) \text{ s.t. } \text{root}(v|_{p'}) = f, \text{root}(v|_p) \in \mathcal{D}, p'.k \leq p, u \not\vdash v|_p\}$.

Considering those usable arguments could be helpful in proofs of innermost termination since they impose weaker monotonicity requirements.

As Fernández noticed, the set of usable arguments can be seen as a replacement map which specifies the arguments to be reduced. According to her results, the μ -termination of a TRS \mathcal{R} implies the innermost termination of \mathcal{R} if $\mu(f) = \mathbf{UA}(f, \mathcal{R}, R)$ for all $f \in \mathcal{F}$ where R only contains rules such that all left-hand sides are argument normalized.

Corollary 4.3 [9, Corollary 11] Let \mathcal{R} be a TRS and $\mu(f) = \mathbf{UA}(f, \mathcal{R}, R')$ for every $f \in \mathcal{F}$ where $R' \subseteq R$ contains all rules $l \rightarrow r \in R$ such that l is argument normalized. If \mathcal{R} is μ -terminating, then \mathcal{R} is innermost terminating.

This observation is very useful since now, all techniques for proving termination of *CSR* can be used for proving innermost termination. Several methods and techniques for proving termination of *CSR* have been developed so far [2,3,14,23].

4.1 Usable arguments for *CSR*

Following Fernández's ideas, in the innermost context-sensitive setting (for a given replacement map μ) we could relax monotonicity requirements by taking into account that reductions only take place on μ -replacing positions of the right-hand sides of the rules which are rooted by a defined symbol. We adapt Fernández's ideas to *CSR*. In sharp contrast to the unrestricted case, we need to take into account that in innermost *CSR* a redex does not need to be argument normalized. Only argument μ -normalization can be assumed. Thus, non- μ -replacing subterms may contain redexes that can be reduced later on if they come to a replacing position.

Proposition 4.4 A CS-TRS (\mathcal{R}, μ) is innermost μ -terminating iff \mathcal{R}' is innermost μ -terminating, where $\mathcal{R}' \subseteq \mathcal{R}$ contains all rules $l \rightarrow r \in \mathcal{R}$ such that l is argument μ -normalized.

Proof. Trivial since the only rules that can be applied in innermost μ -reductions are those whose left-hand sides are argument μ -normalized. \square

In the following, we assume that all rules in any CS-TRS (\mathcal{R}, μ) are argument μ -normalized, i.e., for all rules $l \rightarrow r$ in \mathcal{R} , l is argument μ -normalized. Proposition 4.4 ensures that this entails no lack of generality regarding our research on innermost termination of *CSR*. The straightforward adaptation of Fernandez's criterion to *CSR* yields the following definition.

Definition 4.5 [Basic usable CS-arguments] Let $(\mathcal{R}, \mu) = ((\mathcal{C} \uplus \mathcal{D}, R), \mu)$ be a CS-TRS and \mathcal{P} be a set of pairs of terms s.t. for all $u \rightarrow v \in \mathcal{P}$, u is argument μ -normalized. The *basic usable CS-arguments* for a function symbol $f \in \mathcal{F}$ (w.r.t. \mathcal{R} and \mathcal{P}) are defined as $\mathbf{UA}_\mu(f, \mathcal{R}, \mathcal{P}) = \{i \in \mu(f) \mid \exists u \rightarrow v \in \mathcal{P} \cup \mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P}), \exists p, p' \in \text{Pos}^\mu(v) \text{ s.t. } \text{root}(v|_{p'}) = f, \text{root}(v|_p) \in \mathcal{D}, p'.i \leq p, u \not\vdash_\mu v|_p\}$.

Note that the replacement map given by $\mu'(f) = \mathbf{UA}_\mu(f, \mathcal{R}, \mathcal{P})$ for all $f \in \mathcal{F}$ is more restrictive than μ , i.e., for all symbols $f \in \mathcal{F}$, $\mu'(f) \subseteq \mu(f)$.

The following proposition is the context-sensitive version of [9, Lemma 5].

Proposition 4.6 Let (\mathcal{R}, μ) be a CS-TRS and \mathcal{P} be a set of pairs of terms s.t. for all $u \rightarrow v \in \mathcal{P}$, u is argument μ -normalized and $\mathcal{P} \cup \mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P})$ is μ -conservative. Let innermost μ -rewriting below the root be $\xrightarrow{>\Lambda}_i = (\xrightarrow{>\Lambda} \cap \hookrightarrow_i)$. Let $l \rightarrow r \in \mathcal{P} \cup \mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P})$ be such that $\sigma(r) \xrightarrow{>\Lambda}_i^* \mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P}) t$ for some term t and substitution σ s.t. $\sigma(l)$ is argument μ -normalized. If $t|_p$ is an innermost μ -redex, then for all $p'.k \leq p$, we have that $k \in \mathbf{UA}_\mu(\text{root}(t|_{p'}), \mathcal{R}, \mathcal{P})$.

Proof. By induction on the length n of the rewriting sequence. If $n = 0$, then $\sigma(r) = t$. Then, since $\sigma(l)$ is argument μ -normalized, it follows that for all $x \in \text{Var}^\mu(l)$, $\sigma(x) \in \mathbf{NF}_\mu(\mathcal{R})$. Since the rule $l \rightarrow r$ is μ -conservative (that is $\text{Var}^\mu(r) \subseteq \text{Var}^\mu(l)$), we have that for all $x \in \text{Var}^\mu(r)$, $\sigma(x) \in \mathbf{NF}_\mu(\mathcal{R})$. It follows that p is a nonvariable (μ -replacing) position of r , i.e. $p \in \text{Pos}_{\mathcal{F}}^\mu(r)$. Thus, $\text{root}(r|_p) \in \mathcal{D}$ and the result follows by Definition 4.5.

If $n > 0$, then there is a term s such that $\sigma(r) \xrightarrow{>\Lambda}_i^* s$ and $s \xrightarrow{>\Lambda}_i t$ at some μ -replacing position q . By the induction hypothesis, every μ -replacing position of the term t above, which equal or disjoint to q satisfies the result and we only have to prove it for innermost redexes $t|_p$ s.t. $q < p$, it is say, we have to prove that $k \in \mathbf{UA}_\mu(\text{root}(t|_{p'}), \mathcal{R}, \mathcal{P})$, for all $q < p'.k \leq p$. If $s \xrightarrow{>\Lambda}_i t$, then $s|_q = \sigma'(l')$ and $t|_q = \sigma'(r')$, for some rule $l' \rightarrow r' \in \mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P})$ and substitution σ' s.t. $\sigma'(l')$ is argument μ -normalized. This implies that every innermost redex of $t|_q$ occurs at a position $p'' \in \text{Pos}^\mu(r')$ s.t. $\text{root}(r'|_{p''}) \in \mathcal{D}$ (since the rule $l' \rightarrow r'$ is conservative we have that for all $x \in \text{Var}^\mu(r')$, $\sigma(x) \in \mathbf{NF}_\mu(\mathcal{R})$) and $l' \not\vdash_\mu r'|_{p''}$ (otherwise, $\sigma'(l')$ would not be an innermost redex of s). By definition, when $p'' > \Lambda$, $p'.k \leq p''$, $k \in \mathbf{UA}_\mu(\text{root}(t|_{q.p'}), \mathcal{R}, \mathcal{P})$ which is equivalent to what we needed to prove ($k \in \mathbf{UA}_\mu(\text{root}(t|_{p'}), \mathcal{R}, \mathcal{P})$, for all $q < p'.k \leq p$). \square

Corollary 4.3 suggests that innermost μ -termination of a TRS $\mathcal{R} = (\mathcal{F}, R)$ could be proved as μ' -termination for μ' given by $\mu'(f) = \mathbf{UA}_\mu(f, \mathcal{R}, R)$ for all $f \in \mathcal{F}$. This is true for μ -conservative CS-TRSs, as the following theorem shows.

Theorem 4.7 A μ -conservative CS-TRS (\mathcal{R}, μ) is innermost μ -terminating if \mathcal{R} is μ' -terminating, where for all symbols $f \in \mathcal{F}$, $\mu'(f) = \mathbf{UA}_\mu(f, \mathcal{R}, R)$.

Proof. By contradiction. Assume that \mathcal{R} is not innermost μ -terminating. By the argument of size minimality, there is a infinite innermost μ -rewrite sequence with the first step at position Λ : $s_1 \hookrightarrow_i s_2 \hookrightarrow_i s_3 \hookrightarrow_i \dots$ (without loss of generality).

By Proposition 4.6 (where we let $\mathcal{P} = \mathcal{R}$), every step $s_j \xrightarrow{i}^{>\Lambda} s_{j+1}$ at position p satisfies that $p'.k \leq p$, $k \in \mathbf{UA}_\mu(\text{root}(s_j|_{p'}), \mathcal{R}, \mathcal{P})$. Therefore, there is an infinite μ' -rewrite sequence of terms $s_1 \hookrightarrow_{\mu'} s_2 \hookrightarrow_{\mu'} \dots \hookrightarrow_{\mu'} s_n \hookrightarrow_{\mu'} \dots$ which contradicts the μ' -termination of \mathcal{R} . □

Example 4.8 Consider the TRS \mathcal{R} :

$$\mathbf{f}(\mathbf{a}, \mathbf{b}, x) \rightarrow \mathbf{f}(x, x, x)$$

$$\mathbf{c} \rightarrow \mathbf{a}$$

$$\mathbf{c} \rightarrow \mathbf{b}$$

together with $\mu(\mathbf{f}) = \{1, 3\}$. Note that \mathcal{R} is μ -conservative. The set of ICS-DPs consists of the pair $\mathbf{F}(\mathbf{a}, \mathbf{b}, x) \rightarrow \mathbf{F}(x, x, x)$. By using $\mu'(f) = \mathbf{UA}_\mu(f, \mathcal{R}, \mathcal{P})$ for every $f \in \mathcal{F}$ we obtain $\mu'(\mathbf{f}) = \emptyset$. The CS-TRS (\mathcal{R}, μ') has no ICS-DP now. Thus we easily conclude the μ' -termination of \mathcal{R} and, by Theorem 4.7, the innermost μ -termination of \mathcal{R} .

This fact is important since now, all techniques for proving termination of *CSR* can be used to prove termination of innermost *CSR* for μ -conservative systems. The following example shows that μ -conservativeness cannot be dropped in Theorem 4.7.

Example 4.9 Consider again the TRS \mathcal{R} in Example 4.8 but now together with $\mu(\mathbf{f}) = \{1, 2\}$. If we try to apply Theorem 4.7 to prove innermost μ -termination of \mathcal{R} , we obtain $\mu'(\mathbf{f}) = \emptyset$ and (as discussed in Example 4.8) we would conclude the innermost μ -termination of \mathcal{R} . However, \mathcal{R} is *not* innermost μ -terminating:

$$\underline{\mathbf{f}(\mathbf{a}, \mathbf{b}, \mathbf{c})} \hookrightarrow_i \mathbf{f}(\underline{\mathbf{c}}, \mathbf{c}, \mathbf{c}) \hookrightarrow_i \mathbf{f}(\mathbf{a}, \underline{\mathbf{c}}, \mathbf{c}) \hookrightarrow_i \underline{\mathbf{f}(\mathbf{a}, \mathbf{b}, \mathbf{c})} \hookrightarrow_i \dots$$

Note that the first rule of \mathcal{R} is not μ -conservative now.

5 Relaxing monotonicity with CS-DPs

Fernández's criterion was also adapted to deal with proofs of termination of rewriting using dependency pairs. We have recently investigated how to prove innermost termination of *CSR* by using (context-sensitive) dependency pairs [5]. Now, we can adapt the use of CS-usable arguments to be applied in proofs of innermost μ -termination with CS-dependency pairs. To give a further step, we do it directly by considering the cycles of the ICS-DG.

Theorem 5.1 *Let $\mathcal{R} = (\mathcal{F}, R)$ and $\mathcal{P} = (\mathcal{G}, P)$ be TRSs, $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$ and \mathcal{P} is μ -conservative be such that $\mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P})$ is μ -conservative, and $\mu'(f) = \mathbf{UA}_\mu(f, \mathcal{R}, \mathcal{P})$ for all $f \in \mathcal{F}$. If there is a μ' -reduction pair (\succsim, \sqsubset) such that $\mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P}) \subseteq \succsim$, $\mathcal{P} \subseteq \succsim \cup \sqsubset$, and $\mathcal{P} \cap \sqsubset \neq \emptyset$, then there is no minimal innermost $(\mathcal{R}, \mathcal{P}, \mu)$ -chain.*

Proof. We proceed again by contradiction. Assume that there is a minimal innermost $(\mathcal{R}, \mathcal{P}, \mu)$ -chain:

$$\sigma(u_1) \hookrightarrow_{\mathcal{P}, \mu} \sigma(v_1) \xrightarrow{i}^{>\Lambda} (\mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P}), \mu) \sigma(u_2) \hookrightarrow_{\mathcal{P}, \mu} \sigma(v_2) \xrightarrow{i}^{>\Lambda} (\mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P}), \mu) \sigma(u_3) \hookrightarrow \dots$$

where all pairs are infinitely often used, and, for all $j \geq 1$, all $u_j \rightarrow v_j \in \mathcal{P}$ are μ -conservative and there is a substitution σ such that $\sigma(u_j)$ is argument μ -normalized and $\sigma(v_j)$ is innermost (\mathcal{R}, μ) -terminating. By Proposition 4.6, every innermost step in the sequence $\sigma(v_j) \xrightarrow{i}^{>\Lambda} \sigma(u_{j+1})$ is performed at a μ' -replacing position by means of a rule in $\mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P})$. Since by assumption $\mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P}) \subseteq \gtrsim$, and \gtrsim is stable and μ' -monotonic, $\sigma(v_j) \gtrsim \sigma(u_{j+1})$ holds for all $j > 0$. On the other hand, since $\mathcal{P} \subseteq \gtrsim \cup \sqsubset$ and $\mathcal{P} \cap \sqsubset \neq \emptyset$, the sequence $\sigma(u_1) \gtrsim \cup \sqsubset \sigma(v_1) \gtrsim \sigma(u_2) \dots$ contains an infinite number of \sqsubset steps. By compatibility of \gtrsim and \sqsubset , this contradicts the well-foundedness of \sqsubset . \square

Theorem 4.7 can be generalized to (certain) non- μ -conservative CS-TRSs thanks to Theorem 5.1 and the results for proving innermost termination of *CSR* in [5]. Now, for a given CS-TRS (\mathcal{R}, μ) that satisfies the conditions of Theorem 5.1, we can prove its innermost μ -termination by relaxing μ -monotonicity requirements for each cycle.

6 Implementation and Experiments

We have implemented the techniques described in the previous sections as part of the termination tool MU-TERM [1,22]. In order to evaluate the techniques which are reported in this paper we have made some benchmarks. We have considered the examples in the 2007 Termination Problem Data Base (TPDB, version 4.0) available through the URL:

<http://www.lri.fr/~marche/tpdb>

We are going to comment on the results obtained with each improvement.

6.1 Proving innermost termination of rewriting as termination of *CSR*

We have implemented the use of Corollary 4.3 for proving innermost termination of rewriting as termination of *CSR* (this was one of the main results in Fernández's paper). The relevance of this result in practice had not been tested yet. In order to evaluate it we have considered the examples used in the *innermost category* of the 2006 termination Competition³, which are part of the TPDB (in 2007 the category was not run). There are 69 examples, 66 of them are known to be innermost terminating. With Fernández's criterion (Corollary 4.3) MU-TERM succeeds in 32 examples (success rate of 48.5%). This is acceptable if we think that (except for AProVE, which succeeds in the 100% of the examples) the success rate for all

³ <http://www.lri.fr/~marche/termination-competition/2006>

	MU-TERM Fernández	MU-TERM iDPs [6]
YES score	32	39
YES average time	0.03 sec.	0.03 sec.

Table 1
Summary of benchmarks for innermost termination of rewriting

	MU-TERM Fernández cycle-based	MU-TERM Fernández cycle-based (only)
YES score	38	37
YES average time	0.04 sec.	0.79 sec.

Table 2
Summary of benchmarks for innermost termination of rewriting (based on cycles)

other participants in this category is around 20%. However, we have also implemented the use of (standard) dependency pairs for proving innermost termination (according to [6, Theorem 37]) together with the narrowing refinement (we call this tool MU-TERM iDPs) and we are able to prove 39 examples, including all examples solved with Fernández’s criterion. Moreover, we have included Fernández’s criterion as a technique to be applied when trying to solve a cycle in the innermost termination proof (see [9], Theorem 9). There are two new approximations: when MU-TERM has to solve a cycle, the first version uses Fernández’s criterion and if it fails then it tries to solve it in the usual way, that is, without any replacement map (MU-TERM Fernández cycle-based). The second one tries to *force* MU-TERM to solve the cycle with Fernández’s criterion, there is no other option (MU-TERM Fernández cycle-based only). In both cases the results obtained are similar. With the previous implementation of MU-TERM (MU-TERM iDPs) we solve 39 examples and with these two configurations we obtain better results than with Corollary 4.3; however, they do not improve the performance of MU-TERM iDPs. The results are summarized in Tables 1 and 2.

Therefore, it seems that using Corollary 4.3 to prove innermost termination of rewriting is not as good idea (at least with the considered set of examples) since we lose some examples due to a too restrictive new replacement map, and the average time is the same. Regarding the application of these ideas to cycles, we obtain better results but no essential improvement since we also lose some examples.

Full details for the benchmarks summarized in Table 1 can be found here:

<http://www.dsic.upv.es/~balarcon/prole08/benchmarks/FerInnermost.htm>

In the following URL:

<http://www.dsic.upv.es/~balarcon/prole08/Innermost/benchmarks.html>

more information can be found regarding the benchmarks summarized in Table 2. All this shows that we do not obtain any real improvement over the basic technique of dependency pairs for proving innermost termination at least for the set of considered examples.

6.2 Proving innermost termination of CSR

Although there is no special TPDB category for innermost termination of *CSR*, we have used the TRSs in the *CSR* termination subcategory to test our techniques for proving termination of innermost *CSR* (Theorems 4.7 and 5.1). It contains 90 CS-TRSs.

Since Theorem 4.7 only applies to conservative systems, we restrict the attention to the 27 conservative examples. We solve all of them with an average time of 0.025 seconds (MU-TERM Fernández). Further details can be found here:

<http://www.dsic.upv.es/~balarcon/prole08/benchmarks/FerICSR.html>

On the other hand, we have also implemented the use of Theorem 5.1 to deal with nonconservative systems. We have compared the same configurations explained above: the first one (MU-TERM Fernández cycle-based) tries to solve each μ -conservative cycle (with associated μ -conservative usable rules) by using CS-usable arguments as the new replacement map. If it fails, then the normal configuration of MU-TERM (MU-TERM iCSDPs) is used. The second one only applies CS-usable arguments on cycles when searching for a compatible μ -reduction pair (MU-TERM Fernández cycle-based only). All these versions of MU-TERM succeed over the same 70 examples, the same number of examples that we had already solved using the innermost version of the context-sensitive dependency pairs [5]. The time average rates has no exhibit substantial differences. Further details can be found here:

<http://www.dsic.upv.es/~balarcon/prole08/iCSR/benchmarks.html>

7 Conclusions and future work

In this work we have shown how to relax monotonicity requirements for proving innermost termination of context-sensitive rewriting. Fernández defined the notion of usable arguments to indicate those arguments that can be eventually reduced in innermost computations. This notion is totally equivalent to fix a replacement map where only those arguments are μ -replacing, thus transforming an innermost termination problem into a context-sensitive termination problem. We have adapted Fernández's approach [9] to be used for proving innermost termination of context-sensitive rewriting (Theorem 4.7). Moreover, since we have recently adapted the use of context-sensitive dependency pairs to deal with innermost termination of *CSR* [5], we have also investigated how to take advantage of it to adapt the result over dependency pairs and reduction pairs of [9] to the context-sensitive setting (Theorem 5.1). We have implemented both, the innermost and the innermost context-sensitive approaches in MU-TERM since the original results had not been implemented or tested before in any termination tool. We have performed some benchmarks that show no real improvement over previous (standard) approaches for proving either innermost termination [6] or innermost termination of *CSR* [5].

One of our main motivations to analyze and continue Fernández's work was the strong connection that she found between *CSR* and innermost rewriting, specifically the possibility of using *CSR* for proving innermost termination of rewriting

that she showed in her paper. Since in 2005 there was no really *powerful* technique for automatically proving termination of *CSR*, her work could not be *tested*, although it is interesting from a theoretical point of view. The recent definition of a context-sensitive version of the dependency pairs approach, though, suggested that some practical benefits could be obtained from her work. After achieving great results in proofs of termination of *CSR* we also developed the innermost version of CS-DPs [5]. The results obtained with this approach were also much better than the existing transformational methods for proving innermost termination of *CSR* [13]. This lead us to think about combining the new CS-DP approach with Fernández's ideas to obtain a good result for proving innermost termination as she stated in her paper. Moreover, she left open the problem of adapting the notion of usable argument to the context-sensitive setting for proving innermost termination of *CSR* which was a nontrivial issue due to the *peculiarities* of context-sensitive reductions as she already noticed. Therefore we have tackled the problem in all its open issues: we have implemented her results in the termination tool MU-TERM and we have also investigated and implemented the extension of her results to *CSR* (even the treatment over cycles that she mentioned). Unfortunately, our practical experience has been very different to what we expected: viewing an innermost termination problem as a context-sensitive termination problem, even with the possibility of using CS-DPs to achieve a proof, does *not* improve the performance of *classical* techniques for proving innermost termination of rewriting (dependency pairs, narrowing, etc [6]). On the other hand, trying to prove innermost termination of CS-TRSs by using a more restrictive replacement map do not offer any improvement over the results obtained by using innermost CS-DPs with the original replacement map [5]. Thus, we can only conclude that, nowadays, apart from the theoretical interest of the approach that combines two *different* strategies, it does not provide a better approach over the existing techniques for proving innermost termination [6] and innermost termination of *CSR* [5]. The other motivation for Fernández's work is that relaxing monotonicity requirements allows the use of non-monotonic orderings for direct proofs (see e.g., [10]). However, as far as we know, there is no tool implementing these orderings in automatic proofs of termination. Some approximations for proving innermost termination using *specific* orderings are presented in [10] but they does not consider the argument positions of function symbols. More research in this field could hopefully clarify whether usable arguments can actually improve the existing methods for proving innermost termination of (context-sensitive) rewriting in some way.

References

- [1] Alarcón, B., R. Gutiérrez, J. Iborra and S. Lucas. Proving Termination of Context-Sensitive Rewriting with MU-TERM. *Electronic Notes in Theoretical Computer Science*, volume 188, pages 105–115, 2007.
- [2] Alarcón, B., R. Gutiérrez, and S. Lucas. Context-Sensitive Dependency Pairs. In S. Arun-Kumar and N. Garg, editors, *Proc. of the 26th Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS'06*, LNCS 4337:297-308, Springer-Verlag, Berlin, 2006.
- [3] Alarcón, B., R. Gutiérrez, and S. Lucas. Improving the Context-Sensitive Dependency Graph. *Electronic Notes in Theoretical Computer Science*, 188:91–103, 2007.

- [4] B. Alarcón, R. Gutiérrez, and S. Lucas. Context-Sensitive Dependency Pairs. *Technical Report DSIC II/10/08 (72 pages)*. Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Spain, July 2008.
- [5] Alarcón, B., and S. Lucas. Termination of Innermost Context-Sensitive Rewriting Using Dependency Pairs. In B. Konev and F. Wolter, editors, *Proc. of 6th International Symposium on Frontiers of Combining Systems, FroCoS'07*, LNAI 4720: 73–87, Springer-Verlag, Berlin, 2007.
- [6] Arts, T., and J. Giesl. Termination of Term Rewriting Using Dependency Pairs *Theoretical Computer Science*, 236:133-178, 2000.
- [7] Clavel, M., F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and C. Talcott. All About Maude – A High-Performance Logical Framework. LNCS 4350, 2007.
- [8] Durán, F., S. Lucas, J. Meseguer, C. Marché, and X. Urbain. Proving Operational Termination of Membership Equational Programs. *Higher-Order and Symbolic Computation*, 21(1-2):59-88, 2008.
- [9] Fernández, M. L. Relaxing monotonicity for innermost termination. *Information Processing Letters* 93(3):117-123, 2005.
- [10] Fernández, M. L., G. Godoy, A. Rubio. Orderings Innermost Termination. In J. Giesl, editor, *Proc. 16th Int. Conf. on Rewriting Techniques and Applications Proc. of RTA'05*, LNCS 3467:17-31, Springer-Verlag, Berlin, 2005.
- [11] Futatsugi, K., 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.
- [12] Futatsugi, K., and A. Nakagawa. An Overview of CAFE Specification Environment – An algebraic approach for creating, verifying, and maintaining formal specification over networks –. In *Proc. of 1st International Conference on Formal Engineering Methods*, 1997.
- [13] Giesl, J., and A. Middeldorp. Innermost termination of context-sensitive rewriting. In M. Ito and M. Toyama, editors, *Proc. of 6th International Conference on Developments in Language Theory, DLT'02*, LNCS 2450:231-244, Springer-Verlag, Berlin, 2003.
- [14] Giesl, J., and A. Middeldorp. Transformation techniques for context-sensitive rewrite systems. *Journal of Functional Programming*, 14(4): 379-427, 2004.
- [15] Giesl, J., R. Thiemann, P. Schneider-Kamp, and S. Falke. Mechanizing and Improving Dependency Pairs *Journal of Automated Reasoning*, 37(3): 155-203. Springer-Verlag, 2006.
- [16] Goguen, J.A., 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.
- [17] Hirokawa, N., and A. Middeldorp. Automating the dependency pair method. *Information and Computation*, 199:172-199, 2005.
- [18] Lucas, S. Context-sensitive computations in functional and functional logic programs. *Journal of Functional and Logic Programming* 1998(1):1-61, 1998.
- [19] Lucas, S. Termination of Rewriting With Strategy Annotations. In R. Nieuwenhuis and A. Voronkov, editors, *Proc. of 8th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, LPAR'01*, LNAI 2250:669-684, Springer-Verlag, Berlin, 2001.
- [20] Lucas, S. Termination of on-demand rewriting and termination of OBJ programs. In *Proc. of 3rd International Conference on Principles and Practice of Declarative Programming, Proc. of PPDP'01*, pages 82-93, ACM Press, 2001.
- [21] Lucas, S. Context-sensitive rewriting strategies. *Information and Computation*, 178(1):293-343, 2002.
- [22] Lucas, S. MU-TERM: A Tool for Proving Termination of Context-Sensitive Rewriting. In V. van Oostrom, editor, *Proc. of 15th International Conference on Rewriting Techniques and Applications, RTA'04*, LNCS 3091:200-209, Springer-Verlag, Berlin, 2004. Available at <http://www.dsic.upv.es/~slucas/csr/termination/muterm>.
- [23] Lucas, S. Proving termination of context-sensitive rewriting by transformation. *Information and Computation*, 204(12):1782-1846, 2006.